TRELLISMARK: MULTI-BIT LLM WATERMARKING WITH TRELLIS COUPLING AND ONLINE STRENGTH ALLOCATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-bit watermarking enables large language models to embed rich structured messages, such as user and model identifiers, into generated text, supporting provenance attribution and misuse tracing. Existing methods either let each token contribute evidence to only one payload position, leaving each position with limited evidence, or aggregate evidence over complete messages, making exact decoding costly as the payload grows. We introduce `TrellisMark`, which couples neighboring payload positions through overlapping local windows. Each token provides evidence about the symbols in one selected window, while multiple independent heads over the same window strengthen this evidence. The overlap induces a trellis over candidate payloads, enabling exact maximum-score recovery without enumerating the full message space. We further formulate watermark-strength selection as an online allocation problem driven by the current trellis recovery state. Under a cumulative KL-divergence budget, the policy assigns greater strength to steps with higher recovery value. We prove a recovery bound based on trellis-path separation and a regret bound for online allocation. Experiments show improved exact message recovery and better recovery–quality trade-offs over existing baselines.

1 INTRODUCTION

Large language model (LLM) watermarking embeds verifiable signals into generated text, providing a technical means for provenance authentication and misuse tracing (Rijsbosch et al., 2025). Most existing watermarking methods are zero-bit: the detector determines only whether a given text contains a watermark (Kirchenbauer et al., 2023; Zhao et al., 2023; Liu et al., 2023). While sufficient for binary provenance detection, such signals cannot distinguish among different models or users. Multi-bit watermarking extends this capability by embedding structured messages, such as model identifiers or provenance records, into generated text, enabling richer attribution and tracing beyond binary detection (Yoo et al., 2024; Jiang et al., 2025b; Feng et al., 2025; Qu et al., 2024).

Reliable message recovery becomes increasingly difficult as the message length grows, especially when only a limited number of generated tokens are available for embedding. A common strategy is to associate different tokens with individual payload positions (Yoo et al., 2024; Qu et al., 2024). While this keeps decoding simple, it fragments the available evidence: each payload symbol is supported by only a fraction of the generated tokens, making exact message recovery vulnerable to poorly supported positions. At the other extreme, associating each observation with the entire payload allows every token to provide evidence about the full message (Gilani et al., 2026), but exact decoding over the resulting message space quickly becomes computationally prohibitive as the message length grows. This raises a central question: *how can token-level evidence be shared across payload positions to support reliable message recovery while retaining efficient decoding?*

We address this challenge with `TrellisMark`, which couples neighboring payload positions through overlapping windows of w consecutive payload symbols. Each token provides evidence about the payload symbols within one such window. At each generation step, the selected window is mapped to one of q target colors. To strengthen the evidence from each token, we apply H independent mappings (heads) to the same window. Each head produces its own target color, and a token receives one match when its color under that head equals the target. Tokens matching more

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.

heads receive a larger bias in the next-token distribution. During detection, each candidate window assignment receives a local score based on how many heads agree with the observed token colors. These scores are accumulated across generation steps. Because neighboring windows overlap in $w - 1$ symbols, assignments that agree on the shared symbols form transitions in a trellis, with each complete candidate message corresponding to a path. The maximum-scoring message is therefore the maximum-score trellis path and can be recovered exactly using Viterbi decoding (Forney, 2005), without enumerating the full message space. We further prove a finite-length recovery bound based on the separation between the embedded and competing trellis paths.

Building on this trellis structure, we use the current path scores to identify when additional watermark evidence is needed. A competing path that remains close to the embedded path indicates insufficient recovery margin, while the current local mappings determine whether the present step assigns different target colors to the two paths. Together, these two signals determine the recovery value of the current step. We formulate watermark-strength selection as an online resource allocation problem under a cumulative KL-divergence budget, which controls the deviation of the watermarked next-token distributions from the original model distributions. The policy allocates more strength to steps with higher recovery value and preserves budget otherwise, improving the recovery–quality trade-off. We further show that the online policy achieves vanishing regret relative to the hindsight-optimal allocation.

Our main contributions are as follows:

- We propose `TrellisMark`, which couples neighboring payload positions through overlapping local windows, inducing a trellis over candidate messages. This structure enables exact recovery without enumerating the full message space. Multiple independent heads further strengthen the evidence from each generated token.
- We formulate watermark-strength selection as an online resource allocation problem under a cumulative KL-divergence budget. This adaptive allocation improves the recovery–quality trade-off by allocating more distortion to steps with higher recovery value.
- We prove a finite-length recovery bound for the trellis construction and a regret bound for the online allocation policy.
- Experiments show improved exact message recovery and recovery–quality trade-offs. For 16-bit payloads, `TrellisMark` reaches 100% exact recovery, while the strongest baseline remains below 70% in the same quality range.

2 Related Work

LLM Watermarking. LLM watermarking typically embeds detectable signals by modifying the next-token distribution during autoregressive generation. Kirchenbauer et al. (Kirchenbauer et al., 2023) introduced a representative red–green watermark that pseudorandomly partitions the vocabulary into keyed token groups and biases generation toward tokens in a green list. Subsequent work has focused on improving the quality–detectability trade-off (Lee et al., 2024; Huo et al., 2024), preserving the original generation distribution (Kuditipudi et al., 2023; Dathathri et al., 2024), and enhancing robustness to different attacks (Zhao et al., 2023; Shen et al., 2026).

Multi-bit Watermarking. Multi-bit watermarks embed structured payloads beyond binary detection. MPAC assigns each generation step to one payload position and biases a corresponding vocabulary color (Yoo et al., 2024). RSBH adds balanced assignment and Reed–Solomon coding (Qu et al., 2024). BiMark and StealthInk focus on unbiased reweighting (Feng et al., 2025; Jiang et al., 2025b), while ArcMark encodes the full message into each token using channel coding and optimal transport (Gilani et al., 2026). Existing designs therefore either collect evidence locally for individual payload symbols or operate at the message level, where exact decoding becomes costly as the payload grows. Our approach bridges these regimes by coupling each token to a short payload window while retaining efficient trellis decoding.

Online Resource Allocation. Our adaptive watermark-strength allocation builds on online resource allocation, where sequential decisions consume limited resources and dual prices balance current reward against future scarcity (Mehta et al., 2007; Agrawal et al., 2014; Balseiro et al., 2023). Recent

work extends this framework to nonstationary environments, with performance depending on the extent of distributional variation (Balseiro et al., 2023; Jiang et al., 2025a). The closest work is Cai et al. (2026), who use online dual optimization to balance detection and KL distortion for red–green watermarks. Their optimization ultimately learns a common watermark strength for the generated sequence. Our focus is multi-bit recovery, where different tokens have different value for separating the true payload from competing messages. We therefore allocate the KL budget according to this recovery value, placing more strength where it helps decoding most.

3 Problem Formulation

Let P_θ be an autoregressive language model with vocabulary $\mathcal{V}$. Given a prompt $x_{\le 0}$, the model generates a sequence $Y = (y_1, \dots, y_T)$ autoregressively. At each step t, its next-token distribution conditioned on the prompt and previously generated tokens is $p_t(v) = P_\theta\big(v \mid x_{\le 0}, y_{<t}\big)$ for $v \in \mathcal{V}$. A multi-bit watermark encoder takes as input a secret key κ and an L-bit payload $\mathbf{m} = (m_1, \dots, m_L) \in \{0,1\}^L$, and modifies p_t into a watermarked distribution q_t that encodes information about $\mathbf{m}$. The next token is sampled as $y_t \sim q_t$, and the process repeats until the full sequence Y is generated. Given Y and the secret key κ, the detector reconstructs the payload as $\widehat{\mathbf{m}} = \mathrm{Dec}_\kappa(Y)$. Our primary objective is exact payload recovery, i.e., $\widehat{\mathbf{m}} = \mathbf{m}$.

Improving payload recoverability generally requires stronger perturbations of the original next-token distribution, creating a trade-off between message recovery and distributional fidelity. We measure the distortion at step t using the ***Kullback–Leibler divergence***,

$$D_t = D_{\mathrm{KL}}(q_t \| p_t) = \sum_{v \in \mathcal{V}} q_t(v) \log \frac{q_t(v)}{p_t(v)}. \tag{1}$$

KL is well suited to autoregressive generation because sequence-level KL decomposes exactly into the expected sum of conditional token-level KL divergences (Cai et al., 2026). We impose an average per-token KL budget $\bar{b}$, $\frac{1}{T}\sum_{t=1}^{T} D_t \le \bar{b}$, which controls the average distributional distortion introduced during generation. For a generation horizon T, this is equivalently a cumulative budget $B = T\bar{b}$. We report perplexity separately as a complementary measure of realized text quality; see detailed discussion in Appendix D.

Our goal is therefore to maximize the probability of exact payload recovery subject to the distortion constraint:

$$\max \ \Pr[\widehat{\mathbf{m}} = \mathbf{m}] \qquad \text{s.t.} \qquad \frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}(q_t \| p_t) \le \bar{b}. \tag{2}$$

The recovery objective is defined over the complete generated sequence, whereas the encoder must choose each q_t sequentially while consuming the shared distortion budget. This mismatch between a sequence-level recovery objective and sequential token-level decisions motivates the online strength-allocation problem developed in Section 5.

4 Trellis-Coupled Embedding and Recovery

This section describes the trellis-coupled embedding and recovery mechanism of `TrellisMark`. Figure 1 provides an overview of `TrellisMark`.

4.1 Local Payload Mapping and Trellis Structure

Let $\mathbf{m} \in \{0,1\}^L$ be the payload defined in Section 3. We group its bits into q-ary symbols $\mathbf{s} = (s_1, \dots, s_N) \in \mathbb{F}_q^N$, where $N = L/\log_2 q$. For boundary handling, we set $s_j = 0$ outside $j \in [N]$ and append r zero-termination stages after the payload, giving $N + r$ queryable stages; see Appendix G. At generation step t, a keyed position map applied to the current n-gram context selects $J_t = \rho_\kappa(y_{t-n:t-1}) \in [N + r]$. The watermark depends only on the local window $\mathbf{s}_{J_t - r : J_t}$ of width $w := r + 1$.

The secret key and current n-gram context determine a coefficient vector $\mathbf{a}_t \in \mathbb{F}_q^{r+1}$ and a token-coloring function $\chi_t : \mathcal{V} \to \mathbb{F}_q$. For an assignment of the selected window $\mathbf{u} \in \mathbb{F}_q^{r+1}$, we define the

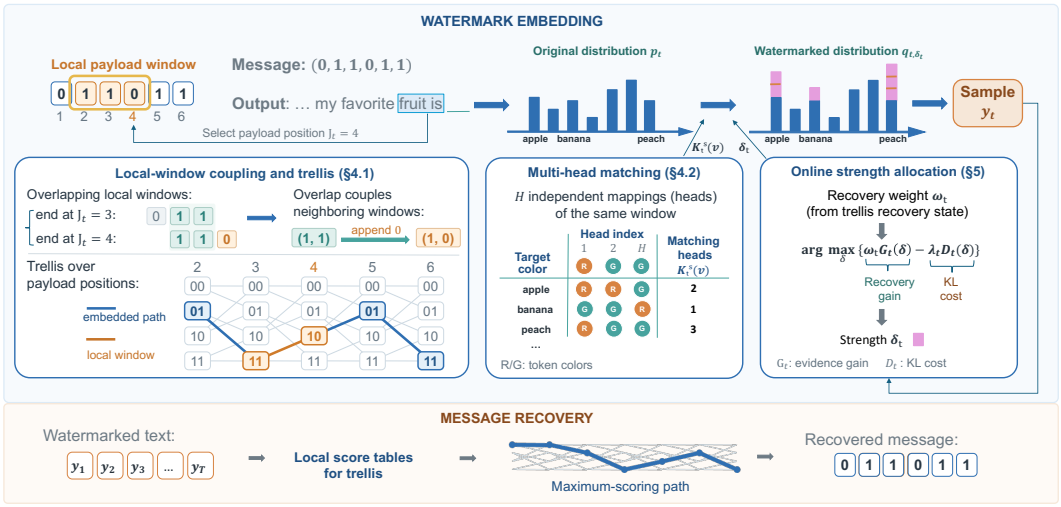


Figure 1: Overview of `TrellisMark`. **Left:** At each generation step, the current context selects a local payload window. Overlap between neighboring windows couples payload positions and induces a trellis over candidate payloads. **Center:** Multiple independent heads map the selected window to target token colors, and the next-token distribution is reweighted according to the number of matching heads. **Right:** The watermark strength is selected online using a recovery weight derived from the current trellis recovery state, balancing evidence gain against KL cost under a cumulative KL-divergence budget. **Bottom:** During recovery, the observed token colors are aggregated into local trellis scores, and the highest-scoring path gives the recovered message.

keyed local payload mapping

$$c_t(\mathbf{u}) = \langle \mathbf{a}_t, \mathbf{u} \rangle_{\mathbb{F}_q}. \tag{3}$$

This mapping assigns the local window to one of q target colors. The target color of the embedded payload is then $c_t(\mathbf{s}) := c_t(\mathbf{s}_{J_t-r:J_t})$. A token $v \in \mathcal{V}$ matches the target when $\chi_t(v) = c_t(\mathbf{s})$.

The overlap between local windows induces a trellis over stages $j = 1, \ldots, N + r$. Although generation queries these stages in the context-dependent order $J_1, \ldots, J_T$, the underlying windows are indexed by their endpoints j. Consecutive windows $\mathbf{s}'_{j-r:j}$ and $\mathbf{s}'_{j-r+1:j+1}$ overlap in r symbols, coupling neighboring assignments. For a candidate payload $\mathbf{s}' = (s'_1, \ldots, s'_N)$, with $s'_j = 0$ outside $j \in [N]$, define the trellis state at stage j as $\sigma_j = (s'_{j-r+1}, \ldots, s'_j) \in \mathbb{F}_q^r$. Appending s'_{j+1} moves to $\sigma_{j+1} = (s'_{j-r+2}, \ldots, s'_{j+1})$ and corresponds to the local window $\mathbf{s}'_{j-r+1:j+1}$. Repeating these transitions across payload positions yields a trellis path for each complete candidate payload; further details are given in Appendix B.

4.2 Multi-Head Matching and Watermark Embedding

A single local payload mapping provides only one target-color comparison for each candidate token. We therefore use H independently keyed heads over the same payload window $\mathbf{s}_{J_t-r:J_t}$. Each head h uses a coefficient vector $\mathbf{a}_{t,h} \in \mathbb{F}_q^{r+1}$ and a token-coloring function $\chi_{t,h} : \mathcal{V} \to \mathbb{F}_q$. Following Eq. 3, head h produces the target color $c_{t,h}(\mathbf{s}) = \langle \mathbf{a}_{t,h}, \mathbf{s}_{J_t-r:J_t} \rangle_{\mathbb{F}_q}$. For a candidate token v, we count the heads for which its token color matches the corresponding target color:

$$K_t^{\mathbf{s}}(v) = \sum_{h=1}^{H} \mathbf{1}\{\chi_{t,h}(v) = c_{t,h}(\mathbf{s})\}. \tag{4}$$

Thus, $K_t^{\mathbf{s}}(v) \in \{0, \ldots, H\}$ is the number of heads on which token v matches the target induced by the embedded payload. We convert this count to a scalar score using $\rho(0) = 0$ and $\rho(b) = 1 + \beta(b-1)$ for $b \geq 1$, where $\beta \in [0, 1]$ controls the additional contribution of multiple matches.

We embed the watermark by reweighting the original next-token distribution according to this score:

$$q_{t,\delta_t}(v) = \frac{p_t(v) \exp(\delta_t \rho(K_t^{\mathbf{s}}(v)))}{Z_t(\delta_t)}, \qquad Z_t(\delta_t) = \sum_{v' \in \mathcal{V}} p_t(v') \exp(\delta_t \rho(K_t^{\mathbf{s}}(v'))). \tag{5}$$

Here, $\delta_t \geq 0$ controls the watermark strength. A larger δ_t shifts more probability mass toward tokens with higher match scores, strengthening evidence for the embedded payload at the cost of greater distortion from $p_t(\cdot)$. Section 5 describes how δ_t is chosen online.

4.3 Trellis Scoring and Message Recovery

Given a generated sequence $\mathbf{Y} = (y_1, \ldots, y_T)$, the detector reconstructs the keyed quantities from the secret key and the corresponding n-gram contexts. For each generation step t and head h, it recovers J_t, $\mathbf{a}_{t,h}$, and $\chi_{t,h}$, and computes the observed token color $z_{t,h} = \chi_{t,h}(y_t)$. For each payload position j, the detector aggregates all observations with $J_t = j$ into a local score table. For a candidate local window assignment $\mathbf{u} \in \mathbb{F}_q^{r+1}$, define

$$\Phi_j(\mathbf{u}) = \sum_{t:J_t=j}^{T} \sum_{h=1}^{H} \mathbf{1}\left\{ z_{t,h} = \langle \mathbf{a}_{t,h}, \mathbf{u} \rangle_{\mathbb{F}_q} \right\}. \tag{6}$$

Thus, $\Phi_j(\mathbf{u})$ counts how often the observed token colors match the target colors predicted by $\mathbf{u}$ across all heads and generation steps querying position j.

For a complete candidate payload $\mathbf{s}' = (s'_1, \ldots, s'_N)$, its decoding score is

$$S_T(\mathbf{s}') = \sum_{j=1}^{N} \Phi_j\left(\mathbf{s}'_{j-r:j}\right). \tag{7}$$

Under the trellis representation in Section 4.1, each local assignment $\mathbf{s}'_{j-r:j}$ identifies the trellis branch ending at payload position j, and $\Phi_j(\mathbf{s}'_{j-r:j})$ is its branch score. Hence, $S_T(\mathbf{s}')$ is the sum of branch scores along the trellis path corresponding to $\mathbf{s}'$. The decoder recovers $\widehat{\mathbf{s}} = \arg\max_{\mathbf{s}' \in \mathbb{F}_q^N} S_T(\mathbf{s}')$. Because the score is additive along trellis branches, the resulting maximum-score path problem can be solved exactly using Viterbi decoding (Forney, 2005), without enumerating all q^N candidate payloads. With q^r states, the resulting decoding complexity is $O((TH + N)q^{r+1})$; further details are given in Appendix B.

Finite-length recovery. Because recovery is based on a finite number of generated tokens, we next quantify how local overlap separates the embedded payload from competing payloads. Under the zero-terminated trellis, a competitor differing in d payload symbols affects at least $r + d$ queryable stages. Combined with sufficient context diversity and a cumulative score-margin condition, this structural separation yields the following finite-length guarantee. Formal assumptions and constants are given in Appendix C.

Theorem 1 (Finite-length recovery under local coupling)**.** *Suppose that, with probability at least $1 - \varepsilon_{\mathrm{ctx}}$, at least a γ-fraction of the T generation contexts are distinct, and that the cumulative target–competitor score-margin condition holds with probability at least $1 - \varepsilon_{\mathrm{mar}}$. Then, for any competing payload $\widetilde{\mathbf{s}} \neq \mathbf{s}$ that differs from the embedded payload $\mathbf{s}$ in $d \geq 1$ symbols,*

$$\Pr[S_T(\widetilde{\mathbf{s}}) \geq S_T(\mathbf{s})] \leq \varepsilon_{\mathrm{ctx}} + \varepsilon_{\mathrm{mar}} + 3\exp\left(-c_{\mathrm{rec}} \frac{\gamma T}{N}(r + d)\right),$$

where $c_{\mathrm{rec}} > 0$ is the recovery-separation constant given in Appendix C. Thus, local coupling strengthens separation according to both the memory length r and the number d of mismatched payload symbols. Summing this bound over all competing payloads yields the sufficient scaling

$$T \gtrsim \frac{N}{\gamma c_{\mathrm{rec}}}\left(1 + \log \frac{N}{r+1}\right),$$

under which the finite-length term vanishes and the total decoding error is at most $\varepsilon_{\mathrm{ctx}} + \varepsilon_{\mathrm{mar}} + o(1)$. The number of heads H enters this guarantee only through c_{rec}. Increasing H can therefore improve the recovery constant when head-wise separation is the bottleneck, but under the constants established in Appendix C, this improvement saturates for $H \geq 2$.

The theorem highlights two effects of the measurement design. First, local overlap amplifies the evidence against nearby competitors: even a payload differing in a single symbol affects $r + 1$ queryable stages, which produces the $r + d$ dependence in the exponent. Consequently, for fixed context-diversity and recovery-separation constants, a generation length of order $O(N \log N)$ is sufficient for recovery. Second, multiple heads strengthen the per-stage separation only until another, H-independent part of the recovery bound becomes limiting. This explains why the worst-case guarantee improves initially with H but saturates beyond two heads, consistent with the diminishing empirical gains in Figure 4c.

5 Online Watermark Strength Allocation

The recovery value of watermarking varies across generation steps. We therefore select the watermark strength online using the current trellis recovery state. The encoder allocates more KL budget when additional evidence is valuable. Otherwise, it preserves budget for unknown future opportunities.

5.1 Recovery Weight from Trellis Scores

Before sampling y_t, define the prefix score of a candidate payload $\widetilde{\mathbf{s}}$ as

$$S_{t-1}(\widetilde{\mathbf{s}}) = \sum_{\tau=1}^{t-1} \sum_{h=1}^{H} \mathbf{1}\left\{z_{\tau,h} = c_{\tau,h}\left(\widetilde{\mathbf{s}}_{J_\tau - r:J_\tau}\right)\right\}. \tag{8}$$

After all T generation steps, this gives the decoding score $S_T(\widetilde{\mathbf{s}})$ defined in Section 4.3. The value of increasing the watermark strength at step t depends on two factors: whether a strong competing path is close to the embedded payload, and whether the current local mappings distinguish that competitor from the target. We quantify these two factors below.

Since step t depends only on the local payload window ending at J_t, we group competing paths by their local assignment $\mathbf{u} \in \mathbb{F}_q^{r+1}$. Define $C_t(\mathbf{u}) = \max_{\widetilde{\mathbf{s}} \neq \mathbf{s}:\, \widetilde{\mathbf{s}}_{J_t - r:J_t} = \mathbf{u}} S_{t-1}(\widetilde{\mathbf{s}})$ as the score of the strongest competing path consistent with $\mathbf{u}$. Its margin from the embedded payload is $M_t(\mathbf{u}) = S_{t-1}(\mathbf{s}) - C_t(\mathbf{u})$. Given a target margin μ, we define the remaining margin deficit as $\mathrm{Def}_t(\mathbf{u}) = \tau_d \,\mathrm{softplus}((\mu - M_t(\mathbf{u}))/\tau_d)$, where $\mathrm{softplus}(x) = \log(1 + e^x)$ and $\tau_d > 0$ controls smoothness. The deficit is large when the target is close to or behind the competitor, and becomes small once the desired margin has been reached.

A large deficit alone does not imply that stronger watermarking is useful at the current step. The local mappings must also distinguish the competing assignment from the target. We measure this separation by

$$\mathrm{Sep}_t(\mathbf{u}) = \frac{1}{H} \sum_{h=1}^{H} \mathbf{1}\left\{c_{t,h}\left(\mathbf{s}_{J_t - r:J_t}\right) \neq c_{t,h}(\mathbf{u})\right\}.$$

We combine the margin deficit and current-step separation into the recovery weight

$$\omega_t = \max_{\mathbf{u} \in \mathbb{F}_q^{r+1}} \mathrm{Def}_t(\mathbf{u}) \,\mathrm{Sep}_t(\mathbf{u}). \tag{9}$$

Thus, ω_t is large when a strong competitor remains close to the embedded payload and the current mappings can distinguish the two. The constrained maximum defining $C_t(\mathbf{u})$ can be computed efficiently using forward and backward trellis scores; details are given in Appendix B.

5.2 Online Strength Selection

The recovery weight ω_t indicates when additional watermark evidence is useful. We next quantify how much evidence a candidate strength δ adds and the KL distortion required to obtain it.

Evidence gain and KL cost. Recall that $K_t^{\mathbf{s}}(v)$ counts the number of heads matched by token v, and that $\rho(K_t^{\mathbf{s}}(v))$ is the match score used for watermark embedding. For a candidate strength δ, define the evidence gain relative to the original next-token distribution as

$$G_t(\delta) = \mathbb{E}_{v \sim q_{t,\delta}}\left[\rho(K_t^{\mathbf{s}}(v))\right] - \mathbb{E}_{v \sim p_t}\left[\rho(K_t^{\mathbf{s}}(v))\right]. \tag{10}$$

This quantity measures the increase in expected target-consistent match score produced by watermarking at the current step. The corresponding distortion cost is $D_t(\delta) = D_{\mathrm{KL}}(q_{t,\delta} \| p_t)$, consistent with Section 3. Both $G_t(\delta)$ and $D_t(\delta)$ depend on tokens only through the $H + 1$ possible values of $K_t^{\mathbf{s}}(v)$, and both are nondecreasing in δ. Together with ω_t, these quantities determine the online allocation of watermark strength. Appendix E characterizes the resulting evidence–distortion relation.

Online strength selection. At step t, the weighted evidence gain from using strength δ is $\omega_t G_t(\delta)$. Since both the recovery weight and the next-token distribution depend on the generated prefix, the encoder must choose δ_t online without knowing the opportunities available at later steps. To balance

current recovery gain against preserving KL budget for the future, we maintain a scalar $\lambda_t \geq 0$ that measures the scarcity of distortion budget: when λ_t is large, spending KL is treated as more costly; when it is small, the encoder can allocate distortion more aggressively.

The exponential-tilting structure gives $D_t'(\delta) = \delta G_t'(\delta)$, so the unconstrained primal–dual choice has the closed form

$$\widetilde{\delta}_t = \min\{\delta_{\max}, \omega_t/\lambda_t\}. \tag{11}$$

We then clip $\widetilde{\delta}_t$ only when necessary to respect the remaining KL budget. Thus, ω_t/λ_t determines the desired watermark strength, while the clipping step guarantees pathwise satisfaction of the cumulative KL constraint. The exact clipping rule and boundary cases are given in Appendix F.

After sampling $y_t \sim q_{t,\delta_t}$, we update the distortion price as

$$\lambda_{t+1} = \left[\lambda_t + \eta_\lambda \left(D_t(\delta_t) - \bar{b}\right)\right]_+, \tag{12}$$

where $\eta_\lambda > 0$ is the update step size and $\bar{b}$ is the target average KL budget per generation step. Spending more than $\bar{b}$ raises the future price of distortion, while spending less lowers it. The encoder can therefore shift KL budget toward more valuable recovery opportunities without imposing the same distortion at every step. Algorithms 1 and 2 in Appendix A summarize the complete embedding and recovery procedures.

Regret guarantee. We compare our policy with the hindsight-optimal allocation of the same KL budget over the realized recovery opportunities. Under a mild temporal-dispersion condition requiring the hindsight-optimal KL expenditure to have sublinear cumulative deviation from proportional spending, the clipped online allocator asymptotically matches the per-token recovery reward of the clairvoyant hindsight allocation while satisfying the KL budget on every trajectory. Specifically, if this deviation grows as $O(T^\alpha)$ for some $\alpha < 1$, an appropriate diminishing stepsize yields regret $O(T^{-(1-\alpha)/2})$. The formal statement and proof are provided in Appendix F.

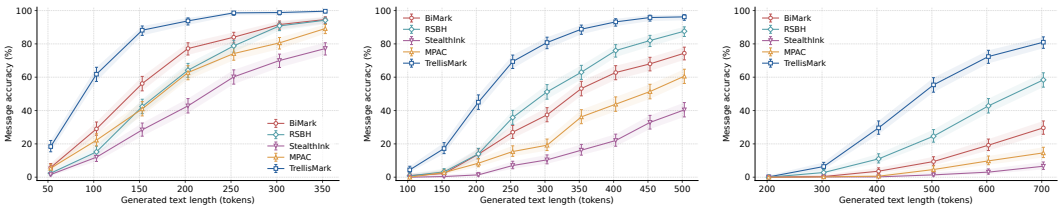
6 Experiments

Experimental Setup. We conduct our main experiments on the RealNewsLike subset of the C4 corpus (Raffel et al., 2020). For each document, we use the first 30 tokens as the prompt and generate the subsequent text with Llama-3.2-3B-Instruct (Meta, 2024), using a temperature of 0.8, top-p of 0.95, and top-k of 50. We compare against four representative multi-bit watermarking methods: MPAC (Yoo et al., 2024), RSBH (Qu et al., 2024), StealthInk (Jiang et al., 2025b), and BiMark (Feng et al., 2025). We additionally evaluate on the LFQA dataset (Krishna et al., 2023) and Qwen-3-4B-Instruct (Qwen, 2025); these results are reported in Appendix H, with implementation details and baseline hyperparameters provided in Appendix G.

Evaluation Metrics. We use exact message recovery as the primary recovery metric, counting a message as correct only when all payload bits are recovered correctly. We measure distributional distortion by the cumulative KL divergence between the watermarked and base-model next-token distributions, which is also the resource controlled by our allocation policy. To separately evaluate the quality of the generated text, we report perplexity under OPT-1.3B model. We discuss the relationship between KL divergence and perplexity in Appendix D.

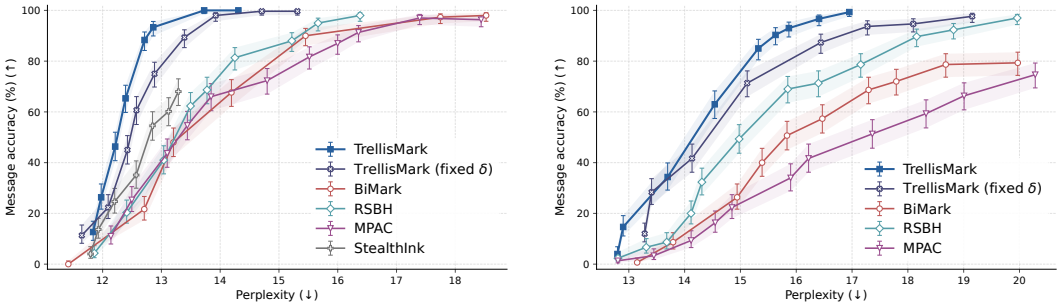
6.1 Message Recovery Performance

We evaluate exact message recovery for 16-, 32-, and 64-bit payloads as the generation length increases. As shown in Figure 2, `TrellisMark` consistently achieves higher recovery accuracy than all baselines across payload sizes, with the gap being especially pronounced at shorter generation lengths. For example, with a 32-bit payload, TrellisMark reaches approximately 89% accuracy at 350 tokens, compared with 63% for the strongest baseline. The advantage remains substantial for longer payloads, reaching 81% accuracy for 64 bits at 700 tokens versus 58% for the strongest baseline. Overall, longer payloads require more generated tokens for accurate recovery, but `TrellisMark` reaches comparable accuracy with substantially shorter generations than the baselines.



(a) 16-bit payload (b) 32-bit payload (c) 64-bit payload

Figure 2: Exact message recovery versus generation length for 16-, 32-, and 64-bit payloads. Each point is evaluated over 500 prompts; error bars denote 95% Wilson confidence intervals. Higher is better.



(a) 16-bit payload. (b) 32-bit payload.

Figure 3: Recovery–quality trade-off for 16- and 32-bit payloads. Each point corresponds to a different watermark-strength setting. Upper left is better; error bars denote 95% Wilson confidence intervals.

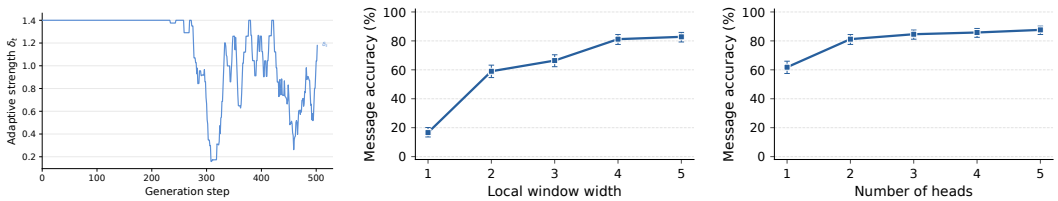
6.2 Recovery–Quality Trade-off

Recovery–Quality Frontier. Figure 3 compares the recovery–quality trade-off for 16- and 32-bit payloads. Across both payload sizes, TrellisMark consistently traces the upper-left frontier, achieving higher exact recovery at comparable perplexity than both the fixed-strength variant and existing baselines. The advantage is especially clear in the high-recovery regime, where adaptive allocation reaches near-perfect recovery at substantially lower perplexity than applying a fixed watermark strength. These results show that online strength allocation improves the recovery–quality trade-off across different payload sizes.

Adaptive Strength Allocation. Figure 4a illustrates the behavior of the online allocation policy during generation. The policy initially uses the maximum strength $\delta_{\max}$, when recovery evidence is still limited, and subsequently varies δ_t over a much wider range as the trellis state evolves. The repeated decreases and renewed increases show that the policy is not following a fixed decay schedule: it reduces strength when additional evidence has lower value and increases it again when new recovery opportunities arise. This adaptive behavior is precisely what Eq. 11 is designed to achieve, reallocating watermark strength in response to the evolving recovery–distortion trade-off.

6.3 Ablation Study

Effect of Trellis Coupling. Figure 4b varies the local window width w, which controls the overlap between neighboring payload windows. With $w = 1$, adjacent windows do not overlap and exact recovery is only about 17%. Introducing overlap at $w = 2$ increases recovery to about 59%, and performance reaches about 81% at $w = 4$, with only a small additional gain at $w = 5$. We therefore use $w = 4$ by default, which captures most of the recovery benefit while avoiding the rapidly growing trellis state space at larger widths.



(a) Adaptive strength over generation.

(b) Local window width w.

(c) Number of heads H.

Figure 4: Analysis of adaptive strength allocation and trellis design. (a) Token-level watermark strength for a representative generation. (b) Effect of the local window width w. (c) Effect of the number of heads H. Error bars in (b) and (c) denote 95% Wilson confidence intervals.

Table 1: Clean perplexity (PPL) and exact recovery rate (%) for 32-bit payloads under text-editing and paraphrase attacks. All methods are evaluated on the same C4 prompts with 450 generated tokens. Parrot (Prithivida, 2022) uses a beam size of 10, while DIPPER (Krishna et al., 2023) uses lexical and order diversity values of 20. Each result uses 500 samples. Bold and underlined values denote the best and second-best results, respectively; lower is better for PPL, while higher is better for recovery.

			Word deletion		Synonym replacement		Paraphrasing	
Method	PPL ↓	Clean	10%	30%	10%	30%	Parrot	DIPPER
BiMark	15.9	62.8	9.4	0.2	9.6	0.4	37.2	1.0
MPAC	16.2	43.8	12.8	0.6	14.8	<u>1.6</u>	28.6	<u>1.8</u>
StealthInk	**13.7**	22.0	1.0	0.0	1.0	0.0	11.8	0.0
RSBH	15.7	<u>76.0</u>	<u>32.4</u>	<u>2.4</u>	<u>26.2</u>	**3.0**	<u>55.4</u>	<u>1.8</u>
`TrellisMark`	<u>15.0</u>	**93.2**	**36.6**	**2.8**	**33.4**	0.4	**75.2**	**4.0**

Effect of Multi-Head Matching. Figure 4c shows that multiple independent heads substantially improve recovery. Increasing H from 1 to 2 raises exact recovery from about 62% to 82%, while further increasing the number of heads yields much smaller gains. This diminishing return is consistent with the recovery analysis in Section 4.3, which predicts that additional heads cease to be the dominant bottleneck once projection separation becomes sufficiently reliable. We therefore use $H = 2$ by default.

6.4 Robustness

Table 1 evaluates exact payload recovery under word deletion, synonym replacement, and paraphrasing attacks for 32-bit payloads. `TrellisMark` achieves the best recovery in six of the seven reported settings. On clean text, it reaches 93.2% exact recovery at a PPL of 15.0, outperforming the best baseline by 17.2 percentage points. The advantage persists under moderate edits, with improvements of 4.2 and 7.2 percentage points under 10% word deletion and synonym replacement, respectively. `TrellisMark` also shows substantially stronger paraphrase robustness, improving over the best baseline by 19.8 percentage points under Parrot and by 2.2 points under DIPPER. The only exception is the 30% synonym-replacement setting. As attack strength increases, exact recovery drops substantially for all methods.

7 Conclusion

We presented `TrellisMark`, a trellis-coupled multi-bit watermarking method for reliable message recovery. We further develop an online watermark-strength allocation strategy under a cumulative KL-divergence budget to improve the recovery–quality trade-off. We prove finite-length recovery and online regret guarantees, and experiments across different payload sizes, datasets, and language models demonstrate improved exact message recovery and recovery–quality trade-offs over existing methods.

AI USE STATEMENT

Generative AI tools were used to polish the manuscript and code for clarity and readability. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final content.

ETHICS STATEMENT

To the best of our knowledge, this work raises no specific ethical concerns. It does not involve human subjects or the collection of private or sensitive data, and all experiments use publicly available datasets and pretrained language models.

REPRODUCIBILITY STATEMENT

The proposed method and algorithms are described in the main paper, with additional proofs, implementation details, and hyperparameter settings provided in the appendix. An anonymized code repository containing the implementation and scripts required to reproduce our experiments is available at `https://anonymous.4open.science/r/TrellisMark-C36B/README.md`.

REFERENCES

Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014. doi: 10.1287/opre.2014.1289.

Santiago R. Balseiro, Haihao Lu, and Vahab Mirrokni. The best of many worlds: Dual mirror descent for online allocation problems. *Operations Research*, 71(1):101–119, 2023. doi: 10.1287/opre.2021.2242. URL `https://doi.org/10.1287/opre.2021.2242`.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015. doi: 10.1287/opre.2015.1408.

Zhongze Cai, Shang Liu, Hanzhao Wang, Huaiyang Zhong, and Xiaocheng Li. Towards better statistical understanding of watermarking llms. *Journal of the American Statistical Association*, 121(553):247–258, 2026.

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.

Xiaoyan Feng, He Zhang, Yanjun Zhang, Leo Yu Zhang, and Shirui Pan. Bimark: Unbiased multilayer watermarking for large language models. *arXiv preprint arXiv:2506.21602*, 2025.

G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 2005.

Atefeh Gilani, Sajani Vithana, Carol Xuan Long, Oliver Kosut, Lalitha Sankar, and Flavio P Calmon. Arcmark: Distortion-free multi-byte llm watermark via optimal transport. *arXiv preprint arXiv:2602.07235*, 2026.

Mingjia Huo, Sai Ashish Somayajula, Youwei Liang, Ruisi Zhang, Farinaz Koushanfar, and Pengtao Xie. Token-specific watermarking with enhanced detectability and semantic coherence for large language models. *arXiv preprint arXiv:2402.18059*, 2024.

Jiashuo Jiang, Xiaocheng Li, and Jiawei Zhang. Online stochastic optimization with wasserstein-based nonstationarity. *Management Science*, 71(11):9104–9122, 2025a. doi: 10.1287/mnsc.2020.03850.

Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. Stealthink: A multi-bit and stealthy watermark for large language models. *arXiv preprint arXiv:2506.05502*, 2025b.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pp. 17061–17084. PMLR, 2023.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36:27469–27500, 2023.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.

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.

Luofeng Liao, Yuan Gao, and Christian Kroer. Nonstationary dual averaging and online fair allocation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 37159–37172, 2022.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. A semantic invariant robust watermark for large language models. *arXiv preprint arXiv:2310.06356*, 2023.

Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5), 2007. doi: 10.1145/1284320.1284321.

Meta. meta-llama/llama-3.2-3b-instruct. `https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct`, 2024.

Prithivida. Parrot: Paraphraser based on t5, 2022. `https://huggingface.co/prithivida/parrot_paraphraser_on_T5`.

Wenjie Qu, Dong Yin, Zixin He, Wei Zou, Tianyang Tao, Jinyuan Jia, and Jiaheng Zhang. Provably robust multi-bit watermarking for ai-generated text via error correction code. *arXiv preprint arXiv:2401.16820*, 2024.

Qwen. Qwen/qwen3-4b-instruct-2507. `https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507`, 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.

Bram Rijsbosch, Gijs van Dijck, and Konrad Kollnig. Adoption of watermarking for generative ai systems in practice and implications under the new eu ai act. *arXiv preprint arXiv:2503.18156*, 2025.

Huanming Shen, Baizhou Huang, and Xiaojun Wan. Enhancing llm watermark resilience against both scrubbing and spoofing attacks. *Advances in Neural Information Processing Systems*, 38: 27957–27993, 2026.

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.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*, 2023.

A Embedding and Recovery Algorithms

For completeness, Algorithms 1 and 2 summarize the watermark embedding and message recovery procedures described in Sections 4–5.

Algorithm 1 Online Watermark Embedding of `TrellisMark`

Require: Language model P_θ, prompt $x_{\le 0}$, payload $\mathbf{s} \in \mathbb{F}_q^N$, secret key κ, average per-token KL budget $\bar{b}$, trellis memory r, number of heads H, maximum strength $\delta_{\max}$, and dual step size η_λ
Ensure: Watermarked sequence Y

Initialize $Y \leftarrow \emptyset$, $\lambda_1 \leftarrow 0$, and $S_0(\cdot) \equiv 0$
for $t = 1, \dots, T$ **do**
  Compute $p_t = P_\theta(\cdot \mid x_{\le 0}, y_{<t})$
  Using κ and $(x_{\le 0}, y_{<t})$, derive J_t and $\{\mathbf{a}_{t,h}, \chi_{t,h}\}_{h=1}^H$
  Compute $c_{t,h}(\mathbf{s}) = \langle \mathbf{a}_{t,h}, \mathbf{s}_{J_t - r:J_t} \rangle_{\mathbb{F}_q}$ for $h = 1, \dots, H$
  Compute the recovery weight ω_t using Eq. 9
  Set $\widetilde{\delta}_t = \min\{\delta_{\max}, \omega_t / \lambda_t\}$, then clip $\widetilde{\delta}_t$ to satisfy the remaining KL budget
  Form q_{t,δ_t} using Eq. 5, sample $y_t \sim q_{t,\delta_t}$, and append y_t to Y
  Compute $z_{t,h} = \chi_{t,h}(y_t)$ for $h = 1, \dots, H$, and update S_t using Eq. 8
  Update $\lambda_{t+1} \leftarrow \left[\lambda_t + \eta_\lambda \left(D_t(\delta_t) - \bar{b}\right)\right]_+$
end for
return Y

Algorithm 2 Message Recovery of `TrellisMark`

Require: Prompt $x_{\le 0}$, watermarked sequence $Y = (y_1, \dots, y_T)$, secret key κ, payload length N, alphabet size q, trellis memory r, and number of heads H
Ensure: Decoded payload $\widehat{\mathbf{m}}$

Initialize $\Phi_j(\cdot) \equiv 0$ for $j = 1, \dots, N$
for $t = 1, \dots, T$ **do**
  Using κ and $(x_{\le 0}, y_{<t})$, reconstruct the keyed payload position J_t and $\{\mathbf{a}_{t,h}, \chi_{t,h}\}_{h=1}^H$
  Compute $z_{t,h} = \chi_{t,h}(y_t)$ for $h = 1, \dots, H$, and for each $\mathbf{u} \in \mathbb{F}_q^{r+1}$ update

$$\Phi_{J_t}(\mathbf{u}) \leftarrow \Phi_{J_t}(\mathbf{u}) + \sum_{h=1}^{H} \mathbf{1}\left\{z_{t,h} = \langle \mathbf{a}_{t,h}, \mathbf{u} \rangle_{\mathbb{F}_q}\right\}$$

end for
Recover the maximum-score trellis path

$$\widehat{\mathbf{s}} \leftarrow \arg\max_{\mathbf{s}' \in \mathbb{F}_q^N} \sum_{j=1}^{N} \Phi_j\left(\mathbf{s}'_{j-r:j}\right)$$

using Viterbi decoding
Convert $\widehat{\mathbf{s}}$ to its binary representation $\widehat{\mathbf{m}}$ and return $\widehat{\mathbf{m}}$

B Trellis Representation and Viterbi Decoding

We provide additional details on the trellis representation used for message recovery. For clarity, we describe the core trellis over the payload positions; the boundary convention used in implementation is specified in Appendix G and follows the same recursion. During generation, the queried positions $J_1, \dots, J_T$ are determined by the evolving text context and need not follow the payload order. Since the recovery score is additive across generation steps, these observations can be regrouped by their queried payload position j without changing the objective. For each j and local assignment $\mathbf{u} \in \mathbb{F}_q^{r+1}$, recall the local score table

$$\Phi_j(\mathbf{u}) = \sum_{\tau: J_\tau = j} \sum_{h=1}^{H} \mathbf{1}\{z_{\tau,h} = \langle \mathbf{a}_{\tau,h}, \mathbf{u} \rangle_{\mathbb{F}_q}\}.$$

The score of a candidate payload $\mathbf{s}'$ therefore decomposes over overlapping local windows as $S_T(\mathbf{s}') = \sum_{j=1}^{N} \Phi_j(\mathbf{s}'_{j-r:j})$.

Trellis states and transitions. Adjacent local windows overlap in exactly r payload symbols. Accordingly, after processing payload position j, the part of a candidate prefix that is relevant to subsequent local windows is its r-symbol suffix $\sigma_j = (s'_{j-r+1}, \ldots, s'_j) \in \mathbb{F}_q^r$. We use this suffix as the trellis state at stage j, giving q^r possible states at each interior stage. A valid transition from stage $j-1$ to j has the form

$$(a, u_1, \ldots, u_{r-1}) \longrightarrow (u_1, \ldots, u_r),$$

for $a, u_1, \ldots, u_r \in \mathbb{F}_q$. Thus, choosing the new symbol u_r shifts the retained suffix by one position, and each state has q possible outgoing transitions. The transition uniquely determines the width-$(r+1)$ local assignment $\mathbf{u}_j = (a, u_1, \ldots, u_r) = \mathbf{s}'_{j-r:j}$, and conversely every local assignment determines one valid transition. The transition associated with $\mathbf{u}_j$ receives local score $\Phi_j(\mathbf{u}_j)$. Therefore, a complete candidate payload induces a path through the trellis, while every valid path determines a candidate payload under the boundary convention specified in Appendix G. The sum of the transition scores along this path is exactly $S_T(\mathbf{s}')$ in Eq. 7.

Viterbi dynamic programming. Although the trellis implicitly contains all q^N payload paths, they need not be enumerated. Let $V_j(\sigma)$ denote the maximum accumulated score among all partial paths ending in state σ at position j. For $\sigma = (u_1, \ldots, u_r)$, every predecessor has the form $(a, u_1, \ldots, u_{r-1})$ for some $a \in \mathbb{F}_q$, so

$$V_j(u_1, \ldots, u_r) = \max_{a \in \mathbb{F}_q} \{V_{j-1}(a, u_1, \ldots, u_{r-1}) + \Phi_j(a, u_1, \ldots, u_r)\}. \tag{13}$$

For each state, the maximizing predecessor is stored as a backpointer.

The recursion is exact because any two partial paths reaching the same state share the same last r symbols. They therefore admit the same future transitions and receive the same future branch-score increments. Their only difference is the score already accumulated, so a lower-scoring path reaching the same state can never overtake the higher-scoring one under any continuation. Retaining only the best path to each state therefore preserves the global optimum.

After the final trellis stage, the decoder selects the highest-scoring admissible terminal path and follows the stored backpointers to recover $\widehat{\mathbf{s}}$. Thus, Viterbi decoding computes $\arg\max_{\mathbf{s}' \in \mathbb{F}_q^N} S_T(\mathbf{s}')$ exactly, without enumerating all q^N payload candidates (Forney, 2005).

This trellis structure is useful beyond decoding. The same forward and backward dynamic programs allow the encoder to evaluate the strongest competing paths in Eq. 14, which is required to compute the recovery weight in Eq. 9. The trellis therefore provides a common computational representation for both exact message recovery and recovery-aware strength allocation. Boundary positions are handled according to the convention in Appendix G.

Forward–backward constrained path scores. Beyond decoding, the same trellis representation supports the constrained maximization used to compute the recovery weight in Section 5.1. Before generation step t, let $\Phi_j^{(t-1)}(\mathbf{u})$ denote the local score accumulated from tokens $y_{<t}$. Let $F_j(\sigma)$ be the maximum score of a partial path through position j ending in state σ, and let $B_j(\sigma)$ be the maximum score obtainable from positions $j+1, \ldots, N$ given state σ at position j. For a local assignment $\mathbf{u} = (a, u_1, \ldots, u_r)$ at position j, define its predecessor and successor states as

$$\sigma^-(\mathbf{u}) = (a, u_1, \ldots, u_{r-1}), \qquad \sigma^+(\mathbf{u}) = (u_1, \ldots, u_r).$$

The maximum score of a complete path constrained to use $\mathbf{u}$ at position j is then

$$C_{t,j}(\mathbf{u}) = F_{j-1}\big(\sigma^-(\mathbf{u})\big) + \Phi_j^{(t-1)}(\mathbf{u}) + B_j\big(\sigma^+(\mathbf{u})\big). \tag{14}$$

Setting $j = J_t$ gives the constrained path score required in Section 5.1. For any $\mathbf{u}$ with $\mathrm{Sep}_t(\mathbf{u}) > 0$, we necessarily have $\mathbf{u} \neq \mathbf{s}_{J_t-r:J_t}$, so every path constrained to use $\mathbf{u}$ is automatically distinct from the embedded payload. Thus, the strongest path consistent with each current local assignment can be obtained from one forward and one backward dynamic program, without enumerating complete payloads.

Online allocation complexity. At generation step t, computing the recovery weight ω_t requires the constrained path scores $C_t(u)$ for the local assignments $u \in \mathbb{F}_q^{r+1}$ at the queried position J_t. These scores are obtained from one forward–backward pass over the trellis, which requires $O(Nq^{r+1})$ time. Given the forward and backward scores, evaluating $C_t(u)$ for all q^{r+1} local assignments requires $O(q^{r+1})$ operations, while computing their current-step separations over H heads requires $O(Hq^{r+1})$ operations. After sampling y_t, updating the local score table at J_t has the same $O(Hq^{r+1})$ cost. Hence, the trellis-related online encoding cost is $O\big(T(N+H)q^{r+1}\big)$ over a length-T generation, or $O((N+H)q^{r+1})$ per generation step.

Decoding complexity. For each generated token, constructing the local score tables $\{\Phi_j\}_{j=1}^{N}$ requires evaluating the H head-level color matches for all q^{r+1} local assignments at its keyed position J_t. This costs $O(Hq^{r+1})$ operations per token and $O(THq^{r+1})$ over the full generated sequence. Once the local score tables have been constructed, each trellis stage contains q^r states with q possible incoming transitions per state, so the Viterbi recursion examines q^{r+1} transitions per stage and requires $O(Nq^{r+1})$ time. The overall decoding complexity is therefore $O\big((TH+N)q^{r+1}\big)$. For fixed q, r, and H, decoding is linear in both T and N, rather than exponential in the payload space size q^N.

C Finite-Length Recovery Analysis

We bound the probability that a competing payload attains a decoding score no smaller than the embedded payload. The analysis separates three mechanisms: whether generation visits enough local windows on which the two payloads differ, whether the keyed projections distinguish their local assignments, and whether token-sampling noise can overcome the resulting expected score advantage. Throughout this section, local payload windows are interpreted according to the zero-termination convention specified in Appendix G.

Random-key model. The secret key is sampled once before generation, and we model the separate keyed mappings used to construct J_t, $\mathbf{a}_{t,h}$, and $\chi_{t,h}$ as independent random functions. Repeated keyed inputs return the same outputs, whereas outputs at distinct previously unseen inputs are independent under the random-key model. We analyze these mappings by deferred decisions. Let $\mathcal{F}_{t-1}$ be the exposure sigma-field generated by the prefix $y_{<t}$ and all keyed outputs revealed through step $t-1$. If the current n-gram context has not appeared previously, then its keyed position has not yet been exposed and, conditional on $\mathcal{F}_{t-1}$, J_t is uniform over the $N+r$ queryable trellis stages. Since $0 \le r \le N$, we have $N + r \le 2N$. After J_t is revealed, the H projection vectors associated with the current keyed input are independent and uniform over the coefficient space, and independent of the keyed position map. All probabilities in this section are taken over the random-key draw and subsequent token-sampling randomness.

We take the q-ary projection arithmetic over the finite field $\mathbb{F}_q$. For any two distinct local assignments $\mathbf{u} \neq \widetilde{\mathbf{u}}$, their difference is nonzero, so $\langle \mathbf{a}_{t,h}, \mathbf{u} - \widetilde{\mathbf{u}} \rangle$ is uniform over $\mathbb{F}_q$. Hence the two assignments have the same projected value with probability $1/q$ and are separated with probability $1 - 1/q$.

For any $\gamma \in (0, 1]$, let $\varepsilon_{\mathrm{ctx}}$ denote the probability that fewer than γT distinct n-gram contexts occur during the length-T generation. This term directly captures finite-length context repetition.

For any competing payload $\widetilde{\mathbf{s}} \neq \mathbf{s}$, define its induced separation from the embedded payload by

$$d_\kappa(\mathbf{s}, \widetilde{\mathbf{s}}) = \sum_{t=1}^{T} \sum_{h=1}^{H} \mathbf{1}\{c_{t,h}(\mathbf{s}) \neq c_{t,h}(\widetilde{\mathbf{s}})\}\,. \tag{15}$$

After the current keyed quantities have been exposed and the watermark strength has been selected, let $\mathcal{G}_t$ denote the corresponding pre-sampling exposure sigma-field. Thus, $\mathcal{G}_t$ contains $\mathcal{F}_{t-1}$, the

current position J_t, the keyed quantities $\{\mathbf{a}_{t,h}, \chi_{t,h}\}_{h=1}^{H}$, the strength δ_t, and the resulting sampling distribution q_{t,δ_t}. The token is then sampled as $y_t \sim q_{t,\delta_t}$, and we set $\mathcal{F}_t = \sigma(\mathcal{G}_t, y_t)$. Hence $\mathcal{F}_{t-1} \subseteq \mathcal{G}_t \subseteq \mathcal{F}_t$.

For a competing payload $\widetilde{\mathbf{s}} \neq \mathbf{s}$, define the one-step target–competitor score difference

$$X_t(\widetilde{\mathbf{s}}) = \big(S_t(\mathbf{s}) - S_{t-1}(\mathbf{s})\big) - \big(S_t(\widetilde{\mathbf{s}}) - S_{t-1}(\widetilde{\mathbf{s}})\big). \tag{16}$$

Its conditional expected advantage is $g_t(\widetilde{\mathbf{s}}) = \mathbb{E}[X_t(\widetilde{\mathbf{s}}) \mid \mathcal{G}_t]$, and its cumulative value is $\Gamma_T(\widetilde{\mathbf{s}}) = \sum_{t=1}^{T} g_t(\widetilde{\mathbf{s}})$.

Assumption 1 (Cumulative target-score advantage)**.** *There exist constants $\eta \in (0, 1]$ and $\varepsilon_{\mathrm{mar}} \in [0, 1)$ such that, with probability at least $1 - \varepsilon_{\mathrm{mar}}$, simultaneously for every competing payload $\widetilde{\mathbf{s}} \neq \mathbf{s}$,*

$$\Gamma_T(\widetilde{\mathbf{s}}) \geq \eta\, d_\kappa(\mathbf{s}, \widetilde{\mathbf{s}}). \tag{17}$$

Assumption 1 is a cumulative condition on the recovery margin induced by the realized sequence of adaptively chosen watermark strengths. The conditional advantage $g_t(\widetilde{\mathbf{s}})$ may vary across generation steps with the context, the queried local payload window, and the selected strength δ_t. The assumption requires only that, after aggregation over the sequence, the expected target–competitor score margin grows proportionally with the number of keyed local mappings that assign different target colors to the two payloads. Individual separating mappings may contribute different amounts of margin, no margin, or even locally favor the competitor, as long as the cumulative advantage remains positive at the stated rate.

Theorem 2 (Finite-length recovery under local coupling)**.** *Suppose Assumption 1 holds and $0 \leq r \leq N$. Define*

$$c_{\mathrm{rec}} := \frac{1}{16} \min\left\{1,\ \frac{(1 - 1/q)H}{2},\ \eta^2(1 - 1/q)\right\}.$$

Then, for any competing payload $\widetilde{\mathbf{s}} \neq \mathbf{s}$ differing from $\mathbf{s}$ in d symbols,

$$\Pr[S_T(\widetilde{\mathbf{s}}) \geq S_T(\mathbf{s})] \leq \varepsilon_{\mathrm{ctx}} + \varepsilon_{\mathrm{mar}} + 3\exp\left(-c_{\mathrm{rec}}\frac{\gamma T}{N}(r + d)\right).$$

Consequently, maximum-score decoding satisfies

$$\Pr[\widehat{\mathbf{s}} \neq \mathbf{s}] \leq \varepsilon_{\mathrm{ctx}} + \varepsilon_{\mathrm{mar}} + 3\exp\left(-c_{\mathrm{rec}}\frac{\gamma T}{N}r\right)\left[\left(1 + (q-1)\exp\left(-c_{\mathrm{rec}}\frac{\gamma T}{N}\right)\right)^{N} - 1\right]. \tag{18}$$

Proof. Fix a competing payload $\widetilde{\mathbf{s}} \neq \mathbf{s}$ that differs from $\mathbf{s}$ in d symbols. We first quantify how often the local construction can distinguish the two payloads, and then show that token-sampling noise is unlikely to overturn the resulting expected score advantage.

Under zero termination, each differing payload symbol affects $r + 1$ consecutive local windows. Among all configurations of d differing symbols, the union of these windows is minimized when the differences are consecutive: the first difference contributes $r + 1$ windows, and each additional consecutive difference contributes one new window. Hence any competitor at Hamming distance d affects at least $r + d$ of the $N + r$ queryable trellis stages.

We next show that sufficiently many such windows are visited. Consider the first occurrences of distinct n-gram contexts and expose the corresponding keyed positions only when those contexts are first encountered. Whenever a new context appears, its keyed position has not previously been revealed. By the random-key model, conditional on all previously exposed key outputs, the corresponding J_t remains uniform over the $N + r$ queryable trellis stages. Therefore, the conditional probability that J_t selects a differing window is at least $\frac{r+d}{N+r} \geq \frac{r+d}{2N}$.

Whether the current context is new is determined before its previously unseen keyed position is exposed. Conditional on the past, each such fresh position therefore selects a differing window with conditional probability at least $(r + d)/(2N)$. A lower-tail Chernoff bound for these adaptively exposed Bernoulli trials implies that, jointly with the event that at least γT distinct contexts occur, the probability that fewer than $\gamma T(r + d)/(4N)$ of them select differing windows is at most

$$\exp\left(-\frac{\gamma T}{16N}(r + d)\right). \tag{19}$$

Thus, except for this event, at least $\gamma T(r+d)/(4N)$ distinct contexts select differing windows. The overlapping construction therefore converts a d-symbol payload difference into a proportional number of opportunities at which the local assignments can be separated.

We next expose the keyed projections associated with these contexts. Conditional on all quantities exposed previously, each fresh projection separates the two distinct local assignments with probability $1-1/q$. A second lower-tail Chernoff bound for the sequentially exposed projections therefore implies that, except with probability

$$\exp\left(-\frac{\gamma(1-1/q)HT}{32N}(r+d)\right), \tag{20}$$

the induced separation satisfies

$$d_\kappa(\mathbf{s},\widetilde{\mathbf{s}}) \geq \frac{\gamma(1-1/q)HT}{8N}(r+d). \tag{21}$$

It remains to control token-sampling noise around the conditional expected score advantage. For each t, let $L_t = \sum_{h=1}^{H} \mathbf{1}\{c_{t,h}(\mathbf{s}) \neq c_{t,h}(\widetilde{\mathbf{s}})\}$ denote the number of separating heads at step t. By definition, $\sum_{t=1}^{T} L_t = d_\kappa(\mathbf{s},\widetilde{\mathbf{s}})$. Conditional on $\mathcal{G}_t$, only these L_t heads can contribute to $X_t(\widetilde{\mathbf{s}})$. Since each such head contributes a difference between two indicators, $X_t(\widetilde{\mathbf{s}}) \in [-L_t, L_t]$. Let $M_t(\widetilde{\mathbf{s}}) = X_t(\widetilde{\mathbf{s}}) - g_t(\widetilde{\mathbf{s}})$. Then $\mathbb{E}[M_t(\widetilde{\mathbf{s}}) \mid \mathcal{G}_t] = 0$.

Hoeffding's lemma therefore gives, for every $\theta > 0$, $\mathbb{E}[\exp(-\theta M_t(\widetilde{\mathbf{s}})) \,|\, \mathcal{G}_t] \leq \exp\left(\frac{\theta^2 L_t^2}{2}\right)$. Since $L_t \leq H$, we have $L_t^2 \leq H L_t$. Moreover, L_t is $\mathcal{G}_t$-measurable, while $\mathcal{F}_{t-1} \subseteq \mathcal{G}_t \subseteq \mathcal{F}_t$. Therefore, iterating the conditional bound over $t = 1, \ldots, T$ and using $\sum_t L_t = d_\kappa(\mathbf{s},\widetilde{\mathbf{s}})$ yields

$$\mathbb{E}\left[\exp\left(-\theta \sum_{t=1}^{T} M_t(\widetilde{\mathbf{s}}) - \frac{\theta^2 H}{2} d_\kappa(\mathbf{s},\widetilde{\mathbf{s}})\right)\right] \leq 1. \tag{22}$$

The cumulative score difference decomposes as $S_T(\mathbf{s}) - S_T(\widetilde{\mathbf{s}}) = \Gamma_T(\widetilde{\mathbf{s}}) + \sum_{t=1}^{T} \left(X_t(\widetilde{\mathbf{s}}) - g_t(\widetilde{\mathbf{s}})\right)$. Hence, whenever $S_T(\widetilde{\mathbf{s}}) \geq S_T(\mathbf{s})$ and the margin condition in Eq. 17 holds, $\sum_{t=1}^{T} \left(X_t(\widetilde{\mathbf{s}}) - g_t(\widetilde{\mathbf{s}})\right) \leq -\Gamma_T(\widetilde{\mathbf{s}}) \leq -\eta\, d_\kappa(\mathbf{s},\widetilde{\mathbf{s}})$. Thus a competitor can overtake the embedded payload only if the centered token-sampling fluctuations offset a cumulative expected margin of at least $\eta d_\kappa(\mathbf{s},\widetilde{\mathbf{s}})$. Applying Eq. 22 with $\theta = \eta/H$ and using Eq. 21 gives

$$\begin{aligned}\Pr\left[S_T(\widetilde{\mathbf{s}}) \geq S_T(\mathbf{s}),\ d_\kappa(\mathbf{s},\widetilde{\mathbf{s}}) \geq \frac{\gamma(1-1/q)HT}{8N}(r+d), \Gamma_T(\widetilde{\mathbf{s}}) \geq \eta d_\kappa(\mathbf{s},\widetilde{\mathbf{s}})\right] \\ \leq \exp\left(-\frac{\gamma\eta^2(1-1/q)T}{16N}(r+d)\right).\end{aligned} \tag{23}$$

Combining the context-diversity and margin failure probabilities with Eqs. 19, 20, and 23 gives

$$\begin{aligned}\Pr[S_T(\widetilde{\mathbf{s}}) \geq S_T(\mathbf{s})] \leq \varepsilon_{\mathrm{ctx}} + \varepsilon_{\mathrm{mar}} + \exp\left(-\frac{\gamma T}{16N}(r+d)\right) \\ + \exp\left(-\frac{\gamma(1-1/q)HT}{32N}(r+d)\right) \\ + \exp\left(-\frac{\gamma\eta^2(1-1/q)T}{16N}(r+d)\right).\end{aligned} \tag{24}$$

By the definition of c_{rec}, each exponential term is at most $\exp\left(-c_{\mathrm{rec}}\frac{\gamma T}{N}(r+d)\right)$.

Therefore,

$$\Pr[S_T(\widetilde{\mathbf{s}}) \geq S_T(\mathbf{s})] \leq \varepsilon_{\mathrm{ctx}} + \varepsilon_{\mathrm{mar}} + 3\exp\left(-c_{\mathrm{rec}}\frac{\gamma T}{N}(r+d)\right).$$

Finally, the context-diversity and margin events are global, so their failure probabilities $\varepsilon_{\mathrm{ctx}}$ and $\varepsilon_{\mathrm{mar}}$ are incurred only once. There are exactly $\binom{N}{d}(q-1)^d$ competing payloads at Hamming distance d

from $\mathbf{s}$. A union bound therefore gives

$$\Pr[\widehat{\mathbf{s}} \neq \mathbf{s}] \leq \varepsilon_{\mathrm{ctx}} + \varepsilon_{\mathrm{mar}} + 3\sum_{d=1}^{N} \binom{N}{d} (q-1)^d \exp\left(-c_{\mathrm{rec}} \frac{\gamma T}{N}(r+d)\right)$$
$$= \varepsilon_{\mathrm{ctx}} + \varepsilon_{\mathrm{mar}} + 3\exp\left(-c_{\mathrm{rec}} \frac{\gamma T}{N} r\right) \left[\left(1 + (q-1)\exp\left(-c_{\mathrm{rec}} \frac{\gamma T}{N}\right)\right)^N - 1\right], \quad (25)$$

where the equality follows from the binomial theorem. This proves the stated exact-recovery bound. □

Interpretation and recovery scaling. The bound separates two model-level failure probabilities from three finite-length concentration effects. The terms $\varepsilon_{\mathrm{ctx}}$ and $\varepsilon_{\mathrm{mar}}$ capture, respectively, insufficient context diversity and failure of the cumulative target-score margin. Conditional on these events not occurring, recovery can fail because too few differing windows are visited, too few keyed projections separate the resulting local assignments, or token-sampling noise overturns the expected score advantage.

The benefit of local overlap enters directly through $r + d$. Under zero termination, a competitor differing in d symbols affects at least $r + d$ of the $N + r$ queryable stages. In particular, a one-symbol competitor affects $r+1$ stages, so its distinguishing opportunities scale as $T(r+1)/N$ up to constants. More generally, each additional consecutive difference contributes one additional affected stage, with all $N + r$ stages affected when the payloads differ in all N symbols.

The number of heads H controls a different bottleneck. The projection separation exponent grows linearly with H, whereas the context-coverage and token-sampling exponents do not. Since $\eta \leq 1$, for $H \geq 2$,

$$c_{\mathrm{rec}} = \frac{\eta^2(1 - 1/q)}{16}.$$

Thus moving from $H = 1$ to $H = 2$ can improve recovery by removing insufficient projection separation as the bottleneck, while further increasing H does not improve this worst-case rate because all heads are evaluated on the same sampled token. This predicts diminishing returns once projection separation is sufficiently reliable.

To see the resulting recovery scaling, write $a := c_{\mathrm{rec}}\gamma T/N$ and $b := q - 1$. Then Eq. 18 gives $\Pr[\widehat{\mathbf{s}} \neq \mathbf{s}] \leq \varepsilon_{\mathrm{ctx}} + \varepsilon_{\mathrm{mar}} + 3e^{-ar}\left[(1 + be^{-a})^N - 1\right]$. Using $(1+x)^N - 1 \leq Nx\, e^{Nx}$ for $x \geq 0$, the finite-length term is at most $3Nb\exp\{-a(r+1) + Nbe^{-a}\}$. For fixed q, choosing $a \gtrsim 1 + \log \frac{N}{r+1}$ ensures, up to constants depending only on q, that Nbe^{-a} is dominated by $a(r+1)$ and that $a(r+1) \gtrsim \log N$. Therefore, for $H \geq 2$, a sufficient recovery scaling is

$$T \gtrsim \frac{N}{\gamma\eta^2(1 - 1/q)} \left(1 + \log \frac{N}{r+1}\right). \quad (26)$$

Hence, for fixed q, γ, and η, a recovery length of order $N \log N$ is sufficient when r is fixed, while $r = \Theta(N)$ reduces the sufficient length to order N.

D Distributional Distortion and Generation Quality

Our optimization uses KL divergence to quantify the distributional distortion introduced by watermarking, while our experiments additionally report perplexity (PPL) evaluated by an external language model. Following Cai et al. (2026), we distinguish distributional distortion of the generation process from the quality of realized generated text. The two metrics therefore serve complementary purposes in our evaluation. Figure 5 shows the empirical relationship betweem the two metrics.

Why KL divergence for optimization. At generation step t, the original and watermarked generators define conditional next-token distributions p_t and q_t, respectively. The KL divergence $D_{\mathrm{KL}}(q_t \| p_t)$ directly measures the extent to which watermarking perturbs the original sampling distribution. More importantly, KL divergence admits a natural chain-rule decomposition for autoregressive generation (Cai et al., 2026). Let P and Q denote the sequence distributions induced by the

original and watermarked generators. Then

$$D_{\mathrm{KL}}(Q\|P) = \sum_{t=1}^{T} \mathbb{E}_{Y_{<t}\sim Q}\left[D_{\mathrm{KL}}\big(q_t(\cdot \mid Y_{<t})\|p_t(\cdot \mid Y_{<t})\big)\right]. \tag{27}$$

Thus, the sequence-level divergence can be expressed as the expected sum of conditional token-level divergences. Correspondingly, the accumulated KL cost along a generated trajectory, $\sum_{t=1}^{T} D_{\mathrm{KL}}(q_t\|p_t)$, provides a sample-path measure of distortion whose expectation recovers the sequence-level KL divergence. This additive structure makes KL particularly suitable as the resource cost in our online token-level allocation problem.

Why perplexity for empirical evaluation. Perplexity instead evaluates the likelihood of a realized sequence under an evaluation language model P_{eval}:

$$\mathrm{PPL}(Y; P_{\mathrm{eval}}) = \exp\left(-\frac{1}{T}\sum_{t=1}^{T} \log P_{\mathrm{eval}}(y_t \mid y_{<t})\right). \tag{28}$$

It therefore evaluates a property of the realized text under the evaluator, rather than directly measuring the divergence between the watermarked and original generation distributions. As discussed by Cai et al. (2026), perplexity-based differences can be misleading when used as a measure of watermark-induced distributional distortion: a modified generator may achieve lower perplexity while still differing substantially from the original generator. We therefore do not use PPL as the resource cost in our optimization. Instead, we report PPL using an external evaluator as a empirical measure of the quality of the generated text as prior works.

Relationship between KL divergence and perplexity. The distinction can also be seen algebraically. Suppose, for illustration, that the original model P itself is used as the perplexity evaluator and that $Y \sim Q$. Then

$$\mathbb{E}_{Y\sim Q}[\log \mathrm{PPL}(Y; P)] = \frac{1}{T}H(Q, P), \tag{29}$$

where $H(Q, P)$ is the cross entropy between the watermarked sequence distribution Q and the original sequence distribution P. Since

$$D_{\mathrm{KL}}(Q\|P) = H(Q, P) - H(Q), \tag{30}$$

we have

$$\frac{1}{T}D_{\mathrm{KL}}(Q\|P) = \mathbb{E}_{Y\sim Q}[\log \mathrm{PPL}(Y; P)] - \frac{1}{T}H(Q). \tag{31}$$

Hence, perplexity alone does not determine the divergence from the original generator because it does not account for the entropy of the watermarked distribution Q.

E Evidence–Distortion Relation

We characterize how watermark strength trades off recovery evidence against distributional distortion. For the exponentially tilted distribution in Eq. 5, standard exponential-family identities give

$$G_t'(\delta) = \mathrm{Var}_{V\sim q_{t,\delta}}[\rho(K_t^{\mathbf{s}}(V))], \qquad D_t'(\delta) = \delta\ \mathrm{Var}_{V\sim q_{t,\delta}}[\rho(K_t^{\mathbf{s}}(V))].$$

Thus both $G_t(\delta)$ and $D_t(\delta)$ are nondecreasing in δ. Whenever $G_t'(\delta) > 0$ and $\delta > 0$, the marginal distortion per unit evidence is $dD_t/dG_t = \delta$, equivalently $dG_t/dD_t = 1/\delta$. Hence additional evidence becomes progressively more distortion-intensive as watermark strength increases. Near $\delta = 0$, evidence increases to first order whenever the match reward has positive variance, whereas $D_t'(0) = 0$, so KL distortion grows only at higher order.

This relation directly motivates the allocation in Section 5.2. Since evidence at step t is weighted by its recovery importance ω_t, its marginal value per unit KL is $d(\omega_t G_t)/dD_t = \omega_t/\delta$. The allocator therefore favors recovery-important opportunities while avoiding excessive strength at any single token. At an interior optimum with positive reward variance and KL shadow price $\lambda > 0$, the first-order condition $\omega_t G_t'(\delta_t) = \lambda D_t'(\delta_t)$ reduces to $\delta_t = \omega_t/\lambda$. Thus ω_t determines where stronger embedding is valuable, while λ calibrates the overall strength under the KL budget. This structure underlies both the hindsight benchmark and the online allocation in Appendix F.

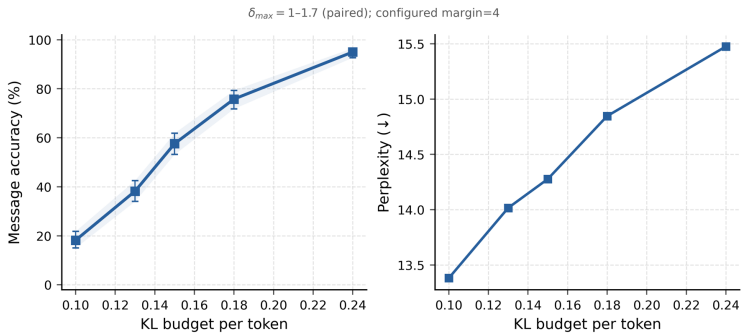


Figure 5: Empirical relationship between the KL-divergence budget and generation quality. Increasing the per-token KL budget allows greater deviation from the original next-token distributions, leading to higher exact message recovery (left) but also higher evaluator-based perplexity (right). Although KL divergence and perplexity measure different quantities, the observed trend shows that larger distributional distortion is associated with reduced empirical text quality in our setting.

F Regret Analysis

We provide a regret guarantee for the recovery-aware online allocation rule. Assume that the recovery weight is uniformly bounded, $0 \le \omega_t \le \omega_{\max}$. This can be enforced directly by capping or normalizing the recovery weight and prevents any single token from dominating the strength allocation. Recall that $\rho(0) = 0$ and $\rho(b) = 1 + \beta(b-1)$ for $b \ge 1$, where $\beta \in [0,1]$. Since $K_t^{\mathbf{s}}(v) \in \{0, \dots, H\}$, we have $0 \le \rho(K_t^{\mathbf{s}}(v)) \le 1 + \beta(H-1) \le H$. As shown in Appendix E, the exponential-tilting structure gives $G_t'(\delta) = \mathrm{Var}_{q_{t,\delta}}[\rho(K_t^{\mathbf{s}}(v))]$ and $D_t'(\delta) = \delta G_t'(\delta)$. Hence both G_t and D_t are nondecreasing on $[0, \delta_{\max}]$, and $0 \le G_t(\delta) \le H$.

The KL cost is also uniformly bounded. Using $D_t(\delta) = \delta \mathbb{E}_{q_{t,\delta}}[\rho(K_t^{\mathbf{s}}(v))] - \log Z_t(\delta)$ and Jensen's inequality, $\log Z_t(\delta) \ge \delta \mathbb{E}_{p_t}[\rho(K_t^{\mathbf{s}}(v))]$. Therefore $0 \le D_t(\delta) \le \delta G_t(\delta) \le \delta_{\max} H$. It follows that the per-step weighted evidence gain and KL expenditure are bounded by $\omega_{\max} H$ and $\delta_{\max} H$, respectively.

Unclipped proposal and hard-budget clipping. Let $\lambda_t \ge 0$ be the distortion price updated according to Eq. 12. Before enforcing the remaining KL budget, consider the virtual primal–dual proposal

$$\widetilde{\delta}_t \in \arg \max_{0 \le \delta \le \delta_{\max}} \{\omega_t G_t(\delta) - \lambda_t D_t(\delta)\}. \tag{32}$$

Its objective has derivative $(\omega_t - \lambda_t \delta) G_t'(\delta)$. Whenever the local match reward is nondegenerate, $G_t'(\delta) > 0$, so the objective increases up to ω_t / λ_t and decreases afterward. Thus, for $\lambda_t > 0$, $\widetilde{\delta}_t = \min\left\{\delta_{\max}, \frac{\omega_t}{\lambda_t}\right\}$. If $\lambda_t = 0 < \omega_t$, we take $\widetilde{\delta}_t = \delta_{\max}$. Let $B_t^{\mathrm{rem}} = B - \sum_{\tau<t} D_\tau(\delta_\tau)$ denote the budget available before step t, and define the largest affordable strength by $\delta_t^B = \max\{\delta \in [0, \delta_{\max}] : D_t(\delta) \le B_t^{\mathrm{rem}}\}$. Since D_t is continuous and nondecreasing, this quantity is well defined. The implemented strength is

$$\delta_t = \min\left\{\widetilde{\delta}_t, \delta_t^B\right\}. \tag{33}$$

Thus the primal–dual rule determines the desired strength, and clipping is applied only when needed to respect the remaining budget. By construction, $\sum_{t=1}^{T} D_t(\delta_t) \le B$ on every realized trajectory.

Hindsight-optimal allocation. For the realized sequence of recovery–distortion opportunities, define the hindsight benchmark

$$\begin{aligned} \mathrm{OPT}_T^{\mathrm{off}}(B) = \max_{\delta_1,\dots,\delta_T} \quad & \frac{1}{T}\sum_{t=1}^{T} \omega_t G_t(\delta_t) \\ \text{s.t.} \quad & \frac{1}{T}\sum_{t=1}^{T} D_t(\delta_t) \le \bar{b}, \\ & 0 \le \delta_t \le \delta_{\max}, \qquad t = 1,\dots,T. \end{aligned} \tag{34}$$

The benchmark knows the entire realized sequence of opportunities and distributes the same KL budget optimally, whereas the online encoder does not know future opportunities. The regret therefore measures the loss from allocating distortion online.

The evidence–distortion identity also gives the benchmark a simple interpretation. At an interior optimum with positive common KL shadow price $\lambda^\star$, the first-order condition gives $\delta_t^\star = \omega_t/\lambda^\star$. Thus both the online and hindsight allocators favor steps with larger recovery value; the difference is that the hindsight allocator knows all future opportunities.

Temporal dispersion of watermarking opportunities. Let $(\delta_1^\star,\dots,\delta_T^\star)$ be a hindsight-optimal allocation in Eq. 34, and define its cumulative KL expenditure by

$$C_k^\star = \sum_{t=1}^{k} D_t(\delta_t^\star), \qquad A_T = \max_{1\le k\le T} \left| C_k^\star - \frac{k}{T} C_T^\star \right|. \tag{35}$$

Assumption (No persistent temporal concentration). For some $\alpha \in [0,1)$, the hindsight-optimal KL expenditure satisfies $\mathbb{E}[A_T] = O(T^\alpha)$.

The condition permits substantial token-level variation and requires only that valuable watermarking opportunities are not persistently concentrated in one portion of the response. It does not require the optimal expenditure to be uniform over time; it only rules out linear-scale deviations from its average expenditure path. Temporal regularity plays an analogous role in online resource allocation under representative arrivals or controlled nonstationarity (Besbes et al., 2015; Balseiro et al., 2023; Jiang et al., 2025a; Liao et al., 2022).

We first isolate the effect of enforcing the hard budget.

Lemma 1 (Effect of hard-budget clipping)**.** *Suppose $\lambda_1 = 0$, and let δ_t be the clipped decision in Eq. 33. Then*

$$0 \le \lambda_t \le \frac{\omega_{\max} H}{\bar{b}} + \eta_\lambda \delta_{\max} H, \qquad |\lambda_{t+1} - \lambda_t| \le \eta_\lambda \left(\delta_{\max} H + \bar{b}\right), \tag{36}$$

where $\bar{b} = B/T$. Moreover, the cumulative Lagrangian loss caused by hard-budget clipping is at most

$$\frac{(\omega_{\max} H)^2}{\eta_\lambda \bar{b}^2} + \frac{2\omega_{\max}\delta_{\max} H^2}{\bar{b}}. \tag{37}$$

Proof of Lemma 1. We first bound the dual price. By definition, $\widetilde{\delta}_t$ maximizes the unrestricted Lagrangian $\omega_t G_t(\delta) - \lambda_t D_t(\delta)$ over $[0,\delta_{\max}]$. Since $\delta = 0$ is always available and has zero reward and zero KL cost, the value at $\widetilde{\delta}_t$ is nonnegative. Thus, whenever $\lambda_t > 0$, $D_t(\widetilde{\delta}_t) \le \omega_t G_t(\widetilde{\delta}_t)/\lambda_t \le \omega_{\max} H/\lambda_t$. Because $\delta_t \le \widetilde{\delta}_t$ and D_t is nondecreasing, the same bound applies to $D_t(\delta_t)$.

It follows that once $\lambda_t \ge \omega_{\max} H/\bar{b}$, the actual expenditure satisfies $D_t(\delta_t) \le \bar{b}$, so the dual price cannot increase further. A single update can increase the price by at most $\eta_\lambda \delta_{\max} H$, which gives the first bound in Eq. 36. The second follows directly from $0 \le D_t(\delta_t) \le \delta_{\max} H$ and the projected update in Eq. 12.

We next bound how many decisions can be affected by the hard budget. Let τ be the first step at which the remaining budget is smaller than $\delta_{\max} H$, and set $\tau = T+1$ if this never occurs. For every $t < \tau$, the remaining budget is at least the largest possible one-step KL expenditure. Hence every $\delta \in [0,\delta_{\max}]$ is feasible and $\delta_t = \widetilde{\delta}_t$.

Suppose $1 < \tau \leq T$. Iterating the projected dual update and using $[x]_+ \geq x$ gives $\lambda_\tau \geq \eta_\lambda[\sum_{t<\tau} D_t(\delta_t) - (\tau-1)\bar{b}]$. By the definition of τ, $\sum_{t<\tau} D_t(\delta_t) > B - \delta_{\max} H$. Since $B = T\bar{b}$, we obtain

$$\lambda_\tau > \eta_\lambda \left[(T - \tau + 1)\bar{b} - \delta_{\max} H\right].$$

Combining this inequality with the dual-price upper bound in Eq. 36 yields

$$T - \tau + 1 < \frac{\omega_{\max} H}{\eta_\lambda \bar{b}^2} + \frac{2\delta_{\max} H}{\bar{b}}. \tag{38}$$

The same bound is immediate when $\tau = 1$, while no clipping can occur when $\tau = T + 1$. Thus the hard budget can affect only the final number of steps bounded in Eq. 38.

Finally, consider any potentially affected step. The unrestricted Lagrangian value is at most $\omega_t G_t(\widetilde{\delta}_t) \leq \omega_{\max} H$. Moreover, because the Lagrangian increases up to its unrestricted maximizer $\widetilde{\delta}_t$, the clipped decision $\delta_t = \min\{\widetilde{\delta}_t, \delta_t^B\}$ maximizes the same Lagrangian over the budget-feasible interval. Its value is therefore nonnegative because $\delta = 0$ remains feasible. The Lagrangian loss due to clipping is consequently at most $\omega_{\max} H$ per affected step. Multiplying by the bound in Eq. 38 proves Eq. 37. ■

Theorem 3 (Per-token regret relative to hindsight-optimal allocation)**.** *Under the boundedness conditions above, for every realized sequence of allocation opportunities and every $\eta_\lambda > 0$, the clipped policy satisfies*

$$\begin{aligned} &\mathrm{OPT}_T^{\mathrm{off}}(B) - \frac{1}{T}\sum_{t=1}^{T} \omega_t G_t(\delta_t) \\ &\qquad \leq \eta_\lambda(\delta_{\max} H + \bar{b}) A_T + \frac{\eta_\lambda(\delta_{\max} H + \bar{b})^2}{2} + \frac{(\omega_{\max} H)^2}{\eta_\lambda T \bar{b}^2} + \frac{2\omega_{\max}\delta_{\max} H^2}{T\bar{b}}. \end{aligned} \tag{39}$$

Consequently, if $\mathbb{E}[A_T] = O(T^\alpha)$ for some $\alpha \in [0, 1)$ and $\eta_\lambda = \eta_0 T^{-(1+\alpha)/2}$ for a constant $\eta_0 > 0$, then

$$\mathbb{E}\left[\mathrm{OPT}_T^{\mathrm{off}}(B) - \frac{1}{T}\sum_{t=1}^{T} \omega_t G_t(\delta_t)\right] = O\Big(T^{-(1-\alpha)/2}\Big) \longrightarrow 0. \tag{40}$$

Thus the online allocator asymptotically matches the per-token reward of the clairvoyant hindsight allocation while satisfying the cumulative KL budget on every trajectory.

Proof of Theorem 3. Let $(\delta_1^\star, \ldots, \delta_T^\star)$ be a hindsight-optimal allocation in Eq. 34. The argument has three steps. We first compare the hindsight allocation with the virtual primal–dual proposals, then use Lemma 1 to pass from the virtual proposals to the implemented decisions, and finally control the remaining dual terms.

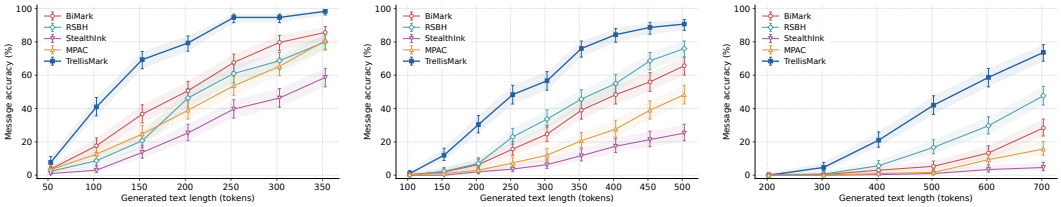
By optimality of $\widetilde{\delta}_t$ in Eq. 32, $\omega_t G_t(\delta_t^\star) - \lambda_t D_t(\delta_t^\star) \leq \omega_t G_t(\widetilde{\delta}_t) - \lambda_t D_t(\widetilde{\delta}_t)$ for every t. Adding and subtracting the Lagrangian value of the actual decision δ_t, summing over time, and applying Lemma 1 therefore give

$$\mathrm{OPT}_T^{\mathrm{off}}(B) - \frac{1}{T}\sum_{t=1}^{T} \omega_t G_t(\delta_t) \leq \frac{1}{T}\sum_{t=1}^{T} \lambda_t \left[D_t(\delta_t^\star) - D_t(\delta_t)\right] + \frac{(\omega_{\max} H)^2}{\eta_\lambda T \bar{b}^2} + \frac{2\omega_{\max}\delta_{\max} H^2}{T\bar{b}}. \tag{41}$$

It remains to control the first term. By feasibility of the hindsight allocation, $C_T^\star \leq B = T\bar{b}$. Since $\lambda_t \geq 0$, adding and subtracting $C_T^\star/T$ and using $C_T^\star/T \leq \bar{b}$ show that this term is at most

$$\frac{1}{T}\sum_{t=1}^{T} \lambda_t \left[D_t(\delta_t^\star) - \frac{C_T^\star}{T}\right] - \frac{1}{T}\sum_{t=1}^{T} \lambda_t \left[D_t(\delta_t) - \bar{b}\right].$$

The first sum measures the interaction between changes in the dual price and temporal variation in the hindsight-optimal expenditure. Since $D_t(\delta_t^\star) - C_T^\star/T = (C_t^\star - tC_T^\star/T) - (C_{t-1}^\star - (t-1)C_T^\star/T)$, summation by parts transforms it into $-\frac{1}{T}\sum_{t=1}^{T-1}(\lambda_{t+1} - \lambda_t)\left(C_t^\star - \frac{t}{T}C_T^\star\right)$. By the definition of A_T and the dual-increment bound in Eq. 36, its absolute value is at most $\eta_\lambda(\delta_{\max} H + \bar{b})A_T$.



(a) 16-bit payload (b) 32-bit payload (c) 64-bit payload

Figure 6: Exact message recovery versus generation length on LFQA with Llama-3.2-3B-Instruct for 16-, 32-, and 64-bit payloads. Each point is evaluated over 300 samples; error bars denote 95% Wilson confidence intervals. Higher is better.

The second sum is controlled directly by the projected dual update. From Eq. 12,

$$\lambda_{t+1}^2 \le \lambda_t^2 + 2\eta_\lambda \lambda_t \left(D_t(\delta_t) - \bar{b}\right) + \eta_\lambda^2 (\delta_{\max} H + \bar{b})^2.$$

Summing over t, using $\lambda_1 = 0$ and $\lambda_{T+1}^2 \ge 0$, gives $-\frac{1}{T}\sum_{t=1}^{T} \lambda_t \left(D_t(\delta_t) - \bar{b}\right) \le \frac{\eta_\lambda}{2}(\delta_{\max} H + \bar{b})^2$. Substituting these two bounds into Eq. 41 proves Eq. 39.

Finally, suppose $\mathbb{E}[A_T] = O(T^\alpha)$ for some $\alpha < 1$. The four terms on the right-hand side of Eq. 39 are of orders $\eta_\lambda T^\alpha$, η_λ, $1/(\eta_\lambda T)$, and $1/T$, respectively. Choosing $\eta_\lambda = \eta_0 T^{-(1+\alpha)/2}$ balances the first and third terms at order $T^{-(1-\alpha)/2}$, while the remaining two terms decay faster. This proves Eq. 40. ■

G Implementation Details

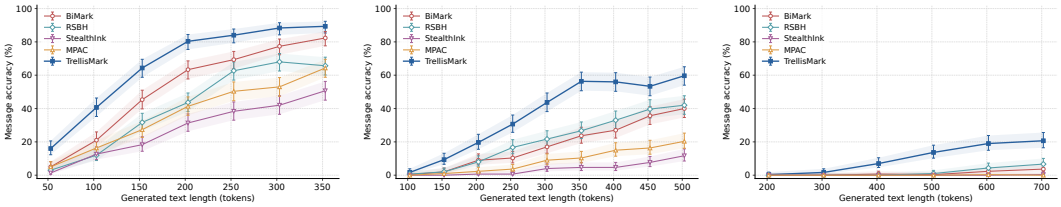
Implementation details. All methods are implemented in a unified generation and evaluation pipeline and are evaluated using the same prompts, payload lengths, decoding lengths, and sampling configuration. For our method, we use $q = 4$ token colors, $H = 2$ heads, and a trellis window width of $w = 4$, corresponding to memory $r = 3$. The keyed position map uses a one-token context ($n = 1$), and the multi-match scaling factor is set to $\beta = 0.5$. We use a softplus smoothing parameter of $\tau_d = 1.0$ and cap the per-token watermark strength at $\delta_{\max} = 1.4$. The target margin is set to 12.0, and the dual variable is updated with a step size of 0.05.

We use zero termination in our implementation. The trellis starts from the all-zero memory state, and r fixed zero symbols are appended after the N payload symbols. Consequently, there are $N + r \le 2N$ queryable trellis stages. This convention ensures that every payload symbol participates in $r + 1$ local windows, including symbols near the payload boundaries. The terminating symbols are fixed and are not part of the payload.

Baseline configurations. We follow the configurations recommended by the original papers or their official implementations, while using the same payload and generation settings as our method whenever supported. For MPAC, we use a color-list ratio of $\gamma = 0.25$, four color lists, and a context width of one. For RSBH, we use the balanced-hash construction with a Reed–Solomon configuration of $(n, k, m) = (6, 4, 8)$, $\gamma = 0.5$, and a context width of one. StealthInk uses one-bit message chunks and a context width of three. BiMark uses ten watermark layers and a context window of two. For baselines with an explicit watermark-strength parameter, we vary that parameter to obtain the recovery–quality trade-off rather than selecting a single operating point.

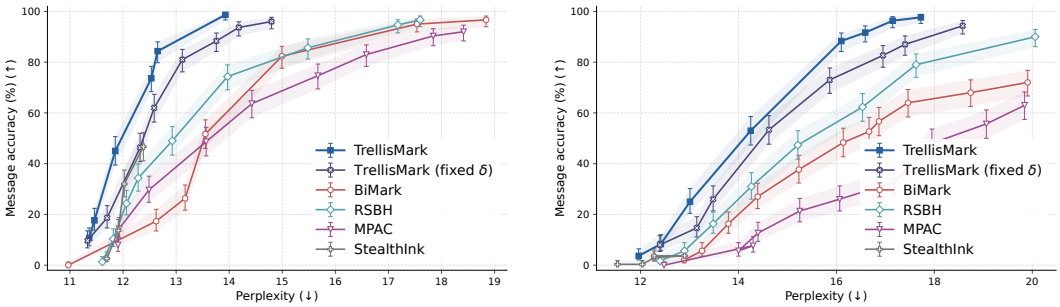
H Additional Dataset and Model Results

Message Recovery Across Datasets and Models. We further evaluate message recovery on LFQA (Krishna et al., 2023) with Llama-3.2-3B-Instruct (Meta, 2024), and on C4 (Raffel et al., 2020) with Qwen-3-4B-Instruct (Qwen, 2025). The former changes the dataset while keeping the language model fixed, whereas the latter changes the language model while retaining the C4 evaluation setting used in the main experiments.

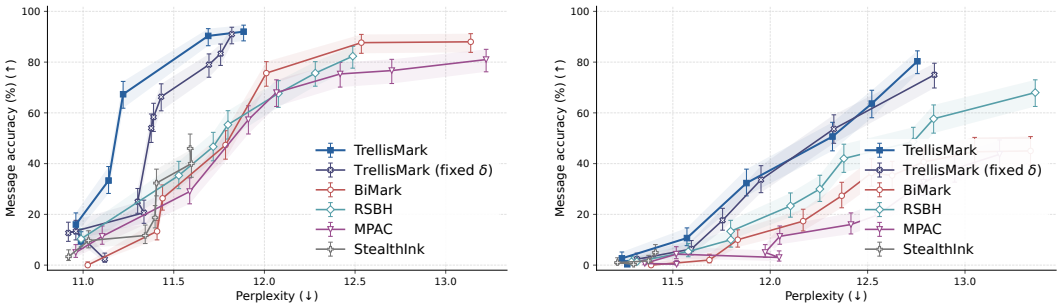


(a) 16-bit payload (b) 32-bit payload (c) 64-bit payload

Figure 7: Exact message recovery versus generation length on C4 with Qwen-3-4B-Instruct for 16-, 32-, and 64-bit payloads. Each point is evaluated over 300 samples; error bars denote 95% Wilson confidence intervals. Higher is better.



(a) LFQA, 16-bit payload. (b) LFQA, 32-bit payload.



(c) C4 with Qwen, 16-bit payload. (d) C4 with Qwen, 32-bit payload.

Figure 8: Recovery–quality trade-offs across additional datasets and language models. The top row evaluates Llama-3.2-3B-Instruct on LFQA, and the bottom row evaluates Qwen-3-4B-Instruct on C4, for 16- and 32-bit payloads. Each point corresponds to a different watermark-strength setting. Upper left is better; error bars denote 95% Wilson confidence intervals.

Figures 6 and 7 exhibit the same overall trend as the main experiments: exact recovery improves with generation length, while larger payloads require longer generations. TrellisMark achieves the highest or tied-highest recovery across the reported lengths in both settings. On LFQA, its recovery at the longest evaluated lengths reaches 98.3%, 90.7%, and 73.7% for 16-, 32-, and 64-bit payloads, compared with 85.7%, 76.0%, and 47.7% for the strongest baseline. On C4 with Qwen, the corresponding results are 89.3%, 59.7%, and 20.7%, compared with 82.3%, 42.0%, and 6.7%. While absolute recovery is lower with Qwen, particularly for the 64-bit payload, TrellisMark maintains a clear advantage across both the dataset and model changes.

Recovery–Quality Trade-off Across Datasets and Models. Figures 8 show that the recovery–quality advantage observed in the main experiments extends to both the additional dataset and language model. Across all four settings, `TrellisMark` traces the upper-left frontier over most of the operating range, achieving higher exact recovery at comparable perplexity than the baselines. The adaptive variant also consistently matches or improves upon fixed-strength TrellisMark, with the

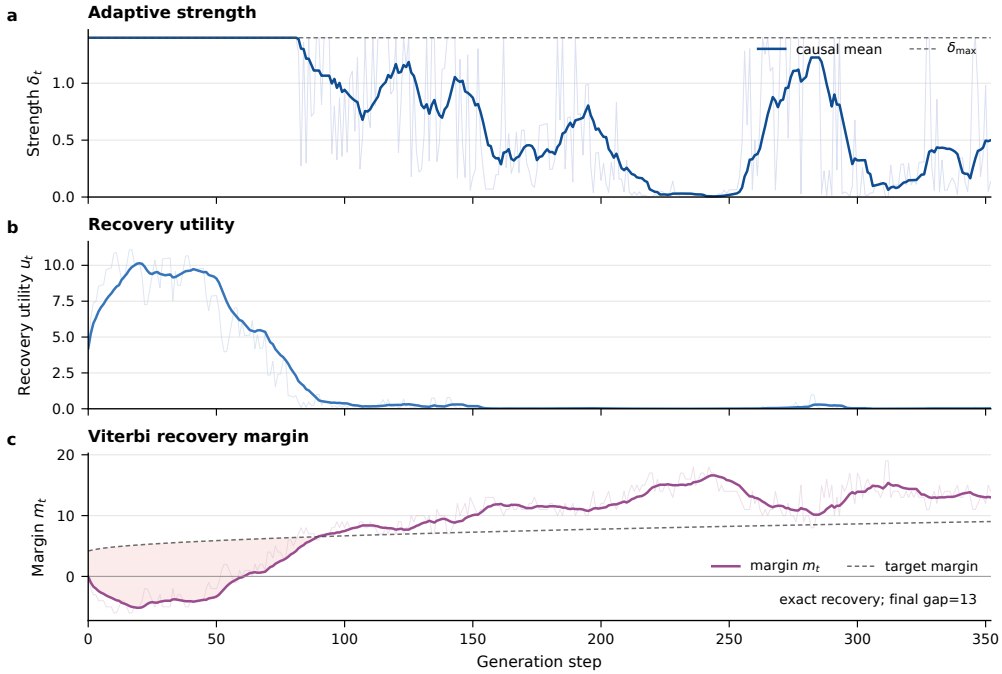


Figure 9: Dynamics of online strength allocation for a representative generation. (a) Per-token watermark strength δ_t, together with its causal moving average and the maximum allowable strength $\delta_{\max}$. (b) Recovery weight (utility) ω_t used by the online allocation policy. (c) Viterbi recovery margin m_t between the embedded message and its strongest competitor, together with the target margin. The allocated strength is initially saturated when recovery demand is high, then decreases as the recovery margin grows, while increasing again when additional recovery evidence is needed.

benefit becoming more pronounced at higher recovery levels and for the more difficult 32-bit payloads. Although the absolute recovery–quality curves shift across datasets and models, the overall ordering remains consistent, indicating that the improved trade-off is not specific to the main C4/Llama setting.

I Dynamics of Online Strength Allocation

Figure 9 illustrates the allocation dynamics for a representative generation. At the beginning of generation, the recovery demand is high and the optimizer selects the maximum allowable strength. As the Viterbi margin grows toward and eventually exceeds its target, the required strength generally decreases. Importantly, the trajectory is not monotonic: later increases in δ_t occur when additional recovery evidence is needed, indicating that the policy responds to the current recovery state rather than following a predetermined position-dependent schedule.

Figure 10 shows that this behavior persists across 500 generations. Both the mean recovery weight (utility) and the mean allocated strength decrease as generation progresses. More directly, Figure 10(c) shows that the allocated strength increases monotonically with recovery utility and reaches $\delta_{\max}$ at sufficiently high utility. This relationship is consistent with the online allocation rule in Eq. equation 11: stronger watermarking is used when additional recovery evidence is more valuable, while weaker strength is sufficient once the message is better separated from competing paths.

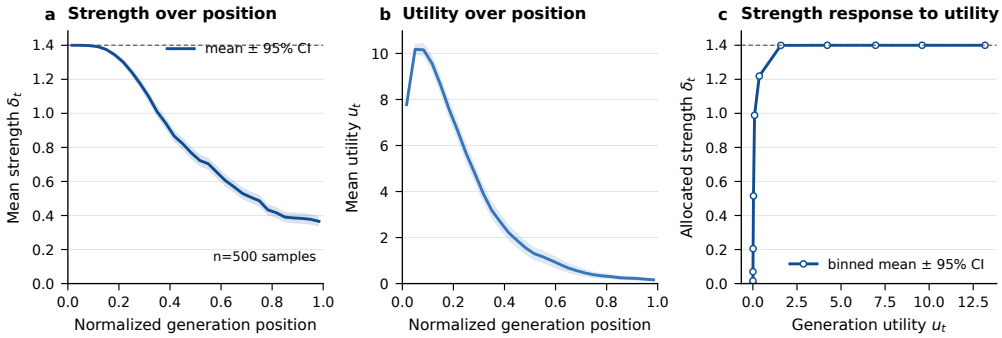


Figure 10: Aggregate behavior of the online strength allocation policy over 500 generations. (a) Mean allocated watermark strength δ_t as a function of normalized generation position. (b) Mean recovery weight (utility) ω_t over the same normalized position. (c) Allocated strength as a function of recovery utility, obtained by grouping generation steps into utility bins. Shaded regions denote 95% confidence intervals, and the dashed line marks the per-step strength limit $\delta_{\max}$. Higher recovery utility leads to stronger allocation, which saturates at $\delta_{\max}$ when the recovery demand is sufficiently large.