TTMARK: PAIRWISE DISTORTION-FREE WATERMARKING BEYOND SINGLE-TOKEN ENTROPY

Anonymous authors
Paper under double-blind review

ABSTRACT

Distortion-free watermarking enables reliable attribution of machine-generated text while preserving output distribution. However, existing methods operate independently on each generated token, making their detection capability fundamentally constrained by the entropy of the next-token distribution. We present **T**andem **T**oken Water**Mark** (TTMARK), a general pairwise watermarking framework that extends distortion-free watermarking from individual tokens to adjacent token pairs. By watermarking the joint distribution of consecutive tokens, TTMARK enlarges the effective watermarking alphabet from V to V^2, allowing the detector to exploit both token entropy and conditional entropy while preserving distortion-freeness over the joint distribution. We further introduce a branch-isolating concatenated tandem generation algorithm that efficiently constructs the joint distribution in a single forward pass. Theoretically, we show that pairwise watermarking achieves better expected detection strength in low-entropy regimes. Extensive experiments across multiple language models, datasets, and three representative distortion-free watermarking schemes demonstrate that TTMARK consistently improves detectability without degrading generation quality, while also improving robustness to edits and enhancing localized watermark detection.

1 INTRODUCTION

The rapid deployment of large language models has necessitated the development of reliable and minimally invasive techniques for identifying machine-generated text. This imperative is further underscored by emerging regulatory frameworks, such as the EU AI Act (European Parliament and Council of the European Union, 2024), which mandate transparency and traceability for AI-generated content. Watermarking (Aaronson, 2023; Kirchenbauer et al., 2023; Zhao et al., 2024) addresses this challenge by embedding a statistical signal during the decoding process, enabling a detector equipped with a secret key to reliably distinguish watermarked outputs from human-written text. Among existing paradigms, distortion-free watermarking (Hu et al., 2024; Dathathri et al., 2024; Chen et al., 2025) is particularly compelling: when marginalized over the secret key, the watermarked decoding distribution remains strictly identical to the original model distribution, thereby precluding the introduction of systematic distributional bias.

Despite these advantages, watermark detectability degrades significantly in practical deployment scenarios. Modern decoding pipelines heavily rely on temperature scaling, top-p sampling, and top-k truncation to improve output quality, which simultaneously reduces the entropy of the next-token distribution. Because standard watermarking schemes operate token-by-token, their statistical power is strictly limited by the available uncertainty at each step (Huang et al., 2023; Dathathri et al., 2024; Chen et al., 2026). This constraint becomes critical during short generations, under conservative sampling, or in localized detection tasks, where the scarcity of high-entropy tokens severely diminishes the detector's observation pool.

To overcome this bottleneck, we propose **T**andem **T**oken Water**Mark** (TTMARK), a novel watermarking mechanism that shifts the domain of distortion-free watermarking from individual tokens to adjacent token pairs. By reweighting the joint distribution of consecutive tokens rather than the standard next-token distribution, TTMARK expands the effective vocabulary size from V to V^2. This formulation naturally surfaces both the entropy of the immediate token and the conditional entropy of its successor. Crucially, TTMARK maintains the distortion-free property over the joint

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.

space, ensuring that the original token marginals are perfectly preserved in expectation. Naively constructing the joint distribution requires independent evaluations for every candidate first token. TTMARK circumvents this by proposing **branch-isolating concatenated tandem generation**. By concatenating the first tokens together with a branch-isolating attention mask, all candidate continuations can be evaluated in a single forward pass, producing the required conditional distributions. For evaluation, we provide a direct detector for unedited text alongside an alignment-based detector robust to perturbations.

Theoretically, we show that in low-entropy regimes, token pairing yields an approximate $\sqrt{2}$ improvement in the aggregate z-score. More broadly, we prove the expected z-score of our distortion-free pairwise watermark is lower-bounded by the single-token baseline. Therefore, TTMARK can further exploit distributional uncertainty that single-token watermarking leaves unutilized. Empirically, we demonstrate that TTMARK universally improves baseline distortion-free watermarking schemes without degrading output quality. When evaluated on the C4 dataset (Raffel et al., 2020) with Llama-3.2-3B-Instruct at a 0.001% false positive rate, the tandem approach elevates the true positive rate of SynthID-Text (Dathathri et al., 2024) from 50.3% to 73.2%, and ENS-MCMark (Wu et al., 2026) from 54.0% to 81.8%. We confirm that these gains generalize consistently across different datasets, model families, and stringent sampling conditions. Moreover, the proposed pairwise watermark enhances robustness to perturbations and substantially reduces false negatives in localized detection contexts, while maintaining generation quality.

In summary, our primary contributions are as follows:

- We introduce TTMARK, a pairwise watermarking mechanism that preserves the distortion-free property while expanding the watermarking alphabet from V to V^2. To operationalize this efficiently, we develop a branch-isolating concatenated tandem generation procedure alongside robust direct and alignment-based detectors.
- We theoretically establish that operating over joint token distributions exploits residual conditional entropy, granting access to stronger detection signals than single-token baselines.
- We present extensive empirical evidence demonstrating that TTMARK significantly improves detectability and robustness across diverse language models, existing watermarking schemes, low-entropy sampling regimes, and semantic-preserving attacks.

2 RELATED WORK

2.1 LANGUAGE MODEL WATERMARKING

The rapid advancement of large language models (LLMs) has raised concerns about content authenticity and potential misuse (Grinbaum & Adomaitis, 2022; Crothers et al., 2023; Pei et al., 2026; Yang et al., 2024). To address these concerns, Aaronson (2023) proposed an unbiased watermarking framework based on the Gumbel-max trick, using prefix n-grams as watermark keys to guide pseudo-random token selection. Christ et al. (2024) extended inverse-sampling-based watermarking to binary language models using position-dependent keys. ITS-edit and EXP-edit (Kuditipudi et al., 2024) further refined these approaches by introducing predefined watermark key sets. Hu et al. (2024) developed an inverse-sampling method with model-driven log-likelihood ratio (LLR) detection, as well as γ-reweight, a logits-based unbiased watermarking scheme. STA-1 (Mao et al., 2025) improved generation quality in low-entropy regimes through rejection sampling, while SynthID (Dathathri et al., 2024) employed tournament sampling to enhance detectability. Building on γ-reweight, DiPmark (Wu et al., 2024) introduced model-agnostic detection, and MCmark (Chen et al., 2025) further improved the detectability of unbiased watermarking.

2.2 JOINT AND MULTI-TOKEN LANGUAGE MODELING

Recent approaches improve inference efficiency by predicting multiple future tokens rather than generating them strictly autoregressively. Block-wise decoding uses parallel prediction heads to reduce sequential dependencies (Stern et al., 2018). Speculative decoding adopts a draft-and-verify paradigm, in which a lightweight model proposes multiple tokens for parallel verification by a larger model while preserving its output distribution (Leviathan et al., 2023). Subsequent work improves sampling efficiency (Chen et al., 2023) and introduces tree-based verification to evaluate multiple speculative branches simultaneously (Miao et al., 2024).

3 METHOD

3.1 PRELIMINARY: DISTORTION-FREE WATERMARKING

Let V denote the vocabulary, with cardinality $N = |V|$. Given a prompt p, a language model M generates tokens autoregressively. At time step t, conditioned on the preceding sequence $\boldsymbol{x}_{1:t}$ and prompt p, the probability of generating the next token $x_{t+1} \in V$ is $P_M(x_{t+1} \mid \boldsymbol{x}_{1:t}, p)$. Watermarking reweights the original distribution $P_M(\cdot \mid \boldsymbol{x}_{1:t}, p)$ into a watermarked distribution $P_w(\cdot \mid \boldsymbol{x}_{1:t}, p, k)$, where $k \in \mathcal{K}$ is a private watermark key sampled from a known distribution $P_{\mathcal{K}}$ over the key space $\mathcal{K}$. Following prior work (Aaronson, 2023; Wu et al., 2024; Dathathri et al., 2024), a watermarking algorithm is *distortion-free* if it preserves the original token distribution in expectation over the key space. Formally, for any $P_M \in \mathcal{P}$ and any token $x_{t+1} \in V$,

$$\mathbb{E}_{k \sim P_{\mathcal{K}}} \left[P_w(x_{t+1} \mid \boldsymbol{x}_{1:t}, p, k) \right] = P_M(x_{t+1} \mid \boldsymbol{x}_{1:t}, p). \tag{1}$$

For detection, distortion-free watermarks typically define a scoring function $s : V \times \mathcal{K} \to \mathbb{R}$, which assigns score $s(x, k)$ to token x under key k. Under the null hypothesis that no watermark is present, the expected score for token x is $\mathbb{E}_{k \sim P_{\mathcal{K}}} s(x, k)$. For model-agnostic watermarks, where the original output logits are unavailable, this expectation is token-independent: $\mathbb{E}_{k \sim P_{\mathcal{K}}} s(x, k) = \mu$ for all $x \in V$. The score variance is likewise independent of x, i.e., $\mathrm{Var}_k[s(x, k)] = \sigma^2$ for all $x \in V$.

Detection then proceeds with a z-score test:

$$z = \frac{\sum_{t=1}^{T} s(x_t, k_t) - \mu T}{\sqrt{T}\sigma}, \tag{2}$$

where k_t is the watermark key associated with token x_t. If the resulting z-score exceeds a predetermined threshold, the sequence is classified as watermarked.

3.2 TANDEM TOKEN WATERMARK

We propose **T**andem **T**oken Water**Mark** (TTMARK), which applies watermarking to a two-token block rather than to each token independently. The full pipeline is illustrated in Fig. 1. At time step t, TTMARK operates on the joint next-token distribution $P_M(x_{t+1}, x_{t+2} \mid \boldsymbol{x}_{1:t}, p)$. By autoregressive factorization, this distribution can be written as

$$P_M(x_{t+1}, x_{t+2} \mid \boldsymbol{x}_{1:t}, p) = P_M(x_{t+1} \mid \boldsymbol{x}_{1:t}, p) P_M(x_{t+2} \mid \boldsymbol{x}_{1:t}, x_{t+1}, p). \tag{3}$$

We describe an efficient procedure for constructing this joint distribution in Sec. 3.3. TTMARK then reweights the joint distribution into $P_{\mathrm{TTMark}}(x_{t+1}, x_{t+2} \mid \boldsymbol{x}_{1:t}, p, k)$, while preserving distortion-freeness at the level of token pairs:

$$\mathbb{E}_{k \sim P_{\mathcal{K}}} \left[P_{\mathrm{TTMark}}(x_{t+1}, x_{t+2} \mid \boldsymbol{x}_{1:t}, p, k) \right] = P_M(x_{t+1}, x_{t+2} \mid \boldsymbol{x}_{1:t}, p). \tag{4}$$

The detector is modified accordingly: instead of scoring a single token, it uses a pairwise scoring function $s : V^2 \times \mathcal{K} \to \mathbb{R}$ and assigns score $s((x_{t+1}, x_{t+2}), k)$ to each generated token pair. Intuitively, TTMARK enlarges the effective vocabulary size from V to $V \times V$, increasing the amount of distributional uncertainty available for watermarking. We next make this intuition precise from two perspectives.

Entropy. Entropy is a primary bottleneck for watermark signal strength, as it limits the expected per-step z-score. To illustrate the effect of two-token watermarking, consider a setting in which each one-token conditional distribution has an average entropy of h. In the low-entropy regime, following prior work (Huang et al., 2023; Dathathri et al., 2024; Chen et al., 2026), the expected watermark strength, represented by the expected single-step z-score, can be approximated as scaling linearly with entropy, i.e., $z_t \propto ah$ for some constant a. For a single-token watermark over T tokens, the aggregate z-score scales as $z \propto \frac{Tah}{\sqrt{T}} = \sqrt{T}ah$. If watermarking is instead applied to $T/2$ token pairs, then for each pair, the joint entropy is approximately $2h$. The corresponding aggregate score scales as:

$$z_{\mathrm{TTMark}} \propto \frac{(T/2)a(2h)}{\sqrt{T/2}} = \sqrt{2T}ah, \tag{5}$$

which suggests a $\sqrt{2}$ gain under this approximation.

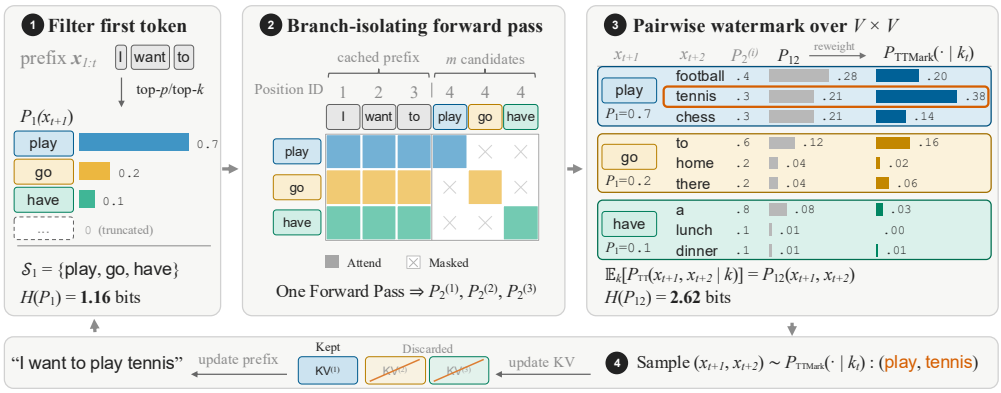


Figure 1: Overview of branch-isolating concatenated tandem generation. TTMARK first forms a filtered set of candidate next tokens and evaluates their continuations in a single batched forward pass using a branch-isolating attention mask. The resulting conditional distributions are combined with the first-token probabilities to construct a joint two-token distribution, enabling efficient pairwise watermarking over adjacent tokens.

Expected z-Score Upper Bound. We further show that, for any distortion-free watermark, grouping two tokens yields no smaller upper bound on the expected normalized z-score. For a single-token watermark, let $z(x,k) = (s(x,k) - \mu)/\sigma$ denote the normalized score for token x.

Theorem 3.1. *Let P be the unwatermarked distribution over vocabulary V, and let $P_w(\cdot \mid k)$ be any distortion-free watermark satisfying $\mathbb{E}_{k\sim P_\mathcal{K}}[P_w(x \mid k)] = P(x)$ for all $x \in V$. Assume that $\mathbb{E}_k[z(x,k)] = 0$ and $\mathrm{Var}_k[z(x,k)] = 1$ for every x. Then*

$$\mathbb{E}_{k\sim P_\mathcal{K},\, x\sim P_w(\cdot|k)}[z(x,k)] \le \sum_{x\in V} \sqrt{P(x) - P^2(x)}. \tag{6}$$

The proof is deferred to Appendix A.1.

Theorem 3.2. *Let $f(Q) = \sum_x \sqrt{Q(x) - Q^2(x)}$ denote the upper bound from the preceding theorem. For any joint distribution P_{12} on $V \times V$, with marginal P_1 and conditional distributions $P_{2|1=x_1}$, the pairwise upper bound satisfies*

$$f(P_{12}) \ge \frac{f(P_1) + \mathbb{E}_{x_1\sim P_1} f(P_{2|1=x_1})}{\sqrt{2}}. \tag{7}$$

The factor $\sqrt{2}$ arises from the normalization when two token-level contributions are combined into one statistic. The proof is deferred to Appendix A.2. Theorem 3.2 provides a justification for watermarking the joint two-token distribution. It shows that for any distortion-free watermark, the pairwise upper bound is at least as large as the normalized aggregate of the single-token bound and the expected conditional bound for the second token.

3.3 BRANCH-ISOLATING CONCATENATED TANDEM GENERATION

Directly constructing the pair distribution required by TTMARK is computationally expensive, since the conditional distribution of x_{t+2} depends on the realized value of x_{t+1}. Under standard decoding, however, the next-token logits are transformed by temperature scaling and truncation rules such as Top-P or Top-K sampling, which produce a sparse decoding distribution. Let P_1 denote the filtered distribution for x_{t+1}, and let $\mathcal{S}_1 = \{x_{t+1}^{(i)}\}_{i=1}^m$ be its nonzero support, where typically $m \ll |V|$. For each candidate $x_{t+1}^{(i)} \in \mathcal{S}_1$, we must compute the filtered distribution of x_{t+2} conditioned on that candidate. To ensure the efficiency of the generation, we propose *Branch-Isolating Concatenated Tandem Generation*, where the candidates are concatenated together, evaluated all in one forward pass, while preventing different candidate branches from attending to one another. The generation algorithm is illustrated in Alg. 1.

Let the current prefix cache have length L, and let the concatenated candidate block be $\boldsymbol{x}_{t+1}^{(1:m)} = [x_{t+1}^{(1)}, \ldots, x_{t+1}^{(m)}]$. The branch-isolating attention mask A allows the query associated with branch

r to attend to all cached prefix positions and to the corresponding candidate token $x_{t+1}^{(r)}$, while preventing attention to other candidate branches:

$$A_{r,j} = \begin{cases} 1, & 1 \le j \le L, \\ 1, & j = L + r, \\ 0, & \text{otherwise}, \end{cases} \qquad r = 1, \dots, m. \tag{8}$$

With this mask, the batched computation is equivalent to executing m independent one-token continuations from the same prefix cache. To preserve the same positional encoding as independent decoding, all candidate branches are assigned position index $L + 1$.

The same decoding parameters are then applied to each branch-specific second-token distribution. Let $P_2^{(i)}$ be the filtered distribution for x_{t+2} after conditioning on $x_{t+1}^{(i)}$, and let $\mathcal{S}_2^{(i)} = \{x \in V : P_2^{(i)}(x) > 0\}$ denote its support. The admissible pair support is therefore $\{(x_{t+1}^{(i)}, x_{t+2}) : x_{t+1}^{(i)} \in \mathcal{S}_1,\ x_{t+2} \in \mathcal{S}_2^{(i)}\}$. Following the autoregressive factorization in Eq. 3, the pre-watermark mass assigned to an admissible pair is $P_1(x_{t+1}^{(i)})P_2^{(i)}(x_{t+2})$. This filtered pair distribution is reweighted into $P_{\mathrm{TTMark}}(x_{t+1}, x_{t+2} \mid \boldsymbol{x}_{1:t}, p, k)$, from which the next token pair is sampled. After sampling, only the KV cache corresponding to the selected first-token branch is retained, and this cache is advanced with the sampled second token to initialize the next iteration. Consequently, each two-token generation step requires one batched branch evaluation and one cache update along the selected branch, rather than m separate conditional forward passes.

Space Efficiency Analysis. This procedure keeps the memory overhead nearly the same. The algorithm only creates temporary KV states and logits for the m_t candidate tokens at the current step, and then discards all but the selected branch. Thus the persistent cache size is essentially the same as ordinary autoregressive decoding.

Time Efficiency Analysis. From a purely serial computation perspective, if a batched step over m candidates were charged as m independent cached forward passes, producing two tokens would cost $m+1$ cached steps instead of the two steps used by standard decoding, giving an approximate time-cost ratio of $(m + 1)/2$. However, in actual GPU decoding, latency is often bounded by memory bandwidth because each cached step must read the model weights and KV states from memory. Branch-isolating generation evaluates the m candidate continuations in one batched pass, which amortizes these memory reads across branches. Therefore, the observed wall-clock overhead can be much smaller than the serial ratio $(m+1)/2$, which is reported in Sec. 5.1. Furthermore, TTMARK reduces computational overhead by applying the watermarking algorithm only once for every two tokens. It should be noted that watermarking algorithms can be highly computationally expensive, frequently necessitating large temporary matrix allocations and the strictly serial execution inherent to multi-layer reweighting architectures, such as SynthID-Text.

3.4 DETECTION

Direct Detection. We first consider the setting in which the generated sequence is observed without token-level corruption. Let $\boldsymbol{y}_{1:L}$ denote the sequence to be tested and let $n = \lfloor L/2 \rfloor$ be the number of complete token pairs. TTMARK partitions the sequence into the same non-overlapping pairs (y_{2i-1}, y_{2i}), for $i = 1, \dots, n$, used during generation. The final unpaired token, if L is odd, is discarded for detection. Given the secret key stream $\{k_t\}_{t=1}^{L}$, the detector evaluates the pairwise score $s((y_{2i-1}, y_{2i}), k_{2i-1})$. Under the null hypothesis that the text is not watermarked, the key is independent of the observed text; hence the pairwise score has key-averaged mean μ and variance σ^2, analogously to the single-token detector. The direct detection statistic is therefore

$$z_{\mathrm{dir}} = \frac{\sum_{i=1}^{n} s((y_{2i-1}, y_{2i}), k_{2i-1}) - n\mu}{\sqrt{n}\sigma}. \tag{9}$$

For sufficiently long sequences, z_{dir} is compared against a threshold calibrated from the standard normal approximation or from an empirical null distribution. This procedure is the exact pair-vocabulary analogue of single-token watermark detection, with V replaced by V^2.

Empirical Detection. Direct detection assumes that the detector observes the same pair boundaries as the generator. In practice, paraphrasing, insertion, deletion, or partial retokenization can

Table 1: Direct detection on C4 using Llama3.2-3B-Instruct. TTMARK improves the detection power of all three underlying watermarking schemes, especially under stringent FPR constraints.

	200 Tokens				400 Tokens			
	TPR@FPR=			Median	TPR@FPR=			Median
	0.1% ↑	0.01% ↑	0.001% ↑	p-value ↓	0.1% ↑	0.01% ↑	0.001% ↑	p-value ↓
ENS-DiPmark	47.0%	32.8%	23.9%	1.89e-3	73.8%	61.2%	46.8%	1.63e-5
+ TTMARK	**68.3%**	**58.9%**	**47.2%**	**2.01e-5**	**94.5%**	**88.6%**	**81.7%**	**4.03e-10**
SynthID-Text	71.8%	60.5%	50.3%	9.59e-6	92.9%	86.5%	80.1%	1.73e-9
+ TTMARK	**87.5%**	**79.5%**	**73.2%**	**5.79e-10**	**99.3%**	**98.5%**	**97.7%**	**8.25e-18**
ENS-MCMark	73.4%	62.5%	54.0%	3.86e-6	95.4%	91.3%	83.5%	2.00e-10
+ TTMARK	**92.0%**	**85.7%**	**81.8%**	**1.84e-11**	**99.6%**	**99.0%**	**98.5%**	**9.36e-21**

shift these boundaries, causing otherwise watermarked pairs to be evaluated under the wrong alignment. To make detection robust to such perturbations, we use a penalized alignment procedure that searches over subsequences of adjacent token pairs while discouraging excessive skipping.

Let $\beta \geq 0$ be a skip penalty. For an observed sequence $\boldsymbol{y}_{1:L}$, we compute the best penalized score by dynamic programming over token positions. Let D_t be the maximum score obtainable after reading the first t observed tokens. The recurrence is

$$D_t = \max\left\{D_{t-1} - \beta,\ D_{t-2} + z((y_{t-1}, y_t), k_{t-1})\right\}, \tag{10}$$

where the first transition skips token y_t, and the second transition retains the adjacent pair (y_{t-1}, y_t) using the positional key k_{t-1}. The boundary condition is $D_0 = 0$, and the pair-retention transition is valid only when $t \geq 2$. The resulting statistic is

$$z_{\text{emp}} = \frac{D_L}{\sqrt{n}}, \qquad n = \lfloor L/2 \rfloor. \tag{11}$$

We normalize by the original number of complete pairs n, so that the detector cannot inflate the statistic by selecting only a small number of high-scoring pairs. Because the maximization over alignments changes the null distribution of the score, thresholds for z_{emp} are calibrated empirically on non-watermarked text generated or collected under the same evaluation protocol. The full algorithm is detailed in Alg. 2.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

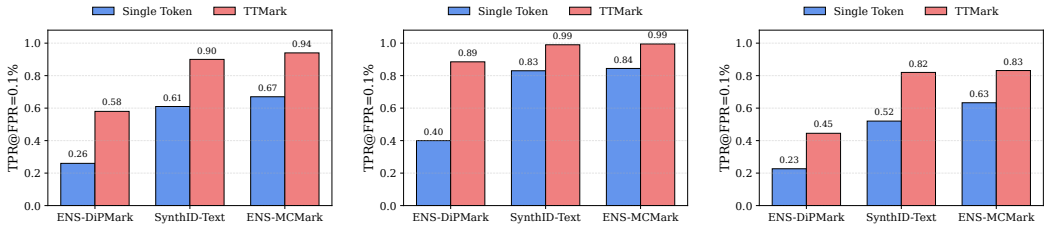
Our experiments focus on three representative distortion-free watermarks: ENS-DiPmark (Wu et al., 2024), SynthID-Text (Dathathri et al., 2024), and ENS-MCMark (Chen et al., 2026). For each baseline, we strictly follow the original watermark parameters and replace the single-token generation and scoring with TTMARK. Unless otherwise specified, generations use Top-$p = 0.95$, Top-$k = 50$, temperature 1.0, and a target length of 200 tokens. The skip penalty β is set to 5.

We evaluate on instruction-tuned models from different families, including Llama3.2-3B-Instruct (Touvron et al., 2023), Mistral-v0.3-7B-Instruct (Jiang et al., 2023), and Qwen2.5-7B-Instruct (Team, 2024). For datasets, following prior works (Hu et al., 2024; Kirchenbauer et al., 2023), we use C4 (Raffel et al., 2020), MMW Book Report (Piet et al., 2025), Dolly Creative Writing (Conover et al., 2023), and Longform QA (Tu et al., 2024).

We report true positive rates at fixed false positive rates (TPR@FPR), focusing on stringent false positive regimes that are most relevant for real-world deployment. We additionally report the median detection p-value over watermarked samples. Lower median p-values indicate stronger separation between watermarked and non-watermarked text.

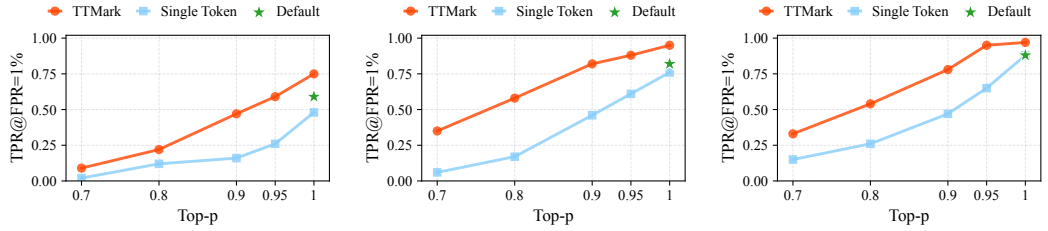
4.2 MAIN RESULTS

Table 1 reports direct detection results on C4 using Llama3.2-3B-Instruct. Across all three watermark families and both sequence lengths, TTMARK improves detection performance without



(a) MMW Book Report (b) Longform QA (c) Dolly CW

Figure 2: Cross-dataset direct detection performance on Llama3.2-3B-Instruct. Each panel reports TPR@FPR $= 0.1\%$ for single-token watermarking and its tandem counterpart.



(a) ENS-DiPmark (b) SynthID-Text (c) ENS-MCMark

Figure 3: Effect of Top-p on detection performance for MMW Book Report using Llama3.2-3B-Instruct. Both tandem and single-token variants use Top-$k = 50$; "Default" denotes the setting without Top-k truncation.

changing the base watermark algorithm. The gains are particularly pronounced at low false positive rates. For 200-token generations, TPR@FPR $= 0.001\%$ increases from 23.9% to 47.2% for ENS-DiPmark, from 50.3% to 73.2% for SynthID-Text, and from 54.0% to 81.8% for ENS-MCMark. At 400 tokens, the tandem variants reach 81.7%, 97.7%, and 98.5% TPR at the same false-positive rate, respectively. The median p-values also decrease by several orders of magnitude, indicating that the improvement is not limited to a particular operating threshold.

Figure 2 shows that the gains transfer beyond C4. Across MMW Book Report, Long-form QA, and Dolly CW, TTMARK consistently improves TPR@FPR $= 0.1\%$ for all three watermark families, with especially clear gains on the more challenging MMW Book Report and Dolly Creative Writing tasks. Additional results across datasets and model families are reported in Appendix C.

5 ANALYSIS

In this section, we analyze TTMARK under varying decoding entropy, stricter sampling parameters, computational costs, editing attacks, and localized detection settings. Additional experiments, including the skip-penalty β ablation, cross-model evaluations, generation-quality measurements, and extension to 3-token blocks, are provided in Appendix C.

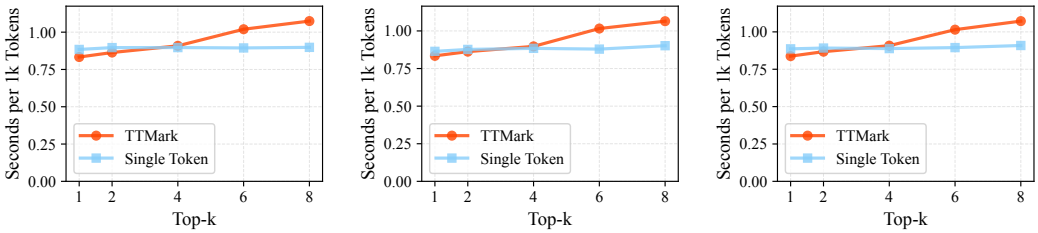
5.1 ABLATION STUDY

Effect of Decoding Entropy. We vary Top-p while keeping Top-$k = 50$ fixed on the MMW Book Report dataset with Llama3.2-3B-Instruct, to evaluate the performance under different decoding entropy. Figure 3 compares the tandem and single-token variants for each base watermark. The tandem variants maintain stronger detection across the evaluated Top-p range, showing that the benefit of TTMARK persists under changes in decoding entropy.

Stricter Sampling Parameters. Practical deployments often use more conservative decoding than the default experimental setting. For example, code generation, tool-using agents, and retrieval-augmented generation commonly reduce temperature in order to improve determinism and factual consistency. In the appendix, Table 5 reports representative recommended sampling parameters from widely used model families. Such settings are challenging for watermarking because the entropy becomes much smaller. Table 2 evaluates this low-entropy regime using Qwen2.5-7B-Instruct on

Table 2: Direct detection under stricter sampling parameters on MMW Book Report using Qwen2.5-7B-Instruct. Generations use the recommended sampling parameters with Top-$p = 0.8$, Top-$k = 20$, and temperature 0.7. Even when the effective candidate support is small, TTMARK improves detection with comparable generation time.

	Average Token Counts	TPR@FPR=			Median p-value ↓	Seconds per 1k Tokens ↓
		0.1% ↑	0.01% ↑	0.001% ↑		
ENS-DiPmark	1.37	9%	2%	0%	1.86e-1	**0.79**
+ TTMARK	1.93	**18%**	**9%**	**2%**	**3.46e-2**	0.80
SynthID-Text	1.39	21%	14%	8%	1.46e-2	0.86
+ TTMARK	1.93	**52%**	**36%**	**20%**	**7.96e-4**	**0.79**
ENS-MCMark	1.38	26%	13%	5%	1.27e-2	0.87
+ TTMARK	1.91	**58%**	**41%**	**28%**	**2.82e-4**	**0.80**



(a) ENS-DiPmark (b) SynthID-Text (c) ENS-MCMark

Figure 4: Runtime of tandem generation as a function of the effective candidate support on MMW Book Report using Llama3.2-3B-Instruct. Measurements are taken with batch size 32. Branch-isolating generation keeps the overhead modest because all first-token branches are evaluated together, and memory bandwidth is the main bottleneck. The accelerated generation speed when m is small stems from a 50% reduction in watermarking function calls.

MMW Book Report with the recommended sampling parameters. TTMARK consistently improves TPR at all reported false-positive rates, with negligible runtime difference. For SynthID-Text and ENS-MCMark, TTMARK more than doubles TPR@FPR $= 0.001\%$, from 8% to 20% and from 5% to 28%, respectively.

Computational Cost. This overhead of TTMARK is bounded by the size of the effective support m after decoding. In practice, truncation makes this support small, and the branch-isolating generation procedure in Sec. 3.3 evaluates all candidate branches in a single batched forward pass. Figure 4 reports the resulting runtime as a function of the number of remaining candidates m. The runtime increases smoothly with support size, but remains close to the single-token baselines in the operating regimes used in our experiments. Because memory bandwidth is the primary bottleneck, the observed overhead is much smaller than the $(m+1)/2$ ratio discussed in Sec. 3.3. Furthermore, the accelerated generation speed when m is small stems from a 50% reduction in watermarking function calls.

5.2 GENERATION QUALITY

Because TTMARK is distortion-free at the token-pair level, it can generally preserve the output quality. We empirically verify this expectation in Table 6 on summarization and machine translation benchmarks. Across ROUGE, BLEU, and BERTScore, the tandem variants remain generally the same compared with the corresponding single-token watermarks and the unwatermarked model.

5.3 ROBUSTNESS TO TEXT EDITING

We next evaluate whether the stronger detection signal persists after edits. Table 3 reports detection after rephrasing and back-translation on C4 using GPT5-Mini. For these experiments, 30% of each generated passage is edited, and detection uses the empirical alignment detection procedure from Sec. 3.4. Across both attack types, TTMARK improves TPR at fixed FPR and AUROC for all three underlying watermark families. Under rephrasing, TTMARK improves TPR at FPR$= 0.1\%$ by 16.1,

Table 3: Robustness to rephrasing and back-translation on C4 using GPT5-Mini. The generation model is Llama3.2-3B-Instruct. Each attack edits 30% of the generated passage. TTMARK improves detection after edits across all base watermarking schemes.

	Rephrase			Back Translation		
	FPR=1%↑	FPR=0.1% ↑	AUROC ↑	FPR=1%↑	FPR=0.1% ↑	AUROC ↑
ENS-DiPmark	59.4%	34.6%	0.9320	55.0%	38.0%	0.9486
+ TTMARK	**69.8%**	**50.7%**	**0.9449**	**72.0%**	**51.8%**	**0.9532**
SynthID-Text	72.2%	59.0%	0.9529	75.6%	60.0%	0.9646
+ TTMARK	**80.0%**	**72.3%**	**0.9570**	**83.1%**	**72.3%**	**0.9726**
ENS-MCMark	75.5%	61.9%	0.9587	78.0%	64.9%	0.9687
+ TTMARK	**88.0%**	**80.7%**	**0.9759**	**89.3%**	**81.3%**	**0.9795**

Table 4: Localized watermark detection. The protocol evaluates whether the detector can recover the watermarked span within a longer document. TTMARK reduces false negatives and improves F1 and IoU on both model families.

	Llama3.2-3B-Instruct				Qwen2.5-7B-Instruct			
	FPR↓	FNR↓	F1↑	IoU↑	FPR↓	FNR↓	F1↑	IoU↑
ENS-DiPmark	**0.000**	0.543	0.627	0.351	0.010	0.567	0.600	0.330
+ TTMARK	**0.000**	**0.277**	**0.839**	**0.590**	**0.003**	**0.303**	**0.820**	**0.574**
SynthID-Text	0.017	0.230	0.862	0.618	**0.007**	0.213	0.877	0.623
+ TTMARK	**0.010**	**0.050**	**0.969**	**0.785**	0.013	**0.063**	**0.961**	**0.787**
ENS-MCMark	0.006	0.210	0.879	0.618	0.037	0.153	0.899	0.637
+ TTMARK	**0.003**	**0.070**	**0.962**	**0.750**	**0.013**	**0.043**	**0.973**	**0.791**

13.3, and 18.8 percentage points for ENS-DiPmark, SynthID-Text, and ENS-MCMark, respectively. Under back translation, the corresponding gains are 13.8, 12.3, and 16.4 percentage points. These results suggest that the additional pairwise signal remains useful even when some token boundaries are corrupted, provided the detector is allowed to search over plausible local alignments.

5.4 LOCALIZED WATERMARK DETECTION

Finally, we consider a localized detection setting in which only a contiguous span of a document is watermarked. The detector must both decide whether a watermark is present and identify the watermarked region. For this evaluation, we strictly follow the settings of WaterSeeker (Pan et al., 2025), including its datasets and hyperparameters. Table 4 reports the resulting localized-detection performance for Llama3.2-3B-Instruct and Qwen2.5-7B-Instruct. Compared with the single-token baselines, TTMARK substantially reduces false negative rates while maintaining low FPR, leading to higher F1 and IoU. The consistent IoU improvements indicate that the stronger pairwise signal helps not only global detection but also finer-grained localization of watermarked spans.

6 CONCLUSION

We introduce TTMARK, a novel framework that extends distortion-free watermarking from individual tokens to adjacent token pairs. By watermarking the joint distribution of consecutive tokens, the proposed method expands the effective watermarking alphabet to V^2, successfully exploiting both immediate token entropy and conditional entropy to overcome the inherent detection bottlenecks of single-token approaches. To operationalize this efficiently, we developed a branch-isolating concatenated tandem generation algorithm that constructs the required joint distribution in a single forward pass. Theoretical analyses formally establish that this pairwise approach yields superior expected detection strength, particularly within challenging low-entropy regimes. Ultimately, extensive empirical evaluations across multiple language models and datasets confirm that TTMARK consistently enhances general detectability, improves robustness against semantic-preserving edits, and substantially boosts localized detection capabilities, all while preserving the underlying generation quality.

AI USE STATEMENT

We used generative AI tools to assist with coding and debugging experimental implementations, as well as to polish and refine the writing of the manuscript. Generative AI tools were not used to generate, develop, or formalize the research ideas or scientific contributions of this work. All AI-assisted code and text were reviewed and verified by the authors. The authors take full responsibility for the research methodology, experimental results, and final content of this paper.

REFERENCES

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

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

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

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

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

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

Evan N Crothers, Nathalie Japkowicz, and Herna L Viktor. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11:70977–71002, 2023.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), 2024. URL https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng.

Alexei Grinbaum and Laurynas Adomaitis. The ethical need for watermarks in machine-generated language. *arXiv preprint arXiv:2209.03118*, 2022.

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

Baihe Huang, Hanlin Zhu, Banghua Zhu, Kannan Ramchandran, Michael I Jordan, Jason D Lee, and Jiantao Jiao. Towards optimal statistical watermarking. *arXiv preprint arXiv:2312.07930*, 2023.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

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*, pp. 17061–17084. PMLR, 2023.

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

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Minjia Mao, Dongjun Wei, Zeyu Chen, Xiao Fang, and Michael Chau. Watermarking large language models: An unbiased and low-risk method. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7939–7960, 2025.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, et al. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pp. 932–949, 2024.

Leyi Pan, Aiwei Liu, Yijian Lu, Zitian Gao, Yichen Di, Lijie Wen, Irwin King, and Philip S Yu. Waterseeker: Pioneering efficient detection of watermarked segments in large documents. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 2866–2882, 2025.

Gan Pei, Jiangning Zhang, Menghan Hu, Zhenyu Zhang, Chengjie Wang, Yunsheng Wu, Guangtao Zhai, Jian Yang, and Dacheng Tao. Deepfake generation and detection: A benchmark and survey. *ACM Computing Surveys*, 58(11):1–41, 2026.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. *Advances in Neural Information Processing Systems*, 31, 2018.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. Waterbench: Towards holistic evaluation of watermarks for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 1517–1542. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.83. URL https://doi.org/10.18653/v1/2024.acl-long.83.

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

Yihan Wu, Ruibo Chen, Georgios Milis, and Heng Huang. An ensemble framework for unbiased language model watermarking. In *International Conference on Learning Representations*, volume 2026, pp. 17899–17917, 2026.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Ruth Petzold, William Yang Wang, and Wei Cheng. A survey on detection of llms-generated content. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9786–9805, 2024.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=SsmT8aO45L.

A Missing Proofs

A.1 Proof of Theorem 3.1

Proof. Recall that the normalized score is defined as $z(x,k) = (s(x,k) - \mu)/\sigma$. For the model-agnostic scoring rules considered in the main text, the null score has the token-independent mean $\mathbb{E}_k[s(x,k)] = \mu$ and variance $\mathrm{Var}_k[s(x,k)] = \sigma^2$ for every x. Therefore, for each fixed token x,

$$\mathbb{E}_k[z(x,k)] = \frac{\mathbb{E}_k[s(x,k)] - \mu}{\sigma} = 0, \qquad \mathrm{Var}_k[z(x,k)] = 1. \tag{12}$$

Using this zero-mean property together with distortion-freeness, $\mathbb{E}_k[P_w(x \mid k)] = P(x)$, we can center the watermarked probability inside the expectation:

$$\begin{aligned}
\mathbb{E}_{k,\, x\sim P_w(\cdot|k)}[z(x,k)] &= \sum_{x\in V} \mathbb{E}_k[P_w(x \mid k) z(x,k)] \\
&= \sum_{x\in V} \left(\mathbb{E}_k[(P_w(x \mid k) - P(x)) z(x,k)] + P(x)\mathbb{E}_k[z(x,k)]\right) \\
&= \sum_{x\in V} \mathbb{E}_k[(P_w(x \mid k) - P(x)) z(x,k)]\,.
\end{aligned} \tag{13}$$

For each x, define the centered random variables $A_x(k) = P_w(x \mid k) - P(x)$ and $B_x(k) = z(x,k)$. Then $\mathbb{E}_k[A_x(k)] = 0$, $\mathbb{E}_k[B_x(k)] = 0$, and $\mathbb{E}_k\left[B_x(k)^2\right] = \mathrm{Var}_k[z(x,k)] = 1$. Applying Cauchy–Schwarz to these two random variables gives

$$\begin{aligned}
\mathbb{E}_k[(P_w(x \mid k) - P(x)) z(x,k)] &\le |\mathbb{E}_k[A_x(k) B_x(k)]| \\
&\le \sqrt{\mathbb{E}_k[A_x(k)^2]}\sqrt{\mathbb{E}_k[B_x(k)^2]} \\
&= \sqrt{\mathrm{Var}_k[P_w(x \mid k)]}.
\end{aligned} \tag{14}$$

It remains to bound the first variance term. Since $0 \le P_w(x \mid k) \le 1$, we have $P_w(x \mid k)^2 \le P_w(x \mid k)$. Together with $\mathbb{E}_k[P_w(x \mid k)] = P(x)$, this implies

$$\mathrm{Var}_k[P_w(x \mid k)] = \mathbb{E}_k\left[P_w(x \mid k)^2\right] - P(x)^2 \le P(x) - P(x)^2. \tag{15}$$

Substituting this bound into the previous inequality and summing over $x \in V$ yields

$$\mathbb{E}_{k\sim P_K,\, x\sim P_w(\cdot|k)}[z(x,k)] \le \sum_{x\in V} \sqrt{P(x) - P(x)^2},$$

which proves the claim. □

A.2 Proof of Theorem 3.2

Proof. We prove the claim by comparing the contribution of each fixed first token x_1. Fix x_1, write $a = P_1(x_1)$, and let $b_y = P_{2|1=x_1}(y)$ for $y \in V$. If $a = 0$, then this x_1 contributes zero to all three quantities in the desired inequality, so the claim is trivial. We therefore assume $a > 0$, in which case $b = (b_y)_{y \in V}$ is a probability distribution.

The contribution of this x_1 to $f(P_{12})$ is

$$S = \sum_{y \in V} \sqrt{P_{12}(x_1, y) - P_{12}(x_1, y)^2} = \sum_{y \in V} \sqrt{a b_y (1 - a b_y)}.$$

The corresponding contribution to $f(P_1)$ is

$$A = \sqrt{P_1(x_1) - P_1(x_1)^2} = \sqrt{a(1-a)},$$

and the contribution to $\mathbb{E}_{x_1 \sim P_1} f(P_{2|1=x_1})$ is

$$B = a \sum_{y \in V} \sqrt{b_y (1 - b_y)}.$$

It suffices to show the pointwise inequality

$$S \geq \frac{A + B}{\sqrt{2}}. \tag{16}$$

We start by expanding S^2. Its diagonal terms are

$$\sum_y a b_y (1 - a b_y) = a - a^2 \sum_y b_y^2 = a(1-a) + a^2 \sum_y b_y (1 - b_y).$$

Here we used $\sum_y b_y = 1$. The first term is A^2, while the second term is the diagonal part of B^2. It remains to compare the cross terms. The cross terms of S^2 are

$$2a \sum_{y < y'} \sqrt{b_y b_{y'} (1 - a b_y)(1 - a b_{y'})},$$

whereas the cross terms of B^2 are

$$2a^2 \sum_{y < y'} \sqrt{b_y b_{y'} (1 - b_y)(1 - b_{y'})}.$$

For any pair $y \neq y'$, because $a \in [0, 1]$ and $b_y, b_{y'} \in [0, 1]$,

$$(1 - a b_y)(1 - a b_{y'}) - a^2 (1 - b_y)(1 - b_{y'}) = (1 - a)\,(1 + a - a(b_y + b_{y'})) \geq 0.$$

Thus

$$a \sqrt{b_y b_{y'} (1 - a b_y)(1 - a b_{y'})} \geq a^2 \sqrt{b_y b_{y'} (1 - b_y)(1 - b_{y'})},$$

so every cross term of S^2 is at least the corresponding cross term of B^2. Combining the diagonal and cross-term comparisons gives

$$S^2 \geq A^2 + B^2.$$

Finally, since $A, B \geq 0$, we have

$$A^2 + B^2 \geq \frac{(A + B)^2}{2},$$

which proves Eq. equation 16. Summing Eq. equation 16 over all $x_1 \in V$ yields

$$\sum_{x_1 \in V} S \geq \frac{\sum_{x_1 \in V} A + \sum_{x_1 \in V} B}{\sqrt{2}}.$$

By construction, $\sum_{x_1 \in V} S = f(P_{12})$, $\sum_{x_1 \in V} A = f(P_1)$, and $\sum_{x_1 \in V} B = \mathbb{E}_{x_1 \sim P_1} f(P_{2|1=x_1})$. Substituting these identities gives the desired inequality. □

B Algorithms for Generation and Detection

Algorithm 1 constructs the filtered two-token distribution by evaluating all admissible first-token branches in a single masked forward pass. The branch-isolating attention mask ensures that each candidate continuation is equivalent to an independent cached decoding step from the same prefix, while allowing the implementation to exploit batching on modern accelerators. Algorithm 2 gives the dynamic program used for edited or partially shifted text.

Algorithm 1 Branch-Isolating Concatenated Tandem Generation

Require: Language model M; prompt p; generation length T; decoding parameters Params; pairwise watermarking rule defining P_{TTMark}; watermark keys $\{k_t\}_{t\geq 0}$

$\ell, \mathrm{KV} \leftarrow M(p)$ ▷ Logits for $P_M(\cdot \mid p)$ and prefix cache
$t \leftarrow 0$, $\boldsymbol{x}_{1:0} \leftarrow [\,]$
while $t < T$ **do**
  $P_1 \leftarrow \mathrm{Filter}(\mathrm{softmax}(\ell), \mathrm{Params})$
  if $t = T - 1$ **then**
    Sample $x_{t+1} \sim P_1$ and set $\boldsymbol{x}_{1:t+1} \leftarrow [\boldsymbol{x}_{1:t}, x_{t+1}]$
    break
  end if
  $\mathcal{S}_1 = \{x_{t+1}^{(i)}\}_{i=1}^{m} \leftarrow \{x \in V : P_1(x) > 0\}$ ▷ Candidate support after truncation
  $\boldsymbol{x}_{t+1}^{(1:m)} \leftarrow [x_{t+1}^{(1)}, \ldots, x_{t+1}^{(m)}]$
  $L \leftarrow \mathrm{length}(\mathrm{KV})$ ▷ Current prefix-cache length
  Construct the branch-isolating attention mask A by Eq. 8 and assign position index $L+1$ to all candidates
  $\{\ell_2^{(i)}, \mathrm{KV}^{(i)}\}_{i=1}^{m} \leftarrow M(\boldsymbol{x}_{t+1}^{(1:m)} \mid \mathrm{KV}, A)$
  for $i = 1, \ldots, m$ **do**
    $P_2^{(i)} \leftarrow \mathrm{Filter}(\mathrm{softmax}(\ell_2^{(i)}), \mathrm{Params})$ ▷ Apply the same decoding parameters to x_{t+2}
    $\mathcal{S}_2^{(i)} \leftarrow \{x \in V : P_2^{(i)}(x) > 0\}$
  end for
  $\mathcal{S}_{12} \leftarrow \{(x_{t+1}^{(i)}, x) : x_{t+1}^{(i)} \in \mathcal{S}_1,\ x \in \mathcal{S}_2^{(i)}\}$
  Define the pre-watermark pair distribution on $\mathcal{S}_{12}$ using $P_1(x_{t+1}^{(i)})P_2^{(i)}(x_{t+2})$, following Eq. 3
  Obtain $P_{\mathrm{TTMark}}(x_{t+1}, x_{t+2} \mid \boldsymbol{x}_{1:t}, p, k_t)$ by reweighting the pre-watermark pair distribution
  Sample $(x_{t+1}, x_{t+2}) \sim P_{\mathrm{TTMark}}(x_{t+1}, x_{t+2} \mid \boldsymbol{x}_{1:t}, p, k_t)$
  $\boldsymbol{x}_{1:t+2} \leftarrow [\boldsymbol{x}_{1:t}, x_{t+1}, x_{t+2}]$
  $\mathrm{KV}^{\star} \leftarrow \mathrm{KV}^{(i^{\star})}$, where $x_{t+1} = x_{t+1}^{(i^{\star})}$
  $\ell, \mathrm{KV} \leftarrow M(x_{t+2} \mid \mathrm{KV}^{\star})$
  $t \leftarrow t + 2$
end while
return Generated sequence $\boldsymbol{x}_{1:T}$

C Additional Experiments

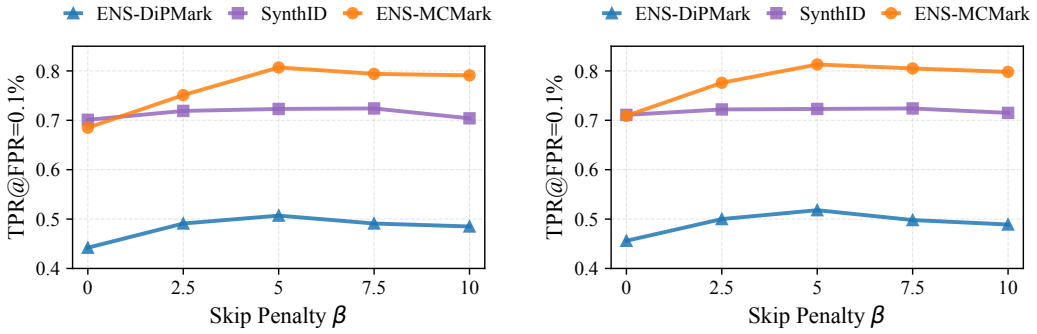
C.1 Ablation of Skip Penalty β

The skip penalty β controls the trade-off in the empirical alignment detector. When β is too small, the detector can skip many tokens and select a small number of unusually high-scoring adjacent pairs, which weakens separation after null calibration. When β is too large, the detector becomes close to direct detection and loses the ability to recover from local insertions, deletions, or modifications. Figure 5 explores the effect of β under rephrasing and back-translation attacks. The results show that moderate skip penalties provide the most reliable behavior across both perturbation types. We therefore use $\beta = 5$ in the main experiments, which preserves most of the pairwise watermark signal while still allowing the detector to realign around locally edited spans.

Algorithm 2 Empirical Detection

Require: Observed tokens $\boldsymbol{y}_{1:L}$; pairwise score s; keys $\{k_i\}$; null mean μ; null standard deviation σ; skip penalty β
Ensure: Empirical detection statistic z_{emp}

$n \leftarrow \lfloor L/2 \rfloor$
Initialize $D_t \leftarrow -\infty$ for $0 \leq t \leq L$
$D_0 \leftarrow 0$
for $t = 1, \ldots, L$ **do**
    $D_t \leftarrow D_{t-1} - \beta$ ▷ Skip y_t
    if $t \geq 2$ **then**
        $r \leftarrow (s((y_{t-1}, y_t), k_{t-1}) - \mu)/\sigma$
        $D_t \leftarrow \max(D_t, D_{t-2} + r)$ ▷ Retain adjacent pair
    end if
end for
return $z_{\text{emp}} \leftarrow D_L/\sqrt{n}$

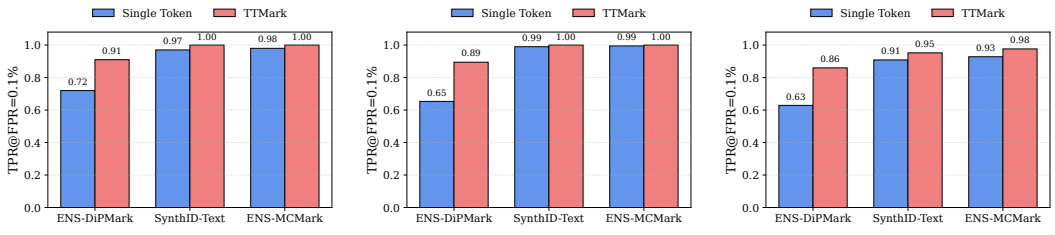


(a) Rephrasing (b) Back translation

Figure 5: Ablation of the empirical detector skip penalty β under edits. Moderate penalties balance alignment flexibility against over-selection of isolated high-scoring pairs.

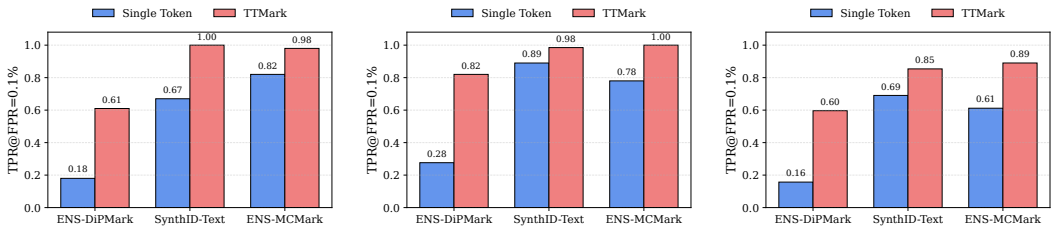
C.2 Cross-Dataset Results on Additional Model Families

The main text reports cross-dataset results for Llama3.2-3B-Instruct. Figures 6 and 7 provide the corresponding results for Qwen2.5-7B-Instruct and Mistral-7B-Instruct-v0.3. Across MMW Book Report, Long-form QA, and Dolly Creative Writing, the tandem variants consistently improve direct detection at FPR $= 0.1\%$ over their single-token counterparts. This pattern holds for all three underlying watermarking schemes, indicating that the gain from TTMark is not specific to a single model family or task distribution.



(a) MMW Book Report (b) Longform QA (c) Dolly CW

Figure 6: Cross-dataset direct detection performance on Qwen2.5-7B-Instruct. Each panel reports TPR@FPR $= 0.1\%$ for single-token watermarking and its tandem counterpart.



(a) MMW Book Report (b) Longform QA (c) Dolly CW

Figure 7: Cross-dataset direct detection performance on Mistral-7B-Instruct-v0.3. Each panel reports TPR@FPR $= 0.1\%$ for single-token watermarking and its tandem counterpart.

Table 5: Representative stricter sampling parameters recommended by model families. These settings reduce decoding entropy and can substantially decrease the effective candidate support available for watermarking. The dash in the Top-k column indicates that no explicit Top-k truncation is recommended in that setting.

	Top-P	Top-K	Temperature
Qwen3 Series	0.95	20	0.6
Qwen2.5 Series	0.8	20	0.7
Llama3 Series	0.9	-	0.6

C.3 Sampling Parameters for Low-Entropy Decoding

Table 5 records representative recommended decoding parameters from common model families. These settings are more conservative than the default experimental configuration and reduce the effective support size available to the watermark. The dash in the Top-k column indicates that no explicit Top-k truncation is recommended in that setting.

C.4 Generation Quality

Because TTMark is distortion-free over token pairs, it should not systematically change the marginal text distribution when averaged over the watermark key. Table 6 evaluates this expectation on summarization and machine translation. Across ROUGE, BLEU, and BERTScore, the tandem variants remain close to both the unwatermarked model and the corresponding single-token watermarking schemes.

C.5 Extension to Longer Blocks

TTMark naturally extends from two-token blocks to longer token blocks. In principle, watermarking r-token blocks further increases the effective vocabulary from V to V^r and can expose additional conditional entropy. Table 7 reports results for a three-token variant on C4 with Llama3.2-3B-Instruct. The three-token variant improves direct detection for all three base watermarks, particularly at short generation lengths and stringent false-positive rates. For example, at 200 tokens and FPR $= 0.001\%$, the three-token variant improves over the two-token variant from 47.2% to 63.0% for ENS-DiPmark, from 73.2% to 82.6% for SynthID-Text, and from 81.8% to 87.0% for ENS-MCMark.

At the same time, longer blocks require more conditional distributions and are more sensitive to boundary disruption under editing attacks. We therefore use two-token blocks as the default design point, which captures a substantial fraction of the available gain while preserving efficient generation and simple detection.

Table 6: Generation quality under watermarking. The tandem variants preserve summarization and translation quality relative to the corresponding single-token watermarking schemes.

	Text Summarization				Machine Translation	
	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	BLEU	BERTScore
No Watermark	0.3849	0.1465	0.2548	0.2733	21.30	0.8336
+ TTMARK	0.3854	0.1473	0.2554	0.2738	21.69	0.8388
ENS-DiPmark	0.3855	0.1333	0.2432	0.2734	20.93	0.8332
+ TTMARK	0.3846	0.1333	0.2430	0.2726	21.55	0.8375
SynthID-Text	0.3850	0.1471	0.2551	0.2735	20.95	0.8324
+ TTMARK	0.3861	0.1475	0.2557	0.2741	21.93	0.8402
ENS-MCMark	0.3842	0.1459	0.2432	0.2725	21.08	0.8325
+ TTMARK	0.3845	0.1466	0.2432	0.2730	22.00	0.8399

Table 7: Direct detection when extending TTMARK from two-token blocks to three-token blocks on C4 using Llama3.2-3B-Instruct. Longer blocks further improve detection, especially at stringent false-positive rates.

	200 Tokens				400 Tokens			
	TPR@FPR=			Median	TPR@FPR=			Median
	0.1% ↑	0.01% ↑	0.001% ↑	p-value ↓	0.1% ↑	0.01% ↑	0.001% ↑	p-value ↓
ENS-DiPmark	47.0%	32.8%	23.9%	1.89e-3	73.8%	61.2%	46.8%	1.63e-5
+ TTMARK 2-Token	68.3%	58.9%	47.2%	2.01e-5	94.5%	88.6%	81.7%	4.03e-10
+ TTMARK 3-Token	**77.0%**	**70.8%**	**63.0%**	**1.43e-7**	**98.7%**	**96.2%**	**93.0%**	**6.42e-14**
SynthID-Text	71.8%	60.5%	50.3%	9.59e-6	92.9%	86.5%	80.1%	1.73e-9
+ TTMARK 2-Token	87.5%	79.5%	73.2%	5.79e-10	99.3%	98.5%	97.7%	8.25e-18
+ TTMARK 3-Token	**92.8%**	**87.7%**	**82.6%**	**3.81e-14**	**99.9%**	**99.6%**	**98.6%**	**6.84e-26**
ENS-MCMark	73.4%	62.5%	54.0%	3.86e-6	95.4%	91.3%	83.5%	2.00e-10
+ TTMARK 2-Token	92.0%	85.7%	81.8%	1.84e-11	99.6%	99.0%	98.5%	9.36e-21
+ TTMARK 3-Token	**93.4%**	**90.5%**	**87.0%**	**9.08e-15**	**99.8%**	**99.5%**	**99.5%**	**1.16e-27**