TRACING MODEL GENERATED DNA WITH POSITION INDEPENDENT WATERMARK VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generation time watermarks support provenance for generative models, but verification becomes a synchronization problem when token alignment is unknown. In genomic sequences, a verifier may lack strand orientation, token phase, generation boundaries, and the location of the generated region; insertions and deletions can further disrupt alignment. We introduce a position independent verification framework that searches both strands, every nucleotide start, and four window lengths. Under an explicit null model for keyed bits, a Bonferroni correction converts the strongest window statistic into one read level significance decision, accounting for the full search without assuming independence between overlapping windows. Verification requires only the observed DNA, a key, and public settings. We instantiate the framework with the existing SynthID tournament watermark in Carbon-500M and GENERator-v2 1.2B under canonical six mer sampling. Across 192 held out prompts with two draws each, the verifier detected 384/384 correct key marked reads per model, both unedited and after each tested single nonadaptive substitution, insertion, or deletion. Each ordinary and wrong key control family yielded at most one positive among 384 reads per model and condition at a declared 1% threshold. Neither model showed a statistically detectable difference across 14 prespecified model likelihood and sequence measures after correction. These results support verification without model access under unknown synchronization in the evaluated six mer and single edit regime.

1 INTRODUCTION

Genomic language models increasingly support the generation of coding sequences, regulatory elements, genome scale constructs, and functionally constrained DNA designs (Nguyen et al., 2024; Wu et al., 2025; Ben Allal et al., 2026; Li et al., 2026; King et al., 2026). A nucleotide sequence, however, carries no inherent record of whether it originated in a biological sample, a computational design process, or a generative model. That provenance can be lost when a sequence is separated from the logs and metadata maintained by its generating system. Provenance mechanisms that remain associated with a sequence throughout design and synthesis are therefore of increasing interest (Baker & Church, 2024).

Watermarking provides one approach to this problem by embedding a statistical signal during generation (Kirchenbauer et al., 2023). A verifier holding the corresponding key can later test for that signal without access to the model, prompt, or generation logs. In DNA, this becomes a synchronization problem. Models that generate nonoverlapping six mer tokens do not encode their token boundaries in the emitted sequence, leaving six possible token phases. Reverse complementation changes token identities and their order, so verification must consider both strand orientations. Surrounding sequence can also obscure the generation boundaries: where the generated region begins and ends within the read. The verifier must search these alternatives while controlling the false positive probability of the final read level decision.

We develop a verifier for unknown strand orientation, token phase, and generation boundaries, and evaluate it using the SynthID tournament watermark (Dathathri et al., 2024) in Carbon-500M (Ben Allal et al., 2026) and GENERator-v2 1.2B (Li et al., 2026). Under the evaluated policy, these models share nonoverlapping six mer tokenization and direct sampling from canonical tokens, but differ in size, training corpus, and tokenizer construction. We ask whether the sampler follows its in-

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.

tended fixed key distribution, whether watermarking changes the prespecified model and sequence proxies, and whether a verifier can recover the mark without knowing the generation boundary, strand, or token phase. The detector searches both strands, every start position, and four window lengths, then applies one correction covering the complete search.

In both models, the verifier detected every held out marked read under the tested unedited and single base edit conditions. The quality proxy comparisons showed no statistically detectable difference after correction, and ordinary and wrong key control counts were compatible with the declared false positive target. In both models, the window level results are consistent with the search selecting regions with preserved or recovered six mer token alignment after insertions and deletions. These findings establish statistical detection within the evaluated corpus, not biological function, secret key security, or equivalence between the models. Distribution preservation denotes an expectation over fresh keyed functions; at a fixed key and context, sampling follows the explicitly reweighted distribution.

2 BACKGROUND

2.1 WATERMARKING DESIGNED DNA

DNA watermarking predates its application to generative models. The first synthetic bacterial genome included watermark sequences encoding identifying information to distinguish the synthetic chromosome from its natural counterpart (Gibson et al., 2010). DNA-Crypt formalized message embedding through synonymous codon choices and error correction (Heider & Barnekow, 2007). Related approaches embed information within an open reading frame while preserving the encoded amino acid sequence, with experimental evidence that embedded messages can persist during viral replication (Liss et al., 2012).

These approaches rely on explicit sequence design, through either payload insertion or modification of selected nucleotides. They require suitable embedding locations and consideration of the biological consequences of those changes. Generation time watermarking addresses a complementary setting: the mark is introduced through the model's sampling distribution as the sequence is generated, without requiring a separate payload region. A fixed, recognizable marker also differs from a distributed keyed signal in its susceptibility to targeted removal or copying.

2.2 WATERMARKING LANGUAGE MODEL OUTPUT

Language model watermarking modifies token sampling to create a statistically detectable pattern. Some methods bias a pseudorandom subset of tokens and detect their excess frequency (Kirchenbauer et al., 2023); others preserve the original distribution in expectation over fresh keyed randomness (Christ et al., 2024; Kuditipudi et al., 2024; Aaronson & Kirchner, 2022; Hu et al., 2024; Wu et al., 2024). SynthID belongs to the latter class. We use its existing construction and study verification when token alignment and generation boundaries are unknown.

Robustness depends on the attacker's capabilities and the permitted edits. Prior work identifies evasion, key recovery, and forgery risks under stronger adaptive capabilities (Zhang et al., 2024; Diaa et al., 2025; Sadasivan et al., 2025; Jovanović et al., 2024; Reynolds et al., 2025). The present study tests statistical detection after one nonadaptive base edit and does not evaluate those attacks.

2.3 SYNTHID

We use the binary tournament construction of SynthID-Text (Dathathri et al., 2024). At each generation step, a secret key and a short context of recently generated tokens determine binary mark bits for each candidate token and tournament layer. Each layer favors candidates with a mark bit of 1. Averaged over fresh keyed functions, the tournament preserves the model's next token distribution; at a fixed key and context, it samples from a reweighted distribution. Detection reconstructs the mark bits from the output and tests for an excess of 1s.

Two properties motivate this construction for genomic sequences. First, the mark bits depend on local token context rather than absolute position, allowing detection to use contexts retained when a generated region is relocated or cropped. Second, the detector requires only the output, key, and

public settings; it does not need the generating model or its probabilities. SynthID-Text has also been evaluated in a large scale text deployment (Dathathri et al., 2024).

2.4 Genomic tokenization and alignment

Several generative genomic models, including both evaluated here, tokenize DNA into nonoverlapping six base tokens to reduce token sequence length (Wu et al., 2025; Ben Allal et al., 2026; Li et al., 2026). The watermark sampler operates on the distribution used to sample those tokens. We therefore use each model's direct canonical 4,096 token distribution; for GENERator, this excludes the released base by base sampling path (GenerTeam, 2026). This shared policy definition makes the sampling comparison meaningful; a shared vocabulary alone would not suffice.

Six mer tokenization also creates the study's central synchronization problem: token boundaries are not represented in the emitted nucleotide sequence. Here, token phase is the nucleotide offset at which the sequence is partitioned into consecutive six mers; it does not refer to alignment to a reference genome. A verifier must therefore consider all six phases, both strand orientations, and candidate locations and lengths of the generated region. Selecting the strongest window without correcting for that search would overstate the evidence. Our verifier therefore uses an observed read, a key, and public settings without assuming a coding region, reference alignment, or known generation boundary. This setting is complementary to genomic watermarking through synonymous codon substitution (Zhang et al., 2025) and keyed watermarking of protein generation (Chen et al., 2025).

3 Methods

3.1 Shared model policy and sampling

Generation used Carbon-500M (Ben Allal et al., 2026; HuggingFaceBio, 2026) and GENERator-v2-eukaryote-1.2b-base (Li et al., 2026; GenerTeam, 2026), two decoder only genomic language models with nonoverlapping six base tokens. At every step, we restricted logits to the 4,096 canonical DNA tokens and normalized once. For GENERator, this specifies its direct canonical token distribution, not the base by base path used by its released code by default. Temperature was 1.0, with no top k or top p truncation.

This policy defines the distribution p used by both experimental arms: the ordinary arm samples directly from p without a key, whereas the marked arm applies the tournament below before sampling. We evaluated the models separately using the same generation, sampler validation, sequence quality, and detection procedures and report results for each model.

Generation ran on GPU hardware. Model scoring ran on CPU for Carbon and on GPU for GENERator, and detection ran on x86-64 CPUs for both.

3.2 Watermark embedding

At each generation step, the model assigns probability p to each of 4,096 tokens. The key, the preceding four tokens of context, and the candidate token determine its binary mark bit g. Mark bits are derived with HMAC-SHA-256 over the key, a fixed domain label, the context, and the candidate token. The tournament formula, the number of layers, the context width, and the repeated context rule follow the public reference implementation of SynthID-Text (Google DeepMind, 2024). One tournament layer reweights the distribution as

$$p'(x) \;=\; p(x)\left(1 + g(x) - \textstyle\sum_y p(y)\, g(y)\right), \tag{1}$$

so that candidates with a mark bit of 1 gain probability according to the total probability mass assigned to such candidates. Iterating this update is equivalent to evaluating the binary tournament without explicitly constructing its 2^m leaves. The sampler applies $m = 30$ layers, each with its own mark bits, and samples a token from the resulting distribution p_k (Figure 1a).

The first four generated tokens are sampled ordinarily so that subsequent watermark contexts can be reconstructed from the continuation alone. A repeated four token context is also sampled ordinarily

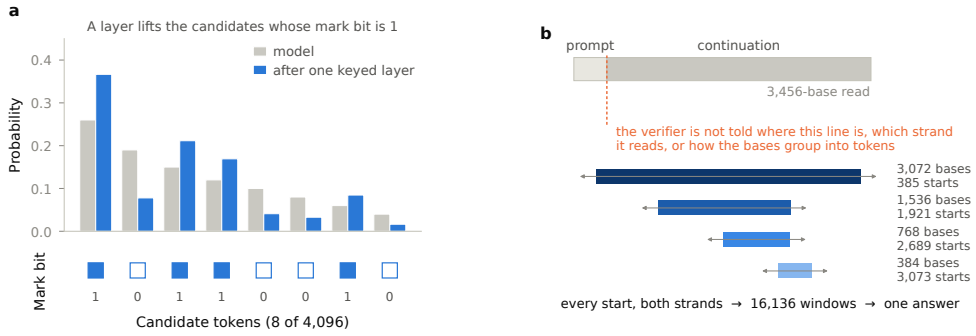


Figure 1: **SynthID embedding and position independent detection.** **a**, One keyed layer, drawn for eight of the 4,096 candidate tokens with an illustrative distribution. Candidates whose mark bit is 1 gain probability; the others lose it. The sampler applies 30 such layers. **b**, Detector inputs and search. The verifier receives the read, key, and fixed public settings without the generation boundary, strand orientation, or token alignment. It scores windows of four lengths at every starting base on both strands, comprising 16,136 windows for a 3,456 base read, and reports one read level result corrected for the full search. Panel **a** illustrates Equation 1 and is not a measurement.

and excluded from the detector score, following the original construction. The sampler maintains a history of 1,024 contexts to detect repetitions.

Averaged over fresh keyed functions, Equation 1 leaves the model's next token distribution unchanged. This identity does not imply equality at a fixed key and context, where the sampler follows the explicitly reweighted distribution p_k. Sampler fidelity was therefore evaluated against p_k.

3.3 Position independent detection

The verifier receives the read, the key, the generation domain label, and fixed public detector settings. It does not receive the prompt, the model, the model's probabilities, the seed, the strand orientation, the token alignment, or the generation boundary (Figure 1b).

For each of the four window lengths L that fit within the read, the verifier evaluates every starting base on the supplied strand and its reverse complement. This search covers candidate generation boundaries and all six token alignments. Within each window, DNA is partitioned into tokens relative to the window start. The first four tokens provide context; the verifier reconstructs mark bits for the remaining eligible tokens and obtains S ones among n scored bits.

Under the null assumption that mark bits behave as independent fair binary values, the window level P value is the exact binomial tail

$$P_{\text{win}} \;=\; \Pr\big[\text{Binomial}(n, \tfrac{1}{2}) \geq S\big]. \tag{2}$$

The verifier scores $M = 2\sum_L (N - L + 1)$ windows, which is 16,136 for a 3,456 base read, and reports a single read level result:

$$\begin{aligned} P_{\text{read}} \;&=\; \min\big(1,\; M \cdot \min_{\text{windows}} P_{\text{win}}\big), \\ \text{mark found} \;&\iff\; P_{\text{read}} \leq \alpha. \end{aligned} \tag{3}$$

The Bonferroni correction by M accounts for all searched windows, including overlapping windows, without requiring independence between their tests. The threshold uses the prespecified $\alpha = 0.01$ and is not fitted to the evaluated reads. Equivalently, a read is positive only if at least one window reaches strength $\log_{10}(M/\alpha) = 6.21$. The window lengths are 384, 768, 1,536, and 3,072 bases. The verifier skips window lengths that exceed the read length and does not score reads shorter than every prespecified window length. Reverse complement coordinates are mapped back onto the supplied read, and log probabilities are retained throughout to avoid floating point underflow.

3.4 PROMPTS AND GENERATION

Both models used the same cohort of 256 prompts: 384 base forward strand windows drawn from 16 accession versioned RefSeq chromosome records across eight model organisms. Before generation, windows were chosen by hashing a fixed label, the accession, and a segment index. Human, patient, controlled access, organelle, plasmid, and ambiguous base spans were excluded. No prompt was added, moved, or removed after any model was run.

For each model, every prompt produced two marked and two ordinary continuations of 512 tokens (3,072 bases), yielding 512 marked and 512 ordinary continuations per model. The two draws used different replay seeds in both arms, and the two marked draws used different keys. Both keys are published for reproducibility. Deployment would require independently generated secret keys shared only with authorized generators and verifiers.

3.5 SAMPLER VALIDATION

For each model and prompt, we fixed the model state at token index 64 of its ordinary draw zero trajectory and drew 5,000 samples per arm from that state. We compared the counts with the exact intended distribution using a Monte Carlo goodness of fit test with 999 replicates. The marked arm was compared against p_k, not against p, because the distribution preservation identity does not hold at a fixed key and context. Eight additional states per model used a deliberately incorrect sampler as a negative control.

3.6 SEQUENCE MEASURES

The model score is the teacher forced negative log likelihood per generated token, using the same model and token restriction as generation. The prompt is supplied as context and excluded from the loss. Each model scores its own outputs, so absolute scores are not used for comparisons between models. Thirteen additional measures were specified before analysis: GC content, purine content, CpG content, base, dinucleotide, and trinucleotide entropy, distinct 6 mer fraction, longest and mean single base run, Jensen Shannon drift from the prompt at $k = 1, 2, 3$, and perplexity. All 14 form one multiple testing family under Benjamini Hochberg correction.

3.7 EDITS AND STATISTICS

Each edited read contains exactly one substituted, inserted, or deleted base within the continuation. A fixed random rule selects the position and, where applicable, the replacement or inserted base without consulting the verifier. Prompts are the independent unit; the two draws from a prompt are paired repetitions. Uncertainty in the sequence measures is assessed using an equal weight bootstrap over prompts with 20,000 replicates and a paired sign flip test with 100,000 replicates. Detection rates are reported with exact two sided 95% binomial intervals over the 192 held out prompts. For control families, a prompt is positive if either draw is positive; marked output recall is also assessed by requiring both draws to be detected. All reported read level P values include the correction over every searched strand, starting base, and window length.

4 RESULTS

4.1 SAMPLER FIDELITY

Each sampler agreed with its intended distribution at the tested model states. Among 256 marked states per model, tests for 13 Carbon states and 5 GENERator states yielded nominal P values below 0.05, compared with 12.8 expected by chance; the corresponding ordinary arm counts were 12 and 14. Neither the marked nor the ordinary arm produced a rejection after correction in either model, whereas the deliberately incorrect sampler was rejected at all eight control states in each model. This supports sampler fidelity at the tested states and at the resolution provided by 5,000 draws per state; it is not a proof of correctness at every possible model state.

4.2 SEQUENCE QUALITY PROXIES

For every model and prompt, we generated two marked and two ordinary continuations of 3,072 bases, giving 512 matched pairs from 256 prompts per model. Marked and ordinary output showed no statistically detectable difference in either model across the prespecified 14 measure family (Figure 3). All intervals contained zero, and no comparison was significant after multiple testing correction; the smallest adjusted P value was 0.85 for Carbon and 0.73 for GENERator.

The paired marked minus ordinary difference in model score was $+0.00852$ nats per token in Carbon (95% interval -0.04426 to $+0.05973$; $P = 0.75$) and -0.00564 in GENERator (95% interval -0.05614 to $+0.04378$; $P = 0.83$). The largest absolute standardized effects were 0.084 in Carbon and 0.111 in GENERator. These results indicate no measurable quality loss under the evaluated proxies, but they do not establish equivalence, relative biological quality between models, or biological function.

4.3 RECALL AND FALSE POSITIVE RATE

The prompt split was specified before examining detection results: 64 prompts were reserved for selection between two window score summaries, and 192 were held out for evaluation. Neither subset was used to fit the detection threshold. For each model, the 384 base prompt was prepended to each held out continuation, producing a 3,456 base read whose generation boundary was withheld from the verifier. Three detection families were evaluated: marked reads with the correct key, ordinary reads, and marked reads with the other draw's key. Each family was evaluated unedited and after exactly one substitution, insertion, or deletion. The search is determined by the read and window lengths and is identical across models. Unedited reads require $M = 16{,}136$ windows and a threshold of 6.21; the window count is recomputed for the read length after an edit.

In both models, the verifier detected all 384 marked reads in every condition, including each tested single base edit (Table 1). Among the controls, Carbon produced one ordinary positive and no wrong key positives; GENERator produced no ordinary positives and one wrong key positive in the unedited, substitution, and insertion conditions. Each positive cell represents 1 of 384 reads (0.26%). Recall and the two control rates are reported separately because a pooled accuracy would depend on the prevalence of marked and control reads in an application.

Figure 2 shows the separation between marked reads and controls relative to the corrected detection threshold. In Carbon, the weakest unedited marked read had strength 124.8, compared with the 6.21 corrected threshold; in GENERator, the corresponding value was 19.6. Control strengths were concentrated near or below the threshold, with the few positive controls only slightly above it. At the prompt level, the exact 95% interval for all marked prompts detected was 98.10–100% in both models. For control families, the upper bound with zero positive prompts was 1.903%, and the interval for one positive prompt was 0.013–2.868%; these data are compatible with, but do not establish, an operational false positive rate below 1%.

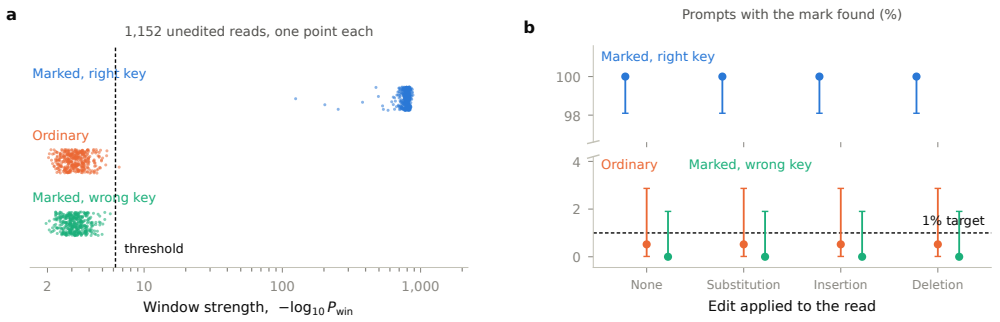
The GENERator aligned detector diagnostic provides a separate check of the analytic null under known token alignment. Across the four evaluated lengths, 2, 6, 6, and 4 ordinary output prompts were positive, versus approximately five expected at each length ($P = 0.25$–0.87). This result is compatible with the analytic null, but it does not replace the full search correction used for the position independent decision.

4.4 DETECTION AFTER SINGLE BASE EDITS

The edit experiment evaluates whether detection persists and how the search responds to disrupted token alignment. Every marked read remained detectable after the tested substitution, insertion, and deletion in both models (Table 1). A substitution perturbs token content and nearby watermark contexts while preserving downstream alignment; an insertion or deletion shifts the grouping of every subsequent token in a window spanning the edit.

The distribution of strongest window lengths shows a consistent pattern across models (Figure 4). For unedited reads, the full 3,072 base window carried the strongest evidence in 383 of 384 Carbon reads and 381 of 384 GENERator reads. After insertion, those counts fell to 203 and 200; after deletion, they were 232 and 227. This shift toward shorter windows is consistent with the search

Carbon-500M



GENERator-v2 1.2B

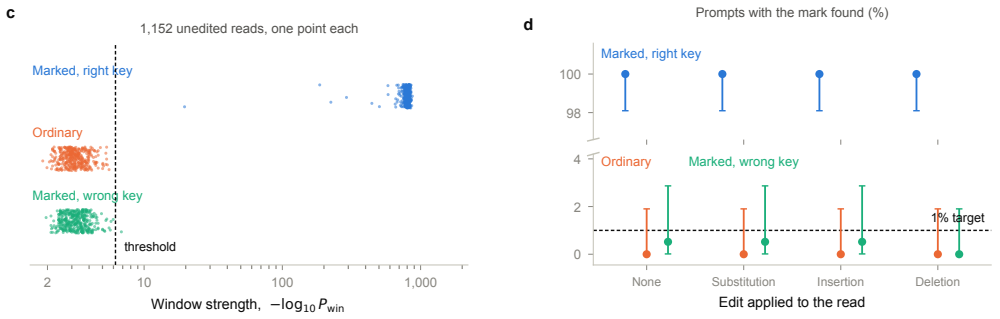


Figure 2: **Position independent detection in Carbon and GENERator. a,c**, Strength of the best window in each unedited read, one point per read, on a log scale. The dashed line denotes the threshold corrected for the search over 16,136 windows. All marked reads exceed the threshold. **b,d**, Proportion of the 192 held out prompts with positive detections, with exact 95% intervals, shown on a broken axis. All marked prompts were detected in all four edit conditions. The GENERator wrong key positive is absent after deletion. The control intervals reach above the 1% target, so these data are consistent with a false positive rate below 1% without establishing one. Panels **a,b** show Carbon and panels **c,d** show GENERator.

Table 1: Positive detections across 192 held out prompts with two draws each per model. Correct and wrong key columns refer to marked reads. Each edit affects exactly one base. Every marked prompt had both draws detected in all four conditions.

Model	Read condition	Correct key	Ordinary	Wrong key
Carbon-500M	Unedited	384/384	1/384	0/384
	One substitution	384/384	1/384	0/384
	One insertion	384/384	1/384	0/384
	One deletion	384/384	1/384	0/384
GENERator-v2 1.2B	Unedited	384/384	0/384	1/384
	One substitution	384/384	0/384	1/384
	One insertion	384/384	0/384	1/384
	One deletion	384/384	0/384	0/384

selecting regions whose alignment is preserved or recovered. It does not imply a general guarantee against more extensive or adversarial modification.

5 DISCUSSION

We evaluated the SynthID tournament watermark in Carbon-500M and GENERator-v2 1.2B using a shared prompt cohort and position independent detector. For both models, the samplers agreed with their intended distributions at the tested states, the declared quality proxies showed no statistically detectable loss, and every held out marked read was detected in the unedited and tested single base edit conditions. The verifier combines a search over strand, token phase, start position, and window length with one correction for the resulting read level decision.

Detection remained effective after correction for alignment and location uncertainty. The verifier searches both strands, every starting base, and four window lengths, and includes every tested window in the read level correction. In both models, the window level results show that the accumulated signal exceeded the corrected threshold. This margin should not be assumed for shorter continuations or longer surrounding reads: shorter generated regions provide fewer mark bits, while longer reads increase the number of candidate windows. The minimum detectable continuation length under these conditions remains to be established.

Applying the same sampler and verifier to two models supports portability across the evaluated model implementations. The detector depends on the observed read, key, and public settings, so its statistical procedure is unchanged between Carbon and GENERator. Both models achieved complete detection in this cohort under the same canonical six mer policy. The shared prompt cohort and token length limit extrapolation to other corpora and tokenization schemes. This comparison also does not establish relative biological quality or equivalence between the models.

The strongest window results help explain how alignment search supports detection after insertions and deletions. A substitution preserves downstream token boundaries, whereas an insertion or deletion changes the alignment of every subsequent token within a window spanning the edit. The increased selection of shorter windows is consistent with the verifier selecting regions whose alignment remains usable in both Carbon-500M and GENERator-v2 1.2B. Exhaustive search over starting bases also permits recovery of the original alignment downstream of an edit. The strongest window does not identify the exact generation boundaries. This interpretation accounts for the observed single edit results but provides no general guarantee against more extensive modification. Theoretical work on watermark removal further emphasizes that robustness depends on the permitted adversarial capabilities (Zhang et al., 2024). Here, each read received exactly one randomly specified edit, without access to the verifier's score.

The alignment problem is related to synchronization errors in coding theory. Synchronization strings explicitly encode indexing information to support recovery from insertions and deletions (Haeupler & Shahrasbi, 2017). Pseudorandom codes combine computational indistinguishability with decodability under specified error models, motivating their use in watermarking (Christ & Gunn, 2024). Our verifier addresses alignment uncertainty through exhaustive search, without adding an explicit indexing code to the sequence. The cost of this search is the number of windows evaluated and the associated multiple testing correction. This analogy concerns the alignment problem; it does not imply the error correction guarantees of those coding constructions.

Watermarking designed coding sequences through synonymous substitution, including approaches that preserve the mark in translated proteins (Zhang et al., 2025), addresses a complementary objective. Such methods assume a coding region and permit explicit sequence modification, whereas our evaluation concerns detection in generated DNA without those assumptions.

Future work should evaluate the tradeoff between watermark strength, detection power, and sequence level distortion across different tournament depths in both Carbon-500M and GENERator-v2 1.2B.

6 LIMITATIONS

This evaluation covers two models, one watermark construction, and one verifier; it does not provide a general proof of watermark robustness, biological safety, or secret key security. The evaluation used one execution per model on a shared prompt cohort. Both models use six base tokens; single base or variable length tokenization would present different alignment requirements. The shared cohort limits extrapolation to other genomic sources, and longer reads would increase the search correction. Generation ran on GPUs and detection on x86-64 CPUs, without independent replication across computing platforms. For Carbon, the retained protocol document differs from the hash recorded in the detection result. This unresolved provenance discrepancy limits verification of that execution.

The observed control counts are compatible with the 1% target but do not establish an operational false positive rate below it. With 192 independent prompts, the 95% interval reaches 2.87% for one positive prompt; the upper bound for zero positives is 1.90%. More independent prompts are needed for precise estimates near the target.

Finally, the experiments use published keys for reproducibility and therefore do not evaluate secrecy, resistance to key recovery, or adversarial removal in deployment. The analytic null assumes independent fair mark bits without the correct key. Whether attacks reported for text watermarking (Reynolds et al., 2025; Jovanović et al., 2024) transfer to this genomic setting is untested; the assumed attacker does not hold the key or observe the verifier's score. Likewise, model likelihood and sequence statistics are biological proxies, not measures of expression, replication, function, viability, or safety. Assessing biological consequences would require experimental assays.

7 CONCLUSION

The SynthID tournament watermark produced no statistically detectable loss in the evaluated model likelihood and sequence quality proxies for Carbon-500M or GENERator-v2 1.2B. With read level correction over both strands, every starting base, and four window lengths, the position independent verifier detected all 384 held out marked reads per model, unedited and after each tested single base substitution, insertion, or deletion. Control counts were compatible with the declared false positive target, but 192 independent prompts limit precision near 1%. In both models, strongest window distributions are consistent with the search selecting regions with preserved or recovered token alignment after insertions and deletions. These findings support statistical provenance detection within the evaluated scope. Broader conclusions require independent replication, wider sequence coverage, and biological validation.

AI USE STATEMENT

Generative AI tools assisted with method implementation, experiment design, mathematical reasoning and proofs, data preparation, and interpretation of results. They also assisted with drafting and revising the manuscript, articulating the verification setting and claim scope, and preparing the LaTeX submission. Carbon-500M and GENERator-v2 1.2B generated the DNA continuations studied in the experiments, as described in Section 3. The authors are responsible for the final text, claims, and artifacts.

ETHICS STATEMENT

The prompt cohort uses public nonhuman RefSeq records and excludes patient and controlled access sequences (Section 3.4). The experiments assess statistical detection and sequence proxies; they do not establish biological function, viability, safety, or sequence authenticity. The experimental keys are public reproducibility fixtures and provide no evidence of secrecy or resistance to key recovery. A detected watermark is evidence under the specified statistical model, not a certificate of biological origin or safe use.

REPRODUCIBILITY STATEMENT

Section 3 specifies the canonical token policy, tournament update, detector search and correction, prompt construction, and statistical comparisons. The figures and tables report both models separately under the same evaluation design. Section 6 records the execution scope, outstanding independent replication, and unresolved Carbon protocol hash discrepancy. The experimental keys are public fixtures rather than deployment secrets.

REFERENCES

Scott Aaronson and Hendrik Kirchner. Watermarking GPT outputs. Talk slides, 2022. `https://www.scottaaronson.com/talks/watermark.ppt`; the source usually credited for exponential-minimum, or Gumbel, keyed sampling.

David Baker and George Church. Protein design meets biosecurity. *Science*, 383(6681):349, 2024. doi: 10.1126/science.ado1671.

Loubna Ben Allal, Qiuyi Li, Maurizio Fiusco, Lewis Tunstall, Kashif Rasul, Edward Beeching, Dana Aubakirova, Carlos Patiño, Thibaud Frere, Anton Lozhkov, Georgia Channing, Thomas Wolf, Diego di Bernardo, and Leandro von Werra. Carbon: decoding the language of life. *bioRxiv*, 2026. doi: 10.64898/2026.05.22.727119. Posted 25 May 2026.

Yanshuo Chen, Zhengmian Hu, Yihan Wu, Ruibo Chen, Yongrui Jin, Marcus Zhan, Chengjin Xie, Wei Chen, and Heng Huang. Enhancing privacy in biosecurity with watermarked protein design. *Bioinformatics*, 41(7):btaf141, 2025. doi: 10.1093/bioinformatics/btaf141.

Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In *Advances in Cryptology — CRYPTO 2024*, 2024. doi: 10.1007/978-3-031-68391-6_10. IACR ePrint 2024/235; robust to substitutions and deletions.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *Proceedings of the 37th Annual Conference on Learning Theory (COLT)*, volume 247 of *PMLR*, pp. 1125–1139, 2024. arXiv:2306.09194.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob Chen, Nenad Kushman, Ilia Shumailov, Eugene Tarassov, Pushmeet Kohli, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024. doi: 10.1038/s41586-024-08025-4.

Abdulrahman Diaa, Toluwani Aremu, and Nils Lukas. Optimizing adaptive attacks against watermarks for language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *PMLR*, pp. 13546–13569, 2025.

GenerTeam. GENERator-v2-eukaryote-1.2b-base model card and released modeling code. https://huggingface.co/GenerTeam/GENERator-v2-eukaryote-1.2b-base, 2026. Revision c41b0018; modeling_generator.py defines _BPLogitsProcessor, which samples each of six bases independently inside generate().

Daniel G. Gibson, John I. Glass, Carole Lartigue, Vladimir N. Noskov, Ray-Yuan Chuang, Mikkel A. Algire, Gwynedd A. Benders, Michael G. Montague, Li Ma, et al. Creation of a bacterial cell controlled by a chemically synthesized genome. *Science*, 329(5987):52–56, 2010. doi: 10.1126/science.1190719.

Google DeepMind. SynthID-Text reference implementation. https://github.com/google-deepmind/synthid-text, 2024. Apache-2.0; revision addb4a15. Source of the tournament formula, default layer depth, context width, and repeated-context masking used here.

Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings: Codes for insertions and deletions approaching the singleton bound. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2017. doi: 10.1145/3055399.3055498.

Dominik Heider and Angelika Barnekow. DNA-based watermarks using the DNA-Crypt algorithm. *BMC Bioinformatics*, 8:176, 2007. doi: 10.1186/1471-2105-8-176.

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.

HuggingFaceBio. Carbon-500M model card. https://huggingface.co/HuggingFaceBio/Carbon-500M, 2026. Main revision 9796b752 is a standard causal LM; the fns revision 974bc8a5 adds base-pair-level generation and scoring.

Nikola Jovanović, Robin Staab, and Martin Vechev. Watermark stealing in large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, 2024.

Samuel H. King, Claudia L. Driscoll, David B. Li, Daniel Guo, Aditi T. Merchant, Garyk Brixi, Max E. Wilkinson, and Brian L. Hie. Generative design of bacteriophages with genome language models. *Science*, 393:eaec2657, 2026. doi: 10.1126/science.aec2657. URL https://pubmed.ncbi.nlm.nih.gov/42561074/. PMID: 42561074.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, pp. 17061–17084, 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html. arXiv:2301.10226.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=FpaCL1MO2C. arXiv:2307.15593.

Qiuyi Li, Zhihao Zhan, Shikun Feng, Yiheng Zhu, Yuan He, Wei Wu, Zhenghang Shi, Shengjie Wang, Zongyong Hu, Zhao Yang, Jiaoyang Li, Jian Tang, Haiguang Liu, and Tao Qin. GENERator-v2: Reconciling coarse tokenization with single-nucleotide resolution in genomic language modeling. *bioRxiv*, 2026. doi: 10.64898/2026.01.27.702015. Version 2.

Michael Liss, Daniela Daubert, Kathrin Brunner, Kristina Kliche, Ulrich Hammes, Andreas Leiherer, and Ralf Wagner. Embedding permanent watermarks in synthetic genes. *PLoS ONE*, 7(8): e42465, 2012. doi: 10.1371/journal.pone.0042465.

Eric Nguyen, Michael Poli, Matthew G. Durrant, Brian Kang, Dhruva Katrekar, David B. Li, Liam J. Bartie, Armin W. Thomas, Samuel H. King, Garyk Brixi, et al. Sequence modeling and design from molecular to genome scale with Evo. *Science*, 386(6723):eado9336, 2024. doi: 10.1126/science.ado9336.

Shayleen Reynolds, Hengzhi He, Dung Daniel T. Ngo, Saheed Obitayo, Niccolò Dalmasso, Guang Cheng, Vamsi K. Potluru, and Manuela Veloso. Breaking distortion-free watermarks in large language models, 2025.

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*, 2025.

Wei Wu, Qiuyi Li, Yuanyuan Zhang, Zhihao Zhan, Ruipu Chen, Mingyang Li, Kun Fu, Junyan Qi, Yongzhou Bao, Chao Wang, et al. GENERator: A long-context generative genomic foundation model. *arXiv preprint arXiv:2502.07272*, 2025.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. A resilient and accessible distribution-preserving watermark for large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, 2024.

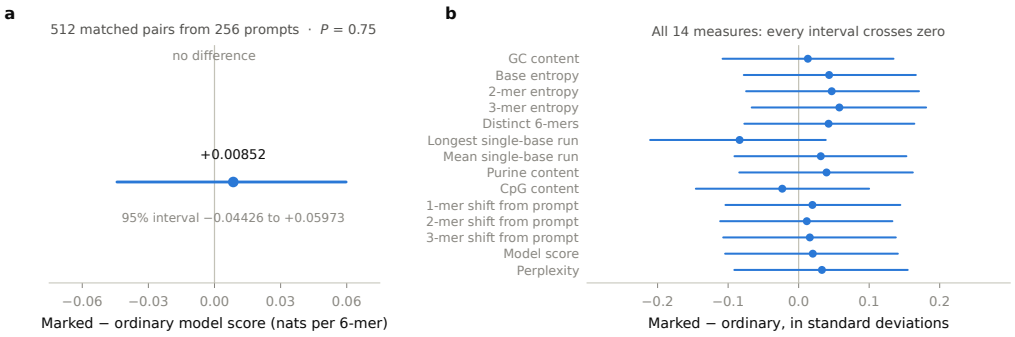
Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for generative models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pp. 58851–58880, 2024. arXiv:2311.04378.

Zaixi Zhang, Ruofan Jin, Le Cong, and Mengdi Wang. Securing the language of life: Inheritable watermarks from DNA language models to proteins. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2509.18207.

A ADDITIONAL SEQUENCE AND DETECTION DIAGNOSTICS

These figures provide the full sequence measure comparisons and the distribution of strongest window lengths for both models. The main text reports the likelihood differences, detection counts, uncertainty, and single edit outcomes.

Carbon-500M



GENERator-v2 1.2B

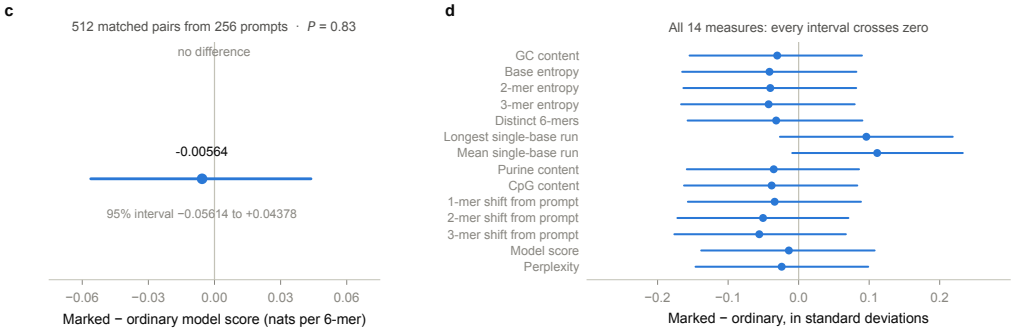


Figure 3: **Model likelihood and sequence quality proxies in Carbon and GENERator. a,c**, Paired difference in the model's score between marked and ordinary continuations, over 512 matched pairs from 256 prompts. Positive values indicate lower model likelihood for marked continuations. The horizontal bar denotes the 95% interval from a bootstrap over prompts. **b,d**, Paired differences for all 14 prespecified measures, expressed in standard deviations, with confidence intervals in the same units. Every interval includes zero and none of the 14 comparisons is significant after multiple testing correction. Panels **a,b** show Carbon and panels **c,d** show GENERator.

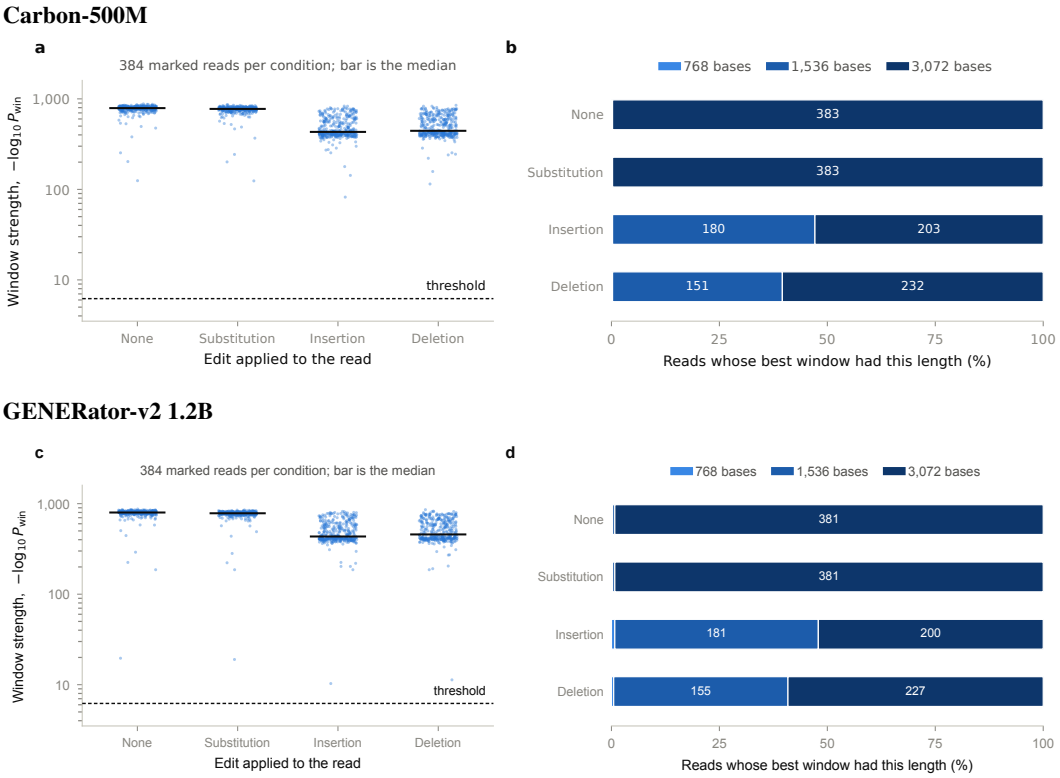


Figure 4: **Effects of single base edits on detection in Carbon and GENERator. a,c**, Strength of the best window in each of the 384 marked reads per condition, on a log scale, with the median shown as a bar. The dashed line is the detection threshold. **b,d**, Length of the strongest window. Numbers inside the bars are read counts out of 384. The full 3,072 base window is strongest for nearly all unedited and substituted reads, whereas shorter windows are more frequent after insertion or deletion. Panels **a,b** show Carbon and panels **c,d** show GENERator.