CERTMARK: DISTORTION-FREE MULTI-BIT WATERMARKING WITH CERTIFIED DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Leading multi-bit watermarking methods for language models encode messages by biasing the model's next-token probabilities, creating a trade-off between message recovery and text quality. Their decoders typically return the highest-scoring candidate from accumulated token-level evidence, without a certified abstention rule that bounds the probability of outputting an incorrect message. We introduce CertMark, a distribution-preserving multi-bit watermark with certified decoding. Rather than modifying probabilities, CertMark uses the embedded message to seed an exact Gumbel-max sampler, thereby preserving the model's original sampling distribution. We propose two scalable decoders: a model-agnostic, text-only decoder and a model-aware variant that leverages the original next-token distributions for stronger recovery. Both support certified abstention with mathematical bounds on the probability of returning an incorrect message. Across text completion, summarization, and story generation, CertMark matches the perplexity of unwatermarked text while reliably recovering multi-bit messages. The model-aware decoder further achieves higher bit accuracy than probability-biasing baselines.

1 INTRODUCTION

Multi-bit watermarks let the provider of a language model attach a payload to text generation, such as a user identifier or a model version, and recover it later from the text alone (Wang et al., 2024; Yoo et al., 2024; Xu et al., 2026). The most prominent schemes (Xu et al., 2026) encode the payload by biasing the model's next-token probabilities toward a message-dependent part of the vocabulary, the multi-bit form of the green-list watermark (Kirchenbauer et al., 2023; Yoo et al., 2024; Xu et al., 2026). The bias introduced into the distribution encodes the watermarking signal. Therefore, watermarking recovery and text quality compete against each other (Kirchenbauer et al., 2023; 2024; Hu et al., 2024; Giboulot & Furon, 2024). Additionally, the watermarking decoder returns a vote with no bound on the probability that the recovered message is wrong (Yoo et al., 2024; Wang et al., 2024; Xu et al., 2026). Alternatively, distortion-free sampling methods do not decrease text quality. They are well established for deciding whether a text is watermarked at all (Kuditipudi et al., 2024; Christ et al., 2024; Hu et al., 2024; Wu et al., 2024), and has been extended to the multi-bit case (Jiang et al., 2025; 2026; Feng et al., 2025) but does not come with a certificate for recovering the message and recovers messages less accurately.

CertMark is the first method that is distortion-free, multi-bit and comes with certificates on message recovery. Our method encodes messages into the seeds for the random sampling during autoregressive decoding. Specifically, the message seeds the random draws of a Gumbel-max sampler (Maddison et al., 2014), every token is still sampled from the deployed model's own distribution, and a text produced this way is, to anyone without the key, distributed exactly like unwatermarked text, the guarantee that distortion-free zero-bit schemes provide (Kuditipudi et al., 2024; Christ et al., 2024). The key holder, in turn, can replay the draws for every candidate message. A wrong candidate scores like pure chance whatever the text, so the decoder can attach to each recovered message an exact bound on the probability that it is wrong, and abstain when the evidence is too thin. When the detector can also run the model, it weighs each token by how surprising the model found it (Lu et al., 2024; Lee et al., 2024) and recovers the payload where text alone cannot.

Our results are twofold. The distortion-free, chance-level and certificate statements are theorems under the standard idealisation of the pseudorandom function as a random function (Christ et al.,

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.

2024; Golowich & Moitra, 2024). The empirical claims come from the XMark benchmark (Xu et al., 2026), text completion on C4 (Raffel et al., 2020), summarization on CNN/DailyMail (Hermann et al., 2015) and story generation on WritingPrompts (Fan et al., 2018) with Qwen3.5-4B and Llama-3.1-8B, together with machine translation, long messages, a copy-paste attack (Kirchenbauer et al., 2024; Xu et al., 2026), a runtime comparison and a direct check of the certificate.

In summary, our contributions are as follows:

i) A distortion-free multi-bit watermark whose text-only decoder comes with an exact certificate on the probability of returning a wrong message, valid for any model, prompt and text.

ii) A model-aware decoder that reads the same generations through the deployed sampler and remains certified and robust when part of the text has been edited.

iii) An evaluation on two model families in which CertMark keeps text quality at the unwatermarked level, recovers payloads at or above every baseline once the model is available, has the fastest text-only decoder, and whose certificate holds empirically.

2 Related Work

Zero-bit watermarking. Zero-bit watermarks detect whether a text is marked but carry no payload. Schemes either bias the next-token distribution toward keyed tokens (Kirchenbauer et al., 2023; Fernandez et al., 2023; Zhao et al., 2024; Kirchenbauer et al., 2024; Liu et al., 2024a;b; Liu & Bu, 2024; Chang et al., 2024), or control sampling while preserving the unwatermarked law (Kuditipudi et al., 2024; Hu et al., 2024; Wu et al., 2024; Christ et al., 2024; Dathathri et al., 2024; Zhao et al., 2025; He et al., 2025; Chen et al., 2025). Token-level signals remain vulnerable to strong rewriting and paraphrasing (Krishna et al., 2023; Zhang et al., 2024; Pang et al., 2024; Jovanović et al., 2024).

Multi-bit watermarking. Multi-bit schemes carry an identifier or metadata rather than a single presence bit. Most use message-conditioned logit shifts, vocabulary partitions, or blockwise allocation, trading payload recovery against decoding cost, robustness, and text quality (Fernandez et al., 2023; Wang et al., 2024; Li et al., 2024; Yoo et al., 2024; Qu et al., 2025; Xu et al., 2025a;b; 2026; Kim et al., 2026). Distribution-preserving alternatives use keyed reweighting or place the message in the sampler's randomness (Jiang et al., 2025; Feng et al., 2025; Boroujeny et al., 2024; Jiang et al., 2026). All return the highest-scoring message with no bound on the probability that it is wrong. CertMark instead uses the payload to seed an exact Gumbel-max draw, couples its text-only decoder to an exact wrong-message null, and optionally uses the model to decode the same generations more accurately.

3 Method

Table 1 collects the notation used in this section.

Preliminaries Write V for the vocabulary, x for the prompt, $y_{<t}$ for the tokens emitted before step t and $p_t(\cdot) = p(\cdot \mid x, y_{<t})$ for the law the deployed sampler actually draws y_t from, after temperature and truncation. A multi-bit watermark embeds a message $m \in \{0,1\}^L$, such as a user identifier, into the generated text so that the holder of a secret key κ can recover it from the text alone (Wang et al., 2024; Yoo et al., 2024; Xu et al., 2026).

Three properties are wanted of such a scheme at once, for a message m, a key κ and the generated text y:

$$\underbrace{\mathcal{L}(y \mid m, \kappa) = \textstyle\prod_t p_t}_{\text{(i) distortion-free}}, \qquad \underbrace{\hat{m}(y, \kappa) = m}_{\text{(ii) recoverable}}, \qquad \underbrace{\Pr[\hat{m} \neq m] \leq \sigma}_{\text{(iii) certified}}. \tag{1}$$

Most existing schemes maximise (ii) alone: green-list and other logit bias methods alters the sampling law at every step, so (i) fails. Typically, the decoder returns the highest-scoring message with no bound of the form (iii). The schemes that do satisfy (i), by reweighting the law so that its mean over the key is unchanged or by moving the message into the sampling randomness (Jiang et al., 2025; Feng et al., 2025; Jiang et al., 2026), still return a vote with no bound of the form (iii). Appendix F compares them with CertMark. In this work we develop CertMark, a distortion-free multi-bit watermark that attains (i)–(iii) that lets the message select the sampler's randomness rather than its law.

Table 1: Notation used in the method.

Model and text		*Encoder*	
V	vocabulary	h	context length in tokens
x	prompt	$c_t = y_{t-h:t-1}$	context of step t
$y_{1:n}$, y_t, $y_{<t}$	text, its t-th token, its prefix	i_t	chunk carried at step t
$Y_{1:n}$	emitted text as a random variable	$\mathcal{C}_t$	contexts used so far
$p_t(\cdot) = p(\cdot \mid x, y_{<t})$	deployed sampler's law at step t	a_t	argument tuple of step t
Key and randomness		$u_t(v)$	uniform attached to token v at step t
$\kappa \in \mathcal{S}$	secret key, $\mathcal{S} = \{0,1\}^{64}$	$S(v)$	Gumbel-max score of token v
PRF_κ, $\langle\cdot\rangle$	keyed PRF, its argument encoding	$\mathcal{E}$	encoder as a deterministic map
$\mathcal{X}$	typed PRF arguments	*Decoder and certificate*	
$\rho\colon \mathcal{S} \to (0,1)$	seed to a grid point in $(0,1)$	$\mathcal{T}$	visited positions (first occurrences)
chunk, msg, fresh	domain-separating tags	$\mathcal{T}_c$, $n_c = \lvert\mathcal{T}_c\rvert$	positions of chunk c, their count
$\Pr_\kappa$	probability over the idealised key	$\{\mathcal{T}_c\}_{c=1}^{C}$	scored positions of all chunks
Message		$u_t^{(m')}(v)$	uniform under candidate m'
$m \in \{0,1\}^L$	message of L bits	$S_c(m')$	text-only score of candidate m'
$m = (m_1, \ldots, m_C)$	its C chunks, $m_c \in \{0,1\}^{k_c}$	$\hat{\delta}_c$, δ	chunk certificate, chosen level
m'	candidate value of a chunk	D_δ, $\perp$	certified decoder, abstention
$\hat{m}_c$, $\hat{m}$	decoded chunk, decoded message	H_1, H_0	hypotheses $m_c = m'$ and $m_c \neq m'$
σ	error budget in (1)		

3.1 Encoding and Model-Agnostic Decoding

CertMark uses the key κ and message $m \in \{0,1\}^L$ to determine the sampler's randomness and lets p_t unchanged. A key holder can reconstruct this randomness, while anyone without the key sees an ordinary sample from p_t. The following identity, the Gumbel-max trick (Gumbel, 1954; Maddison et al., 2014), makes this possible.

Lemma 1 (Gumbel-max sampling)**.** *Fix a step t and recall that p_t is the sampler's law over V at that step, redrawn at every step. Draw $u(v) \sim U(0,1)$ i.i.d. over $v \in V$, form the Gumbel variates and the scores*

$$S(v) \;=\; \log p_t(v) - \log\big(-\log u(v)\big), \tag{2}$$

and emit $y = \arg\max_{v \in V} S(v)$. Then $\Pr[y = v] = p_t(v)$ for every $v \in V$.

The lemma separates the sampling law from the realized draw: p_t fixes the law, while u_t selects the sample. We generate u_t pseudorandomly from (κ, m), so the encoder and detector can reproduce it without changing p_t. Each u_t must remain i.i.d. uniform conditional on the prefix and must be fresh across steps. The pseudorandom function and freshness rule below ensure this. The lemma and chain rule then yield Theorem 1. The decoder uses the same randomness to distinguish the correct message from wrong candidates.

Pseudorandom function. As is standard in multi-bit watermarking, the sampling randomness comes from a pseudorandom function $\mathrm{PRF}_\kappa : \{0,1\}^* \to \mathcal{S}$, $\mathcal{S} = \{0,1\}^{64}$, keyed by the secret key κ (Goldreich et al., 1986). It is deterministic, so anyone holding κ can recompute any of its values, while without the key its values at distinct inputs are computationally indistinguishable from independent uniform elements of $\mathcal{S}$. We evaluate it at tuples of typed arguments from $\mathcal{X}$ (tags, token sequences and integers), mapped to a single input by an injective, prefix-free encoding, $\mathrm{PRF}_\kappa(x_1, \ldots, x_r) := \mathrm{PRF}_\kappa(\langle x_1, \ldots, x_r\rangle)$, so distinct tuples are distinct inputs, and the tags chunk, msg and fresh separate its three uses. An output $z \in \mathcal{S}$, read as an unsigned integer, becomes a number in $(0,1)$ through $\rho(z) = (\lfloor z/2^{11}\rfloor + \frac{1}{2})\,2^{-53}$, which is uniform on the 53-bit midpoint grid of $(0,1)$ when z is uniform. Throughout the analysis we use the standard random-function idealisation: PRF_κ is treated as a uniformly random function, so the values $\rho(\mathrm{PRF}_\kappa(a))$ at distinct inputs a are independent $U(0,1)$ variables. A uniform is therefore fresh exactly when its argument tuple has not been queried before, which the encoder below guarantees. Appendix I gives our instantiation.

Encoder Rather than encoding the full message m at once, the encoder divides it into chunks: for $m \in \{0,1\}^L$, we write $m = (m_1, \ldots, m_C)$ with $m_c \in \{0,1\}^{k_c}$ and $\sum_{c=1}^{C} k_c = L$. When a previously unseen context appears, the encoder uses that step to encode one selected chunk.

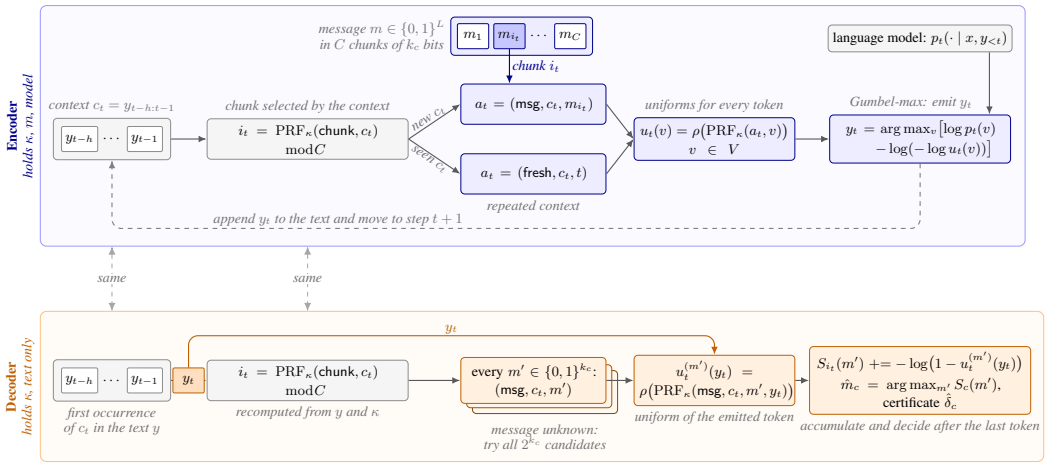


Figure 1: One step of CertMark. The encoder (top) hashes the context to pick a chunk, reads the uniforms at an argument tuple carrying that chunk's value, or at a fresh position tuple when the context recurs, and emits the Gumbel-max token. The decoder (bottom) recomputes context and chunk from text and key alone, tries every value of the chunk, and accumulates $-\log(1-u)$ over the emitted tokens.

Ordinary sampling turns a prefix into $p_t(\cdot|x, y_{<t})$ and then draws $y_t \sim p_t$ using fresh randomness. Below we detail our new encoder

$$\mathcal{E} : (y_{<t},\, p_t,\, \kappa,\, m) \longmapsto y_t\,. \tag{3}$$

$\mathcal{E}$ consumes no randomness of its own but replaces it deterministically by the message m and key κ, yet, by Lemma 1, emits $y_t \sim p_t$.

At step t the model defines the next-token distribution $p_t(v)$ of Section 3, and we take as context the last h emitted tokens, $c_t = y_{t-h:t-1}$. The context alone selects the message chunk,

$$i_t \;=\; \mathrm{PRF}_\kappa(\mathsf{chunk}, c_t) \bmod C, \tag{4}$$

and that chunk's value then enters the arguments at which the uniforms are read.

Lemma 1 requires the uniforms at step t to be new: were u_t to repeat a vector already used in this generation, it would no longer be independent of the prefix. A context occurring for the first time guarantees this, because the tuple $(\mathsf{msg}, c_t, m_{i_t})$ is then one the encoder has never queried. A context that recurs does not, so at those steps the encoder replaces the message by the position t, which is new by construction. Writing $\mathcal{C}_{t-1}$ for the contexts already used in this generation, the step reads its uniforms at the argument tuple a_t given by

$$a_t = \begin{cases} (\mathsf{fresh},\, c_t,\, t), & c_t \in \mathcal{C}_{t-1}, \\ (\mathsf{msg},\, c_t,\, m_{i_t}), & \text{otherwise, and } \mathcal{C}_t = \mathcal{C}_{t-1} \cup \{c_t\}, \end{cases} \qquad u_t(v) = \rho\big(\mathrm{PRF}_\kappa(a_t, v)\big), \tag{5}$$

and the encoder emits the token that Lemma 1 produces from these uniforms,

$$y_t \;=\; \arg\max_{v \in V} \Big[\log p_t(v) - \log\big(-\log u_t(v)\big)\Big]. \tag{6}$$

Thus, a repeated context encodes no message chunk. Figure 1 summarises the encoder and, aligned beneath it, the decoder of the next paragraph. Appendix H gives both as pseudocode.

The key property of the encoder is that it changes the source of the sampling randomness without changing the distribution of the generated text. Because each uniform vector is independent of the prefix, Lemma 1 applies at every step and gives the same sequence distribution as unwatermarked LLM generation, as formalized next.

Theorem 1 (Distortion-free generation)**.** *Treat* PRF *as a random function and draw the key* κ *uniformly from* $\mathcal{S}$*. Let* $Y_{1:n}$ *be the sequence emitted by* (5)–(6) *on prompt* x *carrying message* $m \in \{0,1\}^L$*. Then for every such* m *and* x *and every* $w \in V^n$*,*

$$\Pr_\kappa\big[Y_{1:n} = w\big] \;=\; \prod_{t=1}^{n} p\big(w_t \mid x, w_{<t}\big), \tag{7}$$

which is the law of unwatermarked sampling and, in particular, does not depend on m.

Proof. The proof can be found in Appendix K.

The theorem shows that, over an unknown random key, CertMark generates the same sequence distribution as the unwatermarked LLM, independently of the encoded message. As a corollary, a reader without the key cannot distinguish the two distributions from the text alone.

Remark (Deployment model)**.** *Theorem 1 is a single-generation guarantee. In our experiments, as is standard for PRF-seeded distortion-free samplers (Kuditipudi et al., 2024; Dathathri et al., 2024), we reuse one key. The encoder is therefore deterministic in the prompt and message: repeated prompts reproduce the same text, and shared contexts reuse the same uniforms. Each generation still has the exact unwatermarked marginal law, but generations need not be independent (Fu et al., 2024; Wu et al., 2025; Gloaguen et al., 2025). Theoretically, Theorem 8 in Appendix R resolves this by adding a fresh r-bit nonce to every encoder argument. We leave its evaluation to future work.*

Model-Agnostic Decoder The decoder holds only the text and κ, the same access every baseline detector has. It re-tokenises the text and, with $c_t = y_{t-h:t-1}$ as in the encoder, visits the positions whose context occurs for the first time *in that text*,

$$\mathcal{T} \;=\; \big\{\, t \in \{h+1,\dots,n\} :\; c_t \neq c_s \text{ for all } h+1 \le s < t \,\big\}, \tag{8}$$

recomputing i_t at each $t \in \mathcal{T}$ exactly as the encoder did. Positions $t \le h$ have no full context and a repeated context was read under the tag fresh by (5), so neither carries a message and both are skipped. Writing $\mathcal{T}_c = \{t \in \mathcal{T} : i_t = c\}$ for the positions so assigned to chunk c and $n_c = |\mathcal{T}_c|$, it scores every candidate value $m' \in \{0,1\}^{k_c}$ of that chunk by

$$S_c(m') \;=\; \sum_{t \in \mathcal{T}_c} -\log\big(1 - u_t^{(m')}(y_t)\big), \qquad u_t^{(m')}(v) = \rho\big(\mathrm{PRF}_\kappa(\mathsf{msg}, c_t, m', v)\big), \tag{9}$$

and outputs $\hat{m}_c = \arg\max_{m'} S_c(m')$. This is a joint decision over the whole codebook of 2^{k_c} words. Chunking is what keeps this at $C \cdot 2^{k_c}$ rather than 2^L. After decoding all C chunks, we concatenate $\hat{m}_1, \dots, \hat{m}_C$ to obtain the final message $\hat{m} = (\hat{m}_1, \dots, \hat{m}_C)$.

Remark (Choice of the score)**.** *We use our specific form for scores* (9)*, because it allows us to derive that the score for wrong messages follows a Gamma distribution. Alternatively, we might use the simpler score $\sum_{t \in \mathcal{T}_c} u_t^{(m')}(y_t)$, which would, however, result in harder to handle Irwin-Hall laws (Irwin, 1927; Hall, 1927).*

3.2 Certification and Theoretical Analysis

From this point through the end of Section 3.3, we use the random-function idealisation of PRF from Section 3 (Appendix I). We fix the deployed model, prompt x, sampling settings, embedded message m, and a chunk c with $k_c \ge 1$. Unless stated otherwise, probabilities are over the idealised keyed randomness. For this fixed chunk, the decoder considers every candidate $m' \in \{0,1\}^{k_c}$, calling m_c the correct candidate and every $m' \neq m_c$ a wrong candidate. The decoder returns the candidate with the largest score. To analyze this comparison, Theorem 2 gives the score distribution of every wrong candidate, while Theorem 3 gives the per-token distributions for both cases. Its full statement in Appendix M also gives their expected scores.

Theorem 2 (Score distribution for wrong candidates)**.** *For a wrong candidate $m' \neq m_c$, conditional on the emitted text and the scored positions $\mathcal{T}_c$ selected by the chunk assignments, whenever $n_c = |\mathcal{T}_c| \ge 1$,*

$$S_c(m') \;\sim\; \Gamma(n_c, 1) \qquad \textit{exactly,} \tag{10}$$

where $\Gamma(n_c, 1)$ has shape n_c and rate 1. If $n_c = 0$, then $S_c(m') = 0$.

Proof. The proof can be found in Appendix L.

Theorem 3 (Candidate distributions)**.** *For $t \in \mathcal{T}_c$, condition on $y_{<t}$ and $p = p_t(y_t) \in (0,1]$, the sampler's probability of the emitted token. For every candidate m', its reconstructed value satisfies*

$$u_t^{(m')}(y_t) \sim \begin{cases} \mathrm{Beta}(1/p, 1), & m' = m_c \quad \textit{(correct candidate)}, \\ U(0,1), & m' \neq m_c \quad \textit{(wrong candidate)}, \end{cases} \tag{11}$$

where $u_t^{(m_c)}(y_t) = u_t(y_t)$.

Proof. The proof can be found in Appendix M.

We now describe the certified decoder, which uses the null distribution from Theorem 2 to decide whether to return the predicted chunk or abstain. An abstention means that the decoder returns no value for a particular message chunk because its certificate does not meet the chosen confidence level. Theorem 4 shows that the probability of returning an incorrect chunk is at most the chosen level δ.

Theorem 4 (Certified decoding). *For a level $\delta \in (0, 1]$, let $\hat{m}_c(y) = \arg\max_{m'} S_c(m')$ and define*

$$\hat{\delta}_c(y) \;=\; \min\Big\{1,\; \big(2^{k_c} - 1\big)\, \Pr\big[\Gamma(n_c, 1) \geq S_c(\hat{m}_c(y))\big]\Big\}, \tag{13}$$

with $\hat{\delta}_c(y) = 1$ when $n_c = 0$. Let $\perp$ denote abstention and define the certified decoder $D_\delta \colon V^n \to \{0,1\}^{k_c} \cup \{\perp\}$ by

$$D_\delta(y) \;=\; \begin{cases} \hat{m}_c(y), & \hat{\delta}_c(y) \leq \delta, \\ \perp, & \hat{\delta}_c(y) > \delta, \end{cases} \tag{14}$$

where the scores are computed from the text y and the key. Then, for the text $Y_{1:n}$ emitted by (5)–(6)*,*

$$\Pr_{\kappa}\big[D_\delta(Y_{1:n}) \notin \{m_c, \perp\}\big] \;\leq\; \delta, \tag{15}$$

Here the probability is over the idealised PRF randomness induced by the one-time draw of κ.

Proof. The proof can be found in Appendix N.

The theorem provides certified decoding because the probability of returning an incorrect chunk is at most δ, with abstention when the evidence is insufficient. As a corollary, decoding each of the C chunks at level δ/C bounds the probability of returning any incorrect chunk in the final message by δ.

Appendix O adds a payload-scaling bound, Theorem 7, under which the error of a chunk falls exponentially in the accumulated evidence and each additional bit costs about $\ln 2$ of it.

3.3 Model-Aware Decoder

A model-aware decoder also has access to the prompt and deployed model, allowing it to recompute $p_t = p_t(y_t)$ for each scored token under the complete sampling procedure. By Theorem 3, the candidate distributions in (11) have densities $f_{\text{correct}}(u \mid p) = p^{-1}u^{1/p-1}$ and $f_{\text{wrong}}(u) = 1$ on $(0, 1)$. A single scored token therefore contributes the likelihood ratio

$$f_p(u) \;=\; \frac{f_{\text{correct}}(u \mid p)}{f_{\text{wrong}}(u)} \;=\; \frac{p^{-1}u^{1/p-1}}{1} \;=\; p^{-1}u^{1/p-1}, \tag{16}$$

which is identically 1 when $p = 1$. A token, where the sampler was certain, carries no evidence. This happens for large for u near 1 when $p < 1$, so a large uniform at a token the sampler was unsure of is strong evidence for the candidate.

For the results in this subsection, we additionally condition on the emitted text and the scored-position sets $\{\mathcal{T}_c\}_{c=1}^{C}$, which fixes $p_t = p_t(y_t) \in (0, 1]$ for $t \in \mathcal{T}_c$. For a candidate m' we test the hypothesis $H_1 \colon m_c = m'$ against the alternative $H_0 \colon m_c \neq m'$.

Theorem 5 (Exact likelihood ratio). *For a fixed candidate m', the log-likelihood ratio of H_1 to H_0 based on its scored uniforms is*

$$\ell_c(m') = \sum_{t \in \mathcal{T}_c} \Big[\big(\tfrac{1}{p_t} - 1\big) \log u_t^{(m')}(y_t) + \log \tfrac{1}{p_t}\Big]. \tag{17}$$

Proof. The proof can be found in Appendix P.

Operationally, the model-aware decoder keeps the same candidate enumeration, chunk assignments, and reconstructed uniforms as the model-agnostic decoder. Instead of maximizing the text-only score $S_c(m')$ in (9), it maximizes the log-likelihood score in (17):

$$\hat{m}_c^{\text{MA}} = \arg \max_{m' \in \{0,1\}^{k_c}} \ell_c(m'). \tag{18}$$

Table 2: Main benchmark on Qwen3.5-4B: C4 text completion, CNN/DailyMail summarization and WritingPrompts story generation with $L = 8$ and token budgets $T \in \{150, 200, 250, 300\}$. Perplexity is measured under Qwen3.5-9B; the distortion columns are defined in Section 4. The four CertMark rows are the model-agnostic and the model-aware decoder, each with one 8-bit chunk and with 2-bit chunks. Colours mark the best, second-best and third-best method in each column.

Method	$T = 150$		$T = 200$		$T = 250$		$T = 300$		Avg.		Distortion (avg. over T)			
	BA↑	PPL↓	BA↑	PPL↓	BA↑	PPL↓	BA↑	PPL↓	BA↑	PPL↓	Top-1↑	Top-5↑	R-1↑	R-L↑
Text Completion														
CycleShift	100.00	7.97	100.00	7.58	100.00	7.05	100.00	7.44	100.00	7.51	55.02	84.96	0.290	0.170
DepthW	95.75	7.40	100.00	6.96	100.00	6.52	97.25	6.68	98.25	6.89	57.39	85.87	0.290	0.168
StealthInk	89.75	6.15	90.50	5.79	95.25	5.73	94.25	5.61	92.44	5.82	62.81	89.16	0.308	0.179
MPAC	97.25	7.70	98.00	7.52	98.75	7.03	99.00	7.09	98.25	7.33	55.60	85.34	0.293	0.170
RSBH	96.00	7.11	97.25	6.88	97.75	7.03	97.00	7.03	97.00	7.01	56.21	86.33	0.289	0.167
XMark	100.00	6.93	100.00	7.07	100.00	6.78	100.00	6.81	100.00	6.90	56.66	86.86	0.301	0.173
CertMark model-agnostic 8-bit	100.00	5.65	100.00	5.23	100.00	5.29	100.00	5.11	100.00	5.32	65.48	90.08	0.311	0.184
CertMark model-agnostic 2-bit	99.75	5.63	100.00	5.55	100.00	5.34	99.75	5.20	99.88	5.43	65.59	89.89	0.316	0.184
CertMark model-aware 8-bit	100.00	5.69	100.00	5.20	100.00	5.35	100.00	5.09	100.00	5.33	65.42	90.03	0.312	0.185
CertMark model-aware 2-bit	100.00	5.64	100.00	5.52	100.00	5.36	100.00	5.24	100.00	5.44	65.43	89.94	0.316	0.184
Text Summarization														
CycleShift	95.75	3.58	95.00	3.75	95.00	3.69	99.50	3.71	96.31	3.68	74.56	96.87	0.541	0.348
DepthW	74.75	3.21	75.50	3.43	85.00	3.51	87.00	3.57	80.56	3.43	76.72	97.33	0.551	0.357
StealthInk	75.25	2.95	80.75	3.10	82.25	3.11	88.00	3.14	81.56	3.08	80.59	98.14	0.562	0.372
MPAC	86.25	3.60	89.75	3.50	92.00	3.59	94.75	3.79	90.69	3.62	74.92	96.74	0.546	0.358
RSBH	78.75	3.56	88.75	3.58	96.25	3.80	93.00	3.72	89.19	3.67	74.54	97.09	0.542	0.352
XMark	91.25	3.49	93.75	3.53	97.75	3.64	99.00	3.63	95.44	3.57	74.96	97.45	0.544	0.349
CertMark model-agnostic 8-bit	97.75	3.05	98.75	3.05	98.50	3.06	100.00	3.08	98.75	3.06	81.62	98.25	0.575	0.393
CertMark model-agnostic 2-bit	87.25	2.99	94.50	3.10	95.00	3.13	97.50	3.09	93.56	3.08	81.57	98.17	0.563	0.381
CertMark model-aware 8-bit	100.00	3.03	100.00	3.07	100.00	3.10	100.00	3.10	100.00	3.07	81.50	98.24	0.574	0.392
CertMark model-aware 2-bit	98.75	3.00	99.25	3.12	99.75	3.11	99.75	3.11	99.38	3.08	81.62	98.20	0.564	0.382
Story Generation														
CycleShift	95.00	4.90	100.00	4.85	100.00	4.98	100.00	5.17	98.75	4.97	66.89	92.44	0.349	0.221
DepthW	87.25	4.33	94.75	4.45	96.50	4.79	96.75	4.50	93.81	4.52	69.21	93.35	0.360	0.232
StealthInk	82.50	3.81	84.75	3.93	89.50	4.02	92.50	3.87	87.31	3.91	73.62	95.16	0.367	0.239
MPAC	92.50	4.90	94.50	5.07	94.50	5.24	96.25	4.98	94.44	5.05	66.69	92.65	0.350	0.217
RSBH	92.25	4.64	96.25	4.66	97.25	5.07	97.25	4.98	95.75	4.84	67.08	93.41	0.349	0.219
XMark	96.75	4.72	97.25	4.54	99.50	4.90	99.75	5.00	98.31	4.79	67.55	93.56	0.350	0.224
CertMark model-agnostic 8-bit	100.00	3.63	100.00	3.71	100.00	3.81	100.00	3.87	100.00	3.75	75.89	95.54	0.362	0.230
CertMark model-agnostic 2-bit	95.75	3.71	98.50	3.75	99.25	3.81	99.75	3.82	98.31	3.77	75.81	95.56	0.366	0.236
CertMark model-aware 8-bit	100.00	3.63	100.00	3.71	100.00	3.81	100.00	3.87	100.00	3.75	75.89	95.54	0.362	0.230
CertMark model-aware 2-bit	100.00	3.71	100.00	3.75	100.00	3.81	100.00	3.82	100.00	3.77	75.81	95.56	0.366	0.236

The decoded chunks are concatenated into the final message as before.

Theorem 5 assumes that every scored token was generated by CertMark. This makes its likelihood-ratio decoder vulnerable to editing attacks that introduce foreign tokens. We therefore present the robust decoder in Theorem 6.

Theorem 6 (Robust decoding under contamination)**.** *Fix $\varepsilon \in (0, 1)$. At each scored position t, suppose independently that the true candidate's uniform follows $\mathrm{Beta}(1/p_t, 1)$ with probability $1 - \varepsilon$ and $U(0, 1)$ with probability ε, while every wrong candidate's uniform is $U(0, 1)$. Score each candidate by*

$$r_c(m') \;=\; \sum_{t \in \mathcal{T}_c} \log\Big[(1-\varepsilon)\, f_{p_t}\big(u_t^{(m')}(y_t)\big) + \varepsilon\Big], \tag{19}$$

Under a wrong candidate this score has an exact upper-tail probability $Q_{c,\varepsilon}(s) = \Pr[r_c(m') \geq s]$. The robust decoder and its certificate are

$$\hat{m}_c = \arg\max_{m'} r_c(m'), \qquad \hat{\delta}_{c,\varepsilon} = \min\big\{1, (2^{k_c} - 1) Q_{c,\varepsilon}\big(r_c(\hat{m}_c)\big)\big\}. \tag{20}$$

Returning $\hat{m}_c$ only when $\hat{\delta}_{c,\varepsilon} \leq \delta$ bounds the probability of returning an incorrect chunk by δ. Otherwise the decoder abstains.

Proof. The proof can be found in Appendix Q.

Appendix A derives and computes the null tail. Appendix H provides pseudocode for both decoders.

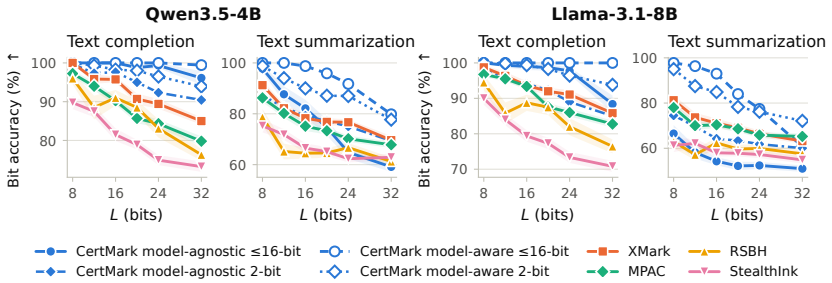


Figure 2: Bit accuracy versus message length L at $T = 150$ on text completion and summarization, one block per model. The default chunk width is the largest divisor of L that is at most 16 bits, hence $8, 12, 16, 10, 12, 16$ for $L = 8, \ldots, 32$. ± 1 standard error shaded.

Table 3: WMT14 German-to-English translation.

	Qwen3.5-4B			Llama-3.1-8B		
Method	BA↑	PPL↓	BLEU↑	BA↑	PPL↓	BLEU↑
RSBH	59.38	2.51	20.64	57.62	3.49	24.84
MPAC	70.00	2.58	20.98	64.62	3.50	26.80
XMark	74.38	2.49	20.32	71.25	3.46	25.99
StealthInk	64.88	2.22	23.00	55.75	3.19	27.08
CertMark model-agnostic 16-bit	61.50	2.18	23.15	48.88	3.08	26.34
CertMark model-agnostic 2-bit	69.88	2.22	22.41	64.00	3.24	26.82
CertMark model-aware 16-bit	85.75	2.18	23.15	88.38	3.08	26.34
CertMark model-aware 2-bit	80.50	2.22	22.41	84.75	3.24	26.82
unwatermarked	–	2.21	22.87	–	3.12	26.70

Table 4: Long C4 completion ($L = 64$).

	Qwen3.5-4B			Llama-3.1-8B		
Method	BA↑	PPL↓	BSc.↑	BA↑	PPL↓	BSc.↑
RSBH	85.03	5.38	.8208	89.25	4.63	.8341
MPAC	87.28	5.46	.8170	88.38	4.94	.8279
XMark	93.38	5.48	.8199	96.00	4.74	.8351
StealthInk	79.81	4.52	.8222	80.69	3.75	.8392
CertMark model-agnostic 16-bit	99.75	4.29	.8262	99.44	3.66	.8365
CertMark model-agnostic 2-bit	96.53	4.30	.8266	94.62	3.63	.8371
CertMark model-aware 16-bit	99.75	4.31	.8246	100.00	3.66	.8365
CertMark model-aware 2-bit	98.84	4.28	.8292	99.25	3.63	.8371
unwatermarked	–	4.22	–	–	3.57	–

4 Experiments

Baselines We compare against the multi-bit watermarking baselines **CycleShift** (Fernandez et al., 2023), **DepthW** (Li et al., 2024), **StealthInk** (Jiang et al., 2025), **MPAC** (Yoo et al., 2024), **RSBH** (Qu et al., 2025), and **XMark** (Xu et al., 2026). Appendix F adds **BiMark** (Feng et al., 2025) and **MirrorMark** (Jiang et al., 2026) and compares the distortion-free schemes on Qwen3.5-4B.

Setup We evaluate on Qwen3.5-4B and Llama-3.1-8B. Full experimental settings and hyperparameters are provided in Appendix B, and a runtime comparison in Appendix D.

Metrics **Bit accuracy** (BA, %) is the percentage of correctly decoded message bits, averaged over users (chance is 50%). **Perplexity** (PPL) is the exponentiated mean negative log-likelihood of the generated tokens under a separate oracle model of the same family; for a distortion-free scheme the reference is the unwatermarked row. **BLEU** (Papineni et al., 2002) scores translations against the WMT14 references. The distortion metrics compare the watermarked text with unwatermarked sampling: **Top-1/Top-5** is the percentage of emitted tokens among the model's 1 or 5 most likely next tokens, while **R-1/R-L** (ROUGE-1/ROUGE-L F-score, Lin, 2004) and **BERTScore** (BSc., Zhang et al., 2020) measure the similarity of each watermarked text to the unwatermarked sample from the same prompt and seed. Appendix C gives full definitions.

Main benchmark and message length. Across all three tasks, CertMark preserves unwatermarked text quality while reliably recovering payloads. Figure 2 shows that this remains true as payloads grow: the model-aware decoder is the most reliable, while smaller chunks mainly help on entropy-limited summarization. Appendix F compares other distortion-free schemes, and Appendix E confirms the same trends on Llama-3.1-8B.

Machine translation. Table 3 evaluates WMT14 German-to-English translation (Bojar et al., 2014). CertMark model-aware 16-bit gives the strongest payload recovery on both models while preserving translation quality. 2-bit chunks help the model-agnostic decoder in this low-entropy task but not the model-aware one.

Long messages and long texts. Table 4 shows that CertMark retains near-complete recovery on long C4 completions ($L = 64$, four 16-bit chunks) for both models without degrading quality. Model-aware decoding removes the remaining errors. Smaller chunks add little with ample evidence.

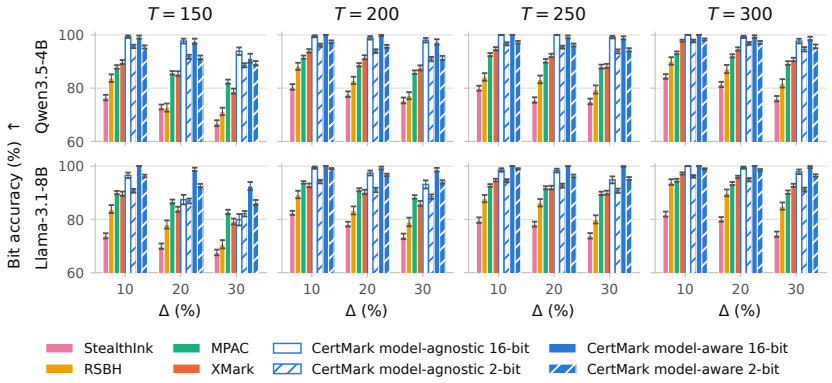


Figure 3: Copy-paste robustness on text completion with $L = 16$: bit accuracy after $\Delta\%$ of each text is replaced by unwatermarked spans, one row per model and one column per token budget T.

Table 5: Empirical check of the certificate (Theorem 4, and its model-aware form in Appendix A). Each column fixes a level δ: the decoder returns a chunk's message only if $\hat{\delta}_c \leq \delta$ and abstains ($\perp$) otherwise. Both decoders read the same saved generations of the message-length sweep ($T = 150$; $L \in \{8, \ldots, 32\}$ on completion and summarization, $L = 8$ on story generation; 50 users each), 950 watermarked chunks and 950 paired unwatermarked chunks per model. Every entry is a fraction of chunks. *Cert. error*: watermarked chunks on which a wrong message is returned; the theorem bounds its probability by δ, and with 950 chunks a single error is .0011. *Null cert. rate*: unwatermarked chunks that are nevertheless certified, which should stay near δ. *Abst.*: watermarked chunks of each task on which the decoder abstains; lower means more messages recovered with a certificate. The model-aware decoder uses the robust score with $\varepsilon = 0.1$. For every row, lower is better.

	CertMark model-agnostic							CertMark model-aware						
δ	10^{-4}	10^{-3}	0.01	0.05	0.1	0.2	0.5	10^{-4}	10^{-3}	0.01	0.05	0.1	0.2	0.5
Qwen3.5-4B														
cert. error ($\leq \delta$)	.0000	.0000	.0063	.0221	.0305	.0537	.1168	.0000	.0011	.0074	.0147	.0168	.0274	.0474
null cert. rate ($\approx \delta$)	.0021	.0063	.0137	.0621	.1200	.2147	.4074	.0000	.0000	.0074	.0442	.1000	.1895	.3863
abst., completion	.178	.129	.082	.056	.044	.027	.013	.040	.022	.009	.004	.004	.002	.000
abst., summarization	.902	.840	.727	.627	.582	.498	.324	.604	.509	.387	.251	.207	.158	.069
abst., story	.160	.060	.020	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
Llama-3.1-8B														
cert. error ($\leq \delta$)	.0000	.0000	.0032	.0211	.0463	.1053	.2126	.0000	.0011	.0042	.0116	.0189	.0411	.0821
null cert. rate ($\approx \delta$)	.0000	.0011	.0116	.0463	.0968	.1789	.3989	.0000	.0011	.0116	.0463	.0979	.1842	.3958
abst., completion	.384	.296	.196	.124	.100	.080	.047	.009	.004	.000	.000	.000	.000	.000
abst., summarization	1.000	1.000	.993	.951	.889	.771	.569	.856	.776	.662	.536	.482	.371	.238
abst., story	.980	.940	.800	.580	.500	.400	.160	.060	.000	.000	.000	.000	.000	.000

Robustness to text-editing attacks. Figure 3 evaluates copy-paste attacks across both models and token budgets. CertMark remains robust and is the strongest method overall. Appendix G extends the comparison to token substitution and deletion.

Empirical check of the certificate. Table 5 checks Theorem 4, and its model-aware form in Appendix A, on saved watermarked chunks and paired unwatermarked chunks; its caption defines each row. The certified error stays within δ up to sampling noise, and the null certification rate tracks the selected level. Relaxing the level reduces abstention but permits more erroneous and null certifications, exhibiting the expected confidence–coverage trade-off. The model-aware decoder abstains less across tasks, particularly when the available evidence is weak, showing that access to the model probabilities improves coverage while retaining the certificate.

5 Conclusion

CertMark is, to our knowledge, the first distortion-free multi-bit watermark with certified recovery. It improves bit accuracy without increasing perplexity and remains fast to encode and decode.

AI USE STATEMENT

Large language models were used for editorial support, including polishing the manuscript's writing and presentation, and for drafting implementation code. The authors made all substantive research decisions and contributed all core technical ideas.

ETHICS STATEMENT

This research adheres to the ICLR Code of Ethics. All experiments use publicly available benchmark datasets and do not involve new data collection or human-subject experiments; no personally identifiable information is intentionally used. Our method is intended to support provenance and accountability by embedding recoverable metadata in language-model outputs. Multi-bit watermarking is nevertheless dual-use: it could facilitate covert user tracking, and a compromised key or forged payload could lead to false attribution. The certified decoder bounds the probability of an incorrect recovery only under the assumptions stated in the paper and does not eliminate these deployment risks. Responsible use therefore requires secure key management, appropriate notice and consent, access controls, clear communication of abstentions and uncertainty, and safeguards against treating watermark evidence alone as conclusive in high-stakes decisions. We report our assumptions, limitations, and robustness results transparently and have taken care to follow best practices for research integrity.

REPRODUCIBILITY STATEMENT

We will make our source code publicly available upon publication to facilitate independent reproduction of our main results. The release will include the CertMark implementation integrated into the XMark evaluation harness, the scripts and configuration files for the reported experiments, the fixed keys and random seeds, and the per-run logs used to produce the tables and figures. Appendix B provides details of the models, datasets, experimental protocol, evaluation metrics, and hyperparameters, while Appendix D describes the runtime evaluation. Additional model, baseline, and robustness results appear in Appendices E–G. Appendix H gives pseudocode for the encoder, both decoders, and the certification procedures, and Appendix I specifies the pseudorandom-function instantiation. Full derivations of the theoretical results are provided in Appendix A and Appendices J–R. To the best of our knowledge, the manuscript and accompanying artifacts contain the details necessary to reproduce our results.

REFERENCES

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3302. URL https://aclanthology.org/W14-3302/.

Massieh Kordi Boroujeny, Ya Jiang, Kai Zeng, and Brian Mark. Multi-bit distortion-free watermarking for large language models. *arXiv preprint arXiv:2402.16578*, 2024.

Yapei Chang, Kalpesh Krishna, Amir Houmansadr, John Frederick Wieting, and Mohit Iyyer. PostMark: A robust blackbox watermark for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8969–8987, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.506.

Ruibo Chen, Yihan Wu, Junfeng Guo, and Heng Huang. Improved unbiased watermark for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025. URL https://aclanthology.org/2025.acl-long.1005/.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 1125–1139. PMLR, 2024.

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. doi: 10.1038/s41586-024-08025-4.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082/.

Xiaoyan Feng, He Zhang, Yanjun Zhang, Leo Yu Zhang, and Shirui Pan. BiMark: Unbiased multilayer watermarking for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 17049–17067. PMLR, 2025.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. In *2023 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6. IEEE, 2023.

Jiayi Fu, Xuandong Zhao, Ruihan Yang, Yuansen Zhang, Jiangjie Chen, and Yanghua Xiao. GumbelSoft: Diversified language model watermarking via the GumbelMax-trick. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5791–5808, 2024. URL https://aclanthology.org/2024.acl-long.315/.

Eva Giboulot and Teddy Furon. WaterMax: Breaking the LLM watermark detectability-robustness-quality trade-off. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Thibaud Gloaguen, Nikola Jovanović, Robin Staab, and Martin Vechev. Black-box detection of language model watermarks. In *The Thirteenth International Conference on Learning Representations*, 2025.

Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986. doi: 10.1145/6490.6503.

Noah Golowich and Ankur Moitra. Edit distance robust watermarks via indexing pseudorandom codes. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Emil J. Gumbel. *Statistical Theory of Extreme Values and Some Practical Applications*, volume 33 of *National Bureau of Standards Applied Mathematics Series*. U.S. Government Printing Office, 1954.

Philip Hall. The distribution of means for samples of size N drawn from a population in which the variate takes values between 0 and 1, all such values being equally probable. *Biometrika*, 19(3/4): 240–245, 1927.

Haiyun He, Yepeng Liu, Ziqiao Wang, Yongyi Mao, and Yuheng Bu. Theoretically grounded framework for LLM watermarking: A distribution-adaptive approach. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.

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*, 2024. URL https://openreview.net/forum?id=7w4RGjzd81.

Joseph O. Irwin. On the frequency distribution of the means of samples from a population having any law of frequency with finite moments, with special reference to Pearson's Type II. *Biometrika*, 19(3/4):225–239, 1927.

Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. StealthInk: A multi-bit and stealthy watermark for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27685–27709. PMLR, 2025.

Ya Jiang, Massieh Kordi Boroujeny, Surender Suresh Kumar, and Kai Zeng. MirrorMark: Generalizable mirrored sampling for multi-bit LLM watermarking. *arXiv preprint arXiv:2601.22246*, 2026.

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

Joeun Kim, HoEun Kim, Dongsup Jin, and Young-Sik Kim. Block-wise codeword embedding for reliable multi-bit text watermarking. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

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*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 2023.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DEJIDCmWOz.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. Who wrote this code? Watermarking for code generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4890–4911, 2024. URL https://aclanthology.org/2024.acl-long.268/.

Liying Li, Yihan Bai, and Minhao Cheng. Where am I from? identifying origin of LLM-generated content. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12218–12229. Association for Computational Linguistics, 2024.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Aiwei Liu, Leyi Pan, Xuming Hu, Shuang Li, Lijie Wen, Irwin King, and Philip S. Yu. An unforgeable publicly verifiable watermark for large language models. In *The Twelfth International Conference on Learning Representations*, 2024a.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. A semantic invariant robust watermark for large language models. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=6p8lpe4MNf.

Yepeng Liu and Yuheng Bu. Adaptive text watermark for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. An entropy-based text watermarking detection method. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11724–11735, 2024. URL https://aclanthology.org/2024.acl-long.630/.

Chris J. Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Qi Pang, Shengyuan Hu, Wenting Zheng, and Virginia Smith. No free lunch in LLM watermarking: Trade-offs in watermarking design choices. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. URL https://aclanthology.org/P02-1040/.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. URL https://aclanthology.org/W18-6319/.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. Provably robust multi-bit watermarking for AI-generated text. In *34th USENIX Security Symposium*, pp. 201–220. USENIX Association, 2025.

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. URL https://www.jmlr.org/papers/v21/20-074.html.

Guy L. Steele Jr., Doug Lea, and Christine H. Flood. Fast splittable pseudorandom number generators. In *Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications (OOPSLA)*, pp. 453–472. ACM, 2014.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. Towards codable watermarking for injecting multi-bits information to LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JYu5Flqm9D.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53443–53470. PMLR, 2024.

Yihan Wu, Ruibo Chen, Zhengmian Hu, Yanshuo Chen, Junfeng Guo, Hongyang Zhang, and Heng Huang. Distortion-free watermarks are not truly distortion-free under watermark key collisions. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jiahao Xu, Rui Hu, and Zikai Zhang. Majority bit-aware watermarking for large language models. *arXiv preprint arXiv:2508.03829*, 2025a.

Jiahao Xu, Rui Hu, Olivera Kotevska, and Zikai Zhang. XMark: Reliable multi-bit watermarking for LLM-generated texts. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14747–14763, San Diego, California, United States, July 2026. Association for Computational Linguistics. URL https://aclanthology.org/2026.acl-long.672/.

Xiaojun Xu, Jinghan Jia, Yuanshun Yao, Yang Liu, and Hang Li. Robust multi-bit text watermark with LLM-based paraphrasers. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025b.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4031–4055, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

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*, 2024.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

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

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. Permute-and-flip: An optimally stable and watermarkable decoder for LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025.

A THE CERTIFICATE OF THE MODEL-AWARE DECODER

The certificate of Theorem 4 rests on one fact: under a wrong candidate the scored uniforms are independent $U(0,1)$ variables whatever the text, so the text-only score has the null law $\Gamma(n_c, 1)$ and the certificate is the number of wrong candidates times the Gamma tail at the observed score. The model-aware decoder changes the per-token score, not this fact. Its certificate therefore has the same form, with the Gamma tail replaced by the null tail of the new score, and it needs the same two extra inputs as the decoder itself, the prompt and the deployed sampler. This appendix spells the construction out in equations; Appendices P and Q hold the proofs.

Throughout, fix a chunk c and condition on the text y, the scored-position sets $\mathbf{T}$, the prompt x and the model. This fixes $p_t = p(y_t \mid x, y_{<t}) \in (0,1]$ for $t \in T_c$ and, with it, the per-token likelihood ratio $f_p(u) = p^{-1}u^{1/p-1}$ of (16).

Score. The model-aware score (17) splits into a candidate-independent constant and a weighted sum of log-uniforms,

$$\begin{aligned} \ell_c(m') &= A_c + R_{c,w}(m'), \qquad w_t = \frac{1}{p_t} - 1 \geq 0, \\ A_c &= \sum_{t \in T_c} \log \frac{1}{p_t}, \qquad R_{c,w}(m') = \sum_{t \in T_c} w_t \log u_t^{(m')}(y_t). \end{aligned} \tag{21}$$

Since A_c is the same for every candidate, $\hat{m}_c = \arg\max_{m'} \ell_c(m') = \arg\max_{m'} R_{c,w}(m')$, and the certificate is built from $R_{c,w}$ alone. A token the sampler was certain of, $p_t = 1$, has $w_t = 0$ and drops out; write $T_c^+ = \{t \in T_c : w_t > 0\}$.

Null law. For a wrong candidate $m' \neq m_c$, the uniforms $u_t^{(m')}(y_t)$, $t \in T_c$, are read at PRF inputs the encoder never queried, so they are independent $U(0,1)$ variables conditional on $(y, \mathbf{T})$, exactly as in Theorem 2; conditioning on x and the model adds nothing, because p_t is a function of the text and the prompt. With $E_t = -\log u_t^{(m')}(y_t) \sim \mathrm{Exp}(1)$ independent,

$$\begin{aligned} Z_{c,w} := -R_{c,w}(m') &= \sum_{t \in T_c^+} w_t E_t, \qquad F_{c,w}(z) = \Pr[Z_{c,w} \leq z], \\ \mathbb{E}\big[e^{-sZ_{c,w}}\big] &= \prod_{t \in T_c^+} \frac{1}{1 + s w_t} \qquad (s \geq 0), \end{aligned} \tag{22}$$

a weighted sum of independent unit exponentials whose law depends on the text only through the weights. Its mean is $\sum_t w_t = \sum_t (1-p_t)/p_t$. For the correct candidate, $u_t(y_t) \sim \mathrm{Beta}(1/p_t, 1)$ by Theorem 3, so $-\log u_t(y_t) \sim \mathrm{Exp}(1/p_t)$ and $-R_{c,w}(m_c)$ has mean $\sum_t w_t p_t = \sum_t (1-p_t)$, smaller by the factor p_t at every token: the less likely the emitted token, the wider the gap the score opens between the correct candidate and every wrong one.

Certificate and decision. A large score is a small $Z_{c,w}$, so the upper tail of the score is the lower tail of the null law, $\Pr[R_{c,w}(m') \geq r] = F_{c,w}(-r)$. The certificate and the certified decoder are

$$\begin{aligned} \hat{\delta}_{c,w} &= \min\Big\{1,\ (2^{k_c} - 1)\, F_{c,w}\big(-R_{c,w}(\hat{m}_c)\big)\Big\} = \min\Big\{1,\ (2^{k_c} - 1)\, F_{c,w}\big(A_c - \ell_c(\hat{m}_c)\big)\Big\}, \\ D_\delta(y) &= \begin{cases} \hat{m}_c, & \hat{\delta}_{c,w} \leq \delta, \\ \perp, & \hat{\delta}_{c,w} > \delta, \end{cases} \end{aligned} \tag{23}$$

with $\hat{\delta}_{c,w} = 1$ when $T_c^+ = \varnothing$. This is (13)–(14) with the Gamma tail replaced by $F_{c,w}$, and the guarantee is the same:

$$\Pr_{\kappa}\big[D_\delta(Y_{1:n}) \notin \{m_c, \perp\}\big] \le \delta. \tag{24}$$

The argument is that of Theorem 4. For $\delta < 1$ let z_δ solve $F_{c,w}(z_\delta) = \delta/(2^{k_c} - 1)$; then $\hat{\delta}_{c,w} \le \delta$ exactly when $-R_{c,w}(\hat{m}_c) \le z_\delta$, a wrong candidate can be accepted only if its own $-R_{c,w}$ is at most z_δ, and the union bound over the $2^{k_c} - 1$ wrong candidates gives $(2^{k_c} - 1)F_{c,w}(z_\delta) = \delta$. Decoding every chunk at level δ/C bounds the probability of any wrong chunk in the message by δ.

Computing the tail. $F_{c,w}$ is the distribution function of a sum of up to a few hundred independent exponentials with distinct rates. Its closed form, an alternating sum of exponentials, is numerically unusable at this size, and the convolution recursion of Appendix P is exact but slow. The implementation evaluates it through the cumulant generating function of the null score,

$$\begin{aligned} K(\lambda) &= \log \mathbb{E}\big[e^{\lambda R_{c,w}(m')}\big] = -\sum_{t \in T_c^+} \log(1 + \lambda w_t), \\ K'(\lambda) &= -\sum_{t \in T_c^+} \frac{w_t}{1 + \lambda w_t}, \qquad K''(\lambda) = \sum_{t \in T_c^+} \frac{w_t^2}{(1 + \lambda w_t)^2}, \end{aligned} \tag{25}$$

by the Lugannani–Rice saddlepoint approximation. For the observed score $r = R_{c,w}(\hat{m}_c)$ above the null mean, let $\hat{\lambda} > 0$ solve $K'(\hat{\lambda}) = r$; then

$$\begin{aligned} F_{c,w}(-r) = \Pr[R_{c,w}(m') \ge r] \;\approx\; 1 - \Phi(\hat{w}) - \phi(\hat{w})\Big(\frac{1}{\hat{w}} - \frac{1}{\hat{u}}\Big), \\ \hat{w} = \sqrt{2\big(\hat{\lambda} r - K(\hat{\lambda})\big)}, \qquad \hat{u} = \hat{\lambda}\sqrt{K''(\hat{\lambda})}, \end{aligned} \tag{26}$$

with Φ and ϕ the standard normal distribution function and density. The approximation is of an exactly known law, not of a modelling assumption, and it is most accurate in the far tail, where certificates are decided; the exact recursion remains available when an exact value is required.

The robust decoder. With $\varepsilon > 0$ the decoder scores $r_c(m') = \sum_{t \in T_c} s_{p_t,\varepsilon}\big(u_t^{(m')}(y_t)\big)$ with $s_{p,\varepsilon}(u) = \log[(1-\varepsilon)f_p(u) + \varepsilon]$, the score (19). The wrong-candidate uniforms are $U(0,1)$ whatever was done to the text, so the null law, its upper tail and the certificate are

$$\begin{aligned} Z_{c,\varepsilon} = \sum_{t \in T_c} s_{p_t,\varepsilon}(U_t), \qquad U_t \text{ i.i.d. } U(0,1), \qquad Q_{c,\varepsilon}(s) = \Pr[Z_{c,\varepsilon} \ge s], \\ \hat{\delta}_{c,\varepsilon} = \min\Big\{1,\ (2^{k_c} - 1)\, Q_{c,\varepsilon}\big(r_c(\hat{m}_c)\big)\Big\}, \end{aligned} \tag{27}$$

and D_δ returns $\hat{m}_c$ when $\hat{\delta}_{c,\varepsilon} \le \delta$, with the guarantee (24) by the same union bound. Each summand is bounded, $\log\varepsilon \le s_{p,\varepsilon}(u) \le \log\big(\varepsilon + (1-\varepsilon)/p\big)$, and for $p < 1$ has the explicit distribution function

$$\Pr\big[s_{p,\varepsilon}(U) \le x\big] = \Big(\frac{p\,(e^x - \varepsilon)}{1 - \varepsilon}\Big)^{p/(1-p)}, \qquad \log\varepsilon < x < \log\Big(\varepsilon + \frac{1-\varepsilon}{p}\Big), \tag{28}$$

so $Q_{c,\varepsilon}$ follows from the convolution recursion of Appendix Q. The implementation carries that recursion out on a lattice of step $\Delta = 0.002$: the mass that (28) assigns to each cell $((k-1)\Delta, k\Delta]$ is placed at the cell's upper end, the lattice masses of the summands are convolved by the fast Fourier transform, and the tail is summed from the first lattice point at or above the observed score. Rounding every summand upwards makes the computed tail an upper bound on $Q_{c,\varepsilon}$, so the certificate stays valid and is at most slightly conservative; it agrees with a 2×10^6-draw Monte Carlo estimate to within a few percent and, unlike Monte Carlo, has no resolution floor, which matters for $2^{16} - 1$ wrong candidates. The contamination model of Theorem 6 shapes the score, not the guarantee: the null law in (27) holds for every wrong candidate on every text, so the certificate of the robust decoder remains valid on edited text, and the floor $\log\varepsilon$ on every summand is what keeps a single pasted low-probability token from deciding the chunk. All model-aware results in this paper use the robust score with $\varepsilon = 0.1$.

B EXPERIMENTAL SETUP AND HYPERPARAMETERS

We evaluate Qwen3.5-4B and Llama-3.1-8B, using their instruction variants when required by the task. Following the XMark protocol, we test text completion on C4 (Raffel et al., 2020), text summarization on CNN/DailyMail (Hermann et al., 2015), and story generation on WritingPrompts (Fan et al., 2018), with 50 users, 8-bit messages, and token budgets $T \in \{150, 200, 250, 300\}$. We use temperature 0.7, top-p sampling with $p = 0.9$, $\delta = 2$ for biased baselines, and a separate model from the same family to measure perplexity. Appendix C defines every reported metric. Every experiment reports the same four CertMark variants, always in this order: CertMark model-agnostic, the decoder of Section 3, and CertMark model-aware, the decoder of Section 3.3, each with the default chunking, whose label gives the chunk width in bits, and with 2-bit chunks. The default chunk width is the largest divisor of L that is at most 16, so one chunk carries the whole message for $L \leq 16$. The two decoders read the generations of the same encoder.

C EVALUATION METRICS

Every number in the paper is computed by the harness of Xu et al. (2026), or offline from the generations it saves, with one recipe for all methods. A run has 50 users. User i holds an L-bit message m_i and two prompts, each answered with exactly $T/2$ tokens, so T is the number of watermarked tokens from which the decoder recovers one message. Each watermarked text is paired with an unwatermarked sample of the same model from the same prompt under the same generation seed and decoding settings. Unless stated otherwise, a reported value is the mean over users, and shaded bands and error bars are ± 1 standard error over users.

Bit accuracy (BA). The decoder reads the two texts of user i and returns an L-bit message $\hat{m}_i$, for CertMark the concatenation of the chunk decisions of Section 3. Bit accuracy is the percentage of correctly recovered bits,

$$\mathrm{BA} = \frac{100}{50\,L} \sum_{i=1}^{50} \sum_{j=1}^{L} \mathbf{1}\big[\hat{m}_{i,j} = m_{i,j}\big],$$

so chance is 50% and 100% means that every user's message was recovered exactly. Everywhere except in Table 5 the decoders always return a message, that is level $\delta = 1$ without abstention. The "Avg." columns and the distortion columns of Table 2 average over the four token budgets. Under an attack (Figures 3 and 4) BA is computed on the edited texts and additionally averaged over three random draws of the edit.

Perplexity (PPL). Fluency is scored by an oracle model of the same family that did not generate the text: Qwen3.5-9B for Qwen3.5-4B, and for Llama-3.1-8B the other 8B sibling, so Llama-3.1-8B-Instruct scores the completions of the base model and the base model scores the two chat tasks. The oracle is run on prompt and response with the prompt tokens masked, and the perplexity of a user is $\exp$ of the mean negative log-likelihood of the generated tokens, pooled over the user's two texts; PPL is the mean over users. Lower is better, but the reference for a distortion-free scheme is the "unwatermarked" row, the same quantity on the paired unwatermarked samples, rather than the smallest attainable value. The two decoders of CertMark read the same generations, so they share PPL and every distortion measure below.

Top-k agreement. At every step of watermarked generation the harness records the k most likely next tokens under the model's own distribution for the current watermarked prefix, before the watermark acts on the logits. Top-k is the percentage of emitted tokens that lie in this set, for $k \in \{1, 5, 10\}$, averaged over texts and users. It measures how far the watermark pulls the sampler away from the model's preferences. Since the unwatermarked sampler at temperature 0.7 emits its top token only part of the time, a distortion-free scheme matches the rate of unwatermarked sampling rather than 100%.

BERTScore (BSc.). BERTScore F1 (Zhang et al., 2020) between each watermarked text and its paired unwatermarked sample, with the package's default English backbone (RoBERTa-large, layer 17), averaged over the texts of a user and then over users. It measures semantic agreement with the untouched sampler's output, not quality against a human reference.

ROUGE. ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-Lsum F-measures (Lin, 2004) with Porter stemming, computed with the `rouge-score` package, the watermarked text as prediction and the paired unwatermarked sample as reference. The harness re-downloads the metric on every call, which fails on our compute nodes and logs NaN, so ROUGE is recomputed offline from the saved generations with the harness's own recipe: the package's bootstrap aggregate over the texts of a user, then the mean over users. Like Top-k and BERTScore, ROUGE here measures agreement with unwatermarked text, so a distortion-free scheme scores like two independent samples of the model rather than 1.

BLEU. Only the translation task has references. BLEU (Papineni et al., 2002) is corpus BLEU computed with sacrebleu (Post, 2018) at its defaults (`13a` tokenization, up to four-grams, exponential smoothing) over all watermarked translations of all users against the English references of WMT14 newstest2014, each translation matched to its reference through the German source in its prompt. The "unwatermarked" row is the same computation on the paired unwatermarked translations. BLEU is therefore the one text metric that scores quality against a human reference.

Certificate metrics. For Table 5 the certified decoder at level δ returns its message only if $\hat{\delta}_c \leq \delta$ and $\perp$ otherwise. Certified error is the fraction of watermarked chunks on which it returns a wrong message; null certification rate is the fraction of paired unwatermarked chunks on which it returns any message; abstention is the fraction of watermarked chunks of a task on which it returns $\perp$.

D Runtime comparison

Table 6: Runtime on one H100. Gen.: ms/token; Dec.: s/user.

	Qwen3.5-4B		Llama-3.1-8B	
Method	Gen.	Dec.	Gen.	Dec.
CycleShift	19.6	.15	15.0	.07
DepthW	19.0	11.56	14.9	12.03
StealthInk	20.0	.18	15.3	.10
MPAC	18.9	.03	14.8	.03
RSBH	19.7	2.18	15.1	1.18
XMark	54.9	.28	27.6	.15
CertMark model-agnostic 8-bit	20.1	.01	15.5	.01
CertMark model-agnostic 2-bit	20.2	.01	15.5	.01
CertMark model-aware 8-bit	20.1	.42	15.5	.20
CertMark model-aware 2-bit	20.2	.35	15.5	.19
unwatermarked	19.1	–	14.1	–

Table 6 measures generation time per emitted token and decoding time per user for two 75-token completions, averaged over 20 users. CertMark generates near the unwatermarked speed, and CertMark model-agnostic 8-bit is the fastest decoder tested. CertMark model-aware adds a model pass but remains faster than the heavier baselines.

E Main benchmark on Llama-3.1-8B

Table 7 gives the large three-task benchmark table for Llama-3.1-8B (completion) and Llama-3.1-8B-Instruct (chat tasks). Its main findings and all other experiments on this model are discussed in Section 4.

F Comparison to other distortion-free methods

CertMark is not the first distortion-free watermark. Zero-bit schemes have preserved the sampling law for some time (Kuditipudi et al., 2024; Christ et al., 2024; Hu et al., 2024; Wu et al., 2024; Dathathri et al., 2024), and StealthInk (Jiang et al., 2025), BiMark (Feng et al., 2025), DISC (Boroujeny et al., 2024) and MirrorMark (Jiang et al., 2026) carry a multi-bit payload without altering it, the first two by a keyed reweighting whose mean over the key is the original law, the last two by moving the message into the sampling randomness. To the best of our knowledge, CertMark is the only method that is distortion-free and also returns a certificate on the decoded message. Table 8 compares the multi-bit schemes on Qwen3.5-4B under the protocol of Table 2: StealthInk repeats its rows of that

Table 7: Table 2 repeated with Llama-3.1-8B (completion) and Llama-3.1-8B-Instruct (chat tasks) as the generating model, with perplexity under the other 8B sibling of the family since the 70B oracle does not fit one GPU. Every number is measured in our harness; layout and colours follow the main table.

Method	$T = 150$		$T = 200$		$T = 250$		$T = 300$		Avg.		Distortion (avg. over T)			
	BA↑	PPL↓	BA↑	PPL↓	BA↑	PPL↓	BA↑	PPL↓	BA↑	PPL↓	Top-1↑	Top-5↑	R-1↑	R-L↑
Text Completion														
CycleShift	100.00	5.63	100.00	5.75	100.00	5.59	100.00	5.51	100.00	5.62	57.62	88.55	0.321	0.186
DepthW	98.25	5.28	99.50	5.12	99.00	5.12	100.00	5.19	99.19	5.18	60.75	89.56	0.337	0.195
StealthInk	90.00	4.61	91.00	4.35	94.25	4.58	96.25	4.28	92.88	4.46	64.98	92.37	0.347	0.202
MPAC	96.75	5.84	98.00	5.48	99.25	5.45	99.00	5.67	98.25	5.61	57.66	88.34	0.329	0.188
RSBH	94.50	5.29	96.25	5.23	95.75	5.35	97.50	5.39	96.00	5.31	58.31	90.17	0.331	0.188
XMark	98.75	5.63	100.00	5.15	100.00	5.04	100.00	5.05	99.69	5.22	59.31	90.27	0.333	0.191
CertMark model-agnostic 8-bit	100.00	4.29	100.00	4.16	100.00	4.09	100.00	4.08	100.00	4.15	68.06	93.58	0.358	0.215
CertMark model-agnostic 2-bit	99.00	4.28	99.75	4.20	100.00	4.11	100.00	3.99	99.69	4.14	67.80	93.47	0.359	0.206
CertMark model-aware 8-bit	100.00	4.29	100.00	4.16	100.00	4.09	100.00	4.08	100.00	4.15	68.06	93.58	0.358	0.215
CertMark model-aware 2-bit	100.00	4.28	100.00	4.20	100.00	4.11	100.00	3.99	100.00	4.14	67.80	93.47	0.359	0.206
Text Summarization														
CycleShift	66.50	4.31	83.00	4.50	88.00	4.58	91.75	4.80	82.31	4.55	82.77	99.08	0.613	0.453
DepthW	57.00	4.31	61.00	4.37	68.50	4.47	68.00	4.65	63.62	4.45	84.29	99.28	0.621	0.460
StealthInk	61.25	4.14	74.00	4.26	73.50	4.26	77.00	4.35	71.44	4.25	87.26	99.74	0.631	0.477
MPAC	78.00	4.57	83.50	4.51	84.00	4.53	91.25	4.74	84.19	4.59	82.53	98.99	0.611	0.451
RSBH	63.00	4.65	72.25	4.52	77.50	4.59	85.25	4.76	74.50	4.63	81.84	99.19	0.608	0.446
XMark	81.25	4.47	81.50	4.52	89.00	4.51	95.50	4.69	86.81	4.55	82.58	99.40	0.619	0.454
CertMark model-agnostic 8-bit	66.50	4.10	73.75	4.10	77.00	4.19	87.50	4.24	76.19	4.16	88.25	99.78	0.633	0.486
CertMark model-agnostic 2-bit	74.50	4.19	78.75	4.19	84.00	4.28	87.25	4.30	81.12	4.24	87.59	99.74	0.636	0.490
CertMark model-aware 8-bit	97.75	4.10	100.00	4.10	100.00	4.19	100.00	4.24	99.44	4.16	88.25	99.78	0.633	0.486
CertMark model-aware 2-bit	95.00	4.19	98.75	4.19	100.00	4.28	99.75	4.30	98.38	4.24	87.59	99.74	0.636	0.490
Story Generation														
CycleShift	85.50	9.04	95.00	8.03	94.75	7.33	97.00	6.72	93.06	7.78	77.56	98.16	0.406	0.264
DepthW	65.25	8.57	77.50	7.87	83.00	7.45	81.75	6.47	76.88	7.59	79.24	98.45	0.410	0.259
StealthInk	72.50	7.86	76.00	7.52	79.50	6.28	80.00	6.03	77.00	6.92	83.28	99.37	0.425	0.281
MPAC	84.00	9.06	86.00	8.10	89.75	7.16	94.25	6.66	88.50	7.75	77.14	97.94	0.400	0.253
RSBH	73.75	8.85	78.75	8.25	87.25	7.47	93.00	6.80	83.19	7.84	76.72	98.55	0.402	0.253
XMark	90.50	8.68	92.50	7.73	97.25	7.20	97.00	6.76	94.31	7.59	77.42	98.68	0.404	0.254
CertMark model-agnostic 8-bit	80.25	7.90	88.50	7.03	93.25	6.35	93.75	5.91	88.94	6.80	85.19	99.48	0.426	0.281
CertMark model-agnostic 2-bit	82.75	8.32	86.75	7.18	89.75	6.51	92.00	6.04	87.81	7.01	85.04	99.42	0.430	0.288
CertMark model-aware 8-bit	100.00	7.90	100.00	7.03	100.00	6.35	100.00	5.91	100.00	6.80	85.19	99.48	0.426	0.281
CertMark model-aware 2-bit	99.00	8.32	99.75	7.18	100.00	6.51	100.00	6.04	99.69	7.01	85.04	99.42	0.430	0.288

table, BiMark runs the authors' code with the paper's defaults, ten layers and $\delta = 1$, and MirrorMark, which has no public code, is our reimplementation from the paper's description with its defaults, carrying $L = 8$ as four 2-bit symbols. Since all of these schemes leave the sampling law unchanged, the perplexity columns agree to within the noise of the 50-user sample and the differences fall on either side, StealthInk sitting slightly higher on completion and story generation. Bit accuracy is what separates them: CertMark model-agnostic 8-bit matches or exceeds every baseline in every cell, with the largest lead on summarization, 98.8 on average against 93.3 for BiMark, 92.7 for MirrorMark and 81.6 for StealthInk, and CertMark model-aware recovers every message on all three tasks. Appendix G includes the three baselines in the attack comparison.

G Further attacks on Qwen3.5-4B

Figure 4 extends Figure 3 on Qwen3.5-4B to two token-level edits, for every method of Table 2 except DepthW, whose decoder enumerates all 2^L codes and needs about an hour per user at $L = 16$, and for BiMark and MirrorMark (Appendix F). CycleShift, BiMark and MirrorMark, absent from Figure 3, are added, with their copy-paste results in the first row. The protocol is that of Figure 3: text completion, $L = 16$, 50 users with two texts each, every method decoded by its own text-only detector on the same edited texts, and three random draws of each edit per text; the model-aware CertMark decoder also reads the prompt.

Token edits. The two edits act alike, because each destroys the contexts that overlap the edited position: about four scored positions per edit for the three-token context of CertMark, two for the one-token contexts of CycleShift and MPAC. Accordingly, CertMark model-aware 16-bit and CertMark model-agnostic 16-bit are the most robust methods at $r = 10\%$, between 96.5 and 99.7 at every budget, where MPAC stays below 93 and XMark below 96; at 20% they still match or exceed

Table 8: Distortion-free multi-bit schemes on Qwen3.5-4B: StealthInk, BiMark, MirrorMark and the four CertMark rows of Table 2, under the same protocol, tasks and budgets, with perplexity under Qwen3.5-9B; every number is measured in our harness. Red marks the best value in each column.

Method	$T = 150$		$T = 200$		$T = 250$		$T = 300$		Avg.	
	BA↑	PPL↓	BA↑	PPL↓	BA↑	PPL↓	BA↑	PPL↓	BA↑	PPL↓
Text Completion										
StealthInk	89.75	6.15	90.50	5.79	95.25	5.73	94.25	5.61	92.44	5.82
BiMark	98.25	5.79	99.75	5.81	99.75	5.40	99.25	5.26	99.25	5.56
MirrorMark	100.00	5.83	100.00	5.66	100.00	5.12	100.00	5.27	100.00	5.47
CertMark model-agnostic 8-bit	100.00	5.65	100.00	5.23	100.00	5.29	100.00	5.11	100.00	5.32
CertMark model-agnostic 2-bit	99.75	5.63	100.00	5.55	100.00	5.34	99.75	5.20	99.88	5.43
CertMark model-aware 8-bit	100.00	5.69	100.00	5.20	100.00	5.35	100.00	5.09	100.00	5.33
CertMark model-aware 2-bit	100.00	5.64	100.00	5.52	100.00	5.36	100.00	5.24	100.00	5.44
Text Summarization										
StealthInk	75.25	2.95	80.75	3.10	82.25	3.11	88.00	3.14	81.56	3.08
BiMark	89.50	2.93	92.00	3.05	95.75	3.07	96.00	3.09	93.31	3.04
MirrorMark	86.25	2.91	90.00	2.99	97.25	3.11	97.25	3.11	92.69	3.03
CertMark model-agnostic 8-bit	97.75	3.05	98.75	3.05	98.50	3.06	100.00	3.08	98.75	3.06
CertMark model-agnostic 2-bit	87.25	2.99	94.50	3.10	95.00	3.13	97.50	3.09	93.56	3.08
CertMark model-aware 8-bit	100.00	3.03	100.00	3.07	100.00	3.10	100.00	3.10	100.00	3.07
CertMark model-aware 2-bit	98.75	3.00	99.25	3.12	99.75	3.11	99.75	3.11	99.38	3.08
Story Generation										
StealthInk	82.50	3.81	84.75	3.93	89.50	4.02	92.50	3.87	87.31	3.91
BiMark	95.25	3.52	97.75	3.75	98.00	3.71	98.75	3.88	97.44	3.71
MirrorMark	98.50	3.52	98.75	3.57	99.25	3.63	99.50	3.70	99.00	3.60
CertMark model-agnostic 8-bit	100.00	3.63	100.00	3.71	100.00	3.81	100.00	3.87	100.00	3.75
CertMark model-agnostic 2-bit	95.75	3.71	98.50	3.75	99.25	3.81	99.75	3.82	98.31	3.77
CertMark model-aware 8-bit	100.00	3.63	100.00	3.71	100.00	3.81	100.00	3.87	100.00	3.75
CertMark model-aware 2-bit	100.00	3.71	100.00	3.75	100.00	3.81	100.00	3.82	100.00	3.77

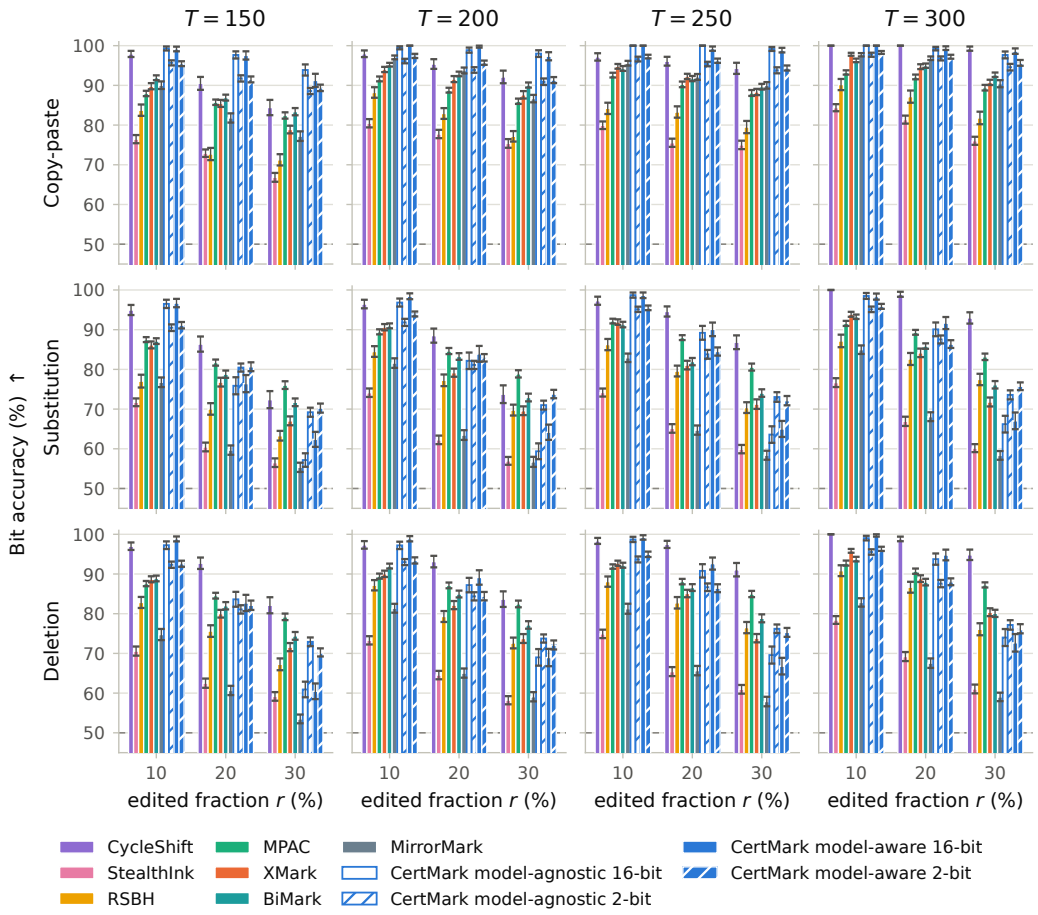


Figure 4: Robustness to token edits on Qwen3.5-4B, text completion, $L = 16$: bit accuracy after a fraction r of the tokens of each watermarked text is replaced by spans of unwatermarked text (copy-paste, as in Figure 3), replaced by random vocabulary tokens, or deleted; one row per attack, one column per token budget T. Colours and markers as in Figure 3; the dashed line is chance. Means over 50 users and three random draws of the edit, ± 1 standard error.

XMark, and MPAC from $T = 200$ on, but trail CycleShift; and at 30%, where a text keeps roughly a quarter of its scored positions, the 16-bit codebook runs out of evidence and they fall below all three, to between 57 and 74. The 2-bit chunks degrade more gracefully, since each chunk needs far less evidence, and keep 69 to 77 at 30%. The model-aware decoder brings no advantage under these edits, because the damage is a loss of contexts rather than a contamination of scored tokens, which is what Theorem 6 addresses. CycleShift, the most robust method under heavy editing, pays for it with a bias on every token and the highest perplexity of Table 2. Of the distortion-free baselines, BiMark degrades like MPAC, from 87 to 94 at 10% to 72 to 80 at 30%, since its two-token context loses few positions per edit and each bit is a majority over its own votes, at the price of never reaching full recovery on clean text. MirrorMark is the least robust: its scheduler assigns positions inside context-anchored frames, so one edit shifts every position up to the next anchor, and it drops to 75 to 85 at 10% and to between 53 and 59 at 30%, close to chance.

H PSEUDOCODE

Algorithms 1–4 restate the method as pseudocode, in the notation of Table 1. There is one encoder: both decoders read its generations and differ only in the per-token score. Each decoder first decodes *without* certification, returning the highest-scoring candidate of every chunk; this is the decoder behind every benchmark table and figure. Given a level δ, it then decodes *with* certification by handing its scores to CERTIFY (Algorithm 2), which keeps the candidate or abstains; this is the decoder of Table 5. Every argument tuple of PRF_κ is encoded by $\langle\cdot\rangle$ of (29), ρ is the map (30), and the loops over the 2^{k_c} candidates are vectorised over the codebook in the implementation.

Algorithm 1 CertMark encoder, one generation, shared by both decoders; Eqs. (4)–(6).

Input:
key κ
message $m = (m_1, \ldots, m_C)$ with $m_c \in \{0,1\}^{k_c}$
prompt x and the deployed sampler $p(\cdot \mid x, \cdot)$, after temperature and truncation
length n and context width h

Output:
text $y_{1:n}$, distributed exactly as unwatermarked sampling (Theorem 1)

$\mathcal{C} \leftarrow \varnothing$ ▷ contexts used so far in this generation
for $t = 1, \ldots, n$ **do**
  $p_t(\cdot) \leftarrow p(\cdot \mid x, y_{<t})$ ▷ after temperature and truncation
  $c_t \leftarrow y_{t-h:t-1}$ ▷ the available prefix when $t \leq h$
  if $t \leq h$ **or** $c_t \in \mathcal{C}$ **then**
    $a_t \leftarrow (\mathsf{fresh}, c_t, t)$ ▷ carries no message; the position makes the argument new
  else
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{c_t\}$; $i_t \leftarrow \mathrm{PRF}_\kappa(\mathsf{chunk}, c_t) \bmod C$; $a_t \leftarrow (\mathsf{msg}, c_t, m_{i_t})$
  end if
  $u_t(v) \leftarrow \rho\big(\mathrm{PRF}_\kappa(a_t, v)\big)$ for every $v \in V$
  $y_t \leftarrow \arg\max_{v \in V} \big[\log p_t(v) - \log(-\log u_t(v))\big]$ ▷ Gumbel-max, Lemma 1
end for
return $y_{1:n}$

Algorithm 2 CERTIFY: certified decision for one chunk, used by Algorithms 3 and 4; Eqs. (13)–(14).

Input:
scores $S(m')$, $m' \in \{0,1\}^k$, and their maximiser $\hat{m}$
number of scored tokens n
level $\delta \in (0,1]$
upper tail $Q(s) = \Pr[S(m') \geq s]$ of the score under a wrong candidate

Output:
$\hat{m}$ when it is certified at level δ, else $\perp$; then $\Pr[\text{output} \notin \{m_c, \perp\}] \leq \delta$ (Theorem 4)

if $n = 0$, or every scored token has $p_t = 1$ in Algorithm 4 **then** $\hat{\delta} \leftarrow 1$
else $\hat{\delta} \leftarrow \min\{1,\ (2^k - 1)\, Q\big(S(\hat{m})\big)\}$ ▷ union bound over the $2^k - 1$ wrong candidates
end if
return $\hat{m}$ if $\hat{\delta} \leq \delta$, else $\perp$ ▷ level δ/C per chunk bounds the whole-message error by δ

Algorithm 3 CertMark model-agnostic decoder, from the text and the key alone; Eqs. (8)–(9).

Input:
key κ
text $y_{1:n}$
chunk widths $k_1, \ldots, k_C$ and context width h
for certified decoding, a level $\delta \in (0,1]$

Output:
decoded message $\hat{m} = (\hat{m}_1, \ldots, \hat{m}_C)$
with certification, each $\hat{m}_c \in \{0,1\}^{k_c} \cup \{\perp\}$

$S_c(m') \leftarrow 0$ for all c and $m' \in \{0,1\}^{k_c}$; $n_c \leftarrow 0$; $\mathcal{T} \leftarrow \varnothing$
for $t = h+1, \ldots, n$ **do**
  $c_t \leftarrow y_{t-h:t-1}$; **if** $c_t = c_s$ for some $s \in \mathcal{T}$ **then continue** ▷ repeated context
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{t\}$; $c \leftarrow \mathrm{PRF}_\kappa(\mathsf{chunk}, c_t) \bmod C$; $n_c \leftarrow n_c + 1$
  for $m' \in \{0,1\}^{k_c}$ **do**
    $u \leftarrow \rho\big(\mathrm{PRF}_\kappa(\mathsf{msg}, c_t, m', y_t)\big)$; $S_c(m') \leftarrow S_c(m') - \log(1-u)$
  end for
end for
for $c = 1, \ldots, C$ **do**
  $\hat{m}_c \leftarrow \arg\max_{m'} S_c(m')$ ▷ without certification: the decoder of every benchmark table
  if certifying **then** $\hat{m}_c \leftarrow$ CERTIFY$\big(S_c(\cdot), \hat{m}_c, n_c, \delta, Q_c\big)$ with $Q_c(s) = \Pr[\Gamma(n_c, 1) \geq s]$ (Theorem 2)
end for
return $\hat{m}$

Algorithm 4 CertMark model-aware decoder, from the text, the key, the prompt and the model; Eqs. (17) and (19).

Input:
as in Algorithm 3
prompt x and the deployed sampler p
contamination rate $\varepsilon \in [0, 1)$

Output:
decoded message $\hat{m} = (\hat{m}_1, \ldots, \hat{m}_C)$
with certification, each $\hat{m}_c \in \{0,1\}^{k_c} \cup \{\perp\}$

$p_t \leftarrow p(y_t \mid x, y_{<t})$ for $t = 1, \ldots, n$ ▷ one forward pass over prompt and text
$r_c(m') \leftarrow 0$ for all c and m'; $\mathcal{T}_c \leftarrow \varnothing$; $\mathcal{T} \leftarrow \varnothing$
for $t = h+1, \ldots, n$ **do**
  $c_t \leftarrow y_{t-h:t-1}$; **if** $c_t = c_s$ for some $s \in \mathcal{T}$ **then continue**
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{t\}$; $c \leftarrow \mathrm{PRF}_\kappa(\mathsf{chunk}, c_t) \bmod C$; $\mathcal{T}_c \leftarrow \mathcal{T}_c \cup \{t\}$
  for $m' \in \{0,1\}^{k_c}$ **do**
    $u \leftarrow \rho\big(\mathrm{PRF}_\kappa(\mathsf{msg}, c_t, m', y_t)\big)$; $r_c(m') \leftarrow r_c(m') + \log\big[(1-\varepsilon)\, p_t^{-1} u^{1/p_t - 1} + \varepsilon\big]$
  end for
end for
$\varepsilon = 0$ gives the log-likelihood ratio ℓ_c of (17); $\varepsilon > 0$ gives the robust score r_c of (19).
for $c = 1, \ldots, C$ **do**
  $\hat{m}_c \leftarrow \arg\max_{m'} r_c(m')$ ▷ without certification
  if certifying **then** $\hat{m}_c \leftarrow$ CERTIFY$\big(r_c(\cdot), \hat{m}_c, |\mathcal{T}_c|, \delta, Q_c\big)$, where $Q_c(s) = F_{c,w}(A_c - s)$ with $w_t = 1/p_t - 1$ and $A_c = \sum_{t \in \mathcal{T}_c} \log(1/p_t)$ if $\varepsilon = 0$ (Theorem 5), and $Q_c = Q_{c,\varepsilon}$ if $\varepsilon > 0$ (Theorem 6)
end for
return $\hat{m}$

I THE PSEUDORANDOM FUNCTION

This appendix gives the details of the pseudorandom function that Section 3 uses. Let $\mathrm{PRF}_\kappa : \{0,1\}^* \to \mathcal{S}$, $\mathcal{S} = \{0,1\}^{64}$, be a pseudorandom function keyed by κ (Goldreich et al., 1986), where $\{0,1\}^* = \bigcup_{n=0}^{\infty} \{0,1\}^n$ is the set of all finite binary strings. Its values at distinct inputs are computationally indistinguishable from independent uniform elements of $\mathcal{S}$. To pass structured arguments to the PRF, let $\mathcal{X}$ contain the typed tags, token sequences and integers used in the method, and fix an injective, prefix-free encoding

$$\begin{aligned} \langle\cdot\rangle : \bigcup_{r=1}^{\infty} \mathcal{X}^r \to \{0,1\}^*, \\ \mathrm{PRF}_\kappa(x_1, \ldots, x_r) := \mathrm{PRF}_\kappa(\langle x_1, \ldots, x_r \rangle)\,. \end{aligned} \tag{29}$$

Thus each tuple becomes one finite binary string, and the three tags chunk, msg and fresh domain-separate the three uses of the PRF. Interpreting $z \in \mathcal{S}$ as an unsigned integer, we convert its output to

$$\rho(z) = \big(\lfloor z/2^{11} \rfloor + \tfrac{1}{2}\big) 2^{-53}, \tag{30}$$

which is uniform on the 53-bit midpoint grid in $(0, 1)$ when z is uniform. Throughout the analysis we use the standard random-function idealisation, treating $\rho(\mathrm{PRF}_\kappa(a))$ at distinct inputs a as independent $U(0, 1)$ variables.

Instantiation. Our implementation realises PRF_κ with 64-bit integer arithmetic. Let ϕ be the SplitMix64 finaliser (Steele Jr. et al., 2014), a bijection of $\{0,1\}^{64}$ made of three xor-shifts and two multiplications by fixed odd constants. A context $c_t = (y_{t-h}, \ldots, y_{t-1})$ is hashed to a seed by starting from $\phi(\kappa \cdot g_1 \oplus g_2)$ and absorbing one token at a time, $s \leftarrow \phi\big(s \oplus (y+1)\, g_3\big)$, with fixed odd constants g_1, g_2, g_3. The three uses of the PRF are separated by fixed salts: the chunk index is $\phi(s \oplus \mathsf{salt}_{\mathsf{chunk}}) \bmod C$, a message tuple (msg, c_t, m) maps to $\phi\big(s \oplus \phi((m+1)\, g_2 \oplus \mathsf{salt}_{\mathsf{msg}})\big)$, and a fresh tuple (fresh, c_t, t) maps to $\phi\big(s \oplus \mathsf{salt}_{\mathsf{fresh}} \oplus \phi(t\, g_3)\big)$. The uniform of token v under a seed σ is $\rho\big(\phi(\sigma + (v+1)\, g_1)\big)$ with ρ as in (30), so the whole vocabulary is scored with one vectorised

pass and the decoder recomputes any single entry from the text and the key. All arithmetic is modulo 2^{64}. This is a fast non-cryptographic hash, chosen for speed; a keyed cryptographic PRF, such as a block cipher in counter mode or HMAC, can be substituted without any change to the method or the analysis, at a higher cost per token.

J Proof of Lemma 1

Lemma 1 (Gumbel-max sampling)**.** *Fix a step t and recall that p_t is the sampler's law over V at that step, redrawn at every step. Draw $u(v) \sim U(0,1)$ i.i.d. over $v \in V$, form the Gumbel variates and the scores*

$$S(v) \;=\; \log p_t(v) - \log\big(-\log u(v)\big), \tag{2}$$

and emit $y = \arg\max_{v \in V} S(v)$. Then $\Pr[y = v] = p_t(v)$ for every $v \in V$.

Proof. Fix the step t and its probability distribution p_t on the finite vocabulary V. Thus $p_t(v) \geq 0$ for every v and $\sum_{v \in V} p_t(v) = 1$. Throughout the proof, p_t is fixed and all probabilities are taken over the independent draws $u(v) \sim U(0,1)$.

Restricting to tokens with positive probability. Let $V_+ = \{v \in V : p_t(v) > 0\}$. This set is nonempty because the probabilities sum to one. With the convention $\log 0 = -\infty$, a token $v \notin V_+$ has $S(v) = -\infty$. In contrast, $S(v)$ is finite for every $v \in V_+$, since $0 < u(v) < 1$ implies $0 < -\log u(v) < \infty$. Consequently, the maximiser must belong to V_+, and $\Pr[y = v] = 0 = p_t(v)$ for every $v \notin V_+$. All divisions by $p_t(v)$ below are restricted to $v \in V_+$.

Transforming the uniforms. For $v \in V_+$, define $E(v) = -\log u(v)$. For every $s \geq 0$,

$$\begin{aligned} \Pr[E(v) > s] &= \Pr[-\log u(v) > s] \\ &= \Pr[u(v) < e^{-s}] = e^{-s}. \end{aligned} \tag{31}$$

The last equality uses $e^{-s} \in (0,1]$ and the distribution function of $U(0,1)$. Thus $E(v)$ is exponential with rate 1. The variables $E(v)$ remain independent, since each is a deterministic function of its own independent uniform draw.

This transformation also verifies the Gumbel terminology in the lemma. For $G(v) = -\log E(v)$ and every $z \in \mathbb{R}$,

$$\Pr[G(v) \leq z] = \Pr[E(v) \geq e^{-z}] = \exp(-e^{-z}), \tag{32}$$

where replacing $>$ by $\geq$ does not change the probability because $E(v)$ has a continuous distribution. Hence the $G(v)$ are independent standard Gumbel variables and $S(v) = \log p_t(v) + G(v)$.

Expressing the maximiser through exponential variables. Define

$$T(v) = \frac{E(v)}{p_t(v)}, \qquad v \in V_+. \tag{33}$$

These variables are independent because the denominators are fixed. For $s \geq 0$, their survival functions and densities are

$$\begin{aligned} \Pr[T(v) > s] &= \Pr[E(v) > p_t(v)s] = e^{-p_t(v)s}, \\ f_{T(v)}(s) &= \frac{\mathrm{d}}{\mathrm{d}s}\big(1 - e^{-p_t(v)s}\big) = p_t(v)e^{-p_t(v)s}, \qquad s > 0. \end{aligned} \tag{34}$$

In particular, $T(v)$ is exponential with rate $p_t(v)$. On V_+ the score can be rewritten as

$$S(v) = \log p_t(v) - \log E(v) = -\log\left(\frac{E(v)}{p_t(v)}\right) = -\log T(v). \tag{35}$$

Since $r \mapsto -\log r$ is strictly decreasing on $(0, \infty)$, the token with the largest score is exactly the token with the smallest value of $T(v)$:

$$y = \arg\max_{v \in V_+} S(v) = \arg\min_{v \in V_+} T(v). \tag{36}$$

This minimiser is unique with probability one. Indeed, for two distinct tokens $v, w \in V_+$, independence implies that conditioning on $T(w)$ leaves $T(v)$ with its continuous distribution. The conditional probability that $T(v)$ equals that particular value of $T(w)$ is therefore zero. Averaging gives $\Pr[T(v) = T(w)] = 0$, and taking the union over the finitely many pairs shows that the probability of any tie is zero.

Computing the probability of selecting each token. Fix $v \in V_+$. Conditional on $T(v) = s$, the token v is selected exactly when $T(w) > s$ for every $w \in V_+ \setminus \{v\}$, apart from the probability-zero ties just considered. Independence gives the conditional probability of this event as $\prod_{w \in V_+ \setminus \{v\}} \Pr[T(w) > s]$. Integrating over the density of $T(v)$ therefore yields

$$\begin{aligned}
\Pr[y = v] &= \int_0^\infty f_{T(v)}(s) \prod_{w \in V_+ \setminus \{v\}} \Pr[T(w) > s] \, \mathrm{d}s \\
&= \int_0^\infty p_t(v) e^{-p_t(v)s} \prod_{w \in V_+ \setminus \{v\}} e^{-p_t(w)s} \, \mathrm{d}s \\
&= p_t(v) \int_0^\infty \exp\left(-s \sum_{w \in V_+} p_t(w) \right) \mathrm{d}s \\
&= p_t(v) \int_0^\infty e^{-s} \, \mathrm{d}s = p_t(v).
\end{aligned} \tag{37}$$

Here the third line combines the exponential factors, the fourth uses $\sum_{w \in V_+} p_t(w) = 1$, and the final equality uses $\int_0^\infty e^{-s} \, \mathrm{d}s = [-e^{-s}]_0^\infty = 1$.

If V_+ contains only v, the product over competing tokens is an empty product, equal to 1, and $p_t(v) = 1$, so the same calculation covers deterministic sampling. Together with the zero-probability case above, this proves $\Pr[y = v] = p_t(v)$ for every $v \in V$.

Equivalence to the implemented score. For each $v \in V_+$, the implementation uses

$$\frac{\log u(v)}{p_t(v)} = -\frac{E(v)}{p_t(v)} = -T(v). \tag{38}$$

Maximising this quantity is again equivalent to minimising $T(v)$, and therefore selects the same token as maximising $S(v)$. Assigning score $-\infty$ to tokens outside V_+ gives the same exclusion as in the original rule. This equivalence is an algebraic identity for the draws in $(0, 1)$ when the scores are evaluated in exact arithmetic. □

K Proof of Theorem 1

Theorem 1 (Distortion-free generation)**.** *Treat* PRF *as a random function and draw the key κ uniformly from $\mathcal{S}$. Let $Y_{1:n}$ be the sequence emitted by* (5)–(6) *on prompt x carrying message $m \in \{0,1\}^L$. Then for every such m and x and every $w \in V^n$,*

$$\Pr_\kappa\big[Y_{1:n} = w\big] \;=\; \prod_{t=1}^{n} p\big(w_t \mid x, w_{<t}\big), \tag{7}$$

which is the law of unwatermarked sampling and, in particular, does not depend on m.

Proof. Fix the prompt x, message m, generation length n, model and sampling settings. We consider one generation, with the seen-context set initially empty and the encoder's context initialization fixed. We work under the continuous random-function idealisation stated after (30): values at distinct token-sampling inputs are independent $U(0, 1)$ variables. All probabilities below are over this idealised keyed randomness, denoted by $\Pr_\kappa$ as in the theorem. We first establish the conditional law of a single token and then derive the law of the entire sequence.

Step 1: condition on the information available before sampling. Fix a step $t \in \{1, \ldots, n\}$. Let $\mathcal{H}_t$ denote the information generated by the emitted prefix $Y_{<t}$, all earlier PRF queries and their answers,

and the current chunk-selection query (4) together with its answer, before any sampling uniforms for step t are read. Including this last query is necessary because i_t depends on the PRF and need not be determined by the prefix alone. Conditional on $\mathcal{H}_t$, the following are fixed:

$$p_t(\cdot) = p(\cdot \mid x, Y_{<t}), \qquad c_t, \qquad \mathcal{C}_{t-1}, \qquad i_t, \qquad a_t. \tag{39}$$

In particular, a_t is chosen using only information already revealed. At $t = 1$, the earlier query transcript and emitted prefix are empty; the same conditioning includes the first chunk-selection query.

Step 2: verify that every sampling input is new. For each $v \in V$, write $b_{t,v} = \langle a_t, v \rangle$ for the encoded input used to obtain $u_t(v)$. We check both branches of (5).

First occurrence of the context. If $c_t \notin \mathcal{C}_{t-1}$, then $a_t = (\mathsf{msg}, c_t, m_{i_t})$. Every earlier message-carrying input has the form $\langle \mathsf{msg}, c_s, m_{i_s}, w \rangle$ for some $s < t$. Because c_t has not appeared before, $c_s \neq c_t$, so none of these inputs equals $b_{t,v}$.

Repeated context. If $c_t \in \mathcal{C}_{t-1}$, then $a_t = (\mathsf{fresh}, c_t, t)$. Every earlier input with the tag fresh contains an earlier position $s < t$. Its position field therefore differs from t, even when its context equals c_t.

In either branch, the tags msg and fresh distinguish the two families of sampling inputs from each other, and both differ from the tag chunk used for chunk selection. Thus $b_{t,v}$ also differs from every chunk-selection input, including the current one. Finally, for distinct tokens $v \neq w$, the inputs $b_{t,v}$ and $b_{t,w}$ differ in their token field. The injective encoding in (29) preserves all these distinctions. Hence the inputs used at step t are pairwise distinct and have never been queried earlier in this generation.

Step 3: obtain independent uniforms conditional on the history. A random function can be revealed one query at a time: at a previously unqueried input, draw an independent value from its prescribed distribution and store it; at a repeated input, return the stored value. This produces the same law as sampling the whole random function in advance. It also shows why choosing a new input from earlier query answers does not change the law of its unrevealed value.

By Step 2, every input $b_{t,v}$ is new. Under the continuous idealisation, for any numbers $r_v \in [0, 1]$, $v \in V$,

$$\Pr_{\kappa}[u_t(v) \leq r_v \text{ for all } v \in V \mid \mathcal{H}_t] = \prod_{v \in V} r_v. \tag{40}$$

This is the joint distribution function of independent $U(0, 1)$ variables. Consequently, conditional on $\mathcal{H}_t$, the entire vector $u_t(\cdot)$ has exactly the distribution required by Lemma 1.

Step 4: identify the next-token distribution. Given $\mathcal{H}_t$, the vector p_t is a fixed probability distribution by Step 1, and the sampling uniforms are independent $U(0, 1)$ variables by Step 3. Applying Lemma 1 to the encoder's rule (6) therefore gives, for every $v \in V$,

$$\Pr_{\kappa}[Y_t = v \mid \mathcal{H}_t] = p_t(v) = p(v \mid x, Y_{<t}). \tag{41}$$

Tokens with $p_t(v) = 0$ have score $-\infty$ and are never emitted; among tokens with positive probability, ties occur with probability zero. Thus the equality includes zero-probability tokens as well.

The prefix $Y_{<t}$ is part of $\mathcal{H}_t$. Averaging over the additional query information by the law of iterated conditional expectation gives

$$\begin{aligned} \Pr_{\kappa}[Y_t = v \mid Y_{<t}] &= \mathbb{E}_{\kappa}\Big[\Pr_{\kappa}[Y_t = v \mid \mathcal{H}_t] \,\Big|\, Y_{<t}\Big] \\ &= \mathbb{E}_{\kappa}[p(v \mid x, Y_{<t}) \mid Y_{<t}] \\ &= p(v \mid x, Y_{<t}), \end{aligned} \tag{42}$$

where the last equality holds because the prompt and prefix already determine the deployed sampling distribution. This is precisely the conditional next-token law of ordinary, unwatermarked generation.

Step 5: derive the sequence distribution. Fix any $w \in V^n$, and write $q_t = \Pr_{\kappa}[Y_{1:t} = w_{1:t}]$, with $q_0 = 1$ for the empty prefix. If $q_{t-1} > 0$, the multiplication rule and Step 4 give

$$\begin{aligned} q_t &= q_{t-1} \Pr_{\kappa}[Y_t = w_t \mid Y_{<t} = w_{<t}] \\ &= q_{t-1}\, p(w_t \mid x, w_{<t}). \end{aligned} \tag{43}$$

If $q_{t-1} = 0$, then $q_t = 0$ because the event $\{Y_{1:t} = w_{1:t}\}$ is contained in $\{Y_{<t} = w_{<t}\}$, so the same recurrence still holds. Starting from $q_0 = 1$ and applying this recurrence successively for $t = 1, \ldots, n$ yields

$$\Pr_{\kappa}[Y_{1:n} = w] = q_n = \prod_{t=1}^{n} p(w_t \mid x, w_{<t}), \tag{44}$$

which proves (7), including sequences with zero probability. The right-hand side is the sequence law of the unwatermarked sampler. It does not depend on m, since the message changes the sampling inputs but leaves $p(\cdot \mid x, w_{<t})$ unchanged. As x and m were arbitrary, the result holds for every prompt and message. □

L Proof of Theorem 2

Theorem 2 (Score distribution for wrong candidates)**.** *For a wrong candidate $m' \neq m_c$, conditional on the emitted text and the scored positions $\mathcal{T}_c$ selected by the chunk assignments, whenever $n_c = |\mathcal{T}_c| \geq 1$,*

$$S_c(m') \;\sim\; \Gamma(n_c, 1) \qquad \textit{exactly,} \tag{10}$$

where $\Gamma(n_c, 1)$ has shape n_c and rate 1. If $n_c = 0$, then $S_c(m') = 0$.

Proof. Fix the prompt, model, sampling settings and embedded message m. Also fix a chunk c and a candidate $m' \neq m_c$. We use the continuous random-function idealisation stated after (30), and take all probabilities over the idealised keyed randomness. We will first show that the values reconstructed for m' are independent uniforms even after conditioning on the information used to generate the text. We then derive the distribution of their score contributions and their sum.

Fixing the text and the scored positions. Let $\mathcal{H}$ denote the information in the complete encoder query transcript: all PRF inputs queried during the generation and all their answers. Include also the chunk-selection queries used by the decoder, before it evaluates any candidate scores. The generated text is determined by the encoder transcript and the fixed prompt and model. Consequently, conditional on $\mathcal{H}$, the text y, its contexts, the visited positions $\mathcal{T}$, their chunk assignments i_t, and the set $\mathcal{T}_c$ are all fixed. This extra conditioning matters because $\mathcal{T}_c$ depends on the chunk-selection PRF values as well as on the text.

If $\mathcal{T}_c = \varnothing$, the sum in (9) is empty and therefore $S_c(m') = 0$. This proves the assertion for $n_c = 0$. In what follows, assume $n_c \geq 1$ and, for every $t \in \mathcal{T}_c$, write

$$b_t = \langle \mathsf{msg}, c_t, m', y_t \rangle, \qquad U_t = \rho\big(\mathrm{PRF}_\kappa(b_t)\big) = u_t^{(m')}(y_t). \tag{45}$$

The inputs b_t are determined by $\mathcal{H}$; their PRF values will be the randomness used to score the wrong candidate.

Showing that the candidate inputs are distinct and unqueried. The decoder retains only the first occurrence of each context in (8). Thus $c_t \neq c_s$ for distinct $t, s \in \mathcal{T}_c$, and the injective encoding in (29) implies $b_t \neq b_s$.

Now fix $t \in \mathcal{T}_c$ and compare b_t with every type of query in $\mathcal{H}$. A chunk-selection query has tag chunk, and a fresh-sampling query has tag fresh; each differs from the tag msg in b_t. A message-carrying encoder query at any position s has the form

$$\langle \mathsf{msg}, c_s, m_{i_s}, v \rangle, \qquad v \in V. \tag{46}$$

If $c_s \neq c_t$, its context field differs from that of b_t. If $c_s = c_t$, the context-based assignment (4) gives $i_s = i_t = c$. The encoder therefore uses $m_{i_s} = m_c$, which differs from m', so the message field differs. This comparison covers every position in the generation, including positions after t, and every token queried by the encoder. Hence none of the b_t was queried in forming $\mathcal{H}$. The conclusion also holds if m' happens to equal the value carried by another chunk, since an identical context always selects the same chunk.

Obtaining independent uniforms under the conditioning. As in Appendix K, a random function can be revealed by drawing and storing an independent value when an input is first queried. Conditional on a complete query transcript, values at all unqueried inputs retain their original independent

distributions. Here the b_t are distinct unqueried inputs chosen from the transcript, so their values can still be drawn independently after that transcript has been fixed. In particular, for any $r_t \in [0,1]$, $t \in \mathcal{T}_c$,

$$\Pr_{\kappa}[U_t \le r_t \text{ for all } t \in \mathcal{T}_c \,|\, \mathcal{H}] = \prod_{t \in \mathcal{T}_c} r_t. \tag{47}$$

Thus the U_t are conditionally independent $U(0,1)$ variables. Conditioning on the generated text does not bias them: the entire generation used other PRF inputs, and its outcome is already part of $\mathcal{H}$.

Deriving the distribution of one score contribution. Set $X_t = -\log(1-U_t)$. Since $0 < U_t < 1$, each X_t is positive. For $z \ge 0$, the conditional survival function is

$$\begin{aligned} \Pr_{\kappa}[X_t > z \mid \mathcal{H}] &= \Pr_{\kappa}[-\log(1-U_t) > z \mid \mathcal{H}] \\ &= \Pr_{\kappa}[1 - U_t < e^{-z} \mid \mathcal{H}] \\ &= \Pr_{\kappa}[U_t > 1 - e^{-z} \mid \mathcal{H}] \\ &= 1 - (1 - e^{-z}) = e^{-z}. \end{aligned} \tag{48}$$

The last line uses $1 - e^{-z} \in [0,1)$ and the uniform law of U_t. For $z < 0$, the survival probability is 1. Therefore each X_t has the exponential distribution with rate 1, with density e^{-z} for $z > 0$ and zero density for $z < 0$. The X_t remain conditionally independent because each is a function of a different U_t.

Deriving the distribution of the sum. Enumerate $\mathcal{T}_c = \{t_1, \dots, t_{n_c}\}$, and let $Z_r = \sum_{j=1}^{r} X_{t_j}$. Conditional on $\mathcal{H}$, we claim that for each integer $1 \le r \le n_c$, its density is

$$f_r(z) = \frac{z^{r-1} e^{-z}}{(r-1)!}, \qquad z > 0, \tag{49}$$

and is zero for $z < 0$. For $r = 1$, this is exactly the exponential density derived above. Suppose it holds for some $r < n_c$. Since Z_r and $X_{t_{r+1}}$ are independent and nonnegative, the density of their sum is the convolution of their densities. For $z > 0$,

$$\begin{aligned} f_{r+1}(z) &= \int_0^z f_r(s) e^{-(z-s)} \, \mathrm{d}s \\ &= \int_0^z \frac{s^{r-1} e^{-s}}{(r-1)!} e^{-(z-s)} \, \mathrm{d}s \\ &= \frac{e^{-z}}{(r-1)!} \int_0^z s^{r-1} \, \mathrm{d}s \\ &= \frac{e^{-z}}{(r-1)!} \frac{z^r}{r} = \frac{z^r e^{-z}}{r!}. \end{aligned} \tag{50}$$

This proves the claim by induction. Taking $r = n_c$ and recalling $S_c(m') = Z_{n_c}$ identifies its conditional density as that of $\Gamma(n_c, 1)$, with shape n_c and rate 1.

Returning to conditioning only on the text and scored positions. Let F_r be the distribution function of $\Gamma(r,1)$ for $r \ge 1$, and set $F_0(z) = \mathbf{1}\{z \ge 0\}$ for the empty score. The preceding argument gives $\Pr_\kappa[S_c(m') \le z \mid \mathcal{H}] = F_{n_c}(z)$. Since $\mathcal{H}$ determines both $Y_{1:n}$ and $\mathcal{T}_c$, iterated conditional expectation yields, for every $z \in \mathbb{R}$,

$$\begin{aligned} \Pr_{\kappa}[S_c(m') \le z \mid Y_{1:n}, \mathcal{T}_c] &= \mathbb{E}_\kappa\Big[\Pr_{\kappa}[S_c(m') \le z \mid \mathcal{H}] \,\Big|\, Y_{1:n}, \mathcal{T}_c\Big] \\ &= \mathbb{E}_\kappa[F_{n_c}(z) \,|\, Y_{1:n}, \mathcal{T}_c] \\ &= F_{n_c}(z). \end{aligned} \tag{51}$$

The last equality holds because $n_c = |\mathcal{T}_c|$ is fixed by the conditioning. This proves (10) for every possible emitted text and scored set, with the empty case handled separately. The resulting law depends on them only through n_c; no model probabilities, prompt, sampling temperature or message values enter its parameters.

The same calculation also applies to a text fixed independently of the PRF. In that case, condition on that text and its chunk-selection queries; the candidate inputs are again distinct and unqueried, so the uniform, exponential and Gamma calculations above apply unchanged. □

M PROOF OF THEOREM 3

The main text states the candidate distributions. The full statement below additionally gives their expected scores; it keeps the theorem's number and the equation numbers reserved in the main text.

Theorem 3 (Candidate distributions and expected scores, full statement)**.** *Treat* PRF *as a random function. Fix the deployed model, prompt x, sampling settings, embedded message m, and a chunk c with $k_c \geq 1$. For $t \in \mathcal{T}_c$, condition on $y_{<t}$ and $p = p_t(y_t) \in (0, 1]$, the sampler's probability of the emitted token. For every candidate m', its reconstructed value and expected per-token score satisfy*

$$u_t^{(m')}(y_t) \sim \begin{cases} \mathrm{Beta}(1/p, 1), & m' = m_c \quad \textit{(correct candidate)}, \\ U(0,1), & m' \neq m_c \quad \textit{(wrong candidate)}, \end{cases} \tag{11}$$

$$\mathbb{E}\Big[-\log\big(1 - u_t^{(m')}(y_t)\big) \,\Big|\, y_{<t}, p\Big] = \begin{cases} g(p) := \psi(1/p + 1) + \gamma_E, & m' = m_c, \\ 1, & m' \neq m_c, \end{cases} \tag{12}$$

where $u_t^{(m_c)}(y_t) = u_t(y_t)$, ψ is the digamma function and γ_E is the Euler–Mascheroni constant; $g(1) = 1$ and g is strictly decreasing on $(0, 1]$.

Proof. Fix the prompt, model, sampling settings and embedded message. Fix a chunk c and a position t, and throughout the proof restrict to the event $t \in \mathcal{T}_c$, as in the theorem. This event says that the context first occurs at t and that its chunk assignment is c; both facts are determined before the token Y_t is sampled. All probabilities are over the continuous random-function idealisation stated after (30).

Conditioning before the token is drawn. Let $\mathcal{H}_t$ be the information available just before reading the sampling uniforms at step t, including the prefix, all earlier PRF queries and their answers, and the current chunk-selection query and its answer, as in Appendix K. Conditional on $\mathcal{H}_t$, the distribution p_t, context c_t and assignment $i_t = c$ are fixed. Since this is the first occurrence of the context, the encoder uses the tuple (msg, c_t, m_c), which is exactly the tuple reconstructed by the correct candidate. Hence

$$u_t^{(m_c)}(v) = u_t(v), \qquad v \in V. \tag{52}$$

Appendix K establishes that this vector has independent $U(0, 1)$ entries conditional on $\mathcal{H}_t$. This is the point at which the fresh-input rule is used. Let $V_+ = \{v \in V : p_t(v) > 0\}$. Tokens outside V_+ cannot be emitted, as shown in Appendix J, so $p = p_t(Y_t)$ belongs to $(0, 1]$ with probability one.

The joint law of the emitted token and its uniform. Appendix J shows that maximising the Gumbel score is equivalent to maximising $\log u_t(v)/p_t(v)$. Exponentiation is strictly increasing, so

$$Y_t = \arg\max_{v \in V_+} \frac{\log u_t(v)}{p_t(v)} = \arg\max_{v \in V_+} u_t(v)^{1/p_t(v)}. \tag{53}$$

Fix $v \in V_+$ and write $q = p_t(v) > 0$. If $u_t(v) = s \in (0, 1)$, then this token is selected precisely when, for every competing token $w \in V_+ \setminus \{v\}$,

$$u_t(w)^{1/p_t(w)} \leq s^{1/q} \quad \Longleftrightarrow \quad u_t(w) \leq s^{p_t(w)/q}. \tag{54}$$

The probability-zero ties can be ignored by Appendix J. Given $\mathcal{H}_t$ and $u_t(v) = s$, the other coordinates remain independent uniforms. Their joint probability of satisfying these inequalities is therefore

$$\begin{aligned} \Pr_\kappa[Y_t = v \mid \mathcal{H}_t, u_t(v) = s] &= \prod_{w \in V_+ \setminus \{v\}} s^{p_t(w)/q} \\ &= s^{\left(\sum_{w \in V_+ \setminus \{v\}} p_t(w)\right)/q} = s^{(1-q)/q}. \end{aligned} \tag{55}$$

The final equality uses $\sum_{w \in V_+} p_t(w) = 1$. When $q = 1$, there are no positive-probability competitors and the empty product is 1, in agreement with the same formula.

The conditional density of $u_t(v)$ is 1 on $(0,1)$. Integrating over its value thus gives, for $0 < x \le 1$,

$$\begin{aligned}\Pr_\kappa[u_t(v) \le x,\ Y_t = v \mid \mathcal{H}_t] &= \int_0^x \Pr_\kappa[Y_t = v \mid \mathcal{H}_t, u_t(v) = s]\,\mathrm{d}s \\ &= \int_0^x s^{1/q-1}\,\mathrm{d}s \\ &= \left[q\,s^{1/q}\right]_{s=0}^{s=x} = q\,x^{1/q}.\end{aligned} \tag{56}$$

Taking $x = 1$ recovers $\Pr_\kappa[Y_t = v \mid \mathcal{H}_t] = q$. Since $q > 0$, division by this probability is valid and yields

$$\begin{aligned}\Pr_\kappa[u_t(Y_t) \le x \mid \mathcal{H}_t, Y_t = v] &= \frac{\Pr_\kappa[u_t(v) \le x,\ Y_t = v \mid \mathcal{H}_t]}{\Pr_\kappa[Y_t = v \mid \mathcal{H}_t]} \\ &= \frac{q\,x^{1/q}}{q} = x^{1/q}, \qquad 0 < x < 1.\end{aligned} \tag{57}$$

Conditioning on the emitted token's probability. The preceding conditional law depends on the token v only through its probability $q = p_t(v)$. Write $p = p_t(Y_t)$ for the probability assigned to the token that was actually emitted. The pair $(\mathcal{H}_t, Y_t)$ determines both $Y_{<t}$ and p, so iterated conditional expectation gives, for $0 < x < 1$,

$$\begin{aligned}\Pr_\kappa[u_t(Y_t) \le x \mid Y_{<t}, p] &= \mathbb{E}_\kappa\Big[\Pr_\kappa[u_t(Y_t) \le x \mid \mathcal{H}_t, Y_t] \,\Big|\, Y_{<t}, p\Big] \\ &= \mathbb{E}_\kappa\Big[x^{1/p_t(Y_t)} \,\Big|\, Y_{<t}, p\Big] = x^{1/p}.\end{aligned} \tag{58}$$

Thus averaging over the additional query information, or over several tokens with the same probability p, leaves this distribution unchanged. Its distribution function is 0 for $x \le 0$ and 1 for $x \ge 1$, and its density on $(0,1)$ is

$$\frac{\mathrm{d}}{\mathrm{d}x}x^{1/p} = \frac{1}{p}x^{1/p-1}. \tag{59}$$

This is the density of $\mathrm{Beta}(1/p, 1)$, proving the correct-candidate case of (11).

The wrong-candidate distribution. Fix $m' \neq m_c$. Appendix L proves that the input $\langle \mathsf{msg}, c_t, m', Y_t\rangle$ is never queried by the encoder: an encoder query with that context uses m_c, and the other query families have different tags. Moreover, the value at this input remains uniform conditional on the complete generation transcript and the chunk-selection queries. Denote that information by $\mathcal{H}$, as in Appendix L. It determines the emitted text and hence also $Y_{<t}$ and p. Consequently,

$$\begin{aligned}\Pr_\kappa[u_t^{(m')}(Y_t) \le x \mid Y_{<t}, p] &= \mathbb{E}_\kappa\Big[\Pr_\kappa[u_t^{(m')}(Y_t) \le x \mid \mathcal{H}] \,\Big|\, Y_{<t}, p\Big] \\ &= \mathbb{E}_\kappa[x \mid Y_{<t}, p] = x, \qquad 0 < x < 1.\end{aligned} \tag{60}$$

Together with the endpoint values 0 and 1, this is the distribution function of $U(0,1)$, proving the second case of (11).

Computing the expected score from the Beta density. Let $B \sim \mathrm{Beta}(a, 1)$ with $a > 0$, so that its density is $a\,x^{a-1}$ on $(0,1)$. For every integer $j \ge 1$, its jth moment is

$$\begin{aligned}\mathbb{E}[B^j] &= \int_0^1 x^j\,a\,x^{a-1}\,\mathrm{d}x \\ &= a\left[\frac{x^{a+j}}{a+j}\right]_{x=0}^{x=1} = \frac{a}{a+j}.\end{aligned} \tag{61}$$

Integrating the geometric series for $1/(1-x)$ on $[0, b]$ gives $-\log(1-b) = \sum_{j=1}^{\infty} b^j/j$ for $0 \le b < 1$.

In particular, the nonnegative partial sums $L_N = \sum_{j=1}^{N} B^j/j$ increase to $-\log(1-B)$ with probability one. The monotone convergence theorem therefore justifies interchanging expectation and this infinite sum:

$$\begin{aligned}\mathbb{E}[-\log(1-B)] &= \lim_{N\to\infty} \mathbb{E}[L_N] \\ &= \lim_{N\to\infty} \sum_{j=1}^{N} \frac{\mathbb{E}[B^j]}{j} \\ &= \sum_{j=1}^{\infty} \frac{a}{j(a+j)} \\ &= \sum_{j=1}^{\infty} \left(\frac{1}{j} - \frac{1}{j+a}\right).\end{aligned} \tag{62}$$

This expectation is finite, since $0 \le a/[j(a+j)] \le a/j^2$ and $\sum_{j=1}^{\infty} j^{-2}$ converges. For the digamma function $\psi(z) = \Gamma'(z)/\Gamma(z)$, its series representation gives

$$\psi(a+1) = -\gamma_E + \sum_{j=1}^{\infty} \left(\frac{1}{j} - \frac{1}{j+a}\right), \qquad a > 0. \tag{63}$$

Combining these two identities yields $\mathbb{E}[-\log(1-B)] = \psi(a+1) + \gamma_E$. Applying this identity to the correct candidate's conditional law, with $a = 1/p$, gives

$$\mathbb{E}\big[-\log\big(1-u_t(Y_t)\big) \,\big|\, Y_{<t}, p\big] = \psi(1/p+1) + \gamma_E = g(p). \tag{64}$$

For a wrong candidate, $U(0,1) = \mathrm{Beta}(1,1)$, so $a = 1$. The corresponding series telescopes:

$$\sum_{j=1}^{\infty} \left(\frac{1}{j} - \frac{1}{j+1}\right) = \lim_{N\to\infty} \left(1 - \frac{1}{N+1}\right) = 1. \tag{65}$$

This proves the wrong-candidate mean in (12) and also shows that $g(1) = 1$. At $p = 1$, the correct-candidate Beta law is itself $U(0,1)$, so the two distributions and their expected scores coincide.

Strict decrease of the expected score. Substituting $a = 1/p$ into the convergent series above gives the equivalent expression

$$g(p) = \sum_{j=1}^{\infty} \frac{1}{j(1+jp)}, \qquad p > 0. \tag{66}$$

For $0 < p_1 < p_2 \le 1$, subtracting the two convergent series yields

$$\begin{aligned}g(p_1) - g(p_2) &= \sum_{j=1}^{\infty} \left(\frac{1}{j(1+jp_1)} - \frac{1}{j(1+jp_2)}\right) \\ &= (p_2 - p_1) \sum_{j=1}^{\infty} \frac{1}{(1+jp_1)(1+jp_2)} > 0.\end{aligned} \tag{67}$$

The last inequality holds because $p_2 - p_1 > 0$ and every summand is positive. Hence g is strictly decreasing on $(0,1]$. Together with $g(1) = 1$, this also gives $g(p) > 1$ whenever $p < 1$, completing the proof. □

N Proof of Theorem 4

Theorem 4 (Certified decoding). *For a level $\delta \in (0,1]$, let $\hat{m}_c(y) = \arg\max_{m'} S_c(m')$ and define*

$$\hat{\delta}_c(y) \;=\; \min\Big\{1,\; \big(2^{k_c} - 1\big)\, \Pr\big[\Gamma(n_c, 1) \ge S_c(\hat{m}_c(y))\big]\Big\}, \tag{13}$$

with $\hat{\delta}_c(y) = 1$ when $n_c = 0$. Let $\perp$ denote abstention and define the certified decoder $D_\delta \colon V^n \to \{0,1\}^{k_c} \cup \{\perp\}$ by

$$D_\delta(y) \;=\; \begin{cases} \hat{m}_c(y), & \hat{\delta}_c(y) \le \delta, \\ \perp, & \hat{\delta}_c(y) > \delta, \end{cases} \tag{14}$$

where the scores are computed from the text y and the key. Then, for the text $Y_{1:n}$ emitted by (5)–(6)*,*

$$\Pr_{\kappa}\big[D_\delta(Y_{1:n}) \notin \{m_c, \perp\}\big] \;\le\; \delta, \tag{15}$$

Here the probability is over the idealised PRF randomness induced by the one-time draw of κ.

Proof. Fix the model, prompt, sampling settings and embedded message m. Also fix the chunk c and the level $\delta \in (0, 1]$, before observing the scores. All probabilities are over the idealised keyed randomness. Write

$$\mathcal{M}_c = \{0,1\}^{k_c}, \qquad K = |\mathcal{M}_c| = 2^{k_c}, \qquad \mathcal{W}_c = \mathcal{M}_c \setminus \{m_c\}. \tag{68}$$

Thus $\mathcal{W}_c$ contains exactly $K - 1$ wrong candidates. Choose any fixed rule for breaking ties in the maximisation defining $\hat{m}_c$. The event whose probability we must bound is

$$E_c = \big\{D_\delta(Y_{1:n}) \notin \{m_c, \perp\}\big\}, \tag{69}$$

namely, the event that the decoder accepts an incorrect chunk.

Boundary cases and conditioning. If $\delta = 1$, then $\Pr_\kappa[E_c] \le 1 = \delta$ simply because E_c is an event. If $K = 1$, there are no wrong candidates, so E_c is empty. We may therefore assume $0 < \delta < 1$ and $K \ge 2$.

Condition on any pair $Y_{1:n} = y$ and $\mathcal{T}_c = \tau$ with positive probability, and put $r = |\tau|$. To keep the notation readable, write

$$\Pr_{y,\tau}[\,\cdot\,] := \Pr_{\kappa}[\,\cdot \mid Y_{1:n} = y,\ \mathcal{T}_c = \tau]. \tag{70}$$

Under this conditioning, the number of scored positions is fixed at $n_c = r$. Conditioning on $\mathcal{T}_c$ as well as on the text is necessary because the scored positions also depend on the chunk-selection PRF values. The candidate scores themselves remain random under this conditioning.

If $r = 0$, the definition gives $\hat{\delta}_c(y) = 1 > \delta$, and the decoder abstains. Hence $\Pr_{y,\tau}[E_c] = 0$. In the remainder of the conditional argument, assume $r \ge 1$.

The conditional null tail and its threshold. Let G_r have the Gamma distribution with shape r and rate 1, and denote its survival function by

$$Q_r(s) := \Pr[G_r \ge s] = \int_s^\infty \frac{z^{r-1}e^{-z}}{(r-1)!}\,\mathrm{d}z, \qquad s \ge 0. \tag{71}$$

Theorem 2 states precisely that every fixed wrong candidate has this conditional score distribution. Consequently,

$$\Pr_{y,\tau}[S_c(m') \ge s] = Q_r(s) \qquad \text{for every } m' \in \mathcal{W}_c \text{ and } s \ge 0. \tag{72}$$

Here Q_r is a deterministic function once r has been fixed; it does not depend on any realised candidate score.

The integral defining Q_r shows that it is continuous, with $Q_r(0) = 1$ and $Q_r(s) \to 0$ as $s \to \infty$. It is also strictly decreasing: whenever $0 \le a < b$,

$$Q_r(a) - Q_r(b) = \int_a^b \frac{z^{r-1}e^{-z}}{(r-1)!}\,\mathrm{d}z > 0, \tag{73}$$

because the integrand is positive for every $z > 0$. Since $0 < \delta/(K-1) < 1$, continuity and strict monotonicity imply that there is exactly one $s_\delta > 0$ satisfying

$$Q_r(s_\delta) = \frac{\delta}{K-1}, \qquad \text{or equivalently} \qquad (K-1)Q_r(s_\delta) = \delta. \tag{74}$$

This threshold depends on r, K and δ, all of which are fixed in the conditional argument.

Rewriting the acceptance rule as a score threshold. Abbreviate $\hat{m} = \hat{m}_c(y)$. Since $r \ge 1$, the certificate in (13) becomes

$$\hat{\delta}_c(y) = \min\{1, (K-1)Q_r(S_c(\hat{m}))\}. \tag{75}$$

For any $a \geq 0$ and $\delta < 1$, the inequality $\min\{1, a\} \leq \delta$ holds if and only if $a \leq \delta$: if $a \geq 1$, the minimum equals $1 > \delta$, and if $a < 1$, the minimum equals a. Applying this observation and then the strict decrease of Q_r gives

$$\begin{aligned} \hat{\delta}_c(y) \leq \delta \quad &\Longleftrightarrow \quad (K-1)Q_r(S_c(\hat{m})) \leq \delta \\ &\Longleftrightarrow \quad Q_r(S_c(\hat{m})) \leq Q_r(s_\delta) \\ &\Longleftrightarrow \quad S_c(\hat{m}) \geq s_\delta. \end{aligned} \tag{76}$$

All scores are nonnegative because their summands are $-\log(1-u)$ with $u \in (0,1)$, so the survival function above is being evaluated on its stated domain.

Controlling an incorrectly selected candidate. Under the current conditioning, the decoder makes a certified error exactly when its selected candidate is wrong and its score meets the threshold. Therefore

$$\begin{aligned} E_c &= \{\hat{m} \in \mathcal{W}_c\} \cap \{S_c(\hat{m}) \geq s_\delta\} \\ &= \bigcup_{m' \in \mathcal{W}_c} \big(\{\hat{m} = m'\} \cap \{S_c(m') \geq s_\delta\}\big) \\ &\subseteq \bigcup_{m' \in \mathcal{W}_c} \{S_c(m') \geq s_\delta\}. \end{aligned} \tag{77}$$

The second line separates the error event according to which wrong candidate is selected. The inclusion then drops the selection condition: accepting a wrong candidate requires at least one wrong candidate to cross the threshold. This step avoids assigning a Gamma law to the selected maximum; Theorem 2 is applied only to each fixed wrong candidate.

By the conditional union bound, the conditional null law, and the definition of s_δ, respectively,

$$\begin{aligned} \Pr_{y,\tau}[E_c] &\leq \sum_{m' \in \mathcal{W}_c} \Pr_{y,\tau}[S_c(m') \geq s_\delta] \\ &= \sum_{m' \in \mathcal{W}_c} Q_r(s_\delta) \\ &= (K-1)Q_r(s_\delta) \\ &= (K-1)\frac{\delta}{K-1} = \delta. \end{aligned} \tag{78}$$

The union bound requires no independence among candidate scores. Nor does the argument require the distribution of the correct candidate's score: that score affects which candidate is selected, but the event inclusion already accounts for the selection. This also shows that the bound holds for the chosen tie-breaking rule.

Removing the conditioning. We have proved $\Pr_{y,\tau}[E_c] \leq \delta$ for every pair (y, τ) with positive probability, including $\tau = \varnothing$. Averaging over both the emitted text and the scored positions, the law of total probability gives

$$\begin{aligned} \Pr_{\kappa}[E_c] &= \mathbb{E}_\kappa\Big[\Pr_{\kappa}[E_c \mid Y_{1:n}, \mathcal{T}_c]\Big] \\ &\leq \mathbb{E}_\kappa[\delta] = \delta. \end{aligned} \tag{79}$$

Together with the boundary cases, this proves (15) for every $\delta \in (0, 1]$. The model, prompt, sampling settings and message were fixed throughout; the averaging is only over the idealised key and the text and scored positions it determines.

The whole-message corollary. For clarity, write $D_{\alpha,c}$ for the decoder applied to chunk c at level α, and set

$$E_c^{\mathrm{msg}} = \big\{D_{\delta/C,c}(Y_{1:n}) \notin \{m_c, \perp\}\big\}. \tag{80}$$

The chunkwise result gives $\Pr_\kappa[E_c^{\mathrm{msg}}] \leq \delta/C$ for each $c = 1, \ldots, C$. Returning any incorrect chunk is exactly the event $\bigcup_{c=1}^{C} E_c^{\mathrm{msg}}$. A second union bound yields

$$\Pr_{\kappa}\left[\bigcup_{c=1}^{C} E_c^{\mathrm{msg}}\right] \leq \sum_{c=1}^{C} \Pr_{\kappa}[E_c^{\mathrm{msg}}] \leq \sum_{c=1}^{C} \frac{\delta}{C} = \delta. \tag{81}$$

This conclusion does not require independence between chunks. □

O PAYLOAD SCALING

The following theorem describes how decoding changes as the number of encoded message bits increases; its exponent Λ_c is reused by Theorem 8.

Theorem 7 (Payload scaling). *For $p \in (0,1]$ and $\lambda \in (0,1)$, define*

$$\eta_p(\lambda) \;=\; \log(1-\lambda) \;-\; \log \frac{\Gamma(1/p+1)\,\Gamma(\lambda+1)}{\Gamma(1/p+\lambda+1)}, \tag{82}$$

$\eta_p(\lambda)$ is nonincreasing in p and $\eta_1(\lambda) = \log(1-\lambda^2) < 0$. Condition on the emitted text $Y_{1:n} = y$ and the scored-position sets $\{\mathcal{T}_c\}_{c=1}^{C}$ selected by the chunk assignments. This fixes n_c and $p_t = p_t(y_t)$ for $t \in \mathcal{T}_c$. Then

$$\begin{aligned} \Pr_{\kappa}\big[\hat{m}_c \neq m_c \,\big|\, Y_{1:n} = y, \{\mathcal{T}_c\}_{c=1}^{C}\big] &\le \big(2^{k_c}-1\big)e^{-\Lambda_c} < \exp\big(k_c \ln 2 - \Lambda_c\big), \\ \Lambda_c &= \sup_{\lambda\in(0,1)} \sum_{t\in\mathcal{T}_c} \eta_{p_t}(\lambda). \end{aligned} \tag{83}$$

An empty scored set has $\Lambda_c = 0$. For the whole message $\hat{m} = (\hat{m}_1, \ldots, \hat{m}_C)$,

$$\Pr_{\kappa}[\hat{m} \neq m \mid Y_{1:n} = y, \{\mathcal{T}_c\}_{c=1}^{C}] \le \sum_{c=1}^{C} (2^{k_c}-1)e^{-\Lambda_c}. \tag{84}$$

Proof. Fix the model, prompt x, sampling settings and embedded message m. We work under the continuous random-function idealisation stated after (30). Throughout the conditional error calculation, fix a realisation of the emitted text $Y_{1:n} = y$ and the scored-position sets $\{\mathcal{T}_c\}_{c=1}^{C}$ with positive probability. Write $\Pr_*$ and $\mathbb{E}_*$ for probability and expectation conditional on this pair. The sets $\mathcal{T}_c$, their sizes n_c, and the probabilities $p_t = p(y_t \mid x, y_{<t})$ are then fixed. Each p_t is positive, since a token with zero sampling probability cannot appear in a text of positive probability.

Theorem 3 gives the correct candidate's law at one step. Here we need its joint law across scored positions after conditioning on the *entire* text and its chunk assignments. We establish that law first, and then compare the correct score with each wrong score.

The joint law of a token and the maximum used to select it. Let $\mathcal{A}$ denote the complete context-to-chunk assignment table determined by the PRF inputs tagged chunk. These inputs are disjoint from all token-sampling inputs, so conditioning on $\mathcal{A}$ leaves the random functions used for token sampling independent with their original laws. For this part of the proof, condition on $\mathcal{A}$ before generating the text.

At step t, let $\mathcal{H}_t$ include $\mathcal{A}$ and the full query history before the sampling uniforms for that step are read. As proved in Appendix K, these sampling inputs are distinct and have never been queried: the first occurrence of a context uses its message tuple, while a repeated context uses a tuple containing the new position t. Thus the coordinates of $u_t(\cdot)$ are independent $U(0,1)$ variables conditional on $\mathcal{H}_t$, and $p_t(\cdot)$ is fixed by that history.

Put $V_{t,+} = \{v : p_t(v) > 0\}$ and, for $v \in V_{t,+}$, define

$$Z_t(v) = u_t(v)^{1/p_t(v)}, \qquad M_t = \max_{v\in V_{t,+}} Z_t(v). \tag{85}$$

The equivalent sampling rule in Appendix J gives $Y_t = \arg\max_{v\in V_{t,+}} Z_t(v)$. For $0 < z < 1$,

$$\Pr_{\kappa}[Z_t(v) \le z \mid \mathcal{H}_t] = z^{p_t(v)}, \qquad f_{Z_t(v)\mid\mathcal{H}_t}(z) = p_t(v) z^{p_t(v)-1}. \tag{86}$$

These variables are conditionally independent, being functions of distinct uniform coordinates, and ties have probability zero. The event $\{M_t \le z, Y_t = v\}$ occurs when $Z_t(v)$ has some value $s \le z$ and every competitor is at most s. Integrating over s gives

$$\begin{aligned} \Pr_{\kappa}[M_t \le z, Y_t = v \mid \mathcal{H}_t] &= \int_0^z p_t(v) s^{p_t(v)-1} \prod_{w\in V_{t,+}\setminus\{v\}} s^{p_t(w)}\,\mathrm{d}s \\ &= p_t(v) \int_0^z s^{\sum_{w\in V_{t,+}} p_t(w)-1}\,\mathrm{d}s \\ &= p_t(v) \int_0^z 1\,\mathrm{d}s = z\,p_t(v). \end{aligned} \tag{87}$$

The third line uses $\sum_{w \in V_{t,+}} p_t(w) = 1$. The same identity holds at $z = 0, 1$ by continuity, and for $p_t(v) = 0$ both sides vanish. If there is only one possible token, the product over competitors is empty and equals 1, so this case is included.

Extending this law to the entire generated text. Fix numbers $z_1, \dots, z_n \in [0, 1]$ and let

$$\mathcal{E}_j = \{Y_{1:j} = y_{1:j},\ M_t \le z_t \text{ for } 1 \le t \le j\}, \qquad \mathcal{E}_0 = \Omega. \tag{88}$$

The event $\mathcal{E}_{j-1}$ is determined by $\mathcal{H}_j$. On this event the prefix is $y_{<j}$, so the preceding single-step identity and iterated conditional expectation yield

$$\begin{aligned} \Pr_\kappa[\mathcal{E}_j \mid \mathcal{A}] &= \mathbb{E}_\kappa\Big[\mathbf{1}_{\mathcal{E}_{j-1}} \Pr_\kappa[M_j \le z_j, Y_j = y_j \mid \mathcal{H}_j] \,\Big|\, \mathcal{A}\Big] \\ &= z_j p(y_j \mid x, y_{<j}) \Pr_\kappa[\mathcal{E}_{j-1} \mid \mathcal{A}]. \end{aligned} \tag{89}$$

Applying this recurrence for $j = 1, \dots, n$ gives

$$\Pr_\kappa[Y_{1:n} = y,\ M_t \le z_t \text{ for all } t \mid \mathcal{A}] = \prod_{t=1}^{n} z_t p(y_t \mid x, y_{<t}). \tag{90}$$

Setting every $z_t = 1$ shows that $\Pr_\kappa[Y_{1:n} = y \mid \mathcal{A}] = \prod_{t=1}^{n} p(y_t \mid x, y_{<t})$. Dividing by this positive probability therefore gives

$$\Pr_\kappa[M_t \le z_t \text{ for all } t \mid Y_{1:n} = y, \mathcal{A}] = \prod_{t=1}^{n} z_t. \tag{91}$$

This is the joint distribution function of independent uniforms. It proves independence after conditioning on the full text, including its future tokens, rather than only on each preceding prefix.

The pair $(y, \mathcal{A})$ determines all the scored sets $\{\mathcal{T}_c\}_{c=1}^{C}$. For a fixed chunk c, set $z_t = 1$ outside $\mathcal{T}_c$ in the last identity and average over assignment tables consistent with $(y, \{\mathcal{T}_c\}_{c=1}^{C})$. The product over $\mathcal{T}_c$ is fixed under this conditioning, so

$$\Pr_*[M_t \le z_t \text{ for all } t \in \mathcal{T}_c] = \mathbb{E}_*\left[\prod_{t \in \mathcal{T}_c} z_t\right] = \prod_{t \in \mathcal{T}_c} z_t. \tag{92}$$

Thus the selected M_t, $t \in \mathcal{T}_c$, remain independent $U(0, 1)$ variables under $\Pr_*$.

At every scored position, the correct candidate reconstructs the uniforms actually used by the encoder. Since Y_t maximises Z_t,

$$B_t := u_t^{(m_c)}(Y_t) = u_t(Y_t) = M_t^{p_t}. \tag{93}$$

Consequently, for $0 < b < 1$,

$$\Pr_*[B_t \le b] = \Pr_*[M_t \le b^{1/p_t}] = b^{1/p_t}. \tag{94}$$

Writing $\alpha_t = 1/p_t$, the variables B_t are therefore independent $\mathrm{Beta}(\alpha_t, 1)$ variables with densities $\alpha_t b^{\alpha_t - 1}$ on $(0, 1)$. In particular, the correct score has the representation

$$S := S_c(m_c) = \sum_{t \in \mathcal{T}_c} X_t, \qquad X_t = -\log(1 - B_t), \tag{95}$$

where the X_t are conditionally independent.

Independence of the wrong scores from the correct score. Write $\mathcal{W}_c = \{0, 1\}^{k_c} \setminus \{m_c\}$ for the wrong candidates. Let $\mathcal{H}$ be the complete encoder query transcript together with the decoder's chunk-selection queries, as in Appendix L. This information determines $(Y_{1:n}, \{\mathcal{T}_c\}_{c=1}^{C})$ and the correct score S.

For $t \in \mathcal{T}_c$ and $m' \in \mathcal{W}_c$, the wrong candidate uses the input

$$b_{t,m'} = \langle \mathsf{msg}, c_t, m', Y_t \rangle. \tag{96}$$

These inputs are distinct over all pairs (t, m'): scored positions have distinct contexts, and distinct candidates have distinct message fields. None is queried by the encoder. Indeed, a message query

with context c_t uses the true value m_c, while a query with another context has a different context field; the remaining query families use different tags. This also rules out collisions with encoder queries for other chunks, even if their true values happen to equal m'.

Let W be the vector of all reconstructed wrong-candidate uniforms $u_t^{(m')}(Y_t)$, indexed by these pairs. By the random-function argument in Appendix L, conditional on $\mathcal{H}$ its entries are independent uniforms. Hence for any vector w with entries $w_{t,m'} \in [0,1]$,

$$\Pr_\kappa[W \le w \mid \mathcal{H}] = \prod_{t \in \mathcal{T}_c} \prod_{m' \in \mathcal{W}_c} w_{t,m'}, \tag{97}$$

where $W \le w$ is interpreted coordinatewise. The right-hand side is constant once the scored sets have been fixed. Since S is determined by $\mathcal{H}$, iterated conditional expectation gives

$$\begin{aligned} \Pr_*[S \le s, W \le w] &= \mathbb{E}_*\Big[\mathbf{1}_{\{S \le s\}} \Pr_\kappa[W \le w \mid \mathcal{H}]\Big] \\ &= \Pr_*[S \le s] \prod_{t \in \mathcal{T}_c} \prod_{m' \in \mathcal{W}_c} w_{t,m'}. \end{aligned} \tag{98}$$

This factorisation proves that W is independent of S under $\Pr_*$, and that all its coordinates remain independent uniforms. Each wrong score is a function of its own coordinates of W. Thus, when $n_c \ge 1$, each $S_c(m')$ is independent of S and has the $\Gamma(n_c, 1)$ law derived in Appendix L. The wrong scores are also mutually independent, although the union bound below does not require that additional fact.

Computing the two exponential moments. Assume for now that $n_c \ge 1$, and fix $\lambda \in (0,1)$. For a wrong candidate, integrate against its Gamma density:

$$\begin{aligned} \mathbb{E}_*[e^{\lambda S_c(m')}] &= \frac{1}{(n_c-1)!} \int_0^\infty s^{n_c-1} e^{-(1-\lambda)s}\, \mathrm{d}s \\ &= \frac{(1-\lambda)^{-n_c}}{(n_c-1)!} \int_0^\infty q^{n_c-1} e^{-q}\, \mathrm{d}q \\ &= (1-\lambda)^{-n_c}. \end{aligned} \tag{99}$$

The second line substitutes $q = (1-\lambda)s$, and the last integral is $\Gamma(n_c) = (n_c-1)!$. The restriction $\lambda < 1$ ensures convergence. This is the moment-generating function of the wrong score, evaluated at the positive argument λ.

For a correct-score contribution with $a = 1/p$, define

$$\begin{aligned} L_p(\lambda) &:= \mathbb{E}[(1-B)^\lambda], \qquad B \sim \mathrm{Beta}(a,1), \\ L_p(\lambda) &= a \int_0^1 b^{a-1}(1-b)^\lambda\, \mathrm{d}b \\ &= a\, \frac{\Gamma(a)\Gamma(\lambda+1)}{\Gamma(a+\lambda+1)} \\ &= \frac{\Gamma(a+1)\Gamma(\lambda+1)}{\Gamma(a+\lambda+1)}. \end{aligned} \tag{100}$$

Here the integral is Euler's beta integral, and the last equality uses $\Gamma(a+1) = a\Gamma(a)$. For completeness, the beta identity for $a, b > 0$ follows from the Gamma integrals by the substitution $u = rz$, $v = r(1-z)$, whose Jacobian is r:

$$\begin{aligned} \Gamma(a)\Gamma(b) &= \int_0^\infty \int_0^\infty u^{a-1} v^{b-1} e^{-(u+v)}\, \mathrm{d}u\, \mathrm{d}v \\ &= \left(\int_0^\infty r^{a+b-1} e^{-r}\, \mathrm{d}r\right)\left(\int_0^1 z^{a-1}(1-z)^{b-1}\, \mathrm{d}z\right) \\ &= \Gamma(a+b) \int_0^1 z^{a-1}(1-z)^{b-1}\, \mathrm{d}z. \end{aligned} \tag{101}$$

All integrands are nonnegative, which justifies separating these integrals; division by $\Gamma(a+b) > 0$ gives the identity used above with $b = \lambda + 1$. Since $e^{-\lambda X_t} = (1-B_t)^\lambda$, we have established

$$\mathbb{E}_*[e^{-\lambda X_t}] = L_{p_t}(\lambda), \qquad \eta_p(\lambda) = \log(1-\lambda) - \log L_p(\lambda). \tag{102}$$

Bounding the probability that one wrong score wins. For a fixed $m' \in \mathcal{W}_c$, the positivity of λ gives

$$\{S_c(m') \geq S\} = \{e^{\lambda(S_c(m')-S)} \geq 1\}. \tag{103}$$

Markov's inequality applies to this nonnegative exponential. Using independence of $S_c(m')$ and S, and then independence of the correct-score contributions, yields

$$\begin{aligned}
\Pr_*[S_c(m') \geq S] &\leq \mathbb{E}_*[e^{\lambda(S_c(m')-S)}] \\
&= \mathbb{E}_*[e^{\lambda S_c(m')}]\mathbb{E}_*[e^{-\lambda S}] \\
&= (1-\lambda)^{-n_c} \prod_{t\in\mathcal{T}_c} \mathbb{E}_*[e^{-\lambda X_t}] \\
&= (1-\lambda)^{-n_c} \prod_{t\in\mathcal{T}_c} L_{p_t}(\lambda) \\
&= \exp\left(-\sum_{t\in\mathcal{T}_c} [\log(1-\lambda) - \log L_{p_t}(\lambda)] \right) \\
&= \exp\left(-\sum_{t\in\mathcal{T}_c} \eta_{p_t}(\lambda) \right).
\end{aligned} \tag{104}$$

The fifth line uses $|\mathcal{T}_c| = n_c$, placing one factor $(1-\lambda)^{-1}$ with each term of the product.

Optimising the exponent. Write $F_c(\lambda) = \sum_{t\in\mathcal{T}_c} \eta_{p_t}(\lambda)$. For each fixed $p > 0$, dominated convergence in the definition of L_p gives

$$\lim_{\lambda\downarrow 0} L_p(\lambda) = 1, \qquad \lim_{\lambda\uparrow 1} L_p(\lambda) = \mathbb{E}[1-B] = \frac{1}{1/p+1} > 0. \tag{105}$$

The integrands are bounded by 1, and the last value also follows by setting $\lambda = 1$ in the Gamma ratio above. Thus F_c is continuous on $(0,1)$, tends to 0 at the left endpoint, and, when $n_c \geq 1$, tends to $-\infty$ at the right endpoint. Continuity on each compact subinterval, together with these endpoint limits, shows that

$$0 \leq \Lambda_c := \sup_{0<\lambda<1} F_c(\lambda) < \infty. \tag{106}$$

The pairwise error bound holds for every $\lambda \in (0,1)$. Taking the infimum of its right-hand side, and using continuity and monotonicity of the exponential, therefore gives

$$\Pr_*[S_c(m') \geq S] \leq \inf_{0<\lambda<1} e^{-F_c(\lambda)} = e^{-\sup_{0<\lambda<1} F_c(\lambda)} = e^{-\Lambda_c}. \tag{107}$$

No maximiser inside $(0,1)$ is required: a sequence of values approaching the supremum gives the same bound.

From pairwise comparisons to chunk and message errors. Fix any rule for breaking ties in the decoder's maximisation. If $\hat{m}_c \neq m_c$, the selected wrong candidate has a score at least as large as $S_c(m_c) = S$. Hence

$$\{\hat{m}_c \neq m_c\} \subseteq \bigcup_{m'\in\mathcal{W}_c} \{S_c(m') \geq S\}. \tag{108}$$

This inclusion remains valid even if a tie occurs. The conditional union bound and $|\mathcal{W}_c| = 2^{k_c} - 1$ now give

$$\begin{aligned}
\Pr_*[\hat{m}_c \neq m_c] &\leq \sum_{m'\in\mathcal{W}_c} \Pr_*[S_c(m') \geq S] \\
&\leq (2^{k_c}-1)e^{-\Lambda_c} \\
&< 2^{k_c} e^{-\Lambda_c} = \exp(k_c \ln 2 - \Lambda_c).
\end{aligned} \tag{109}$$

The strict inequality uses $2^{k_c} - 1 < 2^{k_c}$ and $e^{-\Lambda_c} > 0$, the latter following from finiteness of Λ_c. This proves (83) when $n_c \geq 1$.

If $n_c = 0$, every candidate score is the empty sum 0, and $\Lambda_c = \sup_{0<\lambda<1} 0 = 0$. If there is at least one wrong candidate, the claimed bound is valid because

$$\Pr_*[\hat{m}_c \neq m_c] \leq 1 \leq 2^{k_c} - 1 = (2^{k_c} - 1)e^{-\Lambda_c}. \tag{110}$$

If there are no wrong candidates, the decoder can only select m_c, and both the error probability and the first bound are 0, for any value of n_c. These observations cover all degenerate cases.

All chunks were analysed under the same conditioning on $(Y_{1:n}, \{\mathcal{T}_c\}_{c=1}^C)$. Since the decoded concatenation differs from m precisely when at least one chunk differs, a second conditional union bound yields

$$\begin{aligned}\Pr_*[\hat{m} \neq m] &= \Pr_*\left[\bigcup_{c=1}^{C} \{\hat{m}_c \neq m_c\}\right] \\ &\leq \sum_{c=1}^{C} \Pr_*[\hat{m}_c \neq m_c] \\ &\leq \sum_{c=1}^{C} (2^{k_c} - 1)e^{-\Lambda_c}.\end{aligned} \tag{111}$$

No independence between chunks is needed. If one conditions only on the text, the scored sets are still random, so the corresponding bound is obtained by averaging their right-hand side:

$$\Pr_\kappa[\hat{m} \neq m \mid Y_{1:n} = y] \leq \mathbb{E}_\kappa\left[\sum_{c=1}^{C} (2^{k_c} - 1)e^{-\Lambda_c} \,\middle|\, Y_{1:n} = y\right]. \tag{112}$$

Monotonicity of $\eta_p(\lambda)$ in the token probability. Fix $\lambda \in (0, 1)$ and let $U \sim U(0, 1)$. The variable $B_p = U^p$ has distribution function $b^{1/p}$ on $(0, 1)$, so $B_p \sim \mathrm{Beta}(1/p, 1)$ and

$$L_p(\lambda) = \mathbb{E}[(1 - U^p)^\lambda]. \tag{113}$$

This represents the distributions for different p using the same uniform variable. If $0 < p_1 < p_2 \leq 1$, then for every $U \in (0, 1)$,

$$U^{p_1} > U^{p_2}, \qquad 1 - U^{p_1} < 1 - U^{p_2}, \qquad (1 - U^{p_1})^\lambda < (1 - U^{p_2})^\lambda. \tag{114}$$

Taking expectations gives $L_{p_1}(\lambda) < L_{p_2}(\lambda)$. Since the logarithm is strictly increasing, subtracting these logarithms from the same value $\log(1 - \lambda)$ gives $\eta_{p_1}(\lambda) > \eta_{p_2}(\lambda)$. In particular, $\eta_p(\lambda)$ is nonincreasing in p, as claimed.

The endpoint $p = 1$. When $p = 1$, the Beta variable is uniform and

$$L_1(\lambda) = \int_0^1 (1 - u)^\lambda \, \mathrm{d}u = \left[-\frac{(1 - u)^{\lambda+1}}{\lambda + 1}\right]_0^1 = \frac{1}{\lambda + 1}. \tag{115}$$

Substitution into the definition of η yields

$$\eta_1(\lambda) = \log(1 - \lambda) + \log(1 + \lambda) = \log(1 - \lambda^2) < 0, \tag{116}$$

because $0 < 1 - \lambda^2 < 1$. For example, if every scored token has $p_t = 1$, then $F_c(\lambda) = n_c \log(1 - \lambda^2)$ and $\Lambda_c = 0$, approached as $\lambda \downarrow 0$ when $n_c > 0$. This also illustrates why the supremum in the theorem need not be attained inside the open interval. □

P PROOF OF THEOREM 5

The main text states only the final log-likelihood ratio. The full statement below additionally gives the conditional candidate laws, the product likelihood, the maximum-likelihood and most-powerful-test conclusions, and the certified weighted decoder; it keeps the theorem's number and the main log-likelihood equation number.

Theorem 5 (Exact likelihood ratio, full statement)**.** *Treat* PRF *as a random function. Fix a chunk c with $k_c \geq 1$ and condition on the emitted text and the scored-position sets $\{\mathcal{T}_c\}_{c=1}^{C}$, fixing $p_t = p_t(y_t)$ for $t \in \mathcal{T}_c$. All scored uniforms $u_t^{(m')}(y_t)$ are conditionally independent across positions and candidates, with density $f_{\mathrm{correct}}(\cdot \mid p_t)$ when $m' = m_c$ and density $f_{\mathrm{wrong}} \equiv 1$ otherwise. For a fixed candidate m', the likelihood ratio of $H_1 : m_c = m'$ to $H_0 : m_c \neq m'$, based on that candidate's scored uniforms, is*

$$L_c(m') \;=\; \prod_{t\in\mathcal{T}_c} \frac{f_{\mathrm{correct}}\big(u_t^{(m')}(y_t) \mid p_t\big)}{f_{\mathrm{wrong}}\big(u_t^{(m')}(y_t)\big)} \;=\; \prod_{t\in\mathcal{T}_c} f_{p_t}\big(u_t^{(m')}(y_t)\big), \tag{117}$$

and its logarithm, the quantity the decoder computes, is the sum

$$\begin{aligned} \ell_c(m') = \log L_c(m') &= \sum_{t\in\mathcal{T}_c} \log f_{p_t}\big(u_t^{(m')}(y_t)\big) \\ &= \sum_{t\in\mathcal{T}_c} \Big[\big(\tfrac{1}{p_t} - 1\big) \log u_t^{(m')}(y_t) + \log \tfrac{1}{p_t}\Big]. \end{aligned} \tag{17}$$

The decoder $\arg\max_{m'} \ell_c(m')$ is the conditional maximum-likelihood decoder based on all candidates' scored uniforms. For distinct m', m'', the test of $H_0 : m_c = m''$ against $H_1 : m_c = m'$ that rejects for large $\ell_c(m') - \ell_c(m'')$ is most powerful at its conditional level, with randomisation at the threshold if needed.

More generally, let $w_t \geq 0$ be finite weights determined by the text, prompt and model, and write

$$R_{c,w}(m') = \sum_{t\in\mathcal{T}_c} w_t \log u_t^{(m')}(y_t), \qquad \hat{m}_{c,w} = \arg\max_{m'} R_{c,w}(m'). \tag{118}$$

Under a wrong candidate, $-R_{c,w}(m')$ has the law of $Z_{c,w} = \sum_{t\in\mathcal{T}_c : w_t > 0} w_t E_t$, where the E_t are independent $\mathrm{Exp}(1)$ *variables; zero weights contribute* 0*. Let $F_{c,w}(z) = \Pr[Z_{c,w} \leq z]$. The certificate*

$$\hat{\delta}_{c,w}(y) = \min\big\{1, (2^{k_c} - 1) F_{c,w}\big(-R_{c,w}(\hat{m}_{c,w})\big)\big\}, \tag{119}$$

set to 1 *if all weights are zero, gives the guarantee* (15) *when the decoder returns $\hat{m}_{c,w}$ if $\hat{\delta}_{c,w} \leq \delta$ and abstains otherwise.*

Proof. Fix the model, prompt, sampling settings and the values carried by all chunks other than c. Let $\mathcal{M}_c = \{0,1\}^{k_c}$ and $K = |\mathcal{M}_c| \geq 2$. To compare the possible values of chunk c, write H_a for the hypothesis that $m_c = a$, where $a \in \mathcal{M}_c$. This comparison does not require a prior distribution on the message.

Condition on a realised text $Y_{1:n} = y$ and scored-position sets $\{\mathcal{T}_c\}_{c=1}^{C}$ with positive probability. These fix $\mathcal{T}_c$, n_c and $p_t = p(y_t \mid x, y_{<t}) \in (0,1]$. Appendix O shows that the text law conditional on the chunk-assignment table is $\prod_t p(y_t \mid x, y_{<t})$, independently of the message. The assignment table itself is also independent of the message. Hence $(Y_{1:n}, \{\mathcal{T}_c\}_{c=1}^{C})$ has the same law under every H_a, and the same conditioning is valid for all candidates. Denote conditional probability and expectation under H_a by $\Pr_a$ and $\mathbb{E}_a$.

The joint observation and its conditional density. For $t \in \mathcal{T}_c$ and $b \in \mathcal{M}_c$, put

$$U_{t,b} = u_t^{(b)}(y_t), \qquad U_b = (U_{t,b})_{t\in\mathcal{T}_c}, \qquad \mathcal{U} = (U_b)_{b\in\mathcal{M}_c}. \tag{120}$$

Thus U_b is the column of scored uniforms for candidate b, and $\mathcal{U}$ is the array used to compare all candidates.

Under H_a, Appendix O proves that the coordinates of U_a are independent, with distribution functions u^{1/p_t} on $(0,1)$. Differentiation gives their densities:

$$\frac{\mathrm{d}}{\mathrm{d}u}u^{1/p_t} = \frac{1}{p_t}u^{1/p_t-1} = f_{p_t}(u). \tag{121}$$

The other columns consist of independent uniforms and are also independent of U_a. To justify this last joint assertion, let $\mathcal{H}$ contain the complete generation transcript and the decoder's chunk-selection queries, as in Appendix L. It determines both the conditioning information and the correct column U_a. Every entry in the remaining columns is read at a distinct input that was never queried during generation: at context c_t, the encoder uses a, while a wrong column uses $b \neq a$. Distinct retained contexts and the domain-separating tags rule out the other possible input collisions.

Consequently, conditional on $\mathcal{H}$, the array $U_{-a} = (U_b)_{b\neq a}$ has the product uniform law, denoted by ν, independently of the values in that transcript. For any measurable sets A and B, iterated conditional expectation gives

$$\begin{aligned}\Pr_a[U_a \in A, U_{-a} \in B] &= \mathbb{E}_a\Big[\mathbf{1}_{\{U_a\in A\}}\Pr_a[U_{-a}\in B \mid \mathcal{H}]\Big]\\ &= \mathbb{E}_a\big[\mathbf{1}_{\{U_a\in A\}}\nu(B)\big]\\ &= \Pr_a[U_a\in A]\,\nu(B).\end{aligned} \tag{122}$$

This factorisation establishes the required independence. In particular, for $n_c \geq 1$, the conditional density of the whole array at $u = (u_{t,b}) \in (0,1)^{n_cK}$ under H_a is

$$\begin{aligned}q_a(u) &= \left(\prod_{t\in\mathcal{T}_c} f_{p_t}(u_{t,a})\right)\left(\prod_{b\in\mathcal{M}_c\setminus\{a\}}\prod_{t\in\mathcal{T}_c} 1\right)\\ &= \prod_{t\in\mathcal{T}_c} f_{p_t}(u_{t,a}).\end{aligned} \tag{123}$$

All these densities are positive on the same observation space.

The likelihood ratio for one candidate. Fix a candidate b and consider only its column U_b. Under $H_1 : m_c = b$, its density at $z = (z_t)_{t\in\mathcal{T}_c}$ is

$$h_1(z) = \prod_{t\in\mathcal{T}_c} f_{p_t}(z_t). \tag{124}$$

Under any hypothesis H_a with $a \neq b$, this column instead has density

$$h_0(z) = \prod_{t\in\mathcal{T}_c} 1 = 1. \tag{125}$$

Although $H_0 : m_c \neq b$ contains several possible messages, all of them give exactly this same distribution for U_b. Thus this candidate-specific likelihood ratio is unambiguous without choosing a distribution over the wrong messages. Evaluating h_1/h_0 at the observed column gives

$$\frac{h_1(U_b)}{h_0(U_b)} = \prod_{t\in\mathcal{T}_c}\frac{f_{\mathrm{correct}}(U_{t,b}\mid p_t)}{f_{\mathrm{wrong}}(U_{t,b})} = \prod_{t\in\mathcal{T}_c} f_{p_t}(U_{t,b}) = L_c(b), \tag{126}$$

which proves (117).

Every factor is positive and finite because $p_t > 0$ and $0 < U_{t,b} < 1$. Taking logarithms is therefore valid, and gives

$$\begin{aligned}\ell_c(b) &= \log\prod_{t\in\mathcal{T}_c}\left[p_t^{-1}U_{t,b}^{1/p_t-1}\right]\\ &= \sum_{t\in\mathcal{T}_c}\left[\log p_t^{-1} + \log U_{t,b}^{1/p_t-1}\right]\\ &= \sum_{t\in\mathcal{T}_c}\left[\left(\frac{1}{p_t}-1\right)\log U_{t,b} + \log\frac{1}{p_t}\right].\end{aligned} \tag{127}$$

The second line uses the logarithm of a finite product, and the third uses $\log u^\beta = \beta \log u$ for $u > 0$. This proves (17). If $p_t = 1$, both coefficients in that token's contribution vanish, so its log-likelihood contribution is 0. If $\mathcal{T}_c$ is empty, the product is 1 and the sum is 0 for every candidate.

Maximum likelihood from the full array of scored uniforms. The preceding candidate-specific ratio uses one column at a time. To justify comparing candidates, we use the density of the same full observation $\mathcal{U}$ under each hypothesis. The joint density already derived satisfies

$$q_a(\mathcal{U}) = \prod_{t \in \mathcal{T}_c} f_{p_t}(U_{t,a}) = L_c(a). \tag{128}$$

Hence, as sets of maximisers,

$$\arg\max_{a \in \mathcal{M}_c} q_a(\mathcal{U}) = \arg\max_{a \in \mathcal{M}_c} L_c(a) = \arg\max_{a \in \mathcal{M}_c} \ell_c(a). \tag{129}$$

The last equality holds because the logarithm is strictly increasing. Choosing any fixed rule among tied maximisers thus gives a conditional maximum-likelihood decoder for this observation. The observation here consists of the scored uniforms for all candidates, together with the fixed text, scored sets and model probabilities.

The pairwise test and its rejection direction. Fix two distinct candidates b, d. For the simple hypotheses

$$H_0 : m_c = d, \qquad H_1 : m_c = b, \tag{130}$$

the likelihood ratio based on $\mathcal{U}$ is

$$\mathcal{R}(\mathcal{U}) = \frac{q_b(\mathcal{U})}{q_d(\mathcal{U})} = \frac{L_c(b)}{L_c(d)} = \exp\big(\ell_c(b) - \ell_c(d)\big). \tag{131}$$

A large value therefore favours H_1, so the rejection region for H_0 uses a large value of $\ell_c(b) - \ell_c(d)$.

For completeness, we prove the most-powerful assertion. Let $\alpha \in (0, 1)$ be the prescribed conditional test level. Choose a threshold $\tau > 0$ satisfying

$$\Pr_d[\mathcal{R} > \tau] \le \alpha \le \Pr_d[\mathcal{R} \ge \tau]. \tag{132}$$

Such a threshold is a quantile of the positive, finite random variable $\mathcal{R}$. Define the rejection probability

$$\phi^*(u) = \begin{cases} 1, & \mathcal{R}(u) > \tau, \\ \gamma, & \mathcal{R}(u) = \tau, \\ 0, & \mathcal{R}(u) < \tau, \end{cases} \tag{133}$$

where $\gamma \in [0, 1]$ is chosen so that

$$\mathbb{E}_d[\phi^*] = \Pr_d[\mathcal{R} > \tau] + \gamma \Pr_d[\mathcal{R} = \tau] = \alpha. \tag{134}$$

If the boundary has positive probability, take $\gamma = (\alpha - \Pr_d[\mathcal{R} > \tau])/\Pr_d[\mathcal{R} = \tau]$; the quantile inequalities place this number in $[0, 1]$. If the boundary has probability zero, the tail probability already equals α and any γ works.

Let $\phi(u) \in [0, 1]$ be any other test using this observation with $\mathbb{E}_d[\phi] \le \alpha$. At every u,

$$(\phi^*(u) - \phi(u))(q_b(u) - \tau q_d(u)) \ge 0. \tag{135}$$

Indeed, above the threshold both factors are nonnegative, below it both are nonpositive, and on the boundary the second factor is zero. Integrating this inequality over the observation space yields

$$\begin{aligned} \mathbb{E}_b[\phi^*] - \mathbb{E}_b[\phi] &\ge \tau\big(\mathbb{E}_d[\phi^*] - \mathbb{E}_d[\phi]\big) \\ &= \tau\big(\alpha - \mathbb{E}_d[\phi]\big) \ge 0. \end{aligned} \tag{136}$$

Thus ϕ^* has at least as much power as every competing level-α test. This is the Neyman–Pearson argument in the present conditional experiment. The endpoint levels $\alpha = 0, 1$ are handled by never rejecting and always rejecting, respectively; positivity of both densities ensures that a level-zero test also has zero power.

If $\mathcal{T}_c = \varnothing$, the observation is an empty array and all likelihoods are 1. The same decision problem then consists only of randomisation. Likewise, if every scored token has $p_t = 1$, all hypotheses induce the same product uniform law, $\ell_c(b) - \ell_c(d) = 0$, and a size-$\alpha$ test has power α. Boundary randomisation includes these cases.

The exact null distribution of a weighted log score. Return to a fixed true message m_c, and let the finite weights $w_t \geq 0$ be as in the theorem. Their dependence only on the text, prompt and model makes them fixed under our conditioning. For a wrong candidate $b \neq m_c$, define

$$X_{t,w} = -w_t \log U_{t,b}, \qquad Z_b = \sum_{t \in \mathcal{T}_c} X_{t,w} = -R_{c,w}(b). \tag{137}$$

If $w_t > 0$, then for $z \geq 0$ the uniform null law gives

$$\begin{aligned} \Pr_{m_c}[X_{t,w} > z] &= \Pr_{m_c}[-w_t \log U_{t,b} > z] \\ &= \Pr_{m_c}[\log U_{t,b} < -z/w_t] \\ &= \Pr_{m_c}[U_{t,b} < e^{-z/w_t}] \\ &= e^{-z/w_t}. \end{aligned} \tag{138}$$

Thus $X_{t,w}$ is exponential with rate $1/w_t$ and density $w_t^{-1} e^{-z/w_t}$ for $z > 0$. If $w_t = 0$, then $X_{t,w} = 0$ deterministically; no exponential rate is assigned to that term. The summands are independent because the $U_{t,b}$ are independent and the weights are fixed. Equivalently,

$$Z_b \overset{d}{=} Z_{c,w} = \sum_{t \in \mathcal{T}_c : w_t > 0} w_t E_t, \qquad E_t \text{ independent } \mathrm{Exp}(1). \tag{139}$$

This distribution is specified exactly even when positive weights repeat. To make its computation explicit, enumerate the active positions as $t_1, \ldots, t_r$ and put $\lambda_j = 1/w_{t_j}$. For $r \geq 1$, let g_j be the density of the sum of its first j exponential terms. Independence gives the convolution recursion, for $z > 0$,

$$g_1(z) = \lambda_1 e^{-\lambda_1 z}, \qquad g_j(z) = \int_0^z g_{j-1}(v) \lambda_j e^{-\lambda_j (z-v)} \, \mathrm{d}v \quad (2 \leq j \leq r). \tag{140}$$

Consequently,

$$F_{c,w}(z) = \begin{cases} 0, & z < 0, \\ \displaystyle\int_0^z g_r(v) \, \mathrm{d}v, & z \geq 0. \end{cases} \tag{141}$$

Each g_j is positive on $(0, \infty)$: this holds for g_1, and the recursion integrates a positive integrand over $(0, z)$. Hence $F_{c,w}$ is continuous and strictly increasing on $[0, \infty)$, with $F_{c,w}(0) = 0$ and limit 1 at infinity.

An equivalent expression for the Laplace transform is

$$\mathbb{E}[e^{-sZ_{c,w}}] = \prod_{j=1}^{r} \int_0^\infty \lambda_j e^{-(\lambda_j + s)z} \, \mathrm{d}z = \prod_{j=1}^{r} \frac{\lambda_j}{\lambda_j + s} = \prod_{j=1}^{r} (1 + s w_{t_j})^{-1}, \qquad s \geq 0. \tag{142}$$

If all active weights equal $w > 0$, then $Z_{c,w}/w \sim \Gamma(r, 1)$; unequal weights are covered by the same convolution recursion. If $r = 0$, the sum is zero with probability one.

The appropriate tail and the certification bound. The weighted decoder maximises $R_{c,w}$, whereas its known nonnegative null variable is $Z_{c,w} = -R_{c,w}$. For any fixed score value $s \le 0$,

$$\Pr[-Z_{c,w} \ge s] = \Pr[Z_{c,w} \le -s] = F_{c,w}(-s). \tag{143}$$

Thus the upper tail of the score corresponds to the *lower* tail of its negation. Substituting the selected score into this null tail and multiplying by the number of wrong candidates gives exactly

$$\hat{\delta}_{c,w} = \min\{1, (K-1)F_{c,w}(-R_{c,w}(\hat{m}_{c,w}))\}. \tag{144}$$

We now verify its error guarantee, including the selection of $\hat{m}_{c,w}$. At level $\delta = 1$ the claim is immediate because every error probability is at most 1. For $0 < \delta < 1$, if there are no positive weights, the certificate is defined to be 1 and the decoder always abstains. It remains to consider $r \ge 1$.

Set $q_\delta = \delta/(K-1) \in (0,1)$. The continuity and strict increase of $F_{c,w}$ give a unique $z_\delta > 0$ such that

$$F_{c,w}(z_\delta) = q_\delta. \tag{145}$$

Since $\delta < 1$, clipping the certificate at 1 does not change its acceptance event. Therefore

$$\begin{aligned} \hat{\delta}_{c,w} \le \delta \quad &\Longleftrightarrow \quad F_{c,w}(-R_{c,w}(\hat{m}_{c,w})) \le q_\delta \\ &\Longleftrightarrow \quad -R_{c,w}(\hat{m}_{c,w}) \le z_\delta. \end{aligned} \tag{146}$$

Let $D_{\delta,w}$ denote the decoder that returns $\hat{m}_{c,w}$ on this event and returns $\perp$ otherwise. If it accepts a wrong candidate, that candidate's negated score must meet the last inequality. Hence, under the current conditioning,

$$\{D_{\delta,w}(y) \notin \{m_c, \perp\}\} \subseteq \bigcup_{b \ne m_c} \{Z_b \le z_\delta\}. \tag{147}$$

Every fixed wrong candidate has the same null distribution $F_{c,w}$. Applying the union bound thus gives

$$\begin{aligned} \Pr_{m_c}[D_{\delta,w}(y) \notin \{m_c, \perp\}] &\le \sum_{b \ne m_c} \Pr_{m_c}[Z_b \le z_\delta] \\ &= (K-1)F_{c,w}(z_\delta) \\ &= (K-1)\frac{\delta}{K-1} = \delta. \end{aligned} \tag{148}$$

This step needs no independence between candidate scores and works with any fixed tie-breaking rule. The bound holds for every conditioning pair of positive probability, so averaging over both the text and the scored sets gives

$$\begin{aligned} &\Pr_{\kappa}[D_{\delta,w}(Y_{1:n}) \notin \{m_c, \perp\}] \\ &\quad = \mathbb{E}_\kappa\Big[\Pr_{\kappa}[D_{\delta,w}(Y_{1:n}) \notin \{m_c, \perp\} \mid Y_{1:n}, \{\mathcal{T}_c\}_{c=1}^{C}]\Big] \le \delta. \end{aligned} \tag{149}$$

This is the guarantee (15) for the weighted decoder.

Applying the certificate to the model-aware score. Finally, define

$$w_t^* = \frac{1}{p_t} - 1 \geq 0, \qquad A_c = \sum_{t \in \mathcal{T}_c} \log \frac{1}{p_t}. \tag{150}$$

The log-likelihood formula proved above becomes

$$\ell_c(b) = A_c + R_{c,w^*}(b). \tag{151}$$

The term A_c does not depend on the candidate, so maximising ℓ_c is equivalent to maximising R_{c,w^*}. For the selected candidate $\hat{m}_\ell = \arg\max_b \ell_c(b)$, its negated weighted score is

$$-R_{c,w^*}(\hat{m}_\ell) = A_c - \ell_c(\hat{m}_\ell). \tag{152}$$

The corresponding certificate is therefore

$$\min\left\{1, (K-1)F_{c,w^*}\big(A_c - \ell_c(\hat{m}_\ell)\big)\right\}, \tag{153}$$

with value 1 when all w_t^* vanish. In particular, tokens with $p_t = 1$ have zero weight and contribute neither to the likelihood score nor to its null sum. If all scored tokens are deterministic, or if there are no scored positions, the decoder abstains at every level $\delta < 1$. This completes both the likelihood and certification claims. □

Q Proof of Theorem 6

The main text states the contamination model, the robust score and its guarantees in brief. The full statement, which the proof below establishes, follows; it keeps the theorem's number and equation number.

Theorem 6 (Robust decoding under contamination, full statement)**.** *Fix $\varepsilon \in (0,1)$ and a chunk c with $k_c \geq 1$. Condition on the text and scored-position sets $\{\mathcal{T}_c\}_{c=1}^C$, and assume $p_t = p_t(y_t) \in (0,1]$ for every $t \in \mathcal{T}_c$. In the contamination model, the labels of scored tokens are independent, with probabilities $1-\varepsilon$ of being* watermarked *and ε of being* foreign*. Conditional on these labels, the scored uniforms for each fixed candidate are independent across positions. For the true candidate, their laws are $\mathrm{Beta}(1/p_t, 1)$ at watermarked positions and $U(0,1)$ at foreign positions; for every wrong candidate they are $U(0,1)$ at all positions, regardless of the labels. Then the log-likelihood ratio of $H_1 : m_c = m'$ to $H_0 : m_c \neq m'$, based on candidate m''s scored uniforms, is*

$$r_c(m') \;=\; \sum_{t \in \mathcal{T}_c} \log\Big[(1-\varepsilon)\, f_{p_t}\big(u_t^{(m')}(y_t)\big) + \varepsilon\Big], \tag{19}$$

and (i) each summand lies between $\log\varepsilon$ and $\log\big(\varepsilon + (1-\varepsilon)/p_t\big)$; (ii) a token with $p_t = 1$ contributes exactly 0; and (iii) every wrong candidate's score has the exact null law

$$Z_{c,\varepsilon} = \sum_{t \in \mathcal{T}_c} \log\big[\varepsilon + (1-\varepsilon) f_{p_t}(U_t)\big], \qquad U_t \text{ i.i.d. } U(0,1). \tag{154}$$

Write $Q_{c,\varepsilon}(s) = \Pr[Z_{c,\varepsilon} \geq s]$ and $\hat{m}_{c,\varepsilon} = \arg\max_{m'} r_c(m')$. The certificate

$$\hat{\delta}_{c,\varepsilon} = \min\big\{1, (2^{k_c} - 1) Q_{c,\varepsilon}\big(r_c(\hat{m}_{c,\varepsilon})\big)\big\} \tag{155}$$

gives the error bound of Theorem 4 under this model: returning $\hat{m}_{c,\varepsilon}$ only when $\hat{\delta}_{c,\varepsilon} \leq \delta$ and abstaining otherwise has probability at most δ of returning an incorrect chunk. If $\mathcal{T}_c$ is empty or every $p_t = 1$, this certificate equals 1.

Proof. Fix the chunk c, the contamination parameter $\varepsilon \in (0,1)$, the prompt, model and sampling settings. Condition on the text $Y_{1:n} = y$ and scored-position sets $\{\mathcal{T}_c\}_{c=1}^C$. Thus $\mathcal{T}_c$, n_c and all $p_t = p_t(y_t) \in (0,1]$ are fixed. Write $\mathcal{M}_c = \{0,1\}^{k_c}$ and $K = |\mathcal{M}_c| \geq 2$. The parameter ε is fixed before observing the candidate uniforms. All conditional distributions below are those of the contamination model stated in the theorem.

Marginalising the contamination labels. Fix a candidate $a \in \mathcal{M}_c$ and abbreviate $U_t = u_t^{(a)}(y_t)$ for $t \in \mathcal{T}_c$. Let $B_t = 1$ mean that position t is watermarked and $B_t = 0$ mean that it is foreign. Under the model, the B_t are independent, with

$$\Pr[B_t = 1] = 1 - \varepsilon, \qquad \Pr[B_t = 0] = \varepsilon. \tag{156}$$

For the hypothesis $H_1 : m_c = a$, the conditional density of U_t is $f_{p_t}(u)$ when $B_t = 1$ and 1 when $B_t = 0$. The law of total probability therefore gives the marginal density

$$h_{p_t,\varepsilon}(u) = (1-\varepsilon)f_{p_t}(u) + \varepsilon = \frac{1-\varepsilon}{p_t}u^{1/p_t-1} + \varepsilon, \qquad 0 < u < 1. \tag{157}$$

This is a probability density: both component densities integrate to 1, so

$$\int_0^1 h_{p_t,\varepsilon}(u)\,\mathrm{d}u = (1-\varepsilon)\int_0^1 f_{p_t}(u)\,\mathrm{d}u + \varepsilon\int_0^1 1\,\mathrm{d}u = (1-\varepsilon) + \varepsilon = 1. \tag{158}$$

We also need the *joint* density across scored positions. Independence of the labels and independence of the uniforms conditional on those labels are both used here. For a label vector $b = (b_t)_{t\in\mathcal{T}_c} \in \{0,1\}^{\mathcal{T}_c}$, its probability is $\prod_t (1-\varepsilon)^{b_t}\varepsilon^{1-b_t}$. Conditional on this vector and H_1, the uniforms have joint density $\prod_t f_{p_t}(u_t)^{b_t}$. Summing over all label vectors yields

$$\begin{aligned} q_1((u_t)_{t\in\mathcal{T}_c}) &= \sum_{b\in\{0,1\}^{\mathcal{T}_c}} \prod_{t\in\mathcal{T}_c} \left[(1-\varepsilon)^{b_t}\varepsilon^{1-b_t} f_{p_t}(u_t)^{b_t}\right] \\ &= \prod_{t\in\mathcal{T}_c} \left[\sum_{b_t\in\{0,1\}} (1-\varepsilon)^{b_t}\varepsilon^{1-b_t} f_{p_t}(u_t)^{b_t}\right] \\ &= \prod_{t\in\mathcal{T}_c} \left[\varepsilon + (1-\varepsilon)f_{p_t}(u_t)\right] \\ &= \prod_{t\in\mathcal{T}_c} h_{p_t,\varepsilon}(u_t). \end{aligned} \tag{159}$$

The second equality distributes the finite sum over all binary label choices into one two-term sum per position. It shows explicitly that independence across positions survives marginalisation over the unobserved labels.

Under $H_0 : m_c \neq a$, every U_t is uniform even conditional on the labels, and these uniforms are independent across positions by assumption. Their conditional joint density is therefore 1 for every label vector. Averaging over the labels leaves

$$q_0((u_t)_{t\in\mathcal{T}_c}) = \sum_{b\in\{0,1\}^{\mathcal{T}_c}} \prod_{t\in\mathcal{T}_c} (1-\varepsilon)^{b_t}\varepsilon^{1-b_t} = \prod_{t\in\mathcal{T}_c} \big[(1-\varepsilon) + \varepsilon\big] = 1. \tag{160}$$

This is the same density for every possible true value other than a. Hence the composite hypothesis $m_c \neq a$ induces one common null law for this candidate's observed uniforms; no prior over the other message values is required.

The likelihood ratio and the robust score. Since $h_{p,\varepsilon}(u) \geq \varepsilon > 0$ on $(0,1)$, the likelihood ratio is positive and its logarithm is defined. Evaluating the two densities at the candidate's observed uniforms gives

$$\begin{aligned} \log\frac{q_1((U_t)_{t\in\mathcal{T}_c})}{q_0((U_t)_{t\in\mathcal{T}_c})} &= \log\prod_{t\in\mathcal{T}_c} h_{p_t,\varepsilon}(U_t) \\ &= \sum_{t\in\mathcal{T}_c} \log h_{p_t,\varepsilon}(U_t) \\ &= \sum_{t\in\mathcal{T}_c} \log\big[\varepsilon + (1-\varepsilon)f_{p_t}(U_t)\big] \\ &= r_c(a). \end{aligned} \tag{161}$$

This proves (19). If $\mathcal{T}_c = \varnothing$, the joint likelihoods are both the empty product 1, their ratio is 1, and the robust score is the empty sum 0.

Bounds on a token's score contribution. For $p \in (0,1]$, the exponent $1/p - 1$ is nonnegative. Consequently, for $0 < u < 1$,

$$0 \leq u^{1/p-1} \leq 1, \qquad 0 \leq f_p(u) = \frac{1}{p}u^{1/p-1} \leq \frac{1}{p}. \tag{162}$$

Multiplication by $1-\varepsilon>0$ and addition of ε preserve these inequalities:

$$\varepsilon \le h_{p,\varepsilon}(u) \le \varepsilon + \frac{1-\varepsilon}{p}. \tag{163}$$

The logarithm is increasing on $(0,\infty)$, so the per-token score

$$s_{p,\varepsilon}(u) := \log h_{p,\varepsilon}(u) \tag{164}$$

satisfies

$$\log \varepsilon \le s_{p,\varepsilon}(u) \le \log\left(\varepsilon + \frac{1-\varepsilon}{p}\right). \tag{165}$$

Both endpoints are finite under the stated assumptions. These bounds are algebraic and therefore hold for every candidate and for either contamination label.

To state the negative-contribution bound precisely, let $r_{c,-t}(a)$ be the sum of all the other fixed contributions. Then

$$r_c(a) = r_{c,-t}(a) + s_{p_t,\varepsilon}(U_t) \ge r_{c,-t}(a) + \log\varepsilon = r_{c,-t}(a) - \log(1/\varepsilon). \tag{166}$$

Thus adding this summand can lower the existing sum by at most $\log(1/\varepsilon)$. This proves claim (i) and the stated bound on a token's additive negative contribution.

If $p=1$, then $f_1(u)=u^0=1$ for every $u\in(0,1)$. In this case

$$h_{1,\varepsilon}(u) = (1-\varepsilon)+\varepsilon = 1, \qquad s_{1,\varepsilon}(u) = \log 1 = 0. \tag{167}$$

This proves claim (ii), for watermarked and foreign tokens alike.

The exact null law of one summand. For a wrong candidate, the preceding joint-density calculation shows that its U_t are independent $U(0,1)$ variables. Applying the fixed function $s_{p_t,\varepsilon}$ separately to each coordinate preserves independence. Thus its score has exactly the law

$$Z_{c,\varepsilon} = \sum_{t\in\mathcal{T}_c} s_{p_t,\varepsilon}(U_t), \qquad U_t \text{ independent } U(0,1). \tag{168}$$

This already identifies the null distribution using only the fixed probabilities p_t and ε. We now derive its component distributions and an explicit recursion for their sum.

Let $U\sim U(0,1)$ and first suppose $0<p<1$. Put

$$a_\varepsilon = \log\varepsilon, \qquad b_{p,\varepsilon} = \log\left(\varepsilon + \frac{1-\varepsilon}{p}\right). \tag{169}$$

Because $(1-p)/p>0$, the function

$$u \longmapsto s_{p,\varepsilon}(u) = \log\left(\varepsilon + \frac{1-\varepsilon}{p}u^{(1-p)/p}\right) \tag{170}$$

is continuous and strictly increasing on $(0,1)$, with endpoint limits a_ε and $b_{p,\varepsilon}$. For $a_\varepsilon < x < b_{p,\varepsilon}$, inversion gives

$$\begin{aligned} s_{p,\varepsilon}(U)\le x \quad &\Longleftrightarrow \quad \varepsilon + \frac{1-\varepsilon}{p}U^{(1-p)/p} \le e^x \\ &\Longleftrightarrow \quad U^{(1-p)/p} \le \frac{p(e^x-\varepsilon)}{1-\varepsilon} \\ &\Longleftrightarrow \quad U \le \left(\frac{p(e^x-\varepsilon)}{1-\varepsilon}\right)^{p/(1-p)}. \end{aligned} \tag{171}$$

On this interval, the last threshold lies in $(0,1)$. Using the uniform distribution function therefore yields the exact null distribution function

$$F_{p,\varepsilon}(x) = \begin{cases} 0, & x \le a_\varepsilon, \\ \left(\dfrac{p(e^x-\varepsilon)}{1-\varepsilon}\right)^{p/(1-p)}, & a_\varepsilon < x < b_{p,\varepsilon}, \\ 1, & x \ge b_{p,\varepsilon}. \end{cases} \tag{172}$$

Its endpoint limits are 0 and 1, so this distribution is continuous. On the open support interval its density is

$$d_{p,\varepsilon}(x) = \frac{p}{1-p}\left(\frac{p}{1-\varepsilon}\right)^{p/(1-p)} e^x (e^x - \varepsilon)^{p/(1-p)-1} > 0. \tag{173}$$

This expression follows by differentiating the interior formula with the chain rule. When $p = 1$, the summand is instead identically zero, and its distribution function is $F_{1,\varepsilon}(x) = \mathbf{1}\{x \geq 0\}$.

Obtaining the null distribution of the sum. Enumerate $\mathcal{T}_c = \{t_1, \ldots, t_{n_c}\}$ and let G_j be the distribution function of the sum of its first j null contributions. The initial empty sum gives $G_0(x) = \mathbf{1}\{x \geq 0\}$. Independence of the next uniform and the previous sum gives, for $j = 1, \ldots, n_c$,

$$G_j(x) = \int_0^1 G_{j-1}\big(x - s_{p_{t_j},\varepsilon}(u)\big)\,\mathrm{d}u. \tag{174}$$

Indeed, conditional on $U_{t_j} = u$, the sum is at most x exactly when the previous sum is at most $x - s_{p_{t_j},\varepsilon}(u)$; integration averages over the uniform density 1. Thus $F_{c,\varepsilon} := G_{n_c}$ is the exact null distribution function. A position with $p_t = 1$ leaves the recursion unchanged because its score contribution is zero.

The inclusive upper tail used in the certificate is

$$Q_{c,\varepsilon}(x) = \Pr[Z_{c,\varepsilon} \geq x] = 1 - F_{c,\varepsilon}(x-), \tag{175}$$

where $F(x-) = \lim_{z \uparrow x} F(z)$. The left limit makes this identity valid even for a point mass.

For use in the error bound, let $J_c = \{t \in \mathcal{T}_c : p_t < 1\}$ be the positions with nonconstant score contributions. If $J_c = \varnothing$, the null sum is identically zero. Otherwise its support endpoints are

$$\underline{s}_c = |J_c| \log \varepsilon, \qquad \overline{s}_c = \sum_{t \in J_c} \log\left(\varepsilon + \frac{1-\varepsilon}{p_t}\right). \tag{176}$$

The sum has a continuous distribution and a density positive on $(\underline{s}_c, \overline{s}_c)$. To see positivity, each component has the positive density derived above. If two densities are positive on intervals (A_1, B_1) and (A_2, B_2), then at any $x \in (A_1 + A_2, B_1 + B_2)$ their convolution integrates a positive function over

$$\big(\max\{A_1, x - B_2\},\ \min\{B_1, x - A_2\}\big), \tag{177}$$

which is a nonempty interval. Repeating this argument proves the assertion for every finite number of active positions. The density also shows there are no point masses. Consequently $Q_{c,\varepsilon}$ is continuous, equals 1 at $\underline{s}_c$, equals 0 at $\overline{s}_c$, and is strictly decreasing between them.

Certification after selecting the largest robust score. Fix the actual true chunk m_c. Let $\Pr_*$ denote probability in the contamination model conditional on the text and scored sets fixed at the beginning of the proof. Choose any fixed tie-breaking rule, and write

$$\hat{m} = \arg\max_{a \in \mathcal{M}_c} r_c(a), \qquad \hat{\delta} = \min\{1, (K-1) Q_{c,\varepsilon}(r_c(\hat{m}))\}. \tag{178}$$

The certified decoder $D_{\delta,\varepsilon}$ returns $\hat{m}$ when $\hat{\delta} \leq \delta$ and returns $\perp$ otherwise. We must bound the event that it accepts a wrong candidate.

If $\delta = 1$, the claim follows because every event has probability at most 1. Suppose $0 < \delta < 1$. If $J_c = \varnothing$, every candidate's score is 0, including the case $\mathcal{T}_c = \varnothing$. The inclusive tail gives

$$Q_{c,\varepsilon}(0) = 1, \qquad \hat{\delta} = \min\{1, K-1\} = 1. \tag{179}$$

Since $K \geq 2$, the decoder therefore abstains and its error probability is zero. Using an inclusive tail is essential here: the null law assigns all its mass to the observed score 0.

Now assume $J_c \neq \varnothing$. Since $0 < \delta/(K-1) < 1$, continuity and strict decrease of the null tail give a unique $s_\delta \in (\underline{s}_c, \overline{s}_c)$ such that

$$Q_{c,\varepsilon}(s_\delta) = \frac{\delta}{K-1}. \tag{180}$$

For $\delta < 1$, clipping at 1 does not change whether the certificate is at most δ. Monotonicity of the tail then gives

$$\begin{aligned} \hat{\delta} \le \delta \quad &\Longleftrightarrow \quad (K-1)Q_{c,\varepsilon}(r_c(\hat{m})) \le \delta \\ &\Longleftrightarrow \quad Q_{c,\varepsilon}(r_c(\hat{m})) \le Q_{c,\varepsilon}(s_\delta) \\ &\Longleftrightarrow \quad r_c(\hat{m}) \ge s_\delta. \end{aligned} \tag{181}$$

Therefore, under the fixed conditioning,

$$\begin{aligned} \{D_{\delta,\varepsilon}(y) \notin \{m_c, \perp\}\} &= \{\hat{m} \ne m_c\} \cap \{r_c(\hat{m}) \ge s_\delta\} \\ &\subseteq \bigcup_{a \ne m_c} \{r_c(a) \ge s_\delta\}. \end{aligned} \tag{182}$$

The inclusion holds because an accepted wrong candidate is itself one of the wrong candidates whose score crosses the threshold. It does not assign the null law to the selected maximum.

Bounding and averaging the error probability. For each fixed wrong candidate, the null calculation gives

$$\Pr_*[r_c(a) \ge s_\delta] = Q_{c,\varepsilon}(s_\delta) = \frac{\delta}{K-1}. \tag{183}$$

Applying the conditional union bound over the $K-1$ wrong candidates yields

$$\begin{aligned} \Pr_*[D_{\delta,\varepsilon}(y) \notin \{m_c, \perp\}] &\le \sum_{a \ne m_c} \Pr_*[r_c(a) \ge s_\delta] \\ &= (K-1)\frac{\delta}{K-1} = \delta. \end{aligned} \tag{184}$$

Independence between different candidates is not needed. The mixture law of the correct candidate identifies the score as a likelihood ratio; the certification bound uses only the independent-uniform law within each wrong candidate.

The preceding bound holds for every conditioning pair satisfying the theorem's assumptions. Averaging over the text and scored sets, whenever they are random in the contamination model, gives

$$\begin{aligned} &\Pr[D_{\delta,\varepsilon}(Y_{1:n}) \notin \{m_c, \perp\}] \\ &\quad = \mathbb{E}\big[\Pr[D_{\delta,\varepsilon}(Y_{1:n}) \notin \{m_c, \perp\} \mid Y_{1:n}, \{\mathcal{T}_c\}_{c=1}^{C}]\big] \le \delta. \end{aligned} \tag{185}$$

The probability here is taken in the stated contamination model. This establishes the certified error guarantee and completes claim (iii), together with the likelihood formula, the score bounds, and all degenerate cases. □

R ADDITIONAL THEOREMS

Theorem 1 concerns a single generation. With a fixed key, repeating the same prompt and message produces the same text, and two generations can read the same sampling uniforms whenever their contexts and active message values agree. To separate their sampling randomness, draw a nonce ν uniformly from $\{0,1\}^r$ for every generation, independently of the key and of all other nonces. Replace (5) by

$$\begin{aligned} a_t &= \begin{cases} (\mathsf{fresh},\, \nu,\, c_t,\, t), & c_t \in \mathcal{C}_{t-1}, \\ (\mathsf{msg},\, \nu,\, c_t,\, m_{i_t}), & c_t \notin \mathcal{C}_{t-1}, \end{cases} \\ u_t(v) &= \rho\big(\mathrm{PRF}_\kappa(a_t, v)\big), \qquad \mathcal{C}_t = \mathcal{C}_{t-1} \cup \{c_t\}. \end{aligned} \tag{186}$$

The seen-context set starts empty in each generation. Chunk selection (4) and token emission (6) are unchanged.

The decoder treats the nonce as part of each candidate. For chunk c, it enumerates

$$\mathcal{P}_c = \{0,1\}^r \times \{0,1\}^{k_c}, \qquad K_c = |\mathcal{P}_c| = 2^{r+k_c}, \tag{187}$$

and computes

$$\begin{aligned} u_t^{(\nu',m')}(v) &= \rho\big(\mathrm{PRF}_\kappa(\mathsf{msg}, \nu', c_t, m', v)\big), \\ S_c(\nu', m') &= \sum_{t \in \mathcal{T}_c} -\log\big(1 - u_t^{(\nu',m')}(y_t)\big). \end{aligned} \tag{188}$$

Choose a fixed rule for breaking ties and write $(\hat{\nu}_c, \hat{m}_c) = \arg\max_{(\nu', m') \in \mathcal{P}_c} S_c(\nu', m')$. The returned chunk is the message component $\hat{m}_c$. When $r = 0$, there is a single possible nonce, so this construction reduces to the original scheme up to the fixed extra field in the PRF arguments.

Theorem 8 (Distortion-free generation under one deployed key)**.** *Treat PRF_κ as one random function shared by all generations. Fix a nonnegative integer r, an integer $N \geq 1$, the model and sampling settings, and prompts $x^{(j)}$, messages $m^{(j)} \in \{0,1\}^L$ and lengths n_j for $j = 1, \ldots, N$. Let $Y^{(j)}$ be generated by* (186) *and* (6)*, using independent uniform nonces $\nu_j \in \{0,1\}^r$ that are independent of the random function. Write $\Pr_{\kappa,\nu}$ for probability over that function and the nonces, and define*

$$P_j(w) = \prod_{t=1}^{n_j} p(w_t \mid x^{(j)}, w_{<t}), \qquad D = \{\nu_1, \ldots, \nu_N \text{ are pairwise distinct}\}. \tag{189}$$

Then the following hold.

(i) ***Single-generation law.*** *For every j and $w \in V^{n_j}$, $\Pr_{\kappa,\nu}[Y^{(j)} = w] = P_j(w)$.*

(ii) ***Joint law for distinct nonces.*** *If $N \leq 2^r$, then $\Pr_{\kappa,\nu}[D] > 0$ and, for all $w^{(j)} \in V^{n_j}$,*

$$\Pr_{\kappa,\nu}\left[\bigcap_{j=1}^{N}\{Y^{(j)} = w^{(j)}\} \,\middle|\, D\right] = \prod_{j=1}^{N} P_j(w^{(j)}). \tag{190}$$

Thus, conditional on distinct nonces, the generations are independent unwatermarked samples, even when their fixed prompts or messages coincide.

(iii) ***Collision and total variation bounds.*** *Let P be the joint law of $(Y^{(1)}, \ldots, Y^{(N)})$ and $Q = \bigotimes_{j=1}^{N} P_j$. Then*

$$d_{\mathrm{TV}}(P, Q) \leq \Pr_{\kappa,\nu}[D^c] \leq \min\left\{1, \frac{N(N-1)}{2^{r+1}}\right\}. \tag{191}$$

This holds for every N, including $N > 2^r$.

(iv) ***Decoding and certification for one generation.*** *Fix one generation and a chunk c with $k_c \geq 1$, and write ν for its actual nonce. Conditional on its text, its scored-position sets $\{\mathcal{T}_c\}_{c=1}^{C}$ and ν, every pair $(\nu', m') \neq (\nu, m_c)$ has score law $\Gamma(n_c, 1)$ when $n_c \geq 1$, and score 0 when $n_c = 0$.*

Let $G_n \sim \Gamma(n, 1)$ for $n \geq 1$, let $G_0 = 0$ almost surely, and put $Q_n(s) = \Pr[G_n \geq s]$. The certificate

$$\hat{\delta}_c = \min\left\{1, (K_c - 1) Q_{n_c}\big(S_c(\hat{\nu}_c, \hat{m}_c)\big)\right\} \tag{192}$$

has the guarantee of (15)*: returning $\hat{m}_c$ when $\hat{\delta}_c \leq \delta$ and abstaining otherwise has probability at most δ of returning an incorrect chunk, for $\delta \in (0, 1]$. Moreover, for Λ_c defined in* (83)*,*

$$\Pr_{\kappa,\nu}[\hat{m}_c \neq m_c \mid Y_{1:n} = y, \{\mathcal{T}_c\}_{c=1}^{C}] \leq (K_c - 1)e^{-\Lambda_c} < \exp\big((k_c + r)\ln 2 - \Lambda_c\big). \tag{193}$$

The corresponding whole-message bound is $\sum_{c=1}^{C}(2^{k_c+r} - 1)e^{-\Lambda_c}$. An empty scored set has $\hat{\delta}_c = 1$ and $\Lambda_c = 0$.

The nonce separates the sampling inputs of generations whose nonces differ. The decoder searches 2^r times as many candidates, so its certificate and error bound have the same candidate-count factor as a message with r additional bits. As in Theorem 7, about $r \ln 2$ additional accumulated evidence offsets this factor. The decoder searches for the nonce jointly with the message using the text and key. Our experiments use $r = 0$.

Proof. Keep the model, sampling settings, prompts, messages and lengths fixed as in the theorem. Throughout the proof, $\Pr$ and $\mathbb{E}$ denote probability and expectation over the single shared random function and the independent nonces. The function is drawn once and reused; it is not redrawn between generations.

Let $\mathcal{A}$ be the complete context-to-chunk assignment table determined by the inputs tagged chunk. These inputs are disjoint from all sampling inputs, whose tags are msg or fresh. Under the continuous random-function idealisation following (30), conditioning on $\mathcal{A}$ therefore leaves the sampling values independent $U(0,1)$ variables. The nonces are also independent of $\mathcal{A}$ and of those values. We first work with $\mathcal{A}$ fixed, and later average over it. This conditioning is useful because the assignment table is shared even when the nonces differ.

Step 1: the law of one generation with its nonce fixed. Fix j and a nonce value $a \in \{0,1\}^r$, and condition on $(\mathcal{A}, \nu_j = a)$. Let $\mathcal{H}_t^{(j)}$ contain this information and all queries and answers in generation j before its step-t sampling vector is read. It determines the prefix, the context, the seen-context set, the active chunk and the next-token law.

Every sampling input at step t is new within this generation. To verify this, consider the two cases in (186). On a first occurrence of the context, the input for token v is $\langle \mathsf{msg}, a, c_t, m_{i_t}, v\rangle$. Its context field differs from that of every earlier message input. On a repeated context, the input is $\langle \mathsf{fresh}, a, c_t, t, v\rangle$; its position field differs from that of every earlier fresh input. The tags separate these two families from each other and from chunk-selection inputs. Finally, distinct tokens v give distinct inputs because the tuple encoding is injective.

A random function can be exposed as queries arrive: an unqueried input receives an independent value, and a repeated input returns its stored value. The step-t input is chosen from the preceding history, so adaptively choosing it introduces no conditioning on its as-yet unobserved value. Thus, conditional on $\mathcal{H}_t^{(j)}$, the sampling vector consists of independent uniforms. Lemma 1 gives

$$\Pr[Y_t^{(j)} = v \mid \mathcal{H}_t^{(j)}] = p(v \mid x^{(j)}, Y_{<t}^{(j)}). \tag{194}$$

For a fixed text $w \in V^{n_j}$, put $E_t = \{Y_{1:t}^{(j)} = w_{1:t}\}$ and $E_0 = \Omega$. Iterated conditional expectation gives

$$\begin{aligned} \Pr[E_t \mid \mathcal{A}, \nu_j = a] &= \mathbb{E}\Big[\mathbf{1}_{E_{t-1}} \Pr[Y_t^{(j)} = w_t \mid \mathcal{H}_t^{(j)}] \,\Big|\, \mathcal{A}, \nu_j = a\Big] \\ &= p(w_t \mid x^{(j)}, w_{<t}) \Pr[E_{t-1} \mid \mathcal{A}, \nu_j = a]. \end{aligned} \tag{195}$$

On E_{t-1} the prefix is $w_{<t}$, which justifies the second equality. This recursion also covers zero-probability prefixes without conditioning on such a prefix. Starting at $t = 1$ yields

$$\Pr[Y^{(j)} = w \mid \mathcal{A}, \nu_j = a] = \prod_{t=1}^{n_j} p(w_t \mid x^{(j)}, w_{<t}) = P_j(w). \tag{196}$$

The right-hand side depends on neither the table nor the nonce. Consequently,

$$\Pr[Y^{(j)} = w] = \mathbb{E}\Big[\Pr[Y^{(j)} = w \mid \mathcal{A}, \nu_j]\Big] = P_j(w), \tag{197}$$

which proves (i). This is a marginal calculation for generation j; it does not condition on the outputs or queries of other generations.

Step 2: independence when the nonces are distinct. For a nonce value a, let $\mathcal{I}(a)$ be the set of all sampling inputs of the forms

$$\langle \mathsf{msg}, a, c, b, v\rangle, \qquad \langle \mathsf{fresh}, a, c, t, v\rangle, \tag{198}$$

with admissible contexts, message values, positions and tokens. Injectivity of the tuple encoding implies $\mathcal{I}(a) \cap \mathcal{I}(a') = \varnothing$ when $a \neq a'$: their nonce fields differ. All these sets are also disjoint from the chunk-selection inputs.

Given $\mathcal{A}$ and $\nu_j = a_j$, the text $Y^{(j)}$ is a deterministic function of the random-function values on $\mathcal{I}(a_j)$ and the fixed prompt, message and sampling settings. Although the queried inputs depend on previous tokens, all of them stay within this fixed set. For pairwise distinct $a_1, \ldots, a_N$, the restrictions of the random function to these sets are independent, even conditional on $\mathcal{A}$. Their resulting texts are

therefore independent under that conditioning. With $E_{\mathbf{w}} = \bigcap_{j=1}^{N} \{Y^{(j)} = w^{(j)}\}$, (196) gives

$$\begin{aligned} \Pr[E_{\mathbf{w}} \mid \mathcal{A}, \nu_1 = a_1, \ldots, \nu_N = a_N] &= \prod_{j=1}^{N} \Pr[Y^{(j)} = w^{(j)} \mid \mathcal{A}, \nu_j = a_j] \\ &= \prod_{j=1}^{N} P_j(w^{(j)}). \end{aligned} \tag{199}$$

Conditional independence alone would not in general imply independence after averaging over a shared table. Here it does, because this product does not depend on $\mathcal{A}$ or on the particular distinct nonce values.

When $N \le 2^r$, at least one distinct nonce tuple exists and has positive probability, so $\Pr[D] > 0$. Averaging the preceding identity conditional on D yields

$$\begin{aligned} \Pr[E_{\mathbf{w}} \mid D] &= \mathbb{E}[\Pr[E_{\mathbf{w}} \mid \mathcal{A}, \nu_1, \ldots, \nu_N] \mid D] \\ &= \prod_{j=1}^{N} P_j(w^{(j)}), \end{aligned} \tag{200}$$

proving (ii). When $N > 2^r$, distinct nonces are impossible, so no conditional law given D is asserted.

Step 3: bounding nonce collisions. For any two distinct generations $j < j'$,

$$\begin{aligned} \Pr[\nu_j = \nu_{j'}] &= \sum_{a \in \{0,1\}^r} \Pr[\nu_j = a, \nu_{j'} = a] \\ &= 2^r \cdot 2^{-r} \cdot 2^{-r} = 2^{-r}. \end{aligned} \tag{201}$$

The second equality uses independence and uniformity of the two nonces. Since D^c occurs exactly when at least one pair agrees, the union bound gives

$$\Pr[D^c] \le \sum_{1 \le j < j' \le N} \Pr[\nu_j = \nu_{j'}] = \binom{N}{2} 2^{-r} = \frac{N(N-1)}{2^{r+1}}. \tag{202}$$

Combining this with $\Pr[D^c] \le 1$ gives the stated collision bound. Independence between the pairwise collision events is not needed. For $N = 1$ the union is empty and its probability is zero; for $N > 2^r$ the collision probability is one.

Step 4: converting the collision bound to total variation. Write $\mathbf{Y} = (Y^{(1)}, \ldots, Y^{(N)})$ and $q = \Pr[D^c]$. We use the convention

$$d_{\mathrm{TV}}(P, Q) = \sup_{E \subseteq \prod_{j=1}^{N} V^{n_j}} |P(E) - Q(E)|. \tag{203}$$

If $q = 1$, then $d_{\mathrm{TV}}(P, Q) \le 1 = q$, since $P(E)$ and $Q(E)$ are probabilities. Now suppose $q < 1$, so D has positive probability. By (ii), for every set E of text tuples,

$$\Pr[\mathbf{Y} \in E, D] = (1 - q)\, Q(E). \tag{204}$$

Decomposing according to D and D^c therefore gives

$$\begin{aligned} P(E) - Q(E) &= (1-q)Q(E) + \Pr[\mathbf{Y} \in E, D^c] - Q(E) \\ &= \Pr[\mathbf{Y} \in E, D^c] - qQ(E). \end{aligned} \tag{205}$$

Both terms in the last difference lie in $[0, q]$, so its absolute value is at most q. Taking the supremum over E gives $d_{\mathrm{TV}}(P, Q) \le q$. Step 3 completes (iii), including the case in which conditioning on distinct nonces is impossible.

Step 5: the null law for every wrong nonce–message pair. Fix one generation and drop its superscript j. Fix its true nonce $\nu = a$, a chunk c with $k_c \geq 1$, and write

$$K = 2^{r+k_c}, \qquad \theta_* = (a, m_c), \qquad \mathcal{W}_c = \mathcal{P}_c \setminus \{\theta_*\}. \tag{206}$$

For the remainder of the proof, let $\Pr_*$ and $\mathbb{E}_*$ denote conditioning on $(Y_{1:n} = y, \{\mathcal{T}_c\}_{c=1}^{C}, \nu = a)$, for a realisation of positive probability. This fixes $\mathcal{T}_c$, n_c and $p_t = p(y_t \mid x, y_{<t}) > 0$ at each scored position.

Let $\mathcal{H}$ contain ν, the complete query transcript of this generation, and the decoder's chunk-selection queries before candidate scoring. It determines the text, the scored sets and the correct pair's sampling values. For $\theta = (a', b) \in \mathcal{W}_c$ and $t \in \mathcal{T}_c$, the decoder reads the input

$$d_{t,\theta} = \langle \mathsf{msg}, a', c_t, b, y_t \rangle. \tag{207}$$

These inputs are distinct over all (t, θ). Different scored positions have different contexts, and different candidate pairs differ in their nonce or message field.

None of these inputs appears in $\mathcal{H}$. A chunk or fresh query has a different tag. An encoder message query with context different from c_t differs in its context field. A message query with context c_t uses chunk c, since the chunk assignment is determined by that context, and therefore has the form $\langle \mathsf{msg}, a, c_t, m_c, v \rangle$. It differs from $d_{t,\theta}$ in the nonce field if $a' \neq a$, and in the message field if $a' = a$ and $b \neq m_c$. This comparison covers all encoder positions and tokens, including queries after t.

Conditional on $\mathcal{H}$, the values at these distinct, unqueried inputs are independent uniforms. More explicitly, set $W_{t,\theta} = u_t^{\theta}(y_t)$. For numbers $z_{t,\theta} \in [0, 1]$, the random-function argument gives

$$\Pr[W_{t,\theta} \leq z_{t,\theta} \text{ for all } t, \theta \mid \mathcal{H}] = \prod_{t \in \mathcal{T}_c} \prod_{\theta \in \mathcal{W}_c} z_{t,\theta}. \tag{208}$$

The right-hand side is fixed under $\Pr_*$, so averaging over $\mathcal{H}$ proves the same factorisation under $\Pr_*$. Thus every wrong pair has independent uniform scored values. Other generations may have queried some of these inputs, but their transcripts and outputs are not part of this conditioning. Reading a random-function value does not change it, and integrating out those other generations leaves the marginal law just derived.

For $U \sim U(0, 1)$ and $s \geq 0$,

$$\Pr[-\log(1 - U) \geq s] = \Pr[U \geq 1 - e^{-s}] = e^{-s}. \tag{209}$$

Hence each wrong-pair score is a sum of n_c independent $\mathrm{Exp}(1)$ variables. It has law $\Gamma(n_c, 1)$ for $n_c \geq 1$, by the convolution calculation in Appendix L. When $n_c = 0$, it is the empty sum 0. This proves the null-law assertion in (iv).

Step 6: the correct score and its independence from wrong scores. For the scaling bound we also need the correct pair's joint law after conditioning on the full text. The reasoning in Appendix O continues to apply with the fixed nonce field; we give the factorisation explicitly.

First condition on $\mathcal{A}$ and $\nu = a$ before generation. Write $\mathcal{H}_t$ for the single-generation history from Step 1, with the generation superscript omitted. At each step define

$$M_t = \max_{v : p_t(v) > 0} u_t(v)^{1/p_t(v)}. \tag{210}$$

The maximiser is the emitted token, because this is an equivalent form of the Gumbel-max rule. For a token v with $p_t(v) > 0$ and $z \in [0, 1]$, independence of the fresh uniforms in Step 1 gives

$$\begin{aligned} \Pr[Y_t = v, M_t \leq z \mid \mathcal{H}_t] &= \int_0^z p_t(v) s^{p_t(v)-1} \prod_{\substack{w \neq v \\ p_t(w) > 0}} s^{p_t(w)} \, \mathrm{d}s \\ &= p_t(v) \int_0^z s^{\sum_{w : p_t(w) > 0} p_t(w) - 1} \, \mathrm{d}s \\ &= z \, p_t(v). \end{aligned} \tag{211}$$

The integrand fixes the winning transformed uniform at s and requires every competitor to be at most s. Ties have probability zero. The last equality uses $\sum_w p_t(w) = 1$; for $p_t(v) = 0$, both sides of the identity are zero.

Apply the prefix recursion from Step 1, now also requiring $M_t \le z_t$ at each step. At step t its multiplier is $z_t p(y_t \mid x, y_{<t})$, so

$$\Pr[Y_{1:n} = y,\ M_t \le z_t \text{ for all } t \mid \mathcal{A}, \nu = a] = \prod_{t=1}^{n} z_t p(y_t \mid x, y_{<t}). \tag{212}$$

Dividing by the positive text probability from (196) gives

$$\Pr[M_t \le z_t \text{ for all } t \mid Y_{1:n} = y, \mathcal{A}, \nu = a] = \prod_{t=1}^{n} z_t. \tag{213}$$

For fixed y, the table determines $\{\mathcal{T}_c\}_{c=1}^{C}$. Set $z_t = 1$ outside $\mathcal{T}_c$ and average over the tables consistent with $(y, \{\mathcal{T}_c\}_{c=1}^{C}, a)$. The remaining product is unchanged, proving that $(M_t)_{t\in\mathcal{T}_c}$ are independent uniforms under $\Pr_*$.

At a scored position, the correct pair reconstructs the encoder's uniform and satisfies

$$B_t := u_t^{\theta_*}(y_t) = M_t^{p_t}, \qquad \Pr_*[B_t \le b] = b^{1/p_t} \quad (0 < b < 1). \tag{214}$$

Consequently the B_t are independent $\mathrm{Beta}(1/p_t, 1)$ variables under $\Pr_*$, and the correct score is $S_* = S_c(\theta_*) = \sum_{t\in\mathcal{T}_c} -\log(1 - B_t)$.

The complete transcript $\mathcal{H}$ determines S_*. Combining this fact with the conditional uniform factorisation from Step 5 yields, for any real s,

$$\begin{aligned} \Pr_*[S_* \le s,\ W \le z] &= \mathbb{E}_*\big[\mathbf{1}_{\{S_* \le s\}} \Pr[W \le z \mid \mathcal{H}]\big] \\ &= \Pr_*[S_* \le s] \prod_{t\in\mathcal{T}_c} \prod_{\theta\in\mathcal{W}_c} z_{t,\theta}. \end{aligned} \tag{215}$$

Here inequalities between vectors are coordinatewise. This factorisation proves that the entire array of wrong-pair uniforms is independent of the correct score under $\Pr_*$. In particular, each wrong score is independent of S_*.

Step 7: the certificate after selecting a candidate pair. Write $\hat\theta = (\hat\nu_c, \hat m_c)$ and let D_δ^{pair} return $\hat\theta$ if $\hat\delta_c \le \delta$, and $\perp$ otherwise. We first bound acceptance of a wrong pair. If $\delta = 1$, the claimed bound is automatic. Suppose $0 < \delta < 1$.

If $n_c = 0$, every score is zero and $Q_0(0) = 1$, so $\hat\delta_c = \min\{1, K-1\} = 1$ because $K \ge 2$. The decoder therefore abstains. If $n_c \ge 1$, the Gamma density is positive on $(0,\infty)$, so its upper tail Q_{n_c} is continuous and strictly decreasing there, from 1 to 0. Since $0 < \delta/(K-1) < 1$, there is a unique $s_\delta > 0$ with

$$Q_{n_c}(s_\delta) = \frac{\delta}{K-1}. \tag{216}$$

For $\delta < 1$, clipping the certificate at 1 does not change its acceptance event. Monotonicity of the tail gives

$$\begin{aligned} \hat\delta_c \le \delta \quad &\Longleftrightarrow \quad Q_{n_c}(S_c(\hat\theta)) \le \frac{\delta}{K-1} \\ &\Longleftrightarrow \quad S_c(\hat\theta) \ge s_\delta. \end{aligned} \tag{217}$$

Thus an accepted wrong pair must be one of the $K-1$ wrong pairs whose score crosses this fixed threshold:

$$\{D_\delta^{\mathrm{pair}} \notin \{\theta_*, \perp\}\} \subseteq \bigcup_{\theta\in\mathcal{W}_c} \{S_c(\theta) \ge s_\delta\}. \tag{218}$$

Each fixed wrong pair has the null law from Step 5, so

$$\begin{aligned} \Pr_*[D_\delta^{\mathrm{pair}} \notin \{\theta_*, \perp\}] &\le \sum_{\theta\in\mathcal{W}_c} \Pr_*[S_c(\theta) \ge s_\delta] \\ &= (K-1) Q_{n_c}(s_\delta) = \delta. \end{aligned} \tag{219}$$

The union bound is applied to the fixed candidates, rather than assigning the Gamma law to their selected maximum.

Returning a wrong message requires accepting a wrong pair; a pair with an incorrect nonce but the correct message is not a message error. Hence the certified message decoder D_δ satisfies the same conditional bound. Averaging over this generation's text, scored sets and nonce gives

$$\Pr[D_\delta(Y_{1:n}) \notin \{m_c, \perp\}] = \mathbb{E}\big[\Pr[D_\delta(Y_{1:n}) \notin \{m_c, \perp\} \mid Y_{1:n}, \{\mathcal{T}_c\}_{c=1}^C, \nu]\big] \le \delta. \tag{220}$$

This is the guarantee in (15).

Step 8: the payload-scaling bound for pair candidates. Assume first that $n_c \ge 1$, fix $\theta \in \mathcal{W}_c$, and choose $0 < \lambda < 1$. The exponential moment of a wrong score is

$$\mathbb{E}_*[e^{\lambda S_c(\theta)}] = \prod_{t \in \mathcal{T}_c} \int_0^1 (1-u)^{-\lambda}\, \mathrm{d}u = (1-\lambda)^{-n_c}. \tag{221}$$

For a correct contribution, put $\alpha_t = 1/p_t$. Its negative exponential moment is

$$\begin{aligned} \mathbb{E}_*[(1-B_t)^\lambda] &= \alpha_t \int_0^1 b^{\alpha_t - 1}(1-b)^\lambda \,\mathrm{d}b \\ &= \frac{\Gamma(\alpha_t+1)\Gamma(\lambda+1)}{\Gamma(\alpha_t+\lambda+1)}. \end{aligned} \tag{222}$$

The last equality is the beta integral proved in Appendix O. Step 6 gives the independence needed to multiply these moments. Since $\{S_c(\theta) \ge S_*\} = \{e^{\lambda(S_c(\theta)-S_*)} \ge 1\}$, Markov's inequality gives

$$\begin{aligned} \Pr_*[S_c(\theta) \ge S_*] &\le \mathbb{E}_*[e^{\lambda(S_c(\theta)-S_*)}] \\ &= \mathbb{E}_*[e^{\lambda S_c(\theta)}]\, \mathbb{E}_*[e^{-\lambda S_*}] \\ &= (1-\lambda)^{-n_c} \prod_{t \in \mathcal{T}_c} \frac{\Gamma(1/p_t+1)\Gamma(\lambda+1)}{\Gamma(1/p_t+\lambda+1)} \\ &= \exp\left(-\sum_{t \in \mathcal{T}_c} \eta_{p_t}(\lambda)\right). \end{aligned} \tag{223}$$

The last line is precisely the definition of η_p in (82). Taking the infimum over $0 < \lambda < 1$ yields

$$\Pr_*[S_c(\theta) \ge S_*] \le \exp\left(-\sup_{0<\lambda<1} \sum_{t \in \mathcal{T}_c} \eta_{p_t}(\lambda)\right) = e^{-\Lambda_c}. \tag{224}$$

No maximiser is required: values approaching the supremum give the same bound. Finiteness of Λ_c follows from the endpoint and continuity argument in Appendix O.

If the maximising pair is wrong, some wrong pair has score at least S_*. This implication also holds for a tie, whatever the fixed tie-breaking rule. Therefore

$$\begin{aligned} \Pr_*[\hat{m}_c \ne m_c] &\le \Pr_*[\hat{\theta} \ne \theta_*] \\ &\le \sum_{\theta \in \mathcal{W}_c} \Pr_*[S_c(\theta) \ge S_*] \\ &\le (K-1)e^{-\Lambda_c} < \exp\big((k_c+r)\ln 2 - \Lambda_c\big). \end{aligned} \tag{225}$$

The strict inequality uses $K - 1 < K = 2^{k_c+r}$ and $e^{-\Lambda_c} > 0$.

If $n_c = 0$, then $\Lambda_c = 0$ and every score is zero. The same bound holds because the error probability is at most $1 \le K - 1$. Thus all scored-set sizes are covered. The right-hand side depends on the text and scored sets but not on the actual nonce. Averaging over ν conditional on $(Y_{1:n} = y, \{\mathcal{T}_c\}_{c=1}^C)$ proves the conditional bound stated in (iv). Finally, an error in the concatenated message requires an error in at least one chunk. A union bound under this same conditioning gives

$$\Pr[\hat{m} \ne m \mid Y_{1:n} = y, \{\mathcal{T}_c\}_{c=1}^C] \le \sum_{c=1}^C \Pr[\hat{m}_c \ne m_c \mid Y_{1:n} = y, \{\mathcal{T}_c\}_{c=1}^C] \le \sum_{c=1}^C (2^{k_c+r} - 1)e^{-\Lambda_c}. \tag{226}$$

This completes (iv) and the proof. □