PRESENCE BEFORE PAYLOAD: FALSE-POSITIVE-CONTROLLED MULTI-BIT LLM WATERMARKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-bit LLM watermarking involves two conceptually distinct tasks: detecting whether a watermark is present and, only if so, recovering the payload it encodes. In several existing methods, however, the two decisions are conflated: watermark presence is inferred from the same search-and-decoding procedure used to recover the payload. This can cause unwatermarked text to be accepted whenever payload search or decoding produces a plausible candidate, making nominal false-positive rate (FPR) guarantees unreliable, especially when the assumed null distribution does not match real unwatermarked text. We introduce PIPER, a presence-before-payload watermark that separates the two decisions by design. PIPER partitions the vocabulary into two keyed levels so that the same token observations support both decisions: a coarse level provides payload-independent evidence for presence testing, while a finer level carries payload information for recovery. Presence is decided by an exact binomial test over deduplicated effective contexts, and the payload is decoded only if this test accepts. Under the standard random-key model with an idealized PRF, we prove finite-sample FPR control for a presence test whose statistic and threshold are independent of payload size and decoder choice. We evaluate PIPER across generation lengths from $T = 100$ to $T = 1000$. At the representative $T = 300$ setting and a nominal 1% FPR, PIPER requires no held-out threshold fitting under the stated null model and achieves empirical FPRs of 0.86% and 0.92% on model-generated and natural-text nulls, respectively, with 98.7%–99.0% TPR across 8- and 16-bit payloads. Code is available at this URL.

1 INTRODUCTION

The remarkable capabilities of large language models (LLMs) have transformed content generation and downstream applications (OpenAI et al., 2023; Yang et al., 2024), while raising concerns about content provenance and deceptive misuse (Wu et al., 2025; Liu et al., 2024). LLM watermarking has emerged as a promising approach to establishing provenance by embedding detectable signals into text during generation (Kirchenbauer et al., 2023; Hu et al., 2024; Kuditipudi et al., 2024).

Early work on LLM watermarking focused on *zero-bit watermarks*, which determine whether a text carries a designated watermark (Kirchenbauer et al., 2023; Aaronson, 2023). Subsequent work improved their statistical reliability, robustness, and text quality (Fernandez et al., 2023; Kirchenbauer et al., 2024; Kuditipudi et al., 2024; Hu et al., 2024). However, applications such as source attribution require richer information, including user or tenant identifiers, model versions, and generation metadata (Wang et al., 2024; Yoo et al., 2024). This need motivates *multi-bit watermarking*, which embeds a payload into generated text for later recovery. Multi-bit verification therefore inherently involves two conceptually distinct tasks: determining whether a watermark is present and, if so, recovering the embedded payload.

These two tasks require distinct statistical guarantees: presence testing must control the false-positive rate (FPR), whereas payload recovery is evaluated by decoding accuracy. Some prior multi-bit schemes are designed primarily for payload recovery and do not provide a payload-independent FPR guarantee on unwatermarked inputs (Qu et al., 2025). Crucially, FPR is a property of the full decision rule, not of any individual payload candidate. When presence is inferred from the maximum decoder score over candidate payloads, the null distribution is governed by this maximum, so a per-candidate threshold may substantially underestimate the overall FPR as the candidate space

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.

grows (Yoo et al., 2024; Fu & Russell, 2025). Recent approaches address this issue through different designs: Fu and Russell's Dual Watermark assigns disjoint, hash-selected token subsets to detection and identification (Fu & Russell, 2025), whereas BREW conditions verification on an estimated payload while accounting for blind payload selection (Kim et al., 2026). These designs either separate token evidence across tasks or make presence decisions dependent on payload estimation, leaving open our central question: can the same token observations simultaneously provide payload-independent evidence for presence testing with finite-sample FPR control and payload-specific evidence for payload recovery?

We introduce **PIPER** (Payload-Independent Presence Evaluation and Recovery), a multi-bit watermark built on *presence before payload*. Its nested vocabulary partition divides a coarse parent set into two finer child subsets. Biasing the payload-selected child creates both parent-level presence evidence and child-level payload evidence from the same tokens. Verification applies an exact binomial test to parent membership over deduplicated contexts and decodes the payload only upon acceptance. Under the random-key/ideal-PRF null, we prove finite-sample FPR control uniformly over payload-layer choices, without held-out calibration.

On 8-bit payloads in 300-token OPT-1.3B continuations, PIPER achieves 98.7% presence TPR and a 95.0% correct acceptance-and-recovery rate (CAR), with empirical FPRs of 0.86% on unwatermarked model-generated text and 0.92% on natural text at a nominal 1% level. At 100 tokens, sharing evidence improves CAR by 24.1 percentage points over a balanced disjoint-token design at the same logit bias. Random-key tests yield empirical FPRs below the nominal levels, while fixed-key evaluation on the OpenGen dataset yields an FPR above the nominal level.

Our main contributions are:

- We formalize blind multi-bit verification as presence testing followed by conditional payload recovery, distinguishing FPR, TPR, and CAR.
- We introduce a shared hierarchical partition that supports payload-independent presence testing and payload recovery from the same tokens, without reserving detection-only positions.
- We prove finite-sample FPR and false-attribution bounds under the random-key/ideal-PRF null, uniformly over payload-layer choices, and derive a detection-power bound under a separate signal assumption.
- We benchmark PIPER across payload sizes and generation lengths, assess detection, recovery, and text quality, and examine FPR calibration under random keys and fixed-key distribution shifts.

2 Problem Formulation

2.1 Watermark Embedding

Let $\mathcal{V}$ be the vocabulary of an LLM, and let π denote a prompt. A payload $m \in \{0,1\}^k$ is encoded by an error-correcting code (ECC) into a binary codeword $c = \mathrm{ECCEnc}(m) \in \{0,1\}^n$, providing redundancy for reliable payload recovery in the presence of bit errors.

Under the random-key model, a secret key $\mathsf{SK} \sim \mathrm{Unif}(\mathcal{K})$ is sampled once at setup. We write sk for its realization, which is held fixed throughout generation and the corresponding verification. At each generation step t, a keyed assignment rule selects a codeword coordinate $j_t \in [n]$. The corresponding bit c_{j_t} determines the modification to the model's next-token distribution, yielding the watermarked distribution $P_{\mathsf{sk}}(\cdot \mid \pi, x_{<t}; c_{j_t})$. This induces the sequence distribution

$$P_{\mathsf{sk},m}(x \mid \pi) = \prod_{t=1}^{T} P_{\mathsf{sk}}(x_t \mid \pi, x_{<t}; c_{j_t}), \tag{1}$$

where T denotes the number of generated tokens.

2.2 Watermark Verification

A blind verifier receives a text x and a secret key sk, but not the payload m. It first tests for watermark presence and attempts payload recovery only if the test accepts. Formally, it returns

$$\mathrm{Verify}(x, \mathsf{sk}) = (I, \hat{m}) \in \{0,1\} \times \big(\{0,1\}^k \cup \{\perp\}\big), \qquad I = 0 \Longrightarrow \hat{m} = \perp, \tag{2}$$

where $\perp$ denotes that no payload is recovered.

2.2.1 PRESENCE TESTING

Let P_0 denote the unwatermarked sequence distribution, with prompt dependence suppressed. For a fixed but unknown payload m, define

$$\begin{aligned} H_0 : &\quad X \sim P_0, \qquad X \perp\!\!\!\perp \mathsf{SK}, \\ H_1 : &\quad X \mid \mathsf{SK} = \mathsf{sk} \sim P_{\mathsf{sk},m}(\cdot \mid \pi). \end{aligned} \tag{3}$$

Here H_1 corresponds to the fixed payload m; no prior over payloads is assumed. Probabilities under both hypotheses average over the setup-time key draw and the corresponding text-generation randomness.

Let $S(x, \mathsf{sk})$ denote a detection statistic computed from the text x and the secret key sk, with larger values indicating stronger evidence of a watermark. The verifier then sets

$$I(x, \mathsf{sk}) = \mathbf{1}\{S(x, \mathsf{sk}) \geq \tau_\alpha\}. \tag{4}$$

We write I for the random decision $I(X, \mathsf{SK})$. If $I = 0$, the verifier returns $(0, \perp)$ and stops. The test is *payload-independent* if its statistic and threshold are defined without reference to any payload candidate. Its false-positive rate and true-positive rate are

$$\mathrm{FPR} = \Pr_{H_0}(I = 1), \qquad \mathrm{TPR} = \Pr_{H_1}(I = 1). \tag{5}$$

The threshold τ_α is chosen so that $\mathrm{FPR} \leq \alpha$.

2.2.2 PAYLOAD RECOVERY

If $I = 1$, the verifier produces the $\hat{m}$ component of Verify by first extracting codeword evidence and then applying the ECC decoder:

$$\hat{c} = \mathrm{Extract}(x, \mathsf{sk}), \qquad \hat{m} = \mathrm{ECCDec}(\hat{c}) \in \{0,1\}^k \cup \{\perp\}. \tag{6}$$

Let $\hat{M}$ denote the verifier's payload output on (X, SK), with $\hat{M} = \perp$ whenever $I = 0$.

The *correct acceptance-and-recovery rate* (CAR) is

$$\mathrm{CAR} = \Pr_{H_1}(I = 1, \hat{M} = m). \tag{7}$$

False payload attribution. For any $\mathcal{B} \subseteq \{0,1\}^k$, since $\perp \notin \mathcal{B}$,

$$\Pr_{H_0}(\hat{M} \in \mathcal{B}) = \Pr_{H_0}(I = 1, \hat{M} \in \mathcal{B}) \leq \Pr_{H_0}(I = 1) = \mathrm{FPR} \leq \alpha. \tag{8}$$

Thus, a level-α presence test also bounds the probability of falsely assigning a payload in $\mathcal{B}$ to an unwatermarked text, regardless of the downstream payload decoder.

3 PAYLOAD SEARCH AND FALSE-POSITIVE INFLATION

Before presenting PIPER, we examine a common decode-first design that uses payload search for both recovery and presence testing. Let $\mathcal{R} \subseteq \{0,1\}^k$ be the candidate set, and let $S_u(x, \mathsf{sk})$ denote the score assigned to candidate $u \in \mathcal{R}$. Such a verifier declares watermark presence when the maximum candidate score $\max_{u \in \mathcal{R}} S_u(x, \mathsf{sk})$ exceeds a threshold τ. Its false-positive rate is

$$\mathrm{FPR}_{\mathrm{search}} = \Pr_{H_0}\left(\max_{u \in \mathcal{R}} S_u(X, \mathsf{SK}) \geq \tau \right). \tag{9}$$

Thus, the null distribution is that of the maximum score, not of any individual candidate.

If every candidate has the same null exceedance probability $p_\tau = \Pr_{H_0}(S_u(X, \mathsf{SK}) \geq \tau)$ and the candidate-wise exceedance events are independent, then

$$\mathrm{FPR}_{\mathrm{search}} = 1 - (1 - p_\tau)^{|\mathcal{R}|}, \tag{10}$$

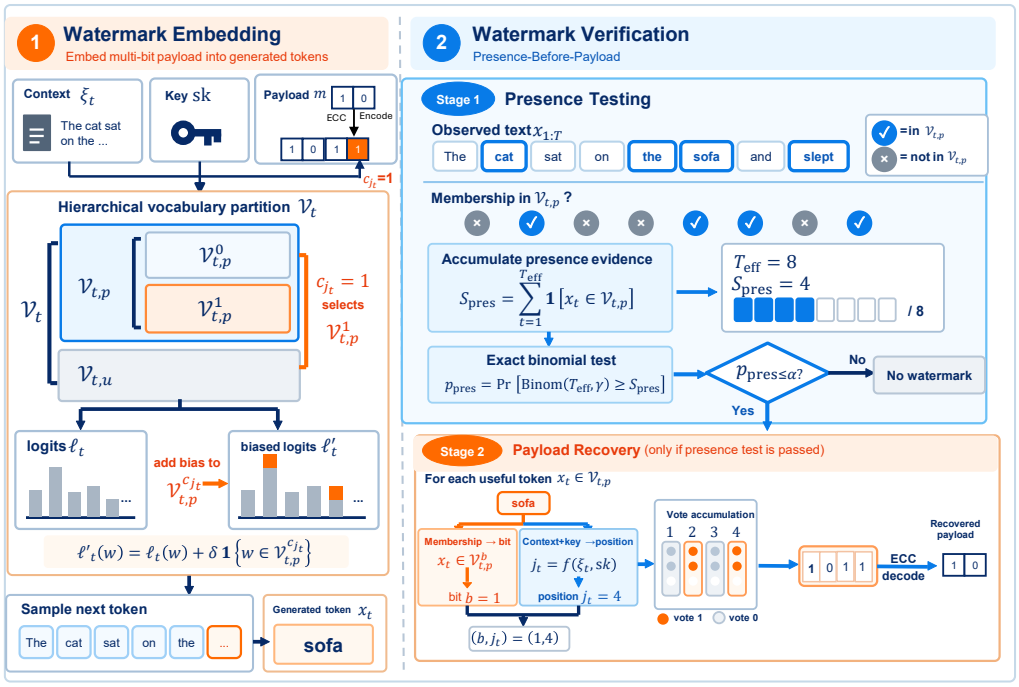


Figure 1: The PIPER pipeline. A shared hierarchical vocabulary partition supports payload-independent presence testing and conditional payload recovery from the same token observations.

which can greatly exceed p_τ as $|\mathcal{R}|$ grows. A threshold calibrated for a single candidate therefore need not control the FPR of the full search-and-decode procedure.

Payload search remains valid if τ is calibrated against the null distribution of the maximum statistic. However, this distribution depends on $\mathcal{R}$, so calibration for one candidate set need not transfer to another; see Appendix B for further discussion. PIPER instead defines its presence statistic and threshold τ_α without reference to any payload candidate or candidate set. Consequently, neither the downstream decoder nor its search space can alter the presence acceptance event or its FPR.

4 The PIPER Method

4.1 Overview

PIPER follows the principle of *presence before payload*, separating presence testing from payload recovery while sharing their token observations. A keyed vocabulary partition defines a parent set containing two child subsets representing bits 0 and 1. Generation biases only the child selected by the encoded bit. Because this child lies within the parent, the same logit bias produces both parent-level presence evidence and child-level payload evidence, without reserving detection-only token positions or applying a separate parent-level bias.

Verification first tests parent membership over distinct contexts, without consulting payload candidates. Only upon acceptance does it use child membership for bit voting and ECC decoding. Under the random-key/ideal-PRF null, this presence test provides finite-sample FPR control through an exact binomial test, without held-out calibration. Figure 1 provides a schematic overview of the full embedding and verification pipeline.

4.2 Shared-Partition Watermark Embedding

Let $c = \text{ECCEnc}(m) \in \{0,1\}^n$ be the encoded payload. At generation step $t \in \{1, \ldots, T\}$, let ξ_t be the last h tokens of the concatenation of the prompt π and the generated prefix $x_{<t}$. We assume

that the prompt contains at least h tokens; for $t > h$, $\xi_t = x_{t-h:t-1}$. Blind verification therefore excludes the first h generation positions.

A keyed hash of ξ_t seeds a pseudorandom permutation of the vocabulary $\mathcal{V}$. The first g entries in this permutation form the parent set $\mathcal{V}_{t,p}$, where $\gamma = g/|\mathcal{V}|$. Within the same permutation order, the first $\lfloor g/2 \rfloor$ parent entries form $\mathcal{V}^0_{t,p}$, and the remaining parent entries form $\mathcal{V}^1_{t,p}$; the two child subsets represent bits 0 and 1, respectively. By default, $g = \lfloor |\mathcal{V}|/2 \rfloor$, giving $\gamma \approx 1/2$.

A second, domain-separated keyed hash maps ξ_t to a codeword coordinate $j_t \in [n]$. The bit c_{j_t} selects the child subset to bias. Let $\ell_t(w)$ denote the base-model logit of token w given $(\pi, x_{<t})$. The watermarked next-token distribution is

$$P_{\mathsf{sk}}(w \mid \pi, x_{<t}; c_{j_t}) = \frac{\exp\big(\ell_t(w) + \delta \mathbf{1}\{w \in \mathcal{V}^{c_{j_t}}_{t,p}\}\big)}{\sum_{v \in \mathcal{V}} \exp\big(\ell_t(v) + \delta \mathbf{1}\{v \in \mathcal{V}^{c_{j_t}}_{t,p}\}\big)}, \qquad w \in \mathcal{V}, \tag{11}$$

where $\delta > 0$ is the logit bias. The next token is sampled from this distribution over the full vocabulary. Because $\mathcal{V}^{c_{j_t}}_{t,p} \subseteq \mathcal{V}_{t,p}$, the same perturbation increases both parent membership and membership in the selected child, thereby producing presence and payload evidence simultaneously. Watermark embedding pseudocode appears in Algorithm 1 in the appendix.

4.3 Presence-Before-Payload Verification

4.3.1 Payload-Independent Presence Testing

Given a candidate text $x = x_{1:T}$ and the secret key sk, the verifier reconstructs $\xi_t = x_{t-h:t-1}$ at each eligible position $t > h$. It retains only the first occurrence of each distinct context, forming the effective-position set $\mathcal{T}_{\text{eff}}$, with $T_{\text{eff}} = |\mathcal{T}_{\text{eff}}|$. Thus, each distinct context contributes at most one observation. This selection depends only on the text, not on the key or any payload candidate.

For each retained position, the verifier reconstructs $\mathcal{V}_{t,p}$ and records parent membership:

$$Y_t = \mathbf{1}\{x_t \in \mathcal{V}_{t,p}\}, \qquad S_{\text{pres}} = \sum_{t \in \mathcal{T}_{\text{eff}}} Y_t. \tag{12}$$

For a fixed T_{eff}, larger values of S_{pres} indicate stronger evidence of watermark presence.

Under the random-key/ideal-PRF null, Theorem 5.1 establishes that

$$S_{\text{pres}} \mid X = x \sim \text{Binomial}(T_{\text{eff}}, \gamma). \tag{13}$$

For a significance level $\alpha \in (0, 1)$, PIPER computes the exact upper-tail p-value

$$p_{\text{pres}} = \sum_{r=S_{\text{pres}}}^{T_{\text{eff}}} \binom{T_{\text{eff}}}{r} \gamma^r (1-\gamma)^{T_{\text{eff}}-r}. \tag{14}$$

Because p_{pres} is nonincreasing in S_{pres}, the presence decision admits the equivalent forms

$$I = \mathbf{1}\{S_{\text{pres}} \geq \tau_\alpha(T_{\text{eff}})\} = \mathbf{1}\{p_{\text{pres}} \leq \alpha\}. \tag{15}$$

Here $\tau_\alpha(T_{\text{eff}})$ is the smallest $s \in \{0, \ldots, T_{\text{eff}}\}$ for which the $\text{Binomial}(T_{\text{eff}}, \gamma)$ upper-tail probability is at most α, with the convention $\tau_\alpha(T_{\text{eff}}) = T_{\text{eff}} + 1$ if no such s exists. Thus, S_{pres} and $\tau_\alpha(T_{\text{eff}})$ instantiate the statistic and threshold defined in Section 2.

Under the stated null, the resulting test controls the finite-sample FPR at level α without held-out calibration. When $I = 1$, the verifier proceeds to conditional payload recovery.

4.3.2 Conditional Payload Recovery

Upon acceptance ($I = 1$), the verifier reconstructs the child subsets and recomputes the codeword coordinates for the same effective positions $\mathcal{T}_{\text{eff}}$. For each codeword coordinate $j \in [n]$, it counts votes for $b \in \{0, 1\}$ and forms a hard bit estimate:

$$N^b_j = \sum_{t \in \mathcal{T}_{\text{eff}}} \mathbf{1}\{j_t = j\} \mathbf{1}\{x_t \in \mathcal{V}^b_{t,p}\}, \qquad \hat{c}_j = \mathbf{1}\{N^1_j > N^0_j\}. \tag{16}$$

Tokens outside the parent set contribute no payload vote. Ties, including coordinates receiving no votes, are resolved as bit 0. The resulting estimate $\hat{c} = (\hat{c}_1, \ldots, \hat{c}_n)$ instantiates $\mathrm{Extract}(x, \mathsf{sk})$ from Section 2 and is passed to the ECC decoder:

$$\hat{m} = \mathrm{ECCDec}(\hat{c}) \in \{0,1\}^k \cup \{\perp\}. \tag{17}$$

The verifier returns $(1, \hat{m})$, with $\hat{m} = \perp$ if the decoder reports failure.

For a fixed text, key, and presence-testing configuration, changing the payload length, decoder, ECC configuration, or registered-message set can affect recovery but cannot alter the presence decision. Verification pseudocode appears in Algorithm 2 in the appendix.

5 Theoretical Guarantees

For PIPER's shared-partition construction (Section 4), we establish finite-sample FPR control and false-attribution bounds under the random-key/ideal-PRF null. We then connect the embedding rule to parent-level evidence and derive a sufficient effective length for reliable detection under an average conditional signal assumption. Throughout, $\alpha \in (0,1)$. Proofs are provided in Appendix D.

Random-key/ideal-PRF null. Under H_0, the setup key $\mathsf{SK} \sim \mathrm{Unif}(\mathcal{K})$ is independent of $X \sim P_0$. Distinct contexts induce independent, uniformly random parent sets of size g, with $\gamma = g/|\mathcal{V}|$.

Theorem 5.1 (Finite-sample FPR control)**.** *Under the stated null, for every fixed text x, which determines T_{eff},*

$$S_{\mathrm{pres}} \mid X = x \sim \mathrm{Binomial}(T_{\mathrm{eff}}, \gamma), \qquad \Pr_{H_0}(I = 1 \mid X = x) \le \alpha. \tag{18}$$

Hence, $\Pr_{H_0}(I = 1) \le \alpha$ for every key-independent unwatermarked distribution P_0.

For a fixed text, context deduplication yields independent membership indicators over the ideal keyed randomness; no independence among text tokens is required. The guarantee averages over setup randomness and need not hold for every fixed deployment key.

Corollary 5.1 (Decoder invariance and false payload attribution)**.** *For a fixed text, key, and presence-testing configuration, PIPER's presence decision is unchanged by the payload length, decoder, ECC configuration, or registered-message set.*

Under the stated null, any downstream decoder whose random payload output $\hat{M}$ satisfies $\hat{M} = \perp$ whenever $I = 0$ obeys

$$\Pr_{H_0}(\hat{M} \in \mathcal{B}) \le \mathrm{FPR} \le \alpha, \qquad \mathcal{B} \subseteq \{0,1\}^k. \tag{19}$$

Connection to the embedding rule. Let u_t and q_t be the base-model probability masses of the parent and selected child, respectively. The child-level bias changes the parent-membership probability to $u_t + \frac{(e^\delta - 1) q_t (1 - u_t)}{1 + (e^\delta - 1) q_t}$. The increment is positive when $\delta > 0$ and $0 < q_t \le u_t < 1$: payload embedding also creates parent-level evidence without a separate bias. Since u_t need not equal γ, power requires a separate condition on the average signal above γ.

Average parent-level signal. For power analysis, fix an integer $N \ge 1$ with $\Pr_{H_1}(T_{\mathrm{eff}} = N) > 0$. Conditional on $T_{\mathrm{eff}} = N$, let $t_1 < \cdots < t_N$ be the retained positions. Assume that, for some $0 < \epsilon \le 1 - \gamma$,

$$\frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{H_1}\left[Y_{t_i} \mid T_{\mathrm{eff}} = N,\, Y_{t_1}, \ldots, Y_{t_{i-1}} \right] \ge \gamma + \epsilon \tag{20}$$

almost surely under this conditional law. Individual conditional probabilities may fall below γ, and independence is not required. The FPR guarantees above do not depend on this condition.

Theorem 5.2 (Effective length for reliable presence detection)**.** *Under the average parent-level signal condition, PIPER's exact binomial test satisfies*

$$\Pr_{H_1}(I = 0 \mid T_{\mathrm{eff}} = N) \le \exp\left[-2N \left(\epsilon - \sqrt{\frac{\log(1/\alpha)}{2N}} \right)_+^2 \right], \tag{21}$$

where $(z)_+ = \max\{z, 0\}$. *For any* $\beta \in (0, 1)$, *conditional detection power is at least* $1-\beta$ *whenever*

$$N \geq \frac{\left(\sqrt{\log(1/\alpha)} + \sqrt{\log(1/\beta)}\right)^2}{2\epsilon^2}. \tag{22}$$

For a common signal margin ϵ, this sufficient effective length scales quadratically in $1/\epsilon$ and logarithmically in the inverse error targets. Although the presence rule is payload-independent, ϵ may depend on the payload and generation configuration.

6 Experiments

6.1 Experimental Setup

Datasets and Models. Following prior LLM watermarking studies, we use prompts from the RealNewsLike subset of C4 and generate continuations of $T = 300$ tokens. We use OPT-1.3B as the watermarking generator and OPT-6.7B as an independent evaluator for continuation-only perplexity (PPL). Unless otherwise specified, we embed an 8-bit payload protected by $\mathrm{BCH}(23, 8, 3)$, and set the logit bias δ, parent fraction γ, and context window h as described in Appendix E.

Baselines. We compare PIPER with three representative multi-bit watermarks: MPAC (Yoo et al., 2024), which uses position-wise voting; CTWL (Wang et al., 2024), which searches over candidate messages; and BREW (Kim et al., 2026), which estimates a payload before designated-codeword verification.

Metrics. We report FPR and TPR for presence testing, CAR for joint acceptance-and-recovery, and PPL for text quality. FPR, TPR, and CAR follow the definitions in Section 2; PPL is computed as described above.

Implementation Details. All methods are evaluated under the same blind text-verification protocol: the verifier receives a candidate text, its required secret keys, and public configuration, but neither the original prompt nor the embedded payload. PIPER performs payload-independent presence testing before decoding; each baseline follows its specified verification procedure. Generation settings, calibration procedures, baseline configurations, and cross-model input construction are described in Appendix E. Attack transformations and robustness results are in Appendix F.6.

6.2 Baseline Comparison

Table 1 compares PIPER with representative multi-bit watermarking baselines under a unified blind verification protocol. All methods are evaluated with the same generation model and text length ($T = 300$). We report FPR, TPR, and CAR. Method configurations are detailed in Appendices E.1 and E.2; supplementary metric definitions are given in Appendix F.4.

PIPER achieves reliable presence detection while maintaining strong payload recovery. For the 8-bit setting, PIPER obtains 98.7% TPR with 0.86%/0.92% FPR on model-generated/natural text and 95.0% CAR. Importantly, increasing the payload length from 8 to 16 bits has little effect on presence detection (99.0% TPR), while the decrease in CAR (from 95.0% to 77.5%) mainly reflects the increased difficulty of recovering longer payloads. This separation demonstrates that the presence decision remains independent of payload decoding.

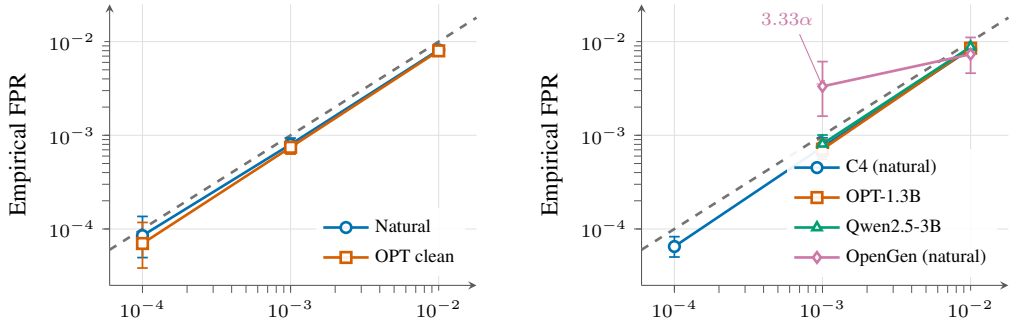
Existing methods exhibit different trade-offs. BREW achieves moderate performance in the 8-bit setting but degrades substantially for 16-bit payloads, where TPR decreases to 19.4%. Under the blind verification protocol used here, this behavior suggests that payload-conditioned verification becomes increasingly challenging as the payload space grows. In contrast, PIPER maintains a stable presence test because its detection statistic is independent of payload length.

6.3 False-Positive Reliability

Random-key null diagnostic. Figure 2(a) evaluates 100 keys on the same 2,000 unwatermarked texts per source. Aggregate empirical FPRs remain below the nominal levels $\alpha \in$

Table 1: Main comparison under different payload lengths. All experiments use $T = 300$ and nominal level $\alpha = 1\%$. For the 8-bit and 16-bit settings, we use BCH$(23, 8, 3)$ and BCH$(31, 16, 3)$, respectively. FPR is reported as model-null/natural-null, and FPR, TPR, and CAR are given in percentages. In the calibration column, Yes denotes held-out threshold calibration, MC denotes native Monte Carlo calibration, and No (exact) denotes a prespecified exact-binomial test.

Method	Held-out calibration	Bits	FPR (M/N) ↓	TPR@1% FPR	CAR ↑	PPL ↓
MPAC (2024)	Yes	8	0.94 / 0.72	95.3	83.4	10.15
	Yes	16	1.78 / 0.92	88.6	49.2	10.65
CTWL (2024)	Yes	8	1.28 / 0.98	95.5	95.4	7.68
	Yes	16	1.44 / 1.32	91.6	91.3	7.64
BREW (2026)	Yes	8	1.00 / 0.80	64.3	63.6	9.26
	Yes	16	0.00 / 0.00	19.4	19.3	9.24
PIPER (Ours)	No (exact)	8	0.86 / 0.92	98.7	95.0	8.82
	No (exact)	16	0.86 / 0.92	99.0	77.5	8.78



(a) Random-key diagnostic **(b) Fixed-key evaluation**

Figure 2: PIPER false-positive reliability. The dashed diagonal denotes the nominal level ($y = x$); both axes use logarithmic scales. **(a)** Each source contains 2,000 unwatermarked texts evaluated under 100 keys. The 95% intervals are descriptive because texts are reused across keys. **(b)** One fixed deployment key is evaluated on natural and model-generated texts without threshold fitting; bars show 95% Clopper–Pearson intervals. OpenGen exceeds the nominal level at $\alpha = 10^{-3}$. Full sample counts, false positives, and intervals appear in Appendix F.1.

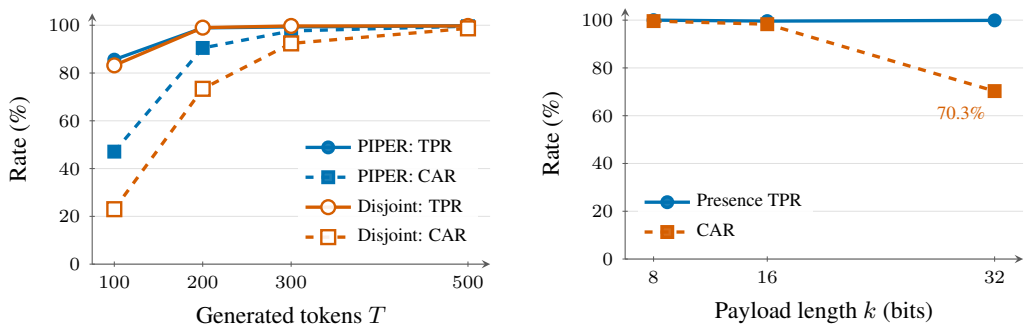
$\{10^{-2}, 10^{-3}, 10^{-4}\}$, consistent with the finite-sample random-key guarantee. Because texts are reused across keys, the plotted intervals are descriptive rather than confidence intervals for independent deployment trials.

Fixed-key deployment. Figure 2(b) evaluates one fixed key without test-set threshold fitting. Empirical FPRs on C4, OPT-1.3B, and Qwen2.5-3B remain below the tested targets. At $\alpha = 10^{-4}$, one million C4 texts yield 65 false positives (FPR 6.50×10^{-5}). In contrast, OpenGen yields 3.33×10^{-3} at $\alpha = 10^{-3}$, with a 95% interval of $[1.60, 6.12] \times 10^{-3}$. This deviation shows that the random-key guarantee does not imply uniform calibration for every fixed key and text distribution.

Full numerical results are in Appendix F.1; MPAC native-threshold diagnostics and cross-method reliability curves are in Appendices F.2 and F.3.

6.4 PIPER vs. Disjoint-Token Ablation

We compare PIPER with a disjoint-token ablation of Fu & Russell (2025). The methods share the model, prompts, payloads, keys, seeds, ECC, detector, and evaluation code, and differ only in evidence routing: PIPER shares all tokens between presence testing and payload recovery,



(a) Shared vs. disjoint evidence **(b) Payload scaling**

Figure 3: Presence detection and payload recovery under two scaling settings. **(a)** PIPER versus the balanced 50/50 disjoint baseline with an 8-bit payload, fixed bias $\delta = 2.0$, and $\alpha = 10^{-3}$. In (a), blue denotes PIPER and orange denotes the disjoint baseline. Solid circles denote TPR and dashed squares denote CAR in both panels. **(b)** PIPER at $T = 500$ and $\alpha = 10^{-2}$, using BCH$(23, 8, 3)$, BCH$(31, 16, 3)$, and BCH$(50, 32, 3)$ for 8, 16, and 32 raw bits. In (b), blue denotes TPR and orange denotes CAR. Presence TPR remains nearly unchanged while CAR decreases at larger payloads. Full numerical results, including FPR, are provided in the appendix.

whereas the disjoint baseline reserves half of the eligible positions for presence testing and half for payload recovery. We fix the generation bias to $\delta = 2.0$ for both methods and evaluate $T \in \{100, 200, 300, 500\}$, an 8-bit payload, and $\alpha = 10^{-3}$.

As shown in Figure 3(a), PIPER raises CAR over the 50/50 disjoint baseline by 24.1, 17.1, 5.2, and 0.9 percentage points at $T = 100, 200, 300, 500$, respectively, while both methods approach saturation at long lengths. The largest gain therefore occurs in the short-text regime, where splitting the token budget most strongly limits payload evidence. Complete numerical results, quality costs, effective sample sizes, and alternative disjoint allocations are reported in Appendix F.5.

6.5 Payload Scaling

Figure 3(b) evaluates payload scaling at a fixed generated-token budget of $T = 500$ and nominal level $\alpha = 10^{-2}$. The payload is encoded using a BCH configuration (n, k, t), where n is the codeword length, k is the number of information bits (i.e., the payload length), and t is the error-correction capability. TPR remains at least 99.6% across all three configurations. FPRs on model-generated and natural text, reported in Appendix Table 7, remain unchanged. In contrast, CAR decreases from 99.6% for BCH$(23, 8, 3)$ to 70.3% for BCH$(50, 32, 3)$, showing that payload recovery, rather than presence detection, becomes the bottleneck as the payload length grows.

Tables 7–10 in Appendix F report extended payload metrics, the full quality–reliability sweep, independent-evaluator results, and cross-model/domain transfer experiments.

7 Conclusion

PIPER separates the two questions that a multi-bit verifier must answer: whether watermark evidence is present and, only then, which payload was embedded. Its nested keyed partition lets the same effective tokens support both decisions without allowing payload search or ECC decoding to enlarge the presence event. The exact-binomial guarantee is finite-sample under the random-key/ideal-PRF null, while the fixed-key and domain-shift experiments make clear where deployment calibration remains necessary. Across detection, attribution, capacity, quality, transfer, and editing tests, the results show that shared evidence can improve short-text attribution while retaining a simple and auditable presence test. Future work should develop fixed-key or distribution-robust calibration and extend shared-partition verification to stronger semantic transformations.

AI USE STATEMENT

We used generative AI tools to assist with literature discovery, research brainstorming, experimental and theoretical discussion, mathematical proof development, drafting, and language polishing. All AI-assisted claims, proofs, code, and experimental decisions were independently checked by the authors, who take full responsibility for the final content.

REFERENCES

Scott Aaronson. Watermarking of large language models, 2023. URL https://simons.berkeley.edu/talks/scott-aaronson-ut-austin-openai-2023-08-17. Simons Institute talk.

Wenbo An, Lianwei Wu, and Zehao Wang. IPMark: A sentence-level watermark for LLMs with hierarchical personalization and efficient detection. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64793.

Xiangyu Feng and Pei-Gen Ye. SemBits: Multi-bit semantic watermarking with sentence-level hashing for LLMs. In *Data Security and Privacy Protection*, pp. 183–196. Springer Nature, 2026. doi: 10.1007/978-981-95-3185-1_12.

Xiaoyan Feng, He Zhang, Yanjun Zhang, Leo Yu Zhang, and Shirui Pan. BiMark: Unbiased multilayer watermarking for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 17049–17067. PMLR, 2025. URL https://proceedings.mlr.press/v267/feng25u.html.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. In *2023 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6. IEEE, 2023. doi: 10.1109/WIFS58808.2023.10374576. URL https://ieeexplore.ieee.org/document/10374576.

Zihao Fu and Chris Russell. Multi-use LLM watermarking and the false detection problem, 2025. URL https://arxiv.org/abs/2506.15975v1.

Thibaud Gloaguen, Robin Staab, Mark Vero, and Martin Vechev. Every bit, everywhere, all at once: A binomial multibit LLM watermark, 2026. URL https://arxiv.org/abs/2605.11653v1.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Baihe Huang, Eric Xu, Kannan Ramchandran, Jiantao Jiao, and Michael I. Jordan. Towards anytime-valid statistical watermarking. *arXiv preprint arXiv:2602.17608*, 2026.

Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. StealthInk: A multi-bit and stealthy watermark for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27685–27709. PMLR, 2025. URL https://proceedings.mlr.press/v267/jiang25j.html.

Ya Jiang, Massieh Kordi Boroujeny, Surender Suresh Kumar, and Kai Zeng. MirrorMark: A distortion-free multi-bit watermark for large language models, 2026. URL https://arxiv.org/abs/2601.22246.

Joeun Kim, HoEun Kim, Dongsup Jin, and Young-Sik Kim. Block-wise codeword embedding for reliable multi-bit text watermarking, 2026. URL http://arxiv.org/abs/2605.00348.

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

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/d78e9e4316e1714fbb0f20be66f8044c-Abstract-Conference.html.

Massieh Kordi Boroujeny, Ya Jiang, Kai Zeng, and Brian L. Mark. Multi-bit distortion-free watermarking for large language models, 2024. URL https://arxiv.org/abs/2402.16578.

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

Xiang Li, Feng Ruan, Huiyuan Wang, Qi Long, and Weijie J. Su. A statistical framework of watermarks for large language models: Pivot, detection efficiency and optimal rules. *The Annals of Statistics*, 53(1):322–351, 2025. doi: 10.1214/24-AOS2468.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip S. Yu. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 57(2):1–36, 2024. doi: 10.1145/3691626.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. Provably robust multi-bit watermarking for AI-generated text. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 201–220. USENIX Association, 2025. URL https://www.usenix.org/conference/usenixsecurity25/presentation/qu-watermarking.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. Towards codable watermarking for injecting multi-bits information to llms. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/abdc8c031aa6c6917c3b593166e5e340-Abstract-Conference.html.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338, 2025.

Xiaojun Xu, Jinghan Jia, Yuanshun Yao, Yang Liu, and Hang Li. Robust multi-bit text watermark with LLM-based paraphrasers. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://icml.cc/virtual/2025/poster/46008.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

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

APPENDIX CONTENTS

A Extended Background and Related Work . . . 13
A.1 Statistical Watermarking for LLMs . . . 13
A.2 Multi-bit Watermarking for LLMs . . . 13
A.3 False-Positive Control in Multi-bit Watermarking . . . 14

B Detailed False-Positive Analysis . . . 14

C Complete PIPER Specification . . . 17
C.1 Scope and Conventions . . . 17
C.2 PIPER Design Details . . . 17

D Proofs of Theoretical Guarantees . . . 19
D.1 Proof of Finite-Sample FPR Control . . . 20
D.2 Proof of Decoder Invariance and False Payload Attribution . . . 20
D.3 Proof of the Shared-Evidence Observation Advantage . . . 21
D.4 Proof of Effective Length for Reliable Presence Detection . . . 22

E Experimental Details . . . 23
E.1 PIPER Main-Experiment Configuration . . . 23
E.2 Baseline Main-Experiment Configurations . . . 23

F Additional Experiments . . . 24
F.1 Random-Key and Fixed-Key Null Results . . . 24
F.2 MPAC Native-Threshold Null Diagnostic . . . 24
F.3 C4 FPR Reliability Data . . . 25
F.4 Extended Capacity, Quality, and Transfer Results . . . 26
F.5 Shared and Disjoint Evidence Allocation . . . 29
F.6 Robustness to Editing, Mixing, and Truncation . . . 30
F.7 Clean-Text Performance and Length Scaling . . . 31

A Extended Background and Related Work

A.1 Statistical Watermarking for LLMs

Early text watermarking methods embedded information through formatting changes, lexical substitutions, or syntactic transformations (Liu et al., 2024). These carriers can preserve the apparent meaning of a text but may be disrupted by canonicalization, reformatting, or editing. Generation-time watermarking incorporates the signal directly into the language model's token-generation process.

A representative construction uses a keyed vocabulary partition and biases generation toward a designated green subset (Kirchenbauer et al., 2023). Detection aggregates green-list membership and tests whether the resulting count is unusually large under an unwatermarked null. This formulation gives the false-positive rate (FPR) a direct interpretation as the Type-I error probability of the presence test.

Fernandez et al. (2023) showed that Gaussian Z-approximations can underestimate false-positive probabilities at stringent significance levels. They derived non-asymptotic null distributions under independence assumptions, including a binomial test for green-list counts and a Gamma test for Aaronson-style scores, and improved empirical calibration through context–token tuple deduplication. Their results highlight the importance of both finite-sample calibration and dependence among scored observations. Kirchenbauer et al. (2024) further investigated detection under paraphrasing, text mixing, and other modifications.

Other constructions seek to preserve the original model distribution. Kuditipudi et al. (2024) couple language-model sampling with randomized key sequences, while Hu et al. (2024) develop unbiased reweighting mechanisms. Their distribution-preservation guarantees average over the prescribed watermark randomness and depend on the construction's independence and randomness-reuse conditions.

Beyond individual constructions, Li et al. (2025) formulate detection through pivotal statistics whose null distributions do not depend on the underlying next-token probabilities. They separate Type-I error control from detection efficiency and study detection rules through the asymptotic decay of Type-II error. For sequential detection, Huang et al. (2026) show that repeatedly applying fixed-horizon tests with data-dependent stopping can invalidate nominal Type-I error guarantees. Their e-value and test-supermartingale framework instead supports anytime-valid detection under optional stopping.

A.2 Multi-bit Watermarking for LLMs

Multi-bit watermarking embeds recoverable payloads for applications such as model identification and user attribution. Early constructions often recover messages by comparing candidate watermark hypotheses: Fernandez et al. (2023) associate payloads with keyed or cyclically shifted hypotheses, while Wang et al. (2024) maximize a payload-dependent likelihood. Exhaustive search over the full k-bit payload space $\{0,1\}^k$ requires evaluating 2^k candidates.

Position- and segment-based constructions reduce this search burden. MPAC assigns tokens to payload coordinates and recovers symbols through coordinate-wise voting (Yoo et al., 2024). The segment-based scheme of Qu et al. (2025) (RSBH) combines frequency-balanced token assignment, candidate enumeration within each segment, and Reed–Solomon decoding. These approaches avoid full-payload enumeration while accumulating evidence for individual coordinates or coded segments.

Another line extends unbiased or distortion-free generation to multi-bit payloads. Kordi Boroujeny et al. introduce DISC, combining distortion-free sampling with efficient multi-bit decoding (Kordi Boroujeny et al., 2024). StealthInk preserves the token distribution in expectation over randomized watermark ciphers (Jiang et al., 2025). BiMark combines bit-flip unbiased reweighting with a multilayer architecture (Feng et al., 2025), while MirrorMark uses measure-preserving mirroring and a context-anchored balanced scheduler, preserving distribution when composed with a distortion-free base sampler (Jiang et al., 2026). These methods seek to preserve generation quality while improving payload recovery; their guarantees and robustness depend on the construction.

Table 2: Selected multi-bit verification designs. Token allocation refers to the evidence used for presence testing and payload recovery.

Method	Token allocation	Presence evidence
Fu–Russell DW	Disjoint	Dedicated indicator statistic
Every Bit Everywhere	Shared	Aggregate bit statistics
BREW	Shared	Codeword verification conditioned on an estimated payload
PIPER	Shared	Parent-membership statistic

A complementary line uses sentence-level carriers. Xu et al. select between two fine-tuned paraphrasers according to successive payload bits and train a classifier to recover them (Xu et al., 2025). SemBits uses semantic locality-sensitive hashing in a sentence-level accept–reject loop, without modifying or fine-tuning the underlying generator (Feng & Ye, 2026). IPMark combines sentence-level candidate selection, semantic–syntactic signals, and hierarchical addressing for personalized attribution (An et al., 2026). These methods target text quality and robustness to editing, with different practical costs: paraphraser-based methods require auxiliary training, rejection sampling can require multiple candidates per accepted sentence, and achievable payload capacity depends on the available sentence-level evidence.

A.3 False-Positive Control in Multi-bit Watermarking

Reliable presence testing requires calibrating the complete decision rule, including any payload selection or alignment search. Existing methods address this requirement through different designs. For example, DISC computes a global p-value accounting for candidate payloads and initial-chunk search (Kordi Boroujeny et al., 2024). Table 2 summarizes selected verification architectures; their statistical guarantees depend on the calibration assumptions described below.

Fu and Russell's Dual Watermark (DW) reserves separate token positions for an indicator watermark, providing presence evidence independently of identity recovery. Their hybrid variants additionally incorporate identification statistics into the final decision (Fu & Russell, 2025).

Gloaguen et al. use an aggregate bit statistic for presence testing and estimate its null distribution through Monte Carlo simulation, empirically validating the resulting calibration (Gloaguen et al., 2026). This detector does not require a selected payload candidate, but its statistic and null calibration depend on the payload dimension. BREW first estimates a payload and then verifies the keyed codewords implied by that estimate. It provides finite-sample FPR bounds accounting for blind payload selection and alignment search under its stated probabilistic assumptions (Kim et al., 2026). Its presence decision nevertheless depends on the estimated payload and verification configuration.

PIPER combines shared token evidence with a presence rule that is invariant to payload-layer choices. For a fixed text, key, and presence-testing configuration, changing the payload length, decoder, ECC configuration, or registered-message set cannot alter the presence decision. Its exact binomial test over deduplicated contexts provides finite-sample FPR control under the random-key/ideal-PRF null. Section 4 presents the construction, and Appendix B further analyzes the statistical mechanisms underlying these verification designs.

B Detailed False-Positive Analysis

We expand Section 3 through four related mechanisms: candidate search, null-model misspecification, position-wise maximization, and ECC acceptance-region expansion. These mechanisms can overlap; each illustrates why FPR calibration must account for the complete verification rule.

(1) **Payload-space search and multiple comparisons.**

Let $\mathcal{R} \subseteq \{0,1\}^k$ be a fixed, nonempty candidate set, and let $S_u(X, \mathsf{SK})$ be the score for candidate u. Without assuming independence among candidate scores, the union bound

gives

$$\begin{aligned}\mathrm{FPR}_{\mathrm{search}} &= \Pr_{H_0}\left(\bigcup_{u\in\mathcal{R}}\{S_u(X,\mathsf{SK})\geq\tau\}\right)\\ &\leq \min\left\{1,\ \sum_{u\in\mathcal{R}}\Pr_{H_0}(S_u(X,\mathsf{SK})\geq\tau)\right\}.\end{aligned} \tag{23}$$

If every candidate has null exceedance probability p_τ, then $\mathrm{FPR}_{\mathrm{search}} \leq \min\{1, |\mathcal{R}|p_\tau\}$. This bound applies even when all scores use the same text and key. It does not imply that inflation is necessarily large: the actual FPR depends on the joint distribution of the scores.

For illustration, independent exceedance events give $1-(1-p_\tau)^{|\mathcal{R}|} \approx 0.632$ when $p_\tau = 10^{-4}$ and $|\mathcal{R}| = 10^4$. Exhaustive search over all k-bit payloads sets $|\mathcal{R}| = 2^k$. Conversely, calibrating each candidate at level $\alpha/|\mathcal{R}|$ is sufficient for overall level-α control regardless of dependence, although it can be conservative.

(2) **Null-model misspecification and detector miscalibration.**

Even without payload search, the population FPR can exceed its nominal level if calibration uses an incorrect null distribution. Let $\Lambda(X,\mathsf{SK})$ denote a generic verifier's complete detection statistic, and let $G_0^\star(s)$ denote its working null upper-tail probability, i.e., $G_0^\star(s) = \Pr_{H_0^\star}[\Lambda(X,\mathsf{SK})\geq s]$. Choose a threshold $\tau_\alpha^\star$ satisfying

$$G_0^\star(\tau_\alpha^\star) \leq \alpha. \tag{24}$$

The actual population FPR is

$$\mathrm{FPR}(\alpha) = \Pr_{H_0}(\Lambda(X,\mathsf{SK})\geq\tau_\alpha^\star), \tag{25}$$

which need not be at most α. In particular, the threshold is anti-conservative if the actual null tail at $\tau_\alpha^\star$ exceeds α.

This is a population probability, not a finite-sample estimate. For R null evaluation pairs $(x^{(r)}, \mathsf{sk}^{(r)})$, the empirical FPR is

$$\widehat{\mathrm{FPR}}_R(\alpha) = \frac{1}{R}\sum_{r=1}^{R}\mathbf{1}\left\{\Lambda(x^{(r)},\mathsf{sk}^{(r)})\geq\tau_\alpha^\star\right\}.$$

Under the random-key evaluation law above, this estimates $\mathrm{FPR}(\alpha)$. If all $\mathsf{sk}^{(r)}$ equal one fixed deployment key, it instead estimates the null rejection probability conditional on that key. Dependence caused by reusing texts or keys must be accounted for when quantifying uncertainty.

The relevant null distribution depends on the conditioning regime. For PIPER, conditioning on a fixed unwatermarked text leaves independent, uniformly random parent sets at distinct contexts under the ideal-PRF model. Retaining one observation per context therefore gives the exact binomial null with parameter $\gamma = g/|\mathcal{V}|$, without requiring the parent sets to have equal language-model probability mass. For a fixed deployment key, however, these membership indicators need not follow the same binomial law across texts, and calibration can depend on the text distribution.

Misspecification can also arise from repeated contexts, inaccurate tail approximations, or omitted selection operations. Fernandez et al. (2023) improve empirical calibration using finite-sample tests and context–token tuple deduplication. Gloaguen et al. (2026) report miscalibrated MPAC and StealthInk p-values on human text and improved empirical calibration after estimating the complete statistics' null distributions by Monte Carlo simulation. Such simulation requires a suitable null model and does not, by itself, establish an exact finite-sample guarantee.

(3) **Post-selection bias from position-wise maximization.**

Position-allocation decoders can select the best-supported symbol separately at each payload position (Yoo et al., 2024). If the selected counts are then used for presence testing, this creates a local form of the candidate search considered above.

Let T_{elig} be the number of token observations eligible under the analyzed allocation rule; it need not equal the generated length T or PIPER's effective length T_{eff}. Suppose these

observations are assigned to q payload positions. Let L_j be the number assigned to position j, so that $\sum_{j=1}^{q} L_j = T_{\text{elig}}$. Conditional on these allocation counts, assume an illustrative null in which all observations have independent, uniformly distributed binary labels. The position-wise count vectors are then independent, with

$$(N_j^0, N_j^1) \sim \text{Multinomial}\left(L_j; \frac{1}{2}, \frac{1}{2}\right). \tag{26}$$

All probabilities and expectations in this calculation are conditional on the allocation counts. Here N_j^b matches the child-vote count defined in Section 4.3.2.

Define the selected support at each position and its aggregate as

$$M_j = \max\{N_j^0, N_j^1\}, \qquad S_{\max} = \sum_{j=1}^{q} M_j. \tag{27}$$

Here M_j is a per-position maximum count, distinct from the verifier's payload output $\hat{M}$ in Section 2. For every position with $L_j > 0$,

$$\mathbb{E}_{H_0}[M_j] > \max_{b \in \{0,1\}} \mathbb{E}_{H_0}[N_j^b] = \frac{L_j}{2}. \tag{28}$$

Moreover, $N_j^0 \sim \text{Binomial}(L_j, 1/2)$, while $S_{\max} \geq \sum_j N_j^0$ for every realization. Thus, the selected statistic has an upper tail at least as large as this fixed-symbol binomial reference. Applying that reference after maximization can produce anti-conservative p-values, even when the underlying multinomial model is correct.

The shift is explicit from the identity

$$M_j = \max\{N_j^0, N_j^1\} = \frac{L_j}{2} + \left| N_j^0 - \frac{L_j}{2} \right|, \qquad N_j^0 \sim \text{Binomial}(L_j, 1/2).$$

For large L_j, a Gaussian approximation gives

$$\mathbb{E}_{H_0}[M_j] \approx \frac{L_j}{2} + \sqrt{\frac{L_j}{2\pi}}.$$

For balanced allocations with integer $T_{\text{elig}}/q \gg 1$,

$$\mathbb{E}_{H_0}[S_{\max}] \approx \frac{T_{\text{elig}}}{2} + \sqrt{\frac{qT_{\text{elig}}}{2\pi}}. \tag{29}$$

Consequently, the naive standardized statistic

$$Z_{\text{naive}} = \frac{S_{\max} - T_{\text{elig}}/2}{\sqrt{T_{\text{elig}}/4}}$$

has approximate mean

$$\mathbb{E}_{H_0}[Z_{\text{naive}}] \approx \sqrt{\frac{2q}{\pi}}, \tag{30}$$

rather than zero. Within this approximation regime, increasing the number of positions increases the normalized selection bias. The approximation should not be extrapolated to allocations with few observations per position.

Maximization therefore changes the null distribution that must be calibrated; it does not make valid testing impossible. Calibrating the complete selected statistic accounts for this effect, although the required threshold and detection power can depend on the allocation and payload configuration.

(4) **Acceptance-region expansion under error-correcting decoding.**

ECC does not inherently invalidate FPR control. Miscalibration can arise when successful decoding is treated as evidence of watermark presence without accounting for the decoder's complete acceptance region.

Consider a binary codebook $\mathcal{C} \subseteq \{0,1\}^n$ of size 2^k and a bounded-distance decoder with correction radius t_{ECC} (the parameter t in $\mathrm{BCH}(n,k,t)$). Its acceptance region is

$$\mathcal{D}_{\mathrm{ECC}} = \left\{ z \in \{0,1\}^n : \min_{c \in \mathcal{C}} d_{\mathrm{H}}(z,c) \le t_{\mathrm{ECC}} \right\}, \tag{31}$$

where d_{H} denotes Hamming distance. This region is the union of Hamming balls around all valid codewords.

Under the illustrative null in which the decoder input is uniform on $\{0,1\}^n$, these balls are disjoint when $2t_{\mathrm{ECC}} < d_{\min}$, where $d_{\min}$ is the minimum distance between distinct codewords. The resulting FPR is

$$\mathrm{FPR}_{\mathrm{ECC}} = \frac{2^k}{2^n} \sum_{i=0}^{t_{\mathrm{ECC}}} \binom{n}{i}. \tag{32}$$

Uniform decoder inputs are an additional illustrative assumption; they are not implied by PIPER's random-key null.

More generally, the FPR is the null probability mass of the entire acceptance region. For a fixed codebook and null distribution, increasing the accepted decoding radius cannot decrease this probability. If a nearest-neighbor decoder always returns a codeword and every returned codeword is accepted as evidence of presence, the FPR is one. An ECC-based presence rule can nevertheless be valid if its complete acceptance region is calibrated under the appropriate null.

PIPER separates these decisions: its presence statistic and threshold are defined without reference to payload candidates or ECC decoding regions. For a fixed text, key, and presence-testing configuration, downstream decoding cannot enlarge the presence acceptance event. The finite-sample FPR guarantee follows from the random-key/ideal-PRF null specified in Section 5; the separation alone does not establish calibration under a different null model. The construction of PIPER is given in Section 4; the full algorithmic details appear in Appendix C.

C Complete PIPER Specification

This appendix supplements Section 4 with implementation conventions, integer partition sizes, the probability-mass calculation, and complete embedding and verification algorithms.

C.1 Scope and Conventions

We use the notation and presence-before-payload rule of Section 4. The specification below makes the prompt boundary, domain-separated PRF inputs, codeword routing, and empty-evidence and tie cases explicit. The false-positive mechanisms motivating this design are analyzed in Section 3 and Appendix B.

C.2 PIPER Design Details

C.2.1 Shared-Partition Watermark Generation

Let $m \in \{0,1\}^k$ be the raw payload and let $c = \mathrm{ECCEnc}(m) = (c_1, \ldots, c_n) \in \{0,1\}^n$ be its length-n ECC codeword, with $[n] = \{1, \ldots, n\}$. Fix an integer parent size $g \in \{1, \ldots, |\mathcal{V}| - 1\}$ for all positions and write $\gamma = g/|\mathcal{V}|$. At generation step $t \in \{1, \ldots, T\}$, the language model produces the next-token logits $\boldsymbol{\ell}_t = \mathcal{M}(\pi, x_{<t})$. Define $\xi_t = \mathrm{suffix}_h(\pi \,\|\, x_{<t})$, where the tokenized prompt π contains at least h tokens and $x_{<1}$ is empty. Thus the prompt supplies context for $t \le h$, and $\xi_t = x_{t-h:t-1}$ for $t > h$. Generation may watermark the first h continuation tokens, but blind verification does not score them because their full contexts require the prompt.

Derive two domain-separated PRF outputs from the same master key,

$$s_t^{\mathrm{part}} = H(\mathsf{sk}, \mathtt{part} \,\|\, \xi_t), \qquad s_t^{\mathrm{idx}} = H(\mathsf{sk}, \mathtt{idx} \,\|\, \xi_t).$$

Algorithm 1 PIPER Multi-bit Watermark Embedding

Require: Language model $\mathcal{M}$; tokenized prompt π with at least h tokens; raw watermark message $m \in \{0,1\}^k$; ECC encoder ECCEnc; generation length T; vocabulary $\mathcal{V}$; fixed parent size $g \in \{1, \dots, |\mathcal{V}| - 1\}$; context length h; payload bias $\delta > 0$; domain-separated keyed hash/PRF H; secret key sk

Output: Watermarked sequence $x_{1:T}$

$c \leftarrow \mathrm{ECCEnc}(m) = (c_1, \dots, c_n)$

for $t = 1, \dots, T$ **do**

  $\boldsymbol{\ell}_t \leftarrow \mathcal{M}(\pi, x_{<t})$

  $\xi_t \leftarrow \mathrm{suffix}_h(\pi \,\|\, x_{<t})$ ▷ $x_{<1}$ is empty

  $s_t^{\mathrm{part}} \leftarrow H(\mathsf{sk}, \mathtt{part} \,\|\, \xi_t)$

  $s_t^{\mathrm{idx}} \leftarrow H(\mathsf{sk}, \mathtt{idx} \,\|\, \xi_t)$

  Use s_t^{part} to permute $\mathcal{V}$; take the first g entries as $\mathcal{V}_{t,p}$ and the remainder as $\mathcal{V}_{t,u}$:

$$\mathcal{V} = \mathcal{V}_{t,p} \cup \mathcal{V}_{t,u}$$

  In the same permutation order, take the first $\lfloor g/2 \rfloor$ parent entries as $\mathcal{V}_{t,p}^0$ and the remainder as $\mathcal{V}_{t,p}^1$:

$$\mathcal{V}_{t,p} = \mathcal{V}_{t,p}^0 \cup \mathcal{V}_{t,p}^1$$

  $j_t \leftarrow 1 + (s_t^{\mathrm{idx}} \bmod n)$

  The encoded bit c_{j_t} selects the target child set $\mathcal{V}_{t,p}^{c_{j_t}}$

  Apply the payload bias only to the selected child set; no separate bias is applied to $\mathcal{V}_{t,p}$:

$$\widetilde{P}_t(w) \leftarrow \frac{\exp\left(\ell_t(w) + \delta \mathbf{1}\{w \in \mathcal{V}_{t,p}^{c_{j_t}}\}\right)}{\sum_{v \in \mathcal{V}} \exp\left(\ell_t(v) + \delta \mathbf{1}\{v \in \mathcal{V}_{t,p}^{c_{j_t}}\}\right)}, \qquad w \in \mathcal{V}$$

  $x_t \sim \widetilde{P}_t$

end for

return $x_{1:T}$

The partition output s_t^{part} deterministically constructs the parent partition

$$\mathcal{V} = \mathcal{V}_{t,p} \cup \mathcal{V}_{t,u}, \qquad |\mathcal{V}_{t,p}| = g, \tag{33}$$

where $\mathcal{V}_{t,u}$ denotes the complement of the parent set. The balanced default uses $g = \lfloor |\mathcal{V}|/2 \rfloor$, so the two parent sizes differ by at most one and $\gamma = 1/2$ whenever $|\mathcal{V}|$ is even. It then constructs a child partition of the parent set,

$$\mathcal{V}_{t,p} = \mathcal{V}_{t,p}^0 \cup \mathcal{V}_{t,p}^1, \qquad |\mathcal{V}_{t,p}^0| = \lfloor |\mathcal{V}_{t,p}|/2 \rfloor, \qquad |\mathcal{V}_{t,p}^1| = |\mathcal{V}_{t,p}| - |\mathcal{V}_{t,p}^0|. \tag{34}$$

The domain-separated index output assigns position t to the encoded-bit index $j_t = 1 + (s_t^{\mathrm{idx}} \bmod n)$, and the encoded bit c_{j_t} selects the target child set $\mathcal{V}_{t,p}^{c_{j_t}}$. For the occupancy analysis below, we idealize $J_t = 1 + (H(\mathsf{sk}, \mathtt{idx} \,\|\, \xi_t) \bmod n)$ as conditionally uniform on $[n]$; the implementation's finite-range mapping has negligible modulo bias and is not used as an exact uniformity claim.

For the mass calculation below, abbreviate the watermarked distribution defined in Section 4 as

$$\widetilde{P}_t(w) := P_{\mathsf{sk}}(w \mid \pi, x_{<t}; c_{j_t}), \qquad w \in \mathcal{V}. \tag{35}$$

Algorithm 1 gives the full-vocabulary normalization and sampling rule.

The increase in parent probability follows directly from the normalization. Let $P_t = \mathrm{softmax}(\boldsymbol{\ell}_t)$, $q_t = P_t(\mathcal{V}_{t,p}^{c_{j_t}})$, and $u_t = P_t(\mathcal{V}_{t,p})$. The bias multiplies the target-child mass by e^δ, so the relative normalizer is $Z_t = (1 - q_t) + e^\delta q_t = 1 + (e^\delta - 1)q_t$. Hence, whenever $0 < q_t \le u_t < 1$,

$$\begin{aligned} \widetilde{P}_t(\mathcal{V}_{t,p}^{c_{j_t}}) &= \frac{e^\delta q_t}{Z_t} > q_t, \\ \widetilde{P}_t(\mathcal{V}_{t,p}) &= \frac{u_t + (e^\delta - 1)q_t}{Z_t} > u_t. \end{aligned} \tag{36}$$

Algorithm 2 PIPER Multi-bit Watermark Verification

Require: Token sequence $x_{1:T}$; vocabulary $\mathcal{V}$; fixed parent size $g \in \{1, \ldots, |\mathcal{V}| - 1\}$; codeword length n; ECC decoder ECCDec; context length h; domain-separated keyed hash/PRF H; secret key sk; target level $0 < \alpha < 1$

Output: Watermark decision I and recovered message $\hat{m}$

$\gamma \leftarrow g/|\mathcal{V}|$ ▷ Default: $g = \lfloor |\mathcal{V}|/2 \rfloor$
$S_{\text{pres}} \leftarrow 0$, $\mathcal{T}_{\text{eff}} \leftarrow \emptyset$
$\mathcal{U}_{\text{seen}} \leftarrow \emptyset$ ▷ Set of previously processed contexts; used for deduplication
for $t = h + 1, \ldots, T$ **do**
  $\xi_t \leftarrow x_{t-h:t-1}$
  if $\xi_t \notin \mathcal{U}_{\text{seen}}$ **then**
    $\mathcal{U}_{\text{seen}} \leftarrow \mathcal{U}_{\text{seen}} \cup \{\xi_t\}$, $\mathcal{T}_{\text{eff}} \leftarrow \mathcal{T}_{\text{eff}} \cup \{t\}$
    $s_t^{\text{part}} \leftarrow H(\mathsf{sk}, \texttt{part} \,\|\, \xi_t)$
    Reconstruct the permutation seeded by s_t^{part}; its first g entries form $\mathcal{V}_{t,p}$
    $S_{\text{pres}} \leftarrow S_{\text{pres}} + \mathbf{1}\{x_t \in \mathcal{V}_{t,p}\}$
  end if
end for
$T_{\text{eff}} \leftarrow |\mathcal{T}_{\text{eff}}|$
if $T_{\text{eff}} = 0$ **then**
  return $(0, \perp)$
end if
$p_{\text{pres}} \leftarrow \sum_{r=S_{\text{pres}}}^{T_{\text{eff}}} \binom{T_{\text{eff}}}{r} \gamma^r (1-\gamma)^{T_{\text{eff}}-r}$
$I \leftarrow \mathbf{1}\{p_{\text{pres}} \le \alpha\}$
if $I = 0$ **then**
  return $(0, \perp)$
end if
$N_j^0 \leftarrow 0$, $N_j^1 \leftarrow 0$ for every $j \in [n]$
for $t \in \mathcal{T}_{\text{eff}}$ in increasing order **do**
  $\xi_t \leftarrow x_{t-h:t-1}$
  $s_t^{\text{part}} \leftarrow H(\mathsf{sk}, \texttt{part} \,\|\, \xi_t)$, $s_t^{\text{idx}} \leftarrow H(\mathsf{sk}, \texttt{idx} \,\|\, \xi_t)$
  Reconstruct the same parent permutation; its first $\lfloor g/2 \rfloor$ entries form $\mathcal{V}_{t,p}^0$ and the remaining parent entries form $\mathcal{V}_{t,p}^1$
  $j_t \leftarrow 1 + (s_t^{\text{idx}} \bmod n)$
  $N_{j_t}^b \leftarrow N_{j_t}^b + \mathbf{1}\{x_t \in \mathcal{V}_{t,p}^b\}$ for $b \in \{0, 1\}$
end for
$\hat{c}_j \leftarrow \mathbf{1}\{N_j^1 > N_j^0\}$ for every $j \in [n]$
$\hat{c} \leftarrow (\hat{c}_1, \ldots, \hat{c}_n)$; $\hat{m} \leftarrow \text{ECCDec}(\hat{c})$
return $(1, \hat{m})$

These inequalities quantify the generation-side coupling; the null guarantee is established separately in Theorem 5.1.

C.2.2 Presence Testing and Conditional Payload Decoding

Algorithm 2 gives the complete verification procedure for the method in Section 4, including context deduplication, empty-evidence rejection, and tie handling. Payload recovery is performed only after presence acceptance. The validity proof and its fixed-key qualification are given in Appendix D.1.

D Proofs of Theoretical Guarantees

Throughout, $\alpha \in (0, 1)$ and the payload m under H_1 is fixed. We use S_{pres}, T_{eff}, and $\tau_\alpha(T_{\text{eff}})$ from Section 4. In the null proof, $N = T_{\text{eff}}(x)$ is determined by the fixed text x. In the power and recovery analyses, N denotes a fixed positive value of T_{eff}, and all stated signal assumptions are imposed under the conditional law given $T_{\text{eff}} = N$.

D.1 Proof of Finite-Sample FPR Control

Under the random-key/ideal-PRF null, condition on $X = x$ and write $N = T_{\text{eff}}(x)$. The contexts and retained positions are fixed. The remaining randomness is the ideal keyed construction, independent of x.

At each retained position, the parent is a uniformly random size-g subset of $\mathcal{V}$, so

$$\Pr_{H_0}(Y_t = 1 \mid X = x) = \frac{g}{|\mathcal{V}|} = \gamma.$$

The retained contexts are distinct and their ideal parent sets are independent. Consequently,

$$S_{\text{pres}} \mid X = x = \sum_{t \in \mathcal{T}_{\text{eff}}} Y_t \sim \text{Binomial}(N, \gamma).$$

This argument makes no independence assumption about the text tokens.

Write

$$\overline{F}_N(s) = \Pr(Z \geq s), \qquad Z \sim \text{Binomial}(N, \gamma).$$

The exact cutoff is

$$\tau_\alpha(N) = \min\{s \in \{0, \ldots, N+1\} : \overline{F}_N(s) \leq \alpha\},$$

where $\overline{F}_N(N+1) = 0$. The value $N+1$ means that no attainable count is accepted. Since the upper tail is nonincreasing,

$$\{p_{\text{pres}} \leq \alpha\} = \{S_{\text{pres}} \geq \tau_\alpha(N)\}.$$

It follows that

$$\Pr_{H_0}(I = 1 \mid X = x) = \overline{F}_N(\tau_\alpha(N)) \leq \alpha.$$

When $N = 0$, $S_{\text{pres}} = 0$, $p_{\text{pres}} = 1$, and $\tau_\alpha(0) = 1$, so the verifier returns $(0, \perp)$. Averaging over $X \sim P_0$ proves Theorem 5.1. The bound can be strict because the statistic is discrete.

Scope of the random-key guarantee. The guarantee averages over ideal setup randomness and need not hold for every fixed deployment key. To express the effect of a changed null law, let $Q_0^\star$ and Q_0 denote the distributions of p_{pres} under the ideal null and deployment, respectively. For the acceptance region $[0, \alpha]$,

$$Q_0([0,\alpha]) \leq Q_0^\star([0,\alpha]) + d_{\text{TV}}(Q_0, Q_0^\star) \leq \alpha + d_{\text{TV}}(Q_0, Q_0^\star).$$

Here d_{TV} denotes total variation distance. Using the p-value accounts for the length-dependent cutoff; the count alone does not specify the acceptance event when T_{eff} varies. This inequality provides no numerical deployment guarantee without controlling the distance. Fixed-key results are reported in Section 6.3.

D.2 Proof of Decoder Invariance and False Payload Attribution

Let D be any downstream procedure returning one payload or $\perp$. For a fixed input, the gated verifier is

$$\text{Verify}_D(x, \mathsf{sk}) = \begin{cases} (0, \perp), & p_{\text{pres}}(x, \mathsf{sk}) > \alpha, \\ (1, D(x, \mathsf{sk})), & p_{\text{pres}}(x, \mathsf{sk}) \leq \alpha. \end{cases}$$

Its presence component is always $\mathbf{1}\{p_{\text{pres}}(x, \mathsf{sk}) \leq \alpha\}$. For fixed text, key, and presence-testing configuration, it is unchanged by the payload length, ECC configuration, registered-message set, or internal decoding procedure. The decoder may use randomized search, internal candidate lists, or erasure handling, provided its reported output remains a single payload or $\perp$.

For random inputs, let $\hat{M}$ denote the reported payload, including any decoder randomness. Since $I = 0$ implies $\hat{M} = \perp$, for every $\mathcal{B} \subseteq \{0,1\}^k$,

$$\{\hat{M} \in \mathcal{B}\} = \{I = 1, \hat{M} \in \mathcal{B}\} \subseteq \{I = 1\}.$$

Therefore,

$$\Pr_{H_0}(\hat{M} \in \mathcal{B}) \leq \Pr_{H_0}(I = 1) \leq \alpha,$$

proving Corollary 5.1. Changing the generation configuration can change the distribution of the text and hence detection power; this does not contradict the pointwise invariance of the presence rule.

D.3 SHARED-EVIDENCE OBSERVATION ADVANTAGE

We first verify the parent-mass identity used in the power analysis.

Parent-mass increment under child bias. Let P_t denote the base-model next-token distribution at position t, and define

$$u_t = P_t(\mathcal{V}_{t,p}), \qquad q_t = P_t(\mathcal{V}_{t,p}^{c_{j_t}}).$$

The child-level bias multiplies the selected child's unnormalized weight by e^δ and leaves all other weights unchanged. Hence the normalizer becomes

$$1 + (e^\delta - 1)q_t,$$

and the parent weight becomes

$$u_t + (e^\delta - 1)q_t.$$

Their ratio is the biased parent-membership probability:

$$\widetilde{P}_t(\mathcal{V}_{t,p}) = \frac{u_t + (e^\delta - 1)q_t}{1 + (e^\delta - 1)q_t}. \tag{37}$$

Subtracting u_t gives

$$\widetilde{P}_t(\mathcal{V}_{t,p}) - u_t = \frac{(e^\delta - 1)q_t(1 - u_t)}{1 + (e^\delta - 1)q_t}. \tag{38}$$

The increment is strictly positive whenever $\delta > 0$ and $0 < q_t \le u_t < 1$. This proves the identity displayed in Section 5.

We now formalize the observation-sharing consequence separately from the signal assumption.

Proposition D.1 (Shared-evidence observation advantage)**.** *Fix a text, key, and presence-testing configuration, and let $N = |\mathcal{T}_{\text{eff}}|$. Count payload votes available before presence gating and any additional decoder filtering. PIPER supplies*

$$N_{\text{pres}} = N, \qquad N_{\text{pay}} = S_{\text{pres}}.$$

For any disjoint allocation $\mathcal{T}_{\text{eff}} = \mathcal{T}_{\text{pres}} \dot\cup \mathcal{T}_{\text{pay}}$ of the same observations, with the same partitions and assignments,

$$N_{\text{pres}}^{\text{disj}} = |\mathcal{T}_{\text{pres}}| \le N, \qquad N_{\text{pay}}^{\text{disj}} = \sum_{t \in \mathcal{T}_{\text{pay}}} Y_t \le S_{\text{pres}}.$$

Proof. Each retained position contributes one parent-membership observation. Since the two child subsets partition the parent, a retained token contributes exactly one child vote if $Y_t = 1$ and no child vote otherwise. Summing gives the two identities for PIPER. Restricting each task to its allocated subset gives the disjoint counts and their inequalities. □

For example, if N is even and a uniformly random half of the retained positions is allocated to recovery, independently of their contents, then, conditional on the fixed observations,

$$\mathbb{E}_{\text{split}}[N_{\text{pay}}^{\text{disj}}] = \frac{1}{2} S_{\text{pres}}, \qquad N_{\text{pres}}^{\text{disj}} = \frac{N}{2}.$$

Thus, allocating $N/2$ positions to recovery is distinct from obtaining $N/2$ actual child votes. These comparisons concern evidence routing on the same observations; they do not establish an ordering of CAR or text quality for separately generated watermark schemes.

Under the average parent-level signal condition of Theorem 5.2, the tower property additionally gives

$$\mathbb{E}_{H_1}[N_{\text{pay}} \mid T_{\text{eff}} = N] = \mathbb{E}_{H_1}[S_{\text{pres}} \mid T_{\text{eff}} = N] \ge N(\gamma + \epsilon).$$

This expectation requires the stated conditional signal assumption. The positive local mass increment alone does not establish it: the base parent mass need not equal γ, and conditioning on the final effective length changes the relevant law. In addition, deduplication among eligible verification positions does not guarantee that a retained context was never queried at an earlier, excluded generation position.

D.4 Proof of Effective Length for Reliable Presence Detection

Fix $N \geq 1$ with $\Pr_{H_1}(T_{\text{eff}} = N) > 0$ and work under this conditional law. Enumerate the retained positions as $t_1 < \cdots < t_N$, set

$$\mathcal{G}_i = \sigma(Y_{t_1}, \ldots, Y_{t_i}), \qquad \mathcal{G}_0 = \{\emptyset, \Omega\},$$

and define

$$\mu_i = \mathbb{E}_{H_1}[Y_{t_i} \mid \mathcal{G}_{i-1}, T_{\text{eff}} = N].$$

The average parent-level signal condition of Theorem 5.2 states that

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{H_1}[Y_{t_i} \mid T_{\text{eff}} = N,\, Y_{t_1}, \ldots, Y_{t_{i-1}}] \geq \gamma + \epsilon$$

almost surely. Since μ_i is the i-th conditional expectation, this is exactly

$$\sum_{i=1}^{N} \mu_i \geq N(\gamma + \epsilon) \quad \text{almost surely.}$$

No lower bound is required for any individual μ_i.

Concentration under the average signal condition. The variables $Y_{t_i} - \mu_i$ are martingale differences. Conditional on the preceding indicators, each lies in an interval of length one. The conditional Hoeffding lemma therefore gives, for every $\lambda > 0$,

$$\mathbb{E}_{H_1}\left[e^{-\lambda(Y_{t_i} - \mu_i)} \mid \mathcal{G}_{i-1}, T_{\text{eff}} = N\right] \leq e^{\lambda^2/8}.$$

Iterating this inequality and applying exponential Markov inequality with $\lambda = 4a/N$ yields, for $a > 0$,

$$\Pr_{H_1}\left(\sum_{i=1}^{N}(Y_{t_i} - \mu_i) \leq -a \,\middle|\, T_{\text{eff}} = N\right) \leq e^{-2a^2/N}.$$

This establishes concentration without independence among the retained indicators.

Relating the exact cutoff to effective length. Let

$$b_N = N\gamma + \sqrt{\frac{N}{2}\log\frac{1}{\alpha}}.$$

For $Z \sim \text{Binomial}(N, \gamma)$, Hoeffding's inequality implies $\Pr(Z \geq b_N) \leq \alpha$. Thus, if $\lceil b_N \rceil \leq N$, this count is accepted by the exact test; otherwise, the sentinel $N + 1$ bounds the cutoff. Hence,

$$\tau_\alpha(N) \leq \min\{N + 1, \lceil b_N \rceil\}.$$

If $\lceil b_N \rceil \leq N$, then $\tau_\alpha(N) \leq \lceil b_N \rceil$, so $\tau_\alpha(N) - 1 < \lceil b_N \rceil - 1 \leq b_N$. If $b_N > N$, then $\tau_\alpha(N) \leq N + 1$, so $\tau_\alpha(N) - 1 \leq N < b_N$. In both cases,

$$\tau_\alpha(N) - 1 < b_N.$$

The integer-valued miss event satisfies

$$I = 0 \quad \Longleftrightarrow \quad S_{\text{pres}} \leq \tau_\alpha(N) - 1 \quad \Longrightarrow \quad S_{\text{pres}} < b_N.$$

On this event,

$$\sum_{i=1}^{N}(Y_{t_i} - \mu_i) < b_N - N(\gamma + \epsilon).$$

If $N(\gamma + \epsilon) > b_N$, apply the concentration inequality with $a = N(\gamma + \epsilon) - b_N$. Otherwise use the trivial bound one. Combining the cases proves

$$\Pr_{H_1}(I = 0 \mid T_{\text{eff}} = N) \leq \exp\left[-2N\left(\epsilon - \sqrt{\frac{\log(1/\alpha)}{2N}}\right)_+^2\right].$$

Using $\tau_\alpha(N)-1$ in the miss event accounts for the discrete cutoff without adding a rounding penalty.

For $\beta \in (0,1)$, the sufficient length condition

$$N \geq \frac{\left(\sqrt{\log(1/\alpha)} + \sqrt{\log(1/\beta)}\right)^2}{2\epsilon^2}$$

implies

$$\epsilon - \sqrt{\frac{\log(1/\alpha)}{2N}} \geq \sqrt{\frac{\log(1/\beta)}{2N}}.$$

Substitution bounds the conditional miss probability by β, completing Theorem 5.2. The signal condition is imposed after conditioning on $T_{\text{eff}} = N$; an unconditional mean bound alone is insufficient.

E Experimental Details

E.1 PIPER Main-Experiment Configuration

The main comparison uses OPT-1.3B for generation and OPT-6.7B for perplexity evaluation, with continuation length $T = 300$. The raw payload contains 8 or 16 information bits before ECC encoding, using $\mathrm{BCH}(23, 8, t_{\mathrm{ECC}} = 3)$ and $\mathrm{BCH}(31, 16, t_{\mathrm{ECC}} = 3)$, respectively.

Generation uses the soft mode with zero separate parent-set bias and payload bias $\delta = 2$. The implementation uses self-hash seeding, the v2 partition implementation, context width $h = 4$, and candidate top-$k = 50$. Sampling uses temperature 1.0, top-$k = 50$, and top-$p = 0.95$. The global seed and message seed are both 42.

Verification follows the blind-text interface. The verifier receives only the generated text and secret key, without access to the original prompt or embedded payload. Presence detection uses the exact binomial test at nominal level $\alpha = 0.01$ and counts only unique contexts. No held-out threshold fitting is used for PIPER. Payload decoding resolves tied votes as bit zero. Configurations explicitly stated for individual ablations take precedence over this default setting.

E.2 Baseline Main-Experiment Configurations

These configurations correspond to the main comparison and are distinct from the native-threshold diagnostics reported later.

Baselines. We compare PIPER with three representative multi-bit watermarks: MPAC (Yoo et al., 2024), which uses coordinate-wise voting; CTWL (Wang et al., 2024), which performs candidate-message likelihood search; and BREW (Kim et al., 2026), which estimates a payload before designated-codeword verification. Baseline thresholds and calibration procedures follow the recommended protocols of their respective methods.

MPAC. The 8-bit comparison uses base 4, $\gamma = 0.25$, watermark strength $\delta = 2.0$, self-hash seeding, and $\mathrm{BCH}(23, 8, t_{\mathrm{ECC}} = 3)$ coding. Repeated n-grams are ignored during detection. Sampling uses temperature 1.0 and top-$p = 0.95$. The detector threshold is calibrated using pooled model-generated and natural calibration negatives and frozen before evaluation.

The 16-bit configuration uses base 4, $\gamma = 0.25$, $\delta = 2.0$, self-hash seeding, and $\mathrm{BCH}(31, 16, t_{\mathrm{ECC}} = 3)$ coding. Its frozen threshold at nominal level $\alpha = 0.01$ is 9.882871850048918. The no-ECC left-hash configuration in Appendix F.2 is a separate native-statistic diagnostic and is not the main comparison setting.

CTWL. The main runs use $\delta = 1.5$, temperature 1.0, top-$p = 0.95$, and the native language-model-based message encoding. The evaluated payload settings use 8-bit and 16-bit message codes with encoding ratios 37.5 and 18.75, respectively, and encoding length 300. The message model is GPT-2 with prefix length 10, 100 random permutations, and the vanilla message-model strategy. The text-level detection statistic is the maximum softmax probability over candidate messages. For each payload setting, the detection threshold is calibrated on 1,000 held-out model-generated negatives at

Table 3: Random-key null diagnostic. Each source contains 2,000 texts evaluated under 100 keys. Confidence intervals are descriptive because the same texts are reused across keys.

Null source	Nominal α	Trials	FP	Empirical FPR	95% CI
Natural	10^{-2}	200,000	1,630	8.15×10^{-3}	$[7.76, 8.55] \times 10^{-3}$
Natural	10^{-3}	200,000	160	8.00×10^{-4}	$[6.81, 9.34] \times 10^{-4}$
Natural	10^{-4}	200,000	17	8.50×10^{-5}	$[4.95, 13.61] \times 10^{-5}$
OPT clean	10^{-2}	200,000	1,593	7.97×10^{-3}	$[7.58, 8.36] \times 10^{-3}$
OPT clean	10^{-3}	200,000	149	7.45×10^{-4}	$[6.30, 8.75] \times 10^{-4}$
OPT clean	10^{-4}	200,000	14	7.00×10^{-5}	$[3.83, 11.74] \times 10^{-5}$

nominal level $\alpha = 0.01$ and frozen before evaluation on model-generated and natural test negatives. The resulting thresholds are 0.9999749660491944 and 0.9983356595039369 for 8-bit and 16-bit payloads, respectively.

BREW. The main runs use OPT-1.3B on C4 RealNewsLike with $T = 300$, $\delta = 2.0$, temperature 1.0, and top-$p = 0.95$. The 8-bit and 16-bit configurations use $\mathrm{BCH}(23, 8, t_{\mathrm{ECC}} = 3)$ and $\mathrm{BCH}(31, 16, t_{\mathrm{ECC}} = 3)$, with block lengths equal to their codeword lengths. The blind two-stage detector searches a shared global offset from -5 to 5. It first estimates the payload from successfully decoded and unmasked block messages, then verifies the designated codewords implied by the estimate. The detection score is the maximum, over offsets, of the number of matching blocks divided by the number of complete blocks.

For each payload setting, the threshold is calibrated on 1,000 held-out model-generated negatives at nominal level $\alpha = 0.01$ and frozen before evaluation. The evaluation uses 5,000 watermarked samples, 5,000 model-null samples, and 5,000 natural-null samples. Calibration uses the least strict observed threshold satisfying the empirical FPR constraint. If no observed score level satisfies the constraint, the next representable floating-point value above the largest calibration score is used. The stored thresholds are 0.16666666666666666 for 8-bit payloads and 0.12500000000000003 for 16-bit payloads. The second threshold is strictly larger than 0.125 and should not be rounded down when reproducing the decision rule. The runs use payload seed 42, data-shuffle seed 2027, and batch size 8.

F Additional Experiments

F.1 Random-Key and Fixed-Key Null Results

Tables 3 and 4 provide the complete numerical results underlying Figure 2. Empirical FPR is defined as the false-positive count divided by the number of null evaluations. In the following tables, R denotes the number of null evaluations, as defined in Appendix B, and is distinct from the effective-position count T_{eff} used in the main text.

The random-key experiment reuses each source's 2,000 texts across 100 independently sampled keys. The reported intervals summarize the displayed evaluation units and should not be interpreted as independent deployment-trial intervals. In the fixed-key experiment, C4 and OpenGen denote natural-text sources, while OPT-1.3B clean and Qwen2.5-3B clean denote unwatermarked model-generated texts. Thresholds are not selected using the reported test samples.

F.2 MPAC Native-Threshold Null Diagnostic

For the MPAC null diagnostic, we use the native MPAC detector with OPT-1.3B, an 8-bit payload, no ECC, base 4, $\gamma = 0.25$, context width 4, and left-hash seeding. The detector applies the one-sided theoretical normal thresholds $z = \Phi^{-1}(1 - \alpha)$, namely 2.3263, 3.0902, and 3.7190 for nominal levels 10^{-2}, 10^{-3}, and 10^{-4}, respectively. No threshold is estimated from the reported test samples. Table 5 gives the complete nine-point result.

Table 4: Fixed deployment-key FPR under domain shift. Thresholds are the theoretical exact-binomial thresholds and are not selected on these test sets.

Null source	Nominal α	R	FP	Empirical FPR	95% CI
C4	10^{-2}	100,000	835	8.35×10^{-3}	$[7.80, 8.93] \times 10^{-3}$
C4	10^{-3}	100,000	75	7.50×10^{-4}	$[5.90, 9.40] \times 10^{-4}$
OPT-1.3B clean	10^{-2}	100,000	851	8.51×10^{-3}	$[7.95, 9.10] \times 10^{-3}$
OPT-1.3B clean	10^{-3}	100,000	71	7.10×10^{-4}	$[5.55, 8.95] \times 10^{-4}$
Qwen2.5-3B clean	10^{-2}	100,000	877	8.77×10^{-3}	$[8.20, 9.37] \times 10^{-3}$
Qwen2.5-3B clean	10^{-3}	100,000	81	8.10×10^{-4}	$[6.43, 10.07] \times 10^{-4}$
OpenGen	10^{-2}	3,000	22	7.33×10^{-3}	$[4.60, 11.08] \times 10^{-3}$
OpenGen	10^{-3}	3,000	10	3.33×10^{-3}	$[1.60, 6.12] \times 10^{-3}$
C4	10^{-4}	1,000,000	65	6.50×10^{-5}	$[5.02, 8.28] \times 10^{-5}$

Table 5: MPAC null behavior under its native theoretical z thresholds. *Generated* denotes unwatermarked model output. Confidence intervals are 95% Clopper–Pearson intervals. These rows diagnose the uncalibrated native statistic and are not controlled-FPR operating points.

Null source	Text type	Nominal α	Threshold z	R	FP	Empirical FPR	95% CI
C4	Natural	10^{-2}	2.3263	100,000	72,058	0.72058	$[0.71779, 0.72336]$
C4	Natural	10^{-3}	3.0902	100,000	27,210	0.27210	$[0.26934, 0.27487]$
OpenGen	Natural	10^{-2}	2.3263	3,000	2,160	0.72000	$[0.70356, 0.73601]$
OpenGen	Natural	10^{-3}	3.0902	3,000	788	0.26267	$[0.24699, 0.27881]$
OPT-1.3B	Generated	10^{-2}	2.3263	100,000	77,427	0.77427	$[0.77167, 0.77686]$
OPT-1.3B	Generated	10^{-3}	3.0902	100,000	34,453	0.34453	$[0.34158, 0.34748]$
Qwen2.5-3B	Generated	10^{-2}	2.3263	100,000	72,294	0.72294	$[0.72015, 0.72571]$
Qwen2.5-3B	Generated	10^{-3}	3.0902	100,000	27,596	0.27596	$[0.27319, 0.27874]$
C4	Natural	10^{-4}	3.7190	1,000,000	79,483	0.079483	$[0.078954, 0.080015]$

Diagnostic interpretation. The native analytic thresholds are anti-conservative for the evaluated complete MPAC detector: at nominal levels 10^{-2} and 10^{-3}, empirical FPRs across the tested null sources range from 0.720 to 0.774 and from 0.263 to 0.345, respectively. At 10^{-4}, one million C4 texts produce 79,483 false positives (7.95×10^{-2}). These results diagnose a mismatch between the native reference threshold and the deployed statistic in this configuration; they neither establish that MPAC cannot be calibrated nor characterize the held-out-calibrated MPAC rows in Table 1. A controlled-FPR comparison must calibrate or otherwise validate the complete verification rule on independent null data.

F.3 C4 FPR Reliability Data

Cross-method reliability. Figure 4 summarizes C4 null reliability under each method's reported operating rule. CTWL remains near the nominal levels, whereas BREW is conservative at the stricter levels. MPAC is shown under its native analytic thresholds, not the held-out-calibrated main-comparison setting. Accordingly, this figure is a null-reliability diagnostic, not a matched-operating-point comparison of detection power or payload recovery.

Table 6 lists the complete C4 values for the four methods plotted in Figure 4. Its PIPER and MPAC counts and sample sizes reproduce the C4 entries in Tables 4 and 5, respectively, for cross-method comparison; these repeated entries provide no additional trials. We compute all intervals from the displayed false-positive counts using the same two-sided 95% Clopper–Pearson convention; this makes the uncertainty notation consistent even where a source result file also reports a different interval estimator.

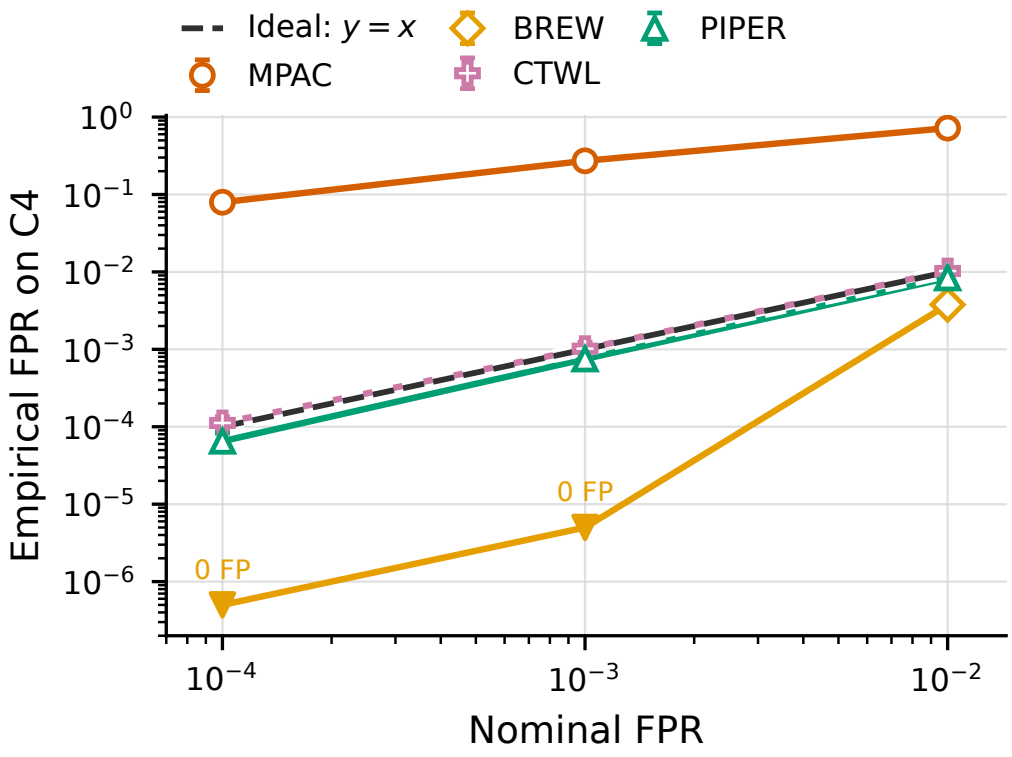


Figure 4: Nominal versus empirical FPR on C4 for MPAC, BREW, CTWL, and PIPER. The dashed diagonal marks ideal calibration ($y = x$). Error bars show 95% Clopper–Pearson intervals. Downward triangles denote zero false positives, drawn at $0.5/R$ for log-scale visibility. MPAC uses native analytic thresholds; other methods use their reported operating rules.

Table 6: Complete nominal-versus-empirical FPR data on C4. All intervals are 95% Clopper–Pearson intervals.

Method	Nominal FPR	R	FP	Empirical FPR	95% CI
MPAC	10^{-2}	100,000	72,058	7.20580×10^{-1}	$[7.17788, 7.23360] \times 10^{-1}$
	10^{-3}	100,000	27,210	2.72100×10^{-1}	$[2.69343, 2.74870] \times 10^{-1}$
	10^{-4}	1,000,000	79,483	7.94830×10^{-2}	$[7.89536, 8.00149] \times 10^{-2}$
BREW	10^{-2}	100,000	378	3.78000×10^{-3}	$[3.40911, 4.18011] \times 10^{-3}$
	10^{-3}	100,000	0	0	$[0, 3.68881] \times 10^{-5}$
	10^{-4}	1,000,000	0	0	$[0, 3.68887] \times 10^{-6}$
CTWL	10^{-2}	100,000	1,027	1.02700×10^{-2}	$[9.65436, 10.91428] \times 10^{-3}$
	10^{-3}	100,000	102	1.02000×10^{-3}	$[8.31762, 12.38075] \times 10^{-4}$
	10^{-4}	1,000,000	112	1.12000×10^{-4}	$[9.22213, 13.47636] \times 10^{-5}$
PIPER	10^{-2}	100,000	835	8.35000×10^{-3}	$[7.79531, 8.93348] \times 10^{-3}$
	10^{-3}	100,000	75	7.50000×10^{-4}	$[5.89967, 9.40042] \times 10^{-4}$
	10^{-4}	1,000,000	65	6.50000×10^{-5}	$[5.01660, 8.28471] \times 10^{-5}$

F.4 Extended Capacity, Quality, and Transfer Results

Metric definitions. For each of the M watermarked samples, let I_i be the presence decision, m_i the true raw payload, and $\widetilde{m}_i$ the tie-zero decoder's diagnostic output, with $\perp$ denoting no decoded payload. Here M denotes the number of watermarked evaluation samples; it is distinct from the null

Table 7: Payload scaling at $T = 500$ and nominal $\alpha = 10^{-2}$. FPR, TPR, CAR, CDA, EMR, and abstention are percentages; goodput is recovered raw payload bits per generated token.

Bits	ECC	Model FPR	Natural FPR	TPR	CAR	CDA	EMR	Abstain	Goodput
8	$\mathrm{BCH}(23, 8, 3)$	0.2	0.6	100.0	99.6	99.6	99.6	0.4	0.0159
16	$\mathrm{BCH}(31, 16, 3)$	0.2	0.6	99.6	98.3	98.7	98.3	1.4	0.0315
32	$\mathrm{BCH}(50, 32, 3)$	0.2	0.6	99.9	70.3	70.4	70.3	26.3	0.0450

evaluation count R used in Appendix B. This diagnostic output is distinguished from the operational verifier output, which is returned only after presence acceptance. Define $C_i = \mathbf{1}\{\widetilde{m}_i = m_i\}$. The reported rates are

$$\mathrm{TPR} = \frac{1}{M}\sum_{i=1}^{M} I_i, \qquad \mathrm{CAR} = \frac{1}{M}\sum_{i=1}^{M} I_i C_i, \qquad \mathrm{CDA} = \frac{\sum_{i=1}^{M} I_i C_i}{\sum_{i=1}^{M} I_i}.$$

CDA is undefined if no sample is accepted. Exact message recovery (EMR) and decoder abstention are

$$\mathrm{EMR} = \frac{1}{M}\sum_{i=1}^{M} C_i, \qquad \mathrm{Abstain} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}\{\widetilde{m}_i = \perp\}.$$

EMR measures exact recovery of all raw payload bits, not bitwise accuracy; CAR additionally requires presence acceptance. Equal displayed EMR and CAR values do not make their definitions interchangeable. Abstain uses all watermarked samples as its denominator and is not the presence-rejection rate $1 - \mathrm{TPR}$. For a k-bit payload and fixed generated length T, the reported decoder goodput is $(k/T)\mathrm{EMR}$ bits per generated token, with EMR expressed as a fraction rather than a percentage; gate-qualified goodput would instead use $(k/T)\mathrm{CAR}$. Model and natural FPR are the fractions of their respective unwatermarked sample sets accepted by the presence test. All rates are displayed as percentages; goodput is not. These definitions also apply to the length sweep in Appendix F.7.

F.4.1 Payload Capacity

Table 7 supplements Figure 3(b) with decoding metrics at $T = 500$, using a fixed generator and presence detector and shared negative sets. As the raw payload grows from 8 to 32 bits, CDA falls from 99.6% to 70.4% and decoder abstention rises from 0.4% to 26.3%. Despite the lower recovery rate, decoder goodput increases from 0.0159 to 0.0450 raw bits per generated token because each successful recovery carries more bits. EMR, abstention, and goodput use the diagnostic decoder convention in Appendix F.4.

F.4.2 Generation Bias and Quality–Reliability Pareto Trade-off

We measure the quality–reliability frontier at $T = 200$ on OPT-1.3B/C4 with an 8-bit $\mathrm{BCH}(23, 8, 3)$ payload. Soft PIPER uses the Cartesian grid $\delta_{\mathrm{pres}} \in \{0, 0.5, 1, 1.5, 2, 3\}$ and $\delta_{\mathrm{pay}} \in \{0, 0.5, 1, 1.5, 2, 3, 4\}$, yielding 42 points; seven hard-parent-mask points use the same payload-bias grid, for 49 settings in total. Each setting contains 1,000 watermarked continuations. The shared reference set contains 1,000 clean OPT-1.3B and 1,000 natural continuations. Watermarked and clean generations use paired prompts and seeds, and quality is scored only on the continuation. For condition $a \in \{\mathrm{wm}, \mathrm{clean}\}$, let $\bar{\mathcal{L}}_i^a$ be the mean continuation-token negative log-likelihood of sample i under the stated evaluator, and let $\bar{\mathcal{L}}^a = M^{-1}\sum_{i=1}^{M} \bar{\mathcal{L}}_i^a$ be its sample mean, with $M = 1000$ paired continuations per setting. We report $\exp(\bar{\mathcal{L}}^{\mathrm{wm}} - \bar{\mathcal{L}}^{\mathrm{clean}}) - 1$ under both the generating OPT-1.3B model and a frozen OPT-6.7B evaluator. The completed external run scored all 49,000 watermarked and 2,000 shared continuations, with zero skipped samples.

Table 8 isolates PIPER's intended no-parent-bias slice. Increasing the child bias improves both presence power and payload recovery, but with a rapidly growing quality cost. The headline point $\delta_{\mathrm{pay}} = 2$ obtains 98.7% TPR and 88.5% CAR at relative perplexity increases of 37.2% under OPT-1.3B and 41.1% under the independent OPT-6.7B evaluator. The external evaluator preserves the same increasing quality-cost trend. In the wider sweep, increasing δ_{pres} from 0 to 3 at $\delta_{\mathrm{pay}} = 2$

Table 8: Quality–reliability sweep for soft PIPER with $\delta_{\text{pres}} = 0$, $T = 200$, and an 8-bit payload. ΔPPL is the paired relative perplexity increase under the generating OPT-1.3B model (Gen.) or independent OPT-6.7B evaluator (Ext.); all entries are percentages.

δ_{pay}	ΔPPL (Gen.)	ΔPPL (Ext.)	TPR	CAR	CDA
0.0	0.0	0.0	1.0	0.0	0.0
0.5	0.2	0.3	13.7	0.1	0.7
1.0	5.8	6.6	63.6	12.8	20.1
1.5	19.2	21.1	94.8	58.8	62.0
2.0	37.2	41.1	98.7	88.5	89.7
3.0	98.9	111.5	99.9	99.5	99.6
4.0	195.7	223.6	100.0	99.9	99.9

Table 9: Cross-model and cross-domain generalization at $T = 300$. FPR, TPR, CAR, CDA, and abstention are percentages. Each row contains 1,000 samples for each of the watermarked, model-null, and natural-null sets.

Model	Domain	Model FPR ↓	Natural FPR ↓	TPR ↑	CAR ↑	CDA ↑	Abstain ↓
OPT-1.3B	C4	0.9	1.0	99.9	98.1	98.2	1.8
OPT-1.3B	OpenGen	0.6	0.4	99.6	98.5	98.9	1.4
OPT-1.3B	LFQA	0.5	1.1	100.0	99.2	99.2	0.8
OPT-6.7B	C4	0.7	1.0	99.0	95.9	96.9	4.1
OPT-6.7B	OpenGen	0.8	0.4	99.2	95.7	96.5	4.3
OPT-6.7B	LFQA	0.4	1.1	99.8	99.5	99.7	0.5
Qwen2.5-3B	C4	0.4	0.8	99.2	97.6	98.4	2.2
Qwen2.5-3B	OpenGen	0.8	0.2	96.0	85.7	89.3	13.3
Qwen2.5-3B	LFQA	1.5	0.7	98.2	89.0	90.6	10.5

raises CAR from 88.5% to 97.0%, while generator-model relative perplexity increase grows from 37.2% to 162.2%. Hard masking at the same payload bias reaches 99.4% CAR but incurs a 336.5% generator-model relative perplexity increase. We therefore retain `soft_p0_m2` as the headline operating point and treat positive parent bias and hard masking only as ablations.

F.4.3 Cross-model and cross-domain evaluation

We evaluate whether PIPER transfers across both model architecture and text domain without changing its operating point. We instantiate the same soft shared-partition configuration on OPT-1.3B, OPT-6.7B, and Qwen2.5-3B, and pair each model with C4 RealNewsLike, OpenGen, and LFQA inputs. Every model–domain cell uses an 8-bit $\mathrm{BCH}(23, 8, 3)$ payload, $T = 300$ generated tokens, $\delta_{\text{pres}} = 0$, $\delta_{\text{pay}} = 2$, the exact-binomial presence test at $\alpha = 10^{-2}$, unique-context counting, and tie-to-zero decoding. Generation in all nine model–domain cells uses top-p sampling with $p = 0.95$ and no top-k truncation. Each cell contains 1,000 watermarked continuations, 1,000 unwatermarked continuations from the evaluated model, and 1,000 natural continuations from the corresponding domain.

Presence detection transfers reliably across the tested grid: TPR ranges from 96.0% to 100.0%, while model-null FPR ranges from 0.4% to 1.5% and natural-null FPR from 0.2% to 1.1%. The largest point estimate, the 1.5% model FPR for Qwen2.5-3B on LFQA, corresponds to 15 false positives out of 1,000 and has a 95% Clopper–Pearson interval of $[0.84\%, 2.46\%]$, which includes the nominal 1% level.

Payload attribution is more sensitive to the model–domain pair. All OPT configurations achieve at least 95.7% CAR, and Qwen2.5-3B reaches 97.6% CAR on C4. In contrast, Qwen2.5-3B falls to 85.7% CAR on OpenGen and 89.0% on LFQA, together with abstention rates of 13.3% and 10.5%, respectively. These results support cross-model and cross-domain generalization of the presence gate at the tested operating point, while exposing a model–domain interaction in payload recovery; the attribution claim should therefore be stated with this boundary rather than as uniform transfer.

Table 10: Independent-evaluator continuation perplexity for the cross-model and cross-domain grid at $T = 300$. Generation uses top-p sampling with $p = 0.95$ and no top-k truncation. PPL is evaluated on paired clean and watermarked continuations; relative PPL increase is $100 \times \left[\exp\left(\frac{1}{M}\sum_{i=1}^{M}(\bar{\mathcal{L}}_i^{\mathrm{wm}} - \bar{\mathcal{L}}_i^{\mathrm{clean}})\right) - 1\right]$. All cells use $M = 1000$ paired clean and watermarked continuations; $\bar{\mathcal{L}}_i^a$ is the mean continuation-token NLL defined above.

Generator	Domain	Independent evaluator	Clean PPL ↓	Watermarked PPL ↓	Rel. PPL ↓
OPT-1.3B	C4	OPT-6.7B	12.90	18.92	+46.7%
OPT-1.3B	OpenGen	OPT-6.7B	14.48	20.48	+41.4%
OPT-1.3B	LFQA	OPT-6.7B	15.92	23.08	+44.9%
OPT-6.7B	C4	OPT-13B	9.47	13.27	+40.2%
OPT-6.7B	OpenGen	OPT-13B	9.60	13.44	+40.0%
OPT-6.7B	LFQA	OPT-13B	12.25	17.58	+43.5%
Qwen2.5-3B	C4	Qwen2.5-7B	12.54	19.00	+51.5%
Qwen2.5-3B	OpenGen	Qwen2.5-7B	8.23	11.84	+43.8%
Qwen2.5-3B	LFQA	Qwen2.5-7B	7.12	11.85	+66.5%

Table 10 evaluates continuation quality for the same nine model–domain cells using a larger held-out model from each generator's family. We score paired clean and watermarked continuations and derive relative PPL increase from their mean NLL difference. The increase ranges from 40.0% for OPT-6.7B on OpenGen to 66.5% for Qwen2.5-3B on LFQA, showing that the quality cost varies across generator–domain pairs.

F.5 Shared and Disjoint Evidence Allocation

For each method and allocation, let $N_{\mathrm{pres},i}$ and $N_{\mathrm{pay},i}$ be the presence-observation and payload-vote counts for text i. Tables 11–13 report the sample means $\overline{N}_{\mathrm{pres}} = M^{-1}\sum_{i=1}^{M} N_{\mathrm{pres},i}$ and $\overline{N}_{\mathrm{pay}} = M^{-1}\sum_{i=1}^{M} N_{\mathrm{pay},i}$ over the M texts in the indicated setting. For PIPER, these per-text counts are N_{pres}^S and N_{pay}^S from Appendix D.3, where S denotes shared evidence; the scheme superscript is omitted in the tables because the method column identifies it.

Table 11 reports the complete fixed-bias comparison used in Section 6.4. Both methods use $\delta = 2.0$; thus, this comparison isolates evidence utilization rather than quality-controlled performance. PIPER approximately doubles the presence sample size and provides about $1.4\times$ the payload sample size of the balanced split, while incurring 7.7–8.4 percentage points more relative perplexity increase.

Table 11: PIPER versus the balanced disjoint baseline at fixed bias $\delta = 2.0$. Rel. PPL is the relative perplexity increase; TPR and CAR are percentages.

T	Method	Rel. PPL	TPR	CAR	$\overline{N}_{\mathrm{pres}}$	$\overline{N}_{\mathrm{pay}}$
100	PIPER shared	$+35.7\%$	**85.6**	**47.1**	95.3	67.9
	Disjoint 50/50	$+28.0\%$	83.2	23.0	47.7	47.5
200	PIPER shared	$+39.2\%$	98.9	**90.5**	193.5	137.5
	Disjoint 50/50	$+31.0\%$	**99.0**	73.4	96.5	96.6
300	PIPER shared	$+43.1\%$	99.3	**97.6**	290.8	206.8
	Disjoint 50/50	$+34.6\%$	**99.7**	92.4	145.3	144.7
500	PIPER shared	$+45.3\%$	**99.8**	**99.6**	481.7	342.8
	Disjoint 50/50	$+36.9\%$	99.7	98.7	240.4	240.1

We additionally evaluate two asymmetric Fu–Russell-style disjoint allocations. The 25/75 setting assigns approximately 25% of effective positions to presence and 75% to payload, whereas 75/25 reverses these roles. Assignment is secret keyed and fixed before the current token is observed. Tables 12 and 13 report the fixed-bias and approximately quality-matched results, respectively.

Table 12: Alternative disjoint allocations at fixed bias $\delta = 2$. TPR and CAR are percentages.

T	Allocation	Rel. PPL	TPR	CAR	$\overline{N}_{\text{pres}}$	$\overline{N}_{\text{pay}}$
100	25/75	+26.9%	37.3	20.5	23.7	71.5
	75/25	+26.8%	95.8	6.3	71.2	23.9
200	25/75	+28.9%	84.6	78.7	48.0	144.3
	75/25	+29.7%	99.1	28.2	144.5	48.4
300	25/75	+31.0%	95.0	93.4	71.9	216.9
	75/25	+32.9%	99.1	54.9	217.2	72.6
500	25/75	+34.3%	97.8	97.4	119.5	359.9
	75/25	+36.2%	99.3	83.7	360.6	120.6

Table 13: Alternative disjoint allocations at approximately quality-matched operating points. The selected δ is the nearest point on the discrete calibration grid.

T	Allocation	δ	Rel. PPL	TPR	CAR	$\overline{N}_{\text{pres}}$	$\overline{N}_{\text{pay}}$
100	25/75	2.5	+39.2%	60.4	45.6	24.0	71.4
	75/25	2.0	+26.8%	95.8	6.3	71.2	23.9
200	25/75	2.5	+41.5%	94.3	93.4	48.4	144.5
	75/25	2.0	+29.7%	99.1	28.2	144.5	48.4
300	25/75	2.5	+45.2%	98.1	97.9	72.4	217.2
	75/25	2.0	+32.9%	99.1	54.9	217.2	72.6
500	25/75	2.0	+34.3%	97.8	97.4	119.5	359.9
	75/25	2.0	+36.2%	99.3	83.7	360.6	120.6

The asymmetric splits make the allocation trade-off explicit. At fixed bias, 25/75 raises CAR from 20.5% at $T = 100$ to 78.7%, 93.4%, and 97.4% at $T = 200, 300, 500$, respectively. Reversing the split strengthens the short-text presence gate (95.8% TPR at $T = 100$) but starves payload recovery, yielding only 6.3% CAR; even at $T = 500$, its CAR is 83.7%. Quality matching selects $\delta = 2.5$ for 25/75 through $T = 300$, giving CAR values of 45.6%, 93.4%, and 97.9%. At $T = 100$, PIPER still has the highest CAR among the three allocations (47.1%) while incurring less relative perplexity increase than quality-matched 25/75 (35.7% versus 39.2%). Thus, moving disjoint evidence between channels trades detection for decoding, whereas PIPER reuses the same effective tokens for both.

F.6 Robustness to Editing, Mixing, and Truncation

The attack study uses 300-token source continuations and the blind-text detector. The transformations below are applied to the watermarked and null continuations before verification.

Lexical transformations. The replacement and insertion conditions use WordNet synonym substitutions. Candidate substitutions are classified by their change in generator-tokenizer length within a local window extending up to 64 characters on either side of the edited word: replacement requires zero change, whereas insertion requires a positive change. Thus, insertion here is a token-increasing synonym substitution, rather than insertion of unrelated text. Word positions are visited in a seeded random order, and an admissible synonym is sampled at each selected position. The requested edit rate is measured against the number of word spans in the original text; the target edit count is the rounded product of this count and the requested rate, with a minimum of one for a nonzero rate. Positions without an admissible candidate are skipped. The achieved rate is the number of completed substitutions divided by the original word-span count. The full edited text is then retokenized for blind verification.

Mixing and truncation. Mixing concatenates a prefix of the watermarked continuation with a prefix of its paired unwatermarked model continuation, keeping the total at 300 tokens. The displayed 50% condition therefore concatenates 150 watermarked tokens and 150 unwatermarked tokens. The corresponding model-null mixture concatenates model-generated and natural prefixes; the natural-null mixture reverses these source roles. Truncation retains the first 200, 100, or 50 tokens of each source continuation, without padding or regeneration.

Table 14: Low-intensity replacement and insertion. Requested and achieved edit rates, TPR, FPR, CDA, and CAR are percentages. Each FPR uses 5,000 negatives from the indicated source.

Attack	Requested	Achieved	TPR	Model FPR	Natural FPR	CDA	CAR
Replacement	5	5.0	99.0	0.76	1.08	91.6	90.7
Insertion	5	5.0	99.0	0.80	1.08	90.0	89.1
Replacement	10	10.0	96.8	1.00	0.76	74.5	72.1
Insertion	10	10.0	96.4	0.94	1.16	69.0	66.5

Table 15: Mixing, truncation, and paraphrase robustness. TPR, FPR, CDA, and CAR are percentages.

Attack	Setting	TPR	Model FPR	Natural FPR	CDA	CAR
Mixing	50% watermarked	83.6	0.78	0.90	42.9	35.9
Truncation	$T = 200$	99.1	0.56	1.02	90.9	90.1
Truncation	$T = 100$	93.0	0.76	1.00	51.8	48.2
Truncation	$T = 50$	60.8	0.72	0.60	12.7	7.7
T5 paraphrase	PAWS model	59.7	0.68	1.00	17.3	10.3

Paraphrase transformation. The T5 PAWS paraphraser rewrites each continuation before it is retokenized by the watermark generator tokenizer and passed to the blind detector. The runner groups sentences into chunks subject to the paraphraser input-length budget and splits an oversized sentence at word boundaries. Decoded chunks are concatenated in their original order. If the complete decoded output is empty, the runner retains the original continuation.

Reported results. Table 14 reports replacement and insertion at 5% and 10% edit rates. At these settings, TPR remains between 96.4% and 99.0%, while CAR decreases from approximately 90% at 5% edits to 66.5–72.1% at 10% edits. This is a selected low-intensity summary: the evaluated 20%/30% edit settings and mixing conditions retaining only 25%/10% watermarked text are omitted from the tables and showed larger performance losses. The displayed results therefore do not establish robustness to those stronger perturbations.

Table 15 reports the selected mixing, truncation, and paraphrase conditions. Retaining half of the watermarked text preserves 83.6% TPR but only 35.9% CAR. Under truncation, 200 observed tokens retain 99.1% TPR and 90.1% CAR, whereas 50 tokens retain 60.8% TPR and 7.7% CAR. T5 paraphrasing retains 59.7% TPR but only 17.3% CDA and 10.3% CAR; its model-null and natural-null FPRs are 0.68% and 1.00%, respectively. Across the displayed conditions, the null FPRs remain near the nominal 1% operating level, while payload recovery degrades earlier and more sharply than presence detection.

F.7 Clean-Text Performance and Length Scaling

The length sweep uses the metric definitions and denominators in Appendix F.4, varying the generated length T while fixing the raw payload at 8 bits. Generation in this sweep uses top-p sampling with $p = 0.95$ and no top-k truncation. Table 16 reports the completed length sweep at nominal $\alpha = 10^{-2}$. Presence detection is already strong at $T = 100$, whereas correct payload attribution requires more evidence: TPR is 95.7% but CAR is only 51.3%. By $T = 300$, TPR reaches 99.9% and CAR reaches 98.1%.

Table 16: Clean-text length scaling for an 8-bit payload at nominal $\alpha = 10^{-2}$. Generation uses top-p sampling with $p = 0.95$ and no top-k truncation. All entries except T are percentages.

T	Model FPR	Natural FPR	TPR	CAR	CDA	EMR
100	1.2	1.2	95.7	51.3	53.6	51.3
200	0.8	1.1	99.7	89.8	90.1	89.8
300	0.9	1.0	99.9	98.1	98.2	98.1
500	0.3	0.6	100.0	99.7	99.7	99.7
1000	0.8	0.9	99.9	99.6	99.7	99.6