IS BINARY ENOUGH? EXPLORING P-ARY ENCODING FOR MULTI-BIT LANGUAGE MODEL WATERMARKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) watermarking provides a practical mechanism for tracing AI-generated text and mitigating misuse. Recent multi-bit watermarking methods enable richer payloads, yet almost all rely on binary encoding, leaving the role of the encoding radix largely unexplored. We ask a fundamental question: can a higher-radix representation outperform binary encoding for multi-bit watermarking? We introduce PAMark, a lightweight watermarking scheme that combines p-ary encoding with a Gumbel-based ranking strategy. Through theoretical analysis, we characterize how the encoding radix affects message recovery and prove that, under certain conditions, an appropriate choice of p achieves higher message accuracy than binary encoding. Extensive experiments across multiple models and settings further validate our analysis. Compared with representative multi-bit watermarking baselines, PAMark achieves consistently stronger message recovery while maintaining competitive text quality and watermark detectability. Our results identify the encoding radix as an important yet overlooked design dimension and provide a principled foundation for moving beyond binary multi-bit LLM watermarking.

kw: Large Language Model Watermarking, Multi-Bit Watermarking, Gumbel Ranking, Message Recovery.

1 INTRODUCTION

LLM watermarking provides a practical mechanism for tracing generated text, attributing content, and mitigating misuse. Green-list watermarking established the feasibility of detecting a keyed signal in generated text Kirchenbauer et al. (2023); distortion-free sampling and production-scale watermarking subsequently broadened the design space Kuditipudi et al. (2024); Dathathri et al. (2024). Beyond binary detection, multi-bit watermarking can encode user identifiers, timestamps, model versions, or other provenance information Yoo et al. (2024); Gloaguen et al. (2026). Many systems start from a binary payload, while some already support larger symbol alphabets. Yet the interaction between radix, full-payload evidence and recovery remains insufficiently characterized. This motivates a basic design question: *is binary encoding always the best choice for message recovery?*

Existing methods distribute evidence across message positions Yoo et al. (2024), balanced segments with error correction Qu et al. (2025), or the full payload Feng et al. (2025); Gloaguen et al. (2026). Other designs preserve sampling randomness while encoding multiple bits Jiang et al. (2026). They use different alphabets and evidence schedules; the useful question is therefore how radix interacts with a particular encoder, decoder and token budget, rather than whether one alphabet universally dominates.

The design problem is especially acute for short continuations. A detector may have only a few dozen generated tokens, yet an identifier still needs to be recovered in full. Reporting average bit accuracy alone can obscure this requirement: a method that recovers nearly every bit on every sample can nevertheless recover very few complete identifiers. We therefore treat exact-message recovery as the primary measure and report bit accuracy and text quality alongside it. These quantities need not move together.

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.

For a fixed B-bit message, base p changes its representation length to $K = \lceil B/\log_2 p \rceil$. Larger p reduces the number of coordinates that must all be correct, but also reduces the chance that a uniformly drawn symbol agrees with any one coordinate. The choice of radix is consequently inseparable from the token budget, the candidate ranking rule, and the number of evidence layers. This is a representation question, not simply a stronger watermark setting.

We introduce PAMark, a lightweight multi-bit watermark that combines p-ary full-payload evidence with Gumbel ranking. Keyed candidate vectors are ranked by message agreement; the highest-ranked candidates receive the largest Gumbel perturbations, which compete with the language-model probabilities. The verifier reconstructs the selected vectors and performs position-wise majority decoding without access to the language model.

Radix affects two opposing factors. A larger radix reduces the number of symbols needed for a fixed payload, but makes any one symbol harder to observe. Additional layers provide more evidence within a token, yet the votes are dependent after selection. We derive a conditional sufficient region for (p, L) and token length, then test actual performance without presenting that idealized bound as a measured prediction.

Our analysis uses candidate-score symmetry to connect full-payload ranking to a positive per-coordinate vote margin. A token-block concentration bound then identifies a sufficient design region and a conservative token budget. The assumptions are deliberately explicit: the model does not turn an idealized guarantee into a universal claim about autoregressive language-model trajectories. Experiments probe where the proposed mechanism helps and where competing watermark designs retain the advantage.

Our contributions are:

- We introduce full-payload p-ary Gumbel ranking with model-free recovery.
- We derive a conditional radix–layer region, token budget and corruption-margin bound.
- We map radix sensitivity, report a matched 24-bit short-text comparison across five watermark families, and evaluate complete-message recovery under Syn-30 on two generators.

2 Related Work

Position Allocation and Balanced Segmentation

MPAC assigns each generation step to a message position and reinforces the corresponding color list Yoo et al. (2024). RSBH balances token-to-segment assignment and uses Reed–Solomon coding to tolerate segment errors Qu et al. (2025). These methods make local evidence easier to decode, but the available token budget must still be divided among message units.

Position assignment provides a clear route from local token evidence to a message symbol. Its cost is that an individual symbol receives only part of the continuation; an error-correcting outer code can mitigate, but not remove, this allocation trade-off. PAMark instead scores every candidate against the entire payload at every generation step. This changes both the evidence available to a short message and the dependence structure that the decoder must handle.

Full-Payload and Multilayer Evidence

BiMark uses multilayer bit-flip unbiased reweighting to increase statistical evidence Feng et al. (2025). Binomial watermarking lets each token contribute Bernoulli evidence to all bits and can use stateful weighting Gloaguen et al. (2026). WeaveMark combines coded payload spreading, soft-decision error correction, and multilayer evidence Park et al. (2026). PAMark also gives every token full-payload evidence, but replaces binary coordinates with a configurable p-ary representation and uses a ranking-based Gumbel coupling rule.

These approaches are important comparators because they vary more than the payload alphabet. Their token-level evidence rules, decoding statistics, and watermark-strength controls also differ. A matched empirical comparison must therefore fix the payload, prompts, token budget, and attack while disclosing method-specific operating points. We use adaptive weighting for Binomial. MirrorMark

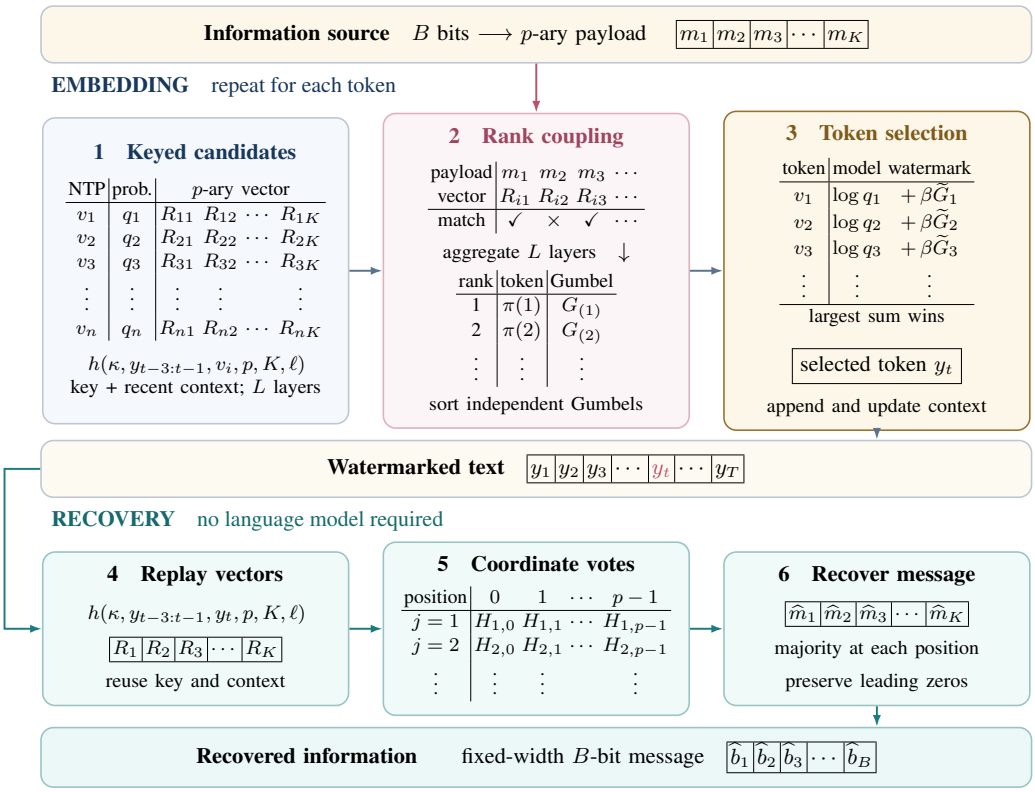

Figure 1: PAMark embedding and model-free recovery. Keyed candidate vectors are ranked by agreement with the whole p-ary payload, then paired with independent Gumbel order statistics. The selected token carries evidence that the decoder regenerates and aggregates by coordinate. Algorithms 1 and 2 specify the inner loops.

uses $p = 2$ in the principal clean table, with base-four and base-eight results identified separately; the attack study and new-prompt verification use $p = 8$.

Randomness-Preserving Sampling

MirrorMark uses mod-1 mirroring and a context-anchored scheduler to encode messages through the sampling randomness Jiang et al. (2026). PAMark is complementary: it ranks candidates by full-payload similarity and assigns Gumbel order statistics according to that ranking. The two designs share the goal of reducing direct logit distortion, but differ in how message evidence is represented and aggregated.

The use of Gumbel order statistics is central to PAMark's sampling path: a message-dependent candidate ranking is paired with independently drawn perturbations, then combined with the original model probabilities. This differs from simply adding a fixed positive logit bias to preferred tokens. It also means that recovery depends on the candidate-rank mechanism as well as on radix; the theory studies their joint effect rather than attributing every observed difference to base conversion alone.

3 PAMark

Figure 1 gives the end-to-end embedding and recovery path before the detailed construction. All components are shown as editable vector elements.

3.1 FULL-PAYLOAD p-ARY REPRESENTATION

Let b be a fixed-width B-bit payload and $m = \mathrm{Radix}_p(b)$ its $K = \lceil B/\log_2 p \rceil$ digit representation, including leading zeros. At generation step t, the model supplies top-n candidates $\{(v_i, q_i)\}_{i=1}^n$ after temperature scaling and renormalization. For candidate v_i, PAMark constructs L reproducible pseudo-random vectors

$$\boldsymbol{R}_{t,i,\ell} = F_\kappa(p, c_t, v_i, \ell) \in [p]^K, \qquad \ell = 1, \ldots, L,$$

where κ is the secret key and c_t consists of the last three token IDs in the reference implementation. The hash input also contains p, K, token identity, and the layer index when $L > 1$. Equal scores receive random tie priorities. The full-payload match score is

$$S_{t,i} = \sum_{\ell=1}^{L} \sum_{j=1}^{K} \mathbf{1}\{R_{t,i,\ell,j} = m_j\}. \tag{1}$$

Candidates are sorted by $S_{t,i}$; averaging or summing layerwise agreement yields the same rank for fixed K, L. Unlike position-allocation methods, every candidate is evaluated against every message coordinate. The experiment code uses a deterministic SHA-256-derived seed with NumPy's pseudo-random generator for reproducibility; this interface should be replaced by a cryptographic keyed PRF for deployment. We make no finite-key security claim for the experimental construction.

3.2 GUMBEL RANKING AND TOKEN SELECTION

Let π_t denote the descending order induced by $S_{t,i}$. Draw n independent standard Gumbel variables and sort them as $g_{(1)} \geq \cdots \geq g_{(n)}$. PAMark assigns the order statistics to candidates according to the message-similarity ranking:

$$\widetilde{g}_{t,\pi_t(r)} = g_{(r)}, \qquad r = 1, \ldots, n.$$

The next token is sampled by

$$y_t = v_{\arg\max_i \{\log q_{t,i} + \beta \widetilde{g}_{t,i}\}}, \tag{2}$$

where β controls the strength of the ranking perturbation. This separates message evidence from a fixed additive bias: the language model probability remains part of the final competition.

3.3 MODEL-FREE RECOVERY

Given the original prompt tokens and observed watermarked token IDs, the verifier regenerates exactly the vectors belonging to those observed tokens. For each digit j and alphabet value a, it accumulates

$$H_{j,a} = \sum_{t=1}^{T} \sum_{\ell=1}^{L} \mathbf{1}\{R_{t,y_t,\ell,j} = a\}, \qquad \widehat{m}_j \in \arg\max_{a \in [p]} H_{j,a}. \tag{3}$$

An independent random choice breaks a final tie. Leading-zero preservation and fixed-width conversion return the B-bit estimate. Verification does not require model logits, but it requires the original prompt and exact tokenization to replay the rolling context.

EMBEDDING AND RECOVERY PROCEDURES

Embedding costs $O(nLK + n \log n)$ per generated token beyond the language-model forward pass. Verification costs $O(TLK)$ time and $O(Kp)$ vote memory, without model calls.

Algorithm 1: PAMark embedding, generation and state update

Input: Prompt tokens h; B-bit payload b; key κ; radix p; length T; layers L; candidate count n; temperature τ; strength β
Output: Watermarked token sequence $y_{1:T}$

$m \leftarrow \mathrm{Radix}_p(b)$ with fixed width $K = \lceil B/\log_2 p \rceil$;
for $t \leftarrow 1$ **to** T **do**
  $(v_i, q_i)_{i=1}^n \leftarrow \mathrm{TopNAndRenormalize}(\mathrm{LM}(h)/\tau, n)$;
  $c_t \leftarrow \mathrm{Suffix}_3(h)$;
  for $i \leftarrow 1$ **to** n **do**
    $S_i \leftarrow 0$;
    for $\ell \leftarrow 1$ **to** L **do**
      $s_{i,\ell} \leftarrow H(\kappa, c_t, v_i, p, K, \ell)$;
      $R_{i,\ell} \leftarrow \mathrm{PRNG}(s_{i,\ell}, p, K)$;
      for $j \leftarrow 1$ **to** K **do**
        $S_i \leftarrow S_i + \mathbf{1}\{R_{i,\ell,j} = m_j\}$;
    Draw independent tie priority $u_i \sim \mathrm{Uniform}(0,1)$;
  $\pi \leftarrow \mathrm{SortDescending}(S_i, u_i)_{i=1}^n$;
  Draw $U_1, \ldots, U_n \overset{\mathrm{iid}}{\sim} \mathrm{Uniform}(0,1)$;
  Sort $G_i = -\log(-\log U_i)$ as $G_{(1)} > \cdots > G_{(n)}$;
  Set $\widetilde{G}_{\pi(r)} \leftarrow G_{(r)}$ for each rank r;
  $y_t \leftarrow v_{\arg\max_i \{\log q_i + \beta \widetilde{G}_i\}}$; append y_t to h;

Algorithm 2: PAMark recovery from observed tokens

Input: Prompt tokens h; observed tokens $y_{1:T}$; key κ; radix p; width K; layers L; bit width B
Output: Recovered B-bit payload $\widehat{b}$

Set $H_{j,a} \leftarrow 0$ for $(j,a) \in [K] \times [p]$;
for $t \leftarrow 1$ **to** T **do**
  $c_t \leftarrow \mathrm{Suffix}_3(h)$;
  for $\ell \leftarrow 1$ **to** L **do**
    $R \leftarrow \mathrm{PRNG}(H(\kappa, c_t, y_t, p, K, \ell), p, K)$;
    for $j \leftarrow 1$ **to** K **do**
      $H_{j,R_j} \leftarrow H_{j,R_j} + 1$;
  Append y_t to h;
for $j \leftarrow 1$ **to** K **do**
  Choose $\widehat{m}_j$ uniformly among maximizers of $H_{j,a}$;
$\widehat{b} \leftarrow \mathrm{FixedWidthBits}(\widehat{m}, p, B)$; return $\widehat{b}$;

4 Recovery Analysis and Design Implications

Our analysis separates three questions: how ranking creates symbol evidence, how dependent evidence accumulates, and how much corruption a vote margin can tolerate. It concerns an explicitly idealized candidate model, not an asserted fit to the experimental recovery rates.

4.1 From Candidate Rank to Coordinate Evidence

Let $K = \lceil B/\log_2 p \rceil$, $d = KL$, and assume fresh independent uniform cells for each of n candidate arrays. The full-payload score satisfies $X_i \sim \mathrm{Bin}(d, 1/p)$. Random ranking ties make the permutation of candidate identities uniform and independent of the ordered scores. For exogenous candidate probabilities q, selection observes the total score through its rank, not which coordinates matched. Conditional on selected score s, the array is consequently uniform over a Hamming shell of size $\binom{d}{s}(p-1)^{d-s}$.

For target digit m_j and competitor $a \neq m_j$, define the whole-token margin

$$V_{t,j,a} = \sum_{\ell=1}^{L} \big(\mathbf{1}\{R_{t,\ell,j} = m_j\} - \mathbf{1}\{R_{t,\ell,j} = a\} \big).$$

Shell symmetry yields the useful identity

$$\mathbb{E}[V_{t,j,a} \mid X_{\text{selected}} = s] = \frac{ps - KL}{K(p-1)}. \tag{4}$$

Thus a selected-score surplus above KL/p translates directly into positive coordinate margins. Layers are not independent after selecting their shared candidate; concentration must operate on entire token blocks.

4.2 A Sufficient Radix–Layer Region and Token Budget

Assume independent, identically distributed token blocks in the reference model. Let $\phi_{p,L}(\theta) = \mathbb{E}e^{-\theta V_{t,j,a}}$ and $I_{p,L} = -\log \inf_{\theta \geq 0} \phi_{p,L}(\theta)$. For $I_{p,L} > 0$, Chernoff's inequality and a union bound give

$$\Pr(\widehat{m} \neq m) \leq K(p-1)e^{-TI_{p,L}}. \tag{5}$$

At target recovery ρ, a sufficient design region and token budget are

$$\mathcal{R}_{\text{suff}} = \left\{ (p, L) \in \mathcal{C} : TI_{p,L} \geq \log \frac{K(p)(p-1)}{1-\rho} \right\}, \tag{6}$$

$$T_{\text{suff}} = \left\lceil \frac{\log(K(p)(p-1)/(1-\rho))}{I_{p,L}} \right\rceil, \tag{7}$$

where $\mathcal{C}$ specifies compute and quality constraints. This is a feasible set, not necessarily a single interval or the exact optimum. Increasing p reduces K but decreases random symbol exposure; increasing L changes both the score distribution and dependence within a token. The combination, not either variable alone, determines $I_{p,L}$. Under symbol symmetry the payload value itself does not matter beyond its width.

A complementary necessary condition follows from candidate exposure:

$$T \geq T_{\text{exp}} = \left\lceil \frac{\log(1 - \rho^{1/K})}{nL \log(1 - 1/p)} \right\rceil.$$

Every target digit must appear among its candidate cells at least once for majority recovery to succeed. This bound can be very weak, while T_{suff} can be conservative; neither is the true minimum for an autoregressive generator.

An Explicit Positive-Margin Regime

Under the additional assumption $\max_i \log q_i - \min_i \log q_i \leq R$ for exogenous positive q, define

$$b_\star = \sum_{s=1}^{KL} [1 - F_X(s-1)^n], \quad m_\star = \frac{pb_\star - KL}{K(p-1)}, \quad g_n(u) = \frac{n}{1 + (n-1)e^u}.$$

Here $b_\star$ is the expected maximum candidate score and $m_\star > 0$ for $n \geq 2$. A Gumbel gap exceeding R/β forces the highest-similarity rank to win. Its probability is exactly $g_n(R/\beta)$. Exchangeability on the complementary event gives

$$\mathbb{E}V_{t,j,a} \geq \mu := g_n(R/\beta)m_\star, \qquad \Pr(\widehat{m} \neq m) \leq K(p-1)\exp\left(-\frac{T\mu^2}{2L^2}\right). \tag{8}$$

This is an explicit sufficient guarantee in the reference model. It exposes the role of candidate log-probability spread and β, but does not prove distribution preservation or predict PPL. Appendix B.8 supplies the gap integral, exchangeability argument and concentration proof.

A Margin Criterion for Corrupted Evidence

For a realized clean sequence, let $D_{\min} = \min_{j, a \neq m_j} \sum_t V_{t,j,a}$. Replacing at most r aligned evidence blocks changes any margin by at most $2Lr$. Consequently,

$$D_{\min} > 2Lr \quad \Longrightarrow \quad \text{exact recovery after any such block replacements.} \tag{9}$$

Under the independent bounded-margin reference assumptions, when $T\mu > 2Lr$,

$$\Pr(\text{failure after at most } r \text{ replacements}) \leq K(p-1)\exp\left[-\frac{(T\mu - 2Lr)^2}{2TL^2}\right]. \tag{10}$$

For a length-preserving token substitution channel with a context of h tokens, e changed token IDs affect at most $\min(T, (h+1)e)$ keyed blocks. This locality observation motivates robustness analysis, but *does not convert a word-selection percentage into r*: WordNet substitution and retokenization need not preserve token alignment. The empirical attack results test the real pipeline; the bound describes its evidence-level mechanism.

The appendix derives the exact joint vote law and a conditional-MGF extension. Neither fresh independent blocks nor a uniform conditional margin is established for real autoregressive trajectories.

5 Experiments

Evaluation protocol. We evaluate Qwen-7B Bai et al. (2023) and Meta-Llama-3-8B with C4 Raffel et al. (2020) and ELI5 Fan et al. (2019) prompts, temperature one and top-128 sampling. Exact recovery requires every payload bit to match. Gemma-2-9B Gemma Team (2024) measures perplexity (PPL) on clean continuations conditioned on the prompt; we exponentiate pooled negative log-likelihood per scored token. PPL is not semantic fidelity. The short-text experiments share two 24-bit payloads and 20 prompts per corpus. Binomial uses adaptive weighting and MirrorMark uses $p = 8$. These local implementations are not independently certified reproductions. Further method diagnostics appear in Appendix A.

A matched radix ablation. Table 1 isolates radix by fixing the 24-bit payloads, 40 tokens, $L = 60$, $\beta = 1.0$ and 20 C4 prompts. Higher-radix encoding substantially improves recovery over binary encoding, showing that the representation itself matters when evidence is limited. A larger alphabet shortens the encoded message but makes each symbol harder to distinguish; the conditional analysis therefore predicts a trade-off rather than an unlimited benefit from increasing p. The irregular empirical curve is consistent with the absence of a simple monotonic rule, although these small cells do not validate its precise theoretical shape. The highest tested recovery occurs at $p = 36$, whereas $p = 12$ has the lowest PPL: the preferred radix depends on whether the objective is complete recovery or scorer-rated text quality. This sweep shares prompts with the Llama/C4 comparison and is not an independent confirmation. Larger-payload follow-ups appear in Appendices A.1 and A.3.

Table 1: PAMark radix ablation on Llama/C4: 24 bits, 40 tokens, two payloads $\times$ 20 shared prompts. Exact recovery is in percent. All rows use the same unwatermarked PPL of 18.95; ΔPPL is watermarked minus unwatermarked PPL. The additional $p > 16$ rows are new completed runs on the identical messages and prompt file.

Radix p	Exact	PPL	ΔPPL
2	7.50	14.16	-4.79
4	37.50	15.00	-3.95
8	42.50	16.84	-2.11
12	42.50	**12.95**	-6.00
16	47.50	14.31	-4.64
20	45.00	13.66	-5.30
24	25.00	13.86	-5.09
32	45.00	14.28	-4.67
36	$\uparrow$ **52.50**	17.97	-0.98

Five-method clean recovery comparison. Table 2 compares five methods on one 54-bit payload, 300 generated tokens and 200 shared Llama/ELI5 prompts. PAMark uses $(p, L, \beta) = (32, 5, 1.15)$, selected after a 40-prompt pilot; MirrorMark uses $p = 2$ and Binomial uses adaptive weighting with $\epsilon = 0.05$. PAMark exceeds binary MirrorMark by 20 percentage points and Binomial by 4.5 points. Its paired discordances with MirrorMark are 59 versus 19. Radix is important for the baseline too: on

the same prompts, MirrorMark at $p = 4$ ties PAMark, while the earlier $p = 8$ arm has higher recovery (Appendix A.1). The binary comparison therefore does not establish superiority over MirrorMark's stronger operating points. MirrorMark and Binomial retain higher mean bit accuracy and lower PPL, illustrating that complete recovery, residual errors and text quality are distinct objectives. MPAC and BiMark likewise have much higher bit accuracy than exact recovery; these are local operating points, not their best achievable performance. New-prompt verification retains MirrorMark $p = 8$ (Appendix A.2) and is not pooled here.

Table 2: Matched Llama/ELI5 clean comparison. Exact and bit recovery are percentages. ΔPPL subtracts unwatermarked PPL 11.38. Method parameters and sampling settings are specified in the text; bold marks the best observed metric, not an overall winner.

Method	Exact recovery	Bit accuracy	ΔPPL
PAMark	**85.50**	96.44	+3.35
MirrorMark	65.50	**99.08**	+0.52
Binomial	81.00	97.97	−3.38
MPAC	6.00	92.62	+5.69
BiMark	28.00	97.30	+0.61

Exact recovery after lexical editing. Our third experiment asks whether complete-message evidence survives lexical editing. It uses 64 bits, 300 tokens and two payloads sharing 60 C4 prompts, excluding the earlier attack pilot. PAMark uses $(p, L, \beta) = (16, 5, 1.1)$, MirrorMark uses $p = 8$ and Binomial uses adaptive weighting. WordNet replaces selected eligible alphabetic word spans before retokenization; words without eligible replacements remain unchanged. Syn-10 and Syn-30 denote requested word-selection rates, not actual model-token change rates. Each attack reuses its arm's clean generations, allowing degradation to be compared within an operating point. Table 3 and Figure 2 show a crossover: Binomial leads PAMark without attacks and under mild editing, but PAMark retains more complete messages under Syn-30 on both generators. The gain over Binomial is approximately 14–18 percentage points and has the same direction for both payloads. This repeat across generators is encouraging, but its interpretation is robustness to this particular edit procedure, not general resistance to arbitrary paraphrasing.

Table 3: C4 validation: exact-message recovery (%). ↑ highlights PAMark only where it exceeds the other listed methods. Experimental and method settings are specified in the text. ΔPPL is clean watermarked minus unwatermarked PPL; the latter is 11.70 for Qwen and 11.08 for Llama.

Generator	Method	Clean	Syn-10	Syn-30	ΔPPL
Qwen	**PAMark**	81.67	66.67	↑ **21.67**	+0.98
Qwen	MirrorMark	80.83	2.50	0.00	−0.17
Qwen	Binomial	84.17	72.50	7.50	−1.31
Llama	**PAMark**	70.83	62.50	↑ **23.33**	+0.13
Llama	MirrorMark	80.00	5.00	1.67	−0.31
Llama	Binomial	80.00	72.50	5.83	−2.14

The scope of the advantage. The attack results distinguish preserving a complete identifier from preserving most of its bits. Under Syn-30, PAMark has higher exact recovery but lower average bit accuracy than Binomial. Consequently, PAMark produces more zero-error messages, while its failed messages contain more errors on average. This matters when even one incorrect bit invalidates an identifier; it does not establish an advantage for applications that can exploit partially recovered payloads. Accumulating full-payload evidence may contribute to this behavior, but the present comparison does not isolate that mechanism from the other implementation differences. The clean ΔPPL values also show a quality cost relative to Binomial. The useful operating region is therefore strong lexical editing with an exact-recovery objective, rather than a simultaneous win in every metric. Appendix A.6 reports realized edits and detector-calibration limitations.

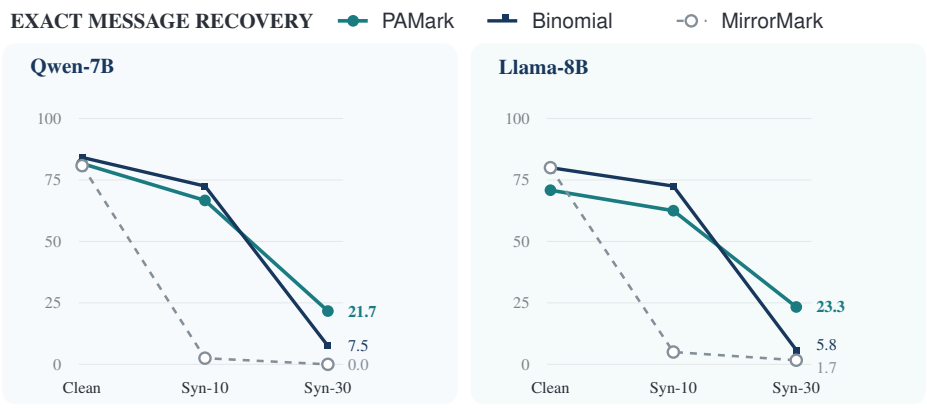

Figure 2: Exact recovery after lexical editing on the same generations (two payloads, 60 shared prompts). Labels show Syn-30 percentages; the PAMark advantage does not extend to clean text or Syn-10.

For erroneous-bit count E, the distinction follows from the exact decomposition

$$\mathbb{E}[E] = (1 - \Pr(E = 0))\,\mathbb{E}[E \mid E > 0]. \tag{11}$$

6 Discussion and Limitations

Comparable clean recovery, stronger recovery after editing. The central benefit is not a universal clean-text lead: PAMark exceeds binary MirrorMark in Table 2, ties its base-four arm on the same prompts, and nearly ties base eight on new prompts. Under the tested Syn-30 attack, PAMark retains more intact messages than base-eight MirrorMark and Binomial on both generators. This motivates PAMark for identifiers exposed to stronger lexical editing. The clean and attack studies use different payloads, corpora and operating points, not one configuration dominating both tasks. Within the attack suite, MirrorMark leads on clean text before the ranking reverses under Syn-30. PAMark's higher PPL and lower mean bit accuracy remain costs.

Radix changes message representation and symbol discrimination, whereas β controls the influence of ranked Gumbel values relative to language-model logits. The scale follow-ups show that stronger watermark influence can improve recovery while substantially increasing PPL. Operating-point selection should therefore impose a quality constraint; neither the largest radix nor the strongest scale is a universal recommendation.

On 200 new ELI5 prompts, PAMark again exceeds Binomial in exact recovery and nearly ties MirrorMark at base eight, with a higher PPL cost (Appendix A.2). This supports a recovery–quality trade-off beyond the selection prompts, but uses only one payload. The short-text cells reuse 20 prompts across two messages; their advantage at 40 tokens does not extend to every budget, as Binomial leads at 60 and 100 tokens.

The edits do not verify semantic preservation, and few payloads limit generalization. Detector 1% FPR lacks independent calibration and is not a validated operating point. Local baselines need official-code cross-checks. The reproducible keyed generator is not a deployment security guarantee, and token independence remains a theoretical assumption rather than a measured fit.

7 Conclusion

PAMark combines p-ary full-payload Gumbel ranking with model-free recovery. Radix tuning improves short-text recovery over binary encoding. Clean recovery is comparable to MirrorMark in the reported comparisons, while stronger synonym editing favors PAMark's complete-message recovery. These setting-specific benefits carry bit-accuracy and PPL trade-offs; broader official-code validation remains necessary.

AI USE STATEMENT

Generative AI (OpenAI Codex) assisted with data preparation, experimental design and code, analysis of recorded results, figures, manuscript writing, and mathematical arguments and proofs. The authors remain responsible for verifying and approving the final content.

REFERENCES

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

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

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3558–3567, 2019. doi: 10.18653/v1/P19-1346. URL https://aclanthology.org/P19-1346/.

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, 2025. URL https://proceedings.mlr.press/v267/feng25u.html.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

Thibaud Gloaguen, Robin Staab, Mark Vero, and Martin Vechev. Every bit, everywhere, all at once: A binomial multibit llm watermark. *arXiv preprint arXiv:2605.11653*, 2026. URL https://arxiv.org/abs/2605.11653.

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. URL https://arxiv.org/abs/2601.22246.

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*, pp. 17061–17084, 2023.

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

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

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. Provably robust multi-bit watermarking for ai-generated text. *arXiv preprint arXiv:2401.16820*, 2025. URL https://arxiv.org/abs/2401.16820.

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.

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, 2024. doi: 10.18653/v1/2024.naacl-long.224. URL https://aclanthology.org/2024.naacl-long.224/.

A Selected Experimental Follow-Ups

This appendix retains recent completed studies that test operating-point sensitivity, new-prompt verification, short-token behavior, model–corpus transfer and attack diagnostics. Earlier small pilots, repeated historical tables and the archive inventory are not reproduced. The original 200-prompt five-method comparison is now Table 2 in the main text; it is not duplicated here.

A.1 Completed Fifty-Four-Bit Scale Follow-Ups

The bounded follow-up queue completed all scheduled arms, including the optional MPAC/BiMark comparison, in approximately 6 hours 48 minutes of wall time. It retained the 54-bit payload 4728523782933237, Llama-8B, 300 generated tokens, top-128 candidates, temperature one and Gemma-2-9B scoring. Table 4 keeps the new $\beta = 1.1$ and 1.15 arms separate from the earlier 1.2 arm. All use the original 200-prompt ELI5 sequence; the unwatermarked PPL is 11.38. Exact counts, mean bit accuracy and pooled NLL/token PPL were checked against the remote per-run records. These records remain on the experiment server; only numerical summaries of this follow-up are included in this source bundle.

Table 4: Measured PAMark scale operating points on the original 200 ELI5 prompts, $p = 32$, $L = 5$, 54 bits and 300 tokens. The $\beta = 1.2$ arm is the earlier selected result, not a new rerun. Each row retains its own recovery and PPL.

β	Correct / total	Bit accuracy (%)	PPL	ΔPPL
1.1	165 / 200	93.77	10.50	-0.88
1.15	171 / 200	96.44	14.73	$+3.35$
1.2	189 / 200	98.39	20.80	$+9.41$

Reducing the Gumbel scale lowers the observed PPL cost but also lowers complete-message recovery. At the selected scale, PAMark exceeds MirrorMark base two (131/200), ties base four (171/200), and trails the earlier base-eight arm (186/200) on the original prompts. Base four has bit accuracy 99.34%, PPL 11.26 and ΔPPL -0.12; its paired discordances with PAMark are 24 versus 24. The new base-two result appears in Table 2. Both reruns used four disjoint 50-run shards, preserved global RNG indices, and completed with Gemma scoring in approximately 19.5 minutes each. Their per-run unwatermarked NLLs match the PAMark reference. This is an operating-point comparison, not a universal ranking or monotonic law. The standalone 40-prompt $\beta = 1.15$ pilot recovers 37/40, whereas its separate 200-prompt run recovers 33/40 on the first 40 entries. Pilot generations were not reused or pooled with the larger run.

A.2 Five-Method Verification on Two Hundred New Prompts

The selected candidate $(p, L, \beta) = (32, 5, 1.15)$ was evaluated on ELI5 entries 201–400, which have no prompt-record overlap with the original first 200 entries. It was selected by highest exact recovery among four 40-prompt candidates, with lower ΔPPL breaking ties; the additional five exploratory arms below did not alter this preconfigured selector. MirrorMark $p = 8$, adaptive Binomial with $\epsilon = 0.05$, MPAC and BiMark were rerun on the identical new prompt sequence and payload. All five arms have 200 distinct run indices. Their per-run message IDs, prompt lengths, unwatermarked NLLs and scored-token counts match. The common unwatermarked PPL is 11.29.

Table 5: New-prompt Llama/ELI5 verification: 54 bits, 300 tokens and 200 common prompts. Bit accuracy and exact recovery are percentages. ΔPPL subtracts unwatermarked PPL 11.29. Bold marks the best listed metric, not an overall winner.

Method	Exact	Bit accuracy	PPL	ΔPPL
PAMark	91.00	98.06	15.87	$+4.58$
MirrorMark	**91.50**	**99.60**	10.86	-0.43
Binomial	84.50	98.28	**8.17**	-3.11
MPAC	3.50	92.25	15.86	$+4.58$
BiMark	25.00	96.72	11.23	-0.06

PAMark recovers 182 messages, MirrorMark 183 and Binomial 169. The PAMark–MirrorMark paired discordances are 16 versus 17, consistent with a near tie rather than a PAMark lead. Against Binomial, the corresponding counts are 27 versus 14; the 6.5-point exact-recovery advantage comes with worse average bit accuracy and higher PPL. This follow-up therefore supports a specific recovery–quality trade-off, not unconditional SOTA. It uses one payload and does not establish broad message-value generalization.

A.3 Exploratory Fifty-Four-Bit Candidate Queue

Table 6 selects six of the nine completed 40-prompt candidates to illustrate distinct radix, layer and quality–recovery trade-offs, rather than reproducing the entire queue. Each is a separately generated arm on the original first 40 ELI5 prompts, with unwatermarked PPL 12.08. The rows are not a full factorial grid and are exploratory, not 200-run estimates. Redundant layer-three/layer-eight cells and the radix-48 continuation are omitted from the manuscript, not pooled into the listed results.

Table 6: Selected 40-prompt sensitivity probes at 54 bits and 300 tokens. ΔPPL subtracts 12.08. Asterisk denotes the candidate chosen by the original four-arm selector.

p	L	β	Exact (%)	PPL	ΔPPL
16	5	1.1	72.50	8.74	-3.35
24	5	1.1	82.50	11.88	-0.21
32	10	1.1	90.00	12.31	$+0.23$
36	5	1.1	82.50	9.52	-2.57
32	5	1.05	87.50	11.06	-1.03
32	5	1.15^{*}	92.50	20.23	$+8.14$

At $\beta = 1.1$, $p = 32, L = 10$ is a useful pilot trade-off (36/40 recovery, ΔPPL $+0.23$), but has not received a 200-prompt validation in this queue. Larger radix or layer count does not uniformly improve recovery. The selected $\beta = 1.15$ arm has the highest pilot recovery and a much larger PPL cost; its independent new-prompt comparison, rather than the pilot alone, determines the qualified conclusion above.

A.4 Twenty-Four-Bit Token-Budget Dependence

The completed Llama/C4 suite uses two 24-bit payloads (15429384 and 10295154), 20 shared prompts, temperature one, top-128 sampling and 20–100 generated tokens. PAMark uses $(p, L, \beta) = (12, 60, 1.0)$, Binomial uses adaptive weighting with $\epsilon = 0.1$ and MirrorMark uses $p = 8$. All five methods are retained. PPL pools continuation NLL and scorer token counts. The unwatermarked PPL values are 16.28, 18.95, 13.19 and 12.03 in ascending token-budget order; they are not a common fixed reference.

Table 7: Matched 24-bit Llama/C4 token-budget study. Exact recovery (%) and ΔPPL use 40 outcomes on 20 prompts per column, with budget-specific unwatermarked PPL. Rows are not pooled across budgets.

	20 tokens		40 tokens		60 tokens		100 tokens	
Method	Exact	ΔPPL	Exact	ΔPPL	Exact	ΔPPL	Exact	ΔPPL
PAMark	**10.00**	+4.64	**42.50**	−6.00	65.00	−0.83	82.50	−0.45
Binomial	5.00	+7.01	35.00	−1.95	**75.00**	+3.56	**92.50**	+1.85
MirrorMark	2.50	+1.99	32.50	−3.60	65.00	−1.40	82.50	+0.47
MPAC	0.00	+9.56	0.00	+0.32	7.50	+3.64	7.50	+3.57
BiMark	0.00	+3.34	0.00	−4.44	7.50	−0.49	22.50	+1.26

At 40 tokens, PAMark recovers 10/20 and 7/20 for the two messages, versus Binomial's 8/20 and 6/20. At 20 tokens, all four PAMark successes occur on one message, limiting payload-wide conclusions. Adaptive Binomial leads at 60 and 100 tokens: the short-token advantage does not extrapolate to every continuation length.

A.5 Matched Forty-Token Model and Corpus Transfer

These three completed 24-bit suites fix the messages, token budget and method settings while varying generator or prompt corpus. Each cell has 40 outcomes from two messages and 20 shared prompts. The Llama/C4 column reuses the 40-token rows above and is not an independent replication. PAMark uses $(12, 60, 1.0)$, adaptive Binomial $\epsilon = 0.1$ and MirrorMark $p = 8$.

Table 8: Matched 24-bit, 40-token transfer results: exact recovery (%) and ΔPPL. Unwatermarked PPL is 18.95 for Llama/C4, 18.21 for Qwen/C4 and 13.33 for Llama/ELI5.

	Llama / C4		Qwen / C4		Llama / ELI5	
Method	Exact	ΔPPL	Exact	ΔPPL	Exact	ΔPPL
PAMark	**42.50**	−6.00	**40.00**	−0.89	50.00	+0.37
Binomial	35.00	−1.95	32.50	+1.13	**52.50**	+1.64
MirrorMark	32.50	−3.60	25.00	−0.65	27.50	−0.97
MPAC	0.00	+0.32	0.00	+5.17	0.00	+7.97
BiMark	0.00	−4.44	0.00	−3.08	2.50	+1.48

Against Binomial, PAMark wins 17/40 versus 14/40 on Llama/C4 and 16/40 versus 13/40 on Qwen/C4, but loses 20/40 versus 21/40 on Llama/ELI5. Shared prompts and only two payloads limit generalization; these differences are descriptive rather than significant superiority claims.

A.6 Realized Lexical Edits and Detector Limitations

This diagnostic supplements the 64-bit, 300-token C4 attack study in Table 3. Two payloads share 60 prompts, giving 120 outcomes per arm. Bit accuracy is pooled over these outcomes; AUC and TPR are macro-averages of two message-specific evaluations. Each detector evaluation has 60 calibration negatives, so empirical FPR changes in steps of $1/60$. The nominal 1% rule selected zero observed calibration FPR; no independent detector test set was used. This is not a validated population 1%-FPR result. The legacy CSV field `bit_accuracy_at_1pct_fpr` is an alias for TPR, not mean bit accuracy.

Table 9: Diagnostics for the principal attack-study methods. Syn is requested word-selection percentage; Edits is mean successfully replaced words, not model-token change count. AUC and nominal TPR are calibration diagnostics only.

Model	Syn	Method	Bit (%)	AUC	TPR (%)	Edits
Llama	10	PAMark	95.21	0.999	96.67	12.2
Llama	10	MirrorMark	66.04	0.886	78.33	11.8
Llama	10	Binomial	97.08	1.000	99.17	11.9
Llama	30	PAMark	90.43	0.998	95.83	36.2
Llama	30	MirrorMark	36.29	0.733	39.17	34.8
Llama	30	Binomial	92.33	1.000	100.00	35.8
Qwen	10	PAMark	96.61	0.995	96.67	12.5
Qwen	10	MirrorMark	72.17	0.926	82.50	12.5
Qwen	10	Binomial	98.62	1.000	100.00	12.2
Qwen	30	PAMark	91.56	0.997	97.50	37.0
Qwen	30	MirrorMark	36.03	0.729	26.67	37.0
Qwen	30	Binomial	93.24	1.000	100.00	37.6

Syn-10 and Syn-30 reuse each arm's clean generations. Their clean results must not be counted as independent repeats. WordNet replacements do not verify word sense or semantic preservation, and retokenization can change several token positions per edited word. The larger exact-recovery gain under Syn-30 therefore remains specific to this lexical-editing procedure.

B Supporting Derivations

This appendix derives the mechanism-level claims under their stated fresh-candidate assumptions. These are not guarantees for every finite-key language-model trajectory.

B.1 Distribution of the Full-Payload Score

Assume that all KL cells of an unselected candidate are independent and uniform on $[p]$. Each cell matches the target symbol with probability $1/p$, so

$$X_i \sim \mathrm{Bin}(KL, 1/p), \qquad \mathbb{E}[X_i] = KL/p. \tag{12}$$

The Gumbel coupling uses the score only through the candidate rank. If several candidates have the same score, an independent continuous tie breaker makes their order uniform. Therefore, conditional on the ordered score values, the candidate identities are exchangeable.

Now condition on the selected candidate having score s. The selection rule does not inspect which coordinates produced the s matches. Every cell assignment with exactly s matches consequently has the same conditional probability. The selected vector is uniform over a shell of size

$$\binom{KL}{s}(p-1)^{KL-s}. \tag{13}$$

This proves the symmetry used when translating a full-payload score into coordinate-wise voting evidence.

B.2 Why Gumbel Ranking Favors Higher Matches

Let $g_{(1)} \geq \cdots \geq g_{(n)}$ be the sorted Gumbel variables and let π be the random candidate order induced by the match scores. PAMark assigns $g_{(r)}$ to candidate $\pi(r)$. For a fixed language-model probability vector, a candidate with a better rank receives a stochastically larger perturbation.

Under fresh exchangeable candidates, the random permutation of candidate identities is independent of the ordered score values. Hence the rank mechanism can be analyzed separately from the coordinate identities. When the language-model probabilities are equal, the highest-ranked candidates are weakly more likely to win the final Gumbel competition. With unequal probabilities, the same ordering competes with $\log q_i$ rather than replacing it. This establishes the direction of the ranking signal without claiming that every selected token must have a larger score.

B.3 A Block-MGF Recovery Bound

Fix a message coordinate j and a wrong symbol $a \neq m_j$. Define the token-level margin

$$V_{t,j,a} = \sum_{\ell=1}^{L} \left(\mathbf{1}\{R_{t,\ell,j} = m_j\} - \mathbf{1}\{R_{t,\ell,j} = a\} \right).$$

Assume that token blocks are independent and identically distributed. For any $\theta > 0$, Markov's inequality gives

$$\Pr\left(\sum_{t=1}^{T} V_{t,j,a} \leq 0 \right) = \Pr\left(\exp\left[-\theta \sum_{t=1}^{T} V_{t,j,a} \right] \geq 1 \right) \tag{14}$$

$$\leq \mathbb{E} \exp\left[-\theta \sum_{t=1}^{T} V_{t,j,a} \right] \tag{15}$$

$$= \left(\mathbb{E} e^{-\theta V_{t,j,a}} \right)^T = \phi_L(\theta)^T. \tag{16}$$

Taking the infimum over $\theta \geq 0$ gives the tightest bound available from this one-dimensional block MGF. If exact decoding fails, at least one message coordinate has at least one competing symbol whose accumulated vote is no smaller than the target vote. A union bound therefore yields

$$\Pr(\widehat{\boldsymbol{m}} \neq \boldsymbol{m}) \leq K(p-1) \inf_{\theta \geq 0} \phi_L(\theta)^T. \tag{17}$$

Continuity permits including $\theta = 0$ in the infimum. The bound preserves within-token layer dependence. It becomes invalid as an unconditional statement when repeated PRF inputs or history-dependent candidate distributions destroy the independent-block assumption.

B.4 The Radix Trade-off

For a fixed information payload, increasing p reduces the number of message coordinates but changes the per-cell baseline from $1/2$ to $1/p$. The first change reduces the number of simultaneous coordinate decisions in the union bound; the second reduces the chance that a random cell agrees with its target. Since the selected-candidate surplus also depends on the candidate ranking and language-model probabilities, the two effects cannot be ordered without specifying the operating regime. This is why radix should be treated as an experimental variable rather than fixed to binary by convention.

B.5 A Conditional Design Region for p and L

With $K(p) = \lceil B / \log_2 p \rceil$ and $I_{p,L} = -\log \inf_{\theta \geq 0} \phi_L(\theta)$, the bound above is

$$\Pr(\widehat{\boldsymbol{m}} = \boldsymbol{m}) \geq 1 - K(p)(p-1) e^{-T I_{p,L}}.$$

The right-hand side is at least ρ precisely when $T I_{p,L} \geq \log[K(p)(p-1)/(1-\rho)]$. Intersecting this condition with any stated resource and text-quality constraint $\mathcal{C}$ gives $\mathcal{R}_{\text{suff}}$. The result is sufficient, not necessary: the union bound may be loose and the block distribution may vary with context. Digit values of the message are interchangeable in the ideal symmetric model, so a strictly message-value-dependent optimal interval does not follow from these assumptions.

B.6 Sufficient and Exposure-Limited Token Budgets

Dividing the preceding inequality by $I_{p,L} > 0$ and rounding up yields T_{suff}. For the exposure bound, assume all $nLTK$ candidate cells are independent and uniform on $[p]$. At one coordinate, the probability that the target digit is absent from every candidate cell is $(1 - 1/p)^{nLT}$. The events for the K coordinates are independent, so the probability that every target digit appears at least once is $[1 - (1 - 1/p)^{nLT}]^K$. Exact majority decoding requires this event; thus

$$\Pr(\widehat{\boldsymbol{m}} = \boldsymbol{m}) \leq [1 - (1 - 1/p)^{nLT}]^K.$$

Solving this upper bound against target ρ gives $T \geq T_{\text{exp}}$. Appearance is far from sufficient for a majority; accordingly this necessary bound may be weak. Both formulas depend on fresh-candidate independence and should not be used as unconditional guarantees for repeated-key autoregressive generation.

B.7 Joint Count-Matrix Law Under Fresh Randomness

Fix an exogenous candidate-probability vector q and independent uniform candidate arrays $X_1, \dots, X_n \in [p]^{L\times K}$. Their similarity scores are independent and identically distributed $\mathrm{Bin}(KL, 1/p)$. Uniform random tie priorities ensure that the permutation assigning original token labels to ordered scores is uniform and independent of the ordered score values. The sorted Gumbels are generated independently. Therefore the rank J of the selected candidate is independent of the ordered similarity-score vector, even though J need not be independent of the token label or of q. Writing $a_r(q,\beta) = \Pr(J = r)$ proves

$$\Pr(S_J = s) = \sum_{r=1}^{n} a_r(q,\beta)\Pr(S_{(r)} = s) = w_s.$$

Conditional on $S_J = s$, the selected candidate array is uniform among the $\binom{KL}{s}(p-1)^{KL-s}$ arrays with s target matches: the selection rule sees only the total score and the random rank. A count matrix c with row sums L corresponds to $\prod_j L!/\prod_a c_{j,a}!$ distinct ordered arrays. With $s(c) = \sum_j c_{j,m_j}$, its probability is

$$\mu(c) = w_{s(c)}\frac{\prod_j L!/\prod_a c_{j,a}!}{\binom{KL}{s(c)}(p-1)^{KL-s(c)}}.$$

If the full selected arrays are independent across tokens and the exogenous q is unchanged, the total vote matrix has convolution law μ^{*T}. For a realized total matrix v, the decoder's independent uniform tie-breaking selects the target digit at coordinate j with probability

$$\tau_j(v) = \frac{\mathbf{1}\{v_{j,m_j} = \max_a v_{j,a}\}}{\#\{a : v_{j,a} = \max_b v_{j,b}\}}.$$

Conditional independence of these final tie choices gives whole-message success $\prod_j \tau_j(v)$; averaging gives $Q_{\mathrm{ref}}(T) = \sum_v \mu^{*T}(v)\prod_j \tau_j(v)$. This conditional product must not be replaced by a product of marginal symbol accuracies. If the exogenous q_t changes but remains independent of candidate arrays, the convolution becomes $\mu_{q_1} * \cdots * \mu_{q_T}$. In autoregressive generation q_t depends on previous selected tokens, so this factorization is not automatically valid.

B.8 An Explicit Bounded-Spread Recovery Guarantee

The previous block-MGF bound requires the selected-block law. A more conservative closed-form result follows when the candidate probabilities are fixed externally and have bounded log spread. The argument combines a Gumbel rank-gap event with a whole-token concentration bound for PAMark's random-tie algorithm; it is *not* a guarantee for the autoregressive experiments.

Assume $n \geq 2$, $\beta > 0$, independent uniform candidate cells, independent Gumbel draws, independent random ranking ties, and independent token blocks. At every block let the exogenous candidate-probability vector q be positive and satisfy

$$\max_i \log q_i - \min_i \log q_i \leq R < \infty.$$

The same argument also applies to exogenous varying q_t if the bound R holds uniformly and the resulting blocks remain independent. Write $d = KL$, let $B_1, \dots, B_n$ be independent $\mathrm{Bin}(d, 1/p)$ variables, and set

$$b_\star = \mathbb{E}\max_i B_i = \sum_{s=1}^{d}[1 - F_B(s-1)^n],$$
$$m_\star = \frac{pb_\star - d}{K(p-1)},$$
$$g_n(u) = \frac{n}{1 + (n-1)e^u}, \qquad u = R/\beta.$$

Since $n \geq 2$ and the binomial law is nondegenerate, $b_\star > d/p$ and hence $m_\star > 0$. For any message position j and wrong digit $a \neq m_j$, the selected token's whole-block vote margin satisfies

$$\mathbb{E}V_{t,j,a} \geq g_n(R/\beta)m_\star > 0. \tag{18}$$

Consequently, for each fixed integer $T \geq 1$,

$$\Pr(\widehat{\boldsymbol{m}} = \boldsymbol{m}) \geq 1 - K(p-1)\exp\left[-\frac{Tg_n(R/\beta)^2 m_\star^2}{2L^2}\right]. \tag{19}$$

For a target probability $\rho \in (0,1)$, one sufficient integer token budget is therefore

$$T \geq \left\lceil \frac{2L^2}{g_n(R/\beta)^2 m_\star^2} \log \frac{K(p-1)}{1-\rho} \right\rceil. \tag{20}$$

This is a conservative sufficient condition, not the true minimum or a measured optimum.

Proof. Random tie priorities make the permutation of candidate labels across similarity ranks uniform and independent of the ordered similarity scores. Since q is exogenous, its coordinates in rank order are an exchangeable random permutation, independent of those scores and of the sorted Gumbels. Let $G_{(1)} > \cdots > G_{(n)}$ denote the descending Gumbels. On the event $E = \{G_{(1)} - G_{(2)} > R/\beta\}$, the top-ranked candidate wins regardless of which q_i is assigned to it. Direct integration of the maximum Gumbel density gives

$$\Pr(E) = n \int_{-\infty}^{\infty} e^{-z} e^{-e^{-z}} \exp\left[-(n-1)e^{R/\beta} e^{-z}\right] dz = g_n(R/\beta).$$

On E^c, exchangeability of the rank-ordered probabilities implies that the probabilities of selecting ranks $1, \ldots, n$ are nonincreasing. Indeed, for fixed sorted Gumbels, swapping two exchangeable probability coordinates cannot make the lower-Gumbel rank more likely to win than the higher-Gumbel rank. A nonincreasing probability vector assigns at least as much expected mass to a nonincreasing score vector as the uniform vector: summation by parts writes their difference as a sum of nonnegative prefix-weight deviations times nonnegative adjacent score gaps. Thus the expected selected score on E^c is at least d/p, while on E it is exactly $b_\star$. Therefore

$$\mathbb{E}S_{\text{selected}} \geq g_n(R/\beta) b_\star + [1 - g_n(R/\beta)]d/p.$$

Conditional on selected total score s, shell symmetry makes each of the d cells correct with probability s/d and equal to any particular wrong digit with probability $(d-s)/[d(p-1)]$. Summing the L cells at position j yields

$$\mathbb{E}(V_{t,j,a} \mid S_{\text{selected}} = s) = \frac{ps - d}{K(p-1)}.$$

The score bound proves (18). Each whole-token margin lies in $[-L, L]$; treating layers as one block, Hoeffding's bounded-variable MGF estimate gives

$$\Pr\left(\sum_{t=1}^{T} V_{t,j,a} \leq 0\right) \leq \exp\left[-\frac{Tg_n(R/\beta)^2 m_\star^2}{2L^2}\right].$$

If the full message is decoded incorrectly, at least one of the $K(p-1)$ target-versus-competitor margins is nonpositive. A union bound gives (19), and solving it for T gives (20). This derivation uses neither independent votes within a token nor independent message coordinates.

B.9 Why Random Ranking Ties Matter

The rank arguments above apply to the implemented random-tie rule, not to an arbitrary fixed-index convention. Consider two binary candidates with one message cell each. Their scores are independent Bernoulli$(1/2)$. If a score tie is always resolved in favor of candidate 1, then candidate 1 is assigned rank 1 with probability

$$\Pr(S_1 \geq S_2) = 1 - \Pr(S_1 = 0, S_2 = 1) = \tfrac{3}{4},$$

not $1/2$. The rank-label permutation is therefore not uniform, and the exchangeable rank-noise argument cannot simply be reused. PAMark's experimental implementation draws random tie priorities; a deployment implementation must retain an independently randomized or suitably keyed tie rule if it relies on these idealized proofs.

B.10 Deterministic and Probabilistic Corruption Bounds

At any coordinate, a token-block margin lies in $[-L, L]$. Replacing a block can therefore decrease its target-versus-competitor margin by no more than $2L$, regardless of how the replacement is chosen. After at most r replacements every margin is at least its clean value minus $2Lr$. If the minimum clean margin is strictly larger than $2Lr$, every target remains the unique winner. This proves (9) without distributional or attack-independence assumptions.

For the probability bound, assume independent clean blocks with each expected margin at least $\mu > 0$. For one coordinate and competitor, Hoeffding's inequality applied to T variables of range width $2L$ gives, when $T\mu > 2Lr$,

$$\Pr\left(\sum_t V_{t,j,a} \le 2Lr\right) \le \exp\left[-\frac{2(T\mu - 2Lr)^2}{T(2L)^2}\right].$$

Taking the union over $K(p-1)$ margins proves (10); the bound is clipped at one if necessary. If $T\mu \le 2Lr$, this derivation supplies only the trivial bound one. The attacker may choose which blocks to replace after seeing the sequence, because the deterministic criterion holds simultaneously for every such choice.

For aligned, length-preserving token substitutions, a changed token at position u alters its own hash input and can alter the preceding-context input of positions $u+1, \dots, u+h$. The union of these sets over e substitutions has size at most $(h+1)e$, truncated by the sequence length. This counting argument assumes unchanged token IDs and alignment outside the substitutions. Word replacement followed by tokenization can violate that assumption, so the Syn-10 and Syn-30 experiments are not certified by this token-level bound.

B.11 A Conditional-MGF Extension beyond Independent Blocks

The independent-block model is sufficient but not logically necessary for the concentration step. Let $\mathcal{F}_{t-1}$ contain the past and suppose, for a fixed $\theta > 0$, that a valid bound

$$\mathbb{E}[e^{-\theta V_{t,j,a}} \mid \mathcal{F}_{t-1}] \le e^{-i_t(\theta)}$$

holds almost surely with deterministic $i_t(\theta)$. Iterating conditional expectations yields

$$\mathbb{E}e^{-\theta \sum_t V_{t,j,a}} \le e^{-\sum_t i_t(\theta)}, \qquad \Pr\left(\sum_t V_{t,j,a} \le 2Lr\right) \le e^{2\theta Lr - \sum_t i_t(\theta)}.$$

One may optimize over a common θ and sum over coordinates and competitors. Importantly, positivity of the *unconditional* empirical mean does not imply this almost-sure conditional bound. Establishing it for the real autoregressive PRNG construction is an open analysis task, not an assumption verified by the reported experiments.

B.12 Why Mean Bit Accuracy Does Not Order Exact Recovery

Let E count erroneous bits and write $a = 1 - \mathbb{E}E/B$. Splitting the expectation according to whether $E = 0$ proves (11). Since $1 \le E \le B$ on a failed message,

$$\max\{0, 1 - B(1-a)\} \le \Pr(E = 0) \le a.$$

For example, distributing one error across every message yields higher average bit accuracy than concentrating several errors in a small fraction of messages, yet the former can have zero exact recovery. Thus neither a^B nor a ranking by mean bit accuracy is a justified substitute for measured whole-message recovery unless a suitable independent-error model is established.