WATERMARKS THAT CERTIFY THEIR COST BEFORE DEPLOYMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

A provider deploying an LLM watermark has to commit to two numbers before the first token is generated: how much task accuracy the watermark will cost, and how much output must accumulate before detection is reliable. Existing watermarks answer both only after the fact, by running the watermark and measuring. We formalize the requirement as a *Predictability Standard*, give a certification procedure that applies to any watermark, and give a watermark designed to meet the full standard. HIDDENMARK perturbs generation along a single known direction in vocabulary space, derived from the LM head, so that its effect on every next-token distribution is available in closed form. This lets a deployer *specify* a budget that the controller enforces, and lets HIDDENMARK *certify* the outcome before generation. From one 30-prompt probe the procedure issues two certificates with finite-sample coverage guarantees, validated on HIDDENMARK, KGW and SynthID deployments: the output length needed to reach 95% detection (88% realized coverage at 90% nominal, within a factor of 2.0 over a 996-fold range) and a lower bound on the change in task accuracy. Both are as tight as the outcome can be measured: the detection error matches the disagreement between two independent 100-prompt evaluations of the same deployment. Under the single key a provider actually deploys, the usual closed-form threshold inflates the false positive rate by up to $778\times$; one precomputed scalar brings the inflation to 5–$7\times$. Building the certificates exposed why watermarks destroy task accuracy while perplexity stays flat, and fixed the design: a spread-spectrum direction that bounds how far any token can be promoted. On GSM8K with a model that passes a qualification statistic computed from plain generation, HIDDENMARK matches KGW and SynthID on the accuracy–detection trade-off, and unlike them exposes a budget that is enforced during generation and a detector whose null holds under a deployed key; on a model the statistic rejects, it falls behind KGW.

1 INTRODUCTION

LLM watermarking has moved from research into deployment (Kirchenbauer et al., 2023a; Aaronson, 2022; Dathathri et al., 2024). Deployment asks a question evaluation does not: before any text is generated, what will the watermark cost, and will it work? A provider bound by an agreement on answer quality needs the accuracy cost in advance. A provider who must be able to attribute text needs to know how much of it has to accumulate before detection is reliable. Today both numbers are obtained by running the watermark on labeled data and measuring.

A concrete failure mode. Consider a provider whose service agreement reads "answer accuracy drops by at most 5 points." Which KGW hyperparameters (γ, δ) honor it? The answer depends on the model, the task and the prompt distribution, so it has to be found by grid search for every deployment. Perplexity offers no shortcut; it is a known poor predictor of downstream accuracy under watermarking (Rieff et al., 2026; Hastuti et al., 2025), and we observe the failure at its sharpest. Watermarking Llama-3.1-8B on MMLU-Pro lowers accuracy by 28.3 points ($p < 10^{-4}$, $n = 120$) while the per-token cross-entropy overhead is -0.107 nats: the quality measure reports that the watermarked text is more probable under the model than unwatermarked text. Detection is no easier to anticipate. On AceReason-7B the output needed to reach 95% detection ranges from 2,420

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.

tokens under KGW to 286,750 under SynthID. A provider who deploys without knowing where its configuration falls either cannot attribute its outputs or spends quality it did not need to spend.

The Predictability Standard. We require that every property a deployer commits to be available before generation, as a guarantee rather than a measurement. A database engineer states latency and throughput targets at deployment and the system is built to meet them. A watermark deployer should likewise state an accuracy tolerance and a detection target, and receive a certificate for each before generating. We formalize this along three axes (§3):

- **Quality predictability.** The deployer specifies a budget that the system enforces, and before deployment the system certifies the resulting change in task accuracy with a stated coverage probability.
- **Detection predictability.** The threshold at level α comes from a closed-form null that holds under the single key a provider deploys, and the output length needed for a target detection rate is certified before generation.
- **Deployment reliability.** A statistic computed from plain generation decides whether a model can carry the certificates at all.

None of KGW, Unigram, SWEET or SynthID ships with a pre-deployment certificate on either quantity (§2). The procedure we give issues one for each of them; HIDDENMARK is built so that it meets all three axioms.

Specifying and certifying. A certificate needs a quantity the deployer can set and whose consequences can be computed before generation. HIDDENMARK supplies it by perturbing generation along a fixed direction $\mathbf{w} \in \mathbb{R}^V$ derived from the LM head. Because the head is linear ($\boldsymbol{\ell}_t = \mathbf{W}\mathbf{h}_t$), a signed shift of strength Δ_t moves the logits by exactly $\Delta_t s_t \mathbf{w}$, so the divergence it induces, the signal it leaves in the emitted token, and the largest factor by which it can raise any token's probability are all closed-form functions of the unperturbed distribution and this one vector. The deployer sets a per-token budget; the controller enforces it at every step; and a 30-prompt probe turns the enforced budget into two certificates with finite-sample coverage. Our probe-based predictor of task damage transfers across watermark families, so the costs of KGW and SynthID can be *estimated* the same way. What they lack is a knob whose per-token effect is known: their perturbation at step t depends on the realized prefix, so a deployer can forecast their cost but cannot set it.

What certification exposed. Requiring closed forms made two failures visible that aggregate metrics hide. The first explains the Llama result above. A direction shaped like $\mathbf{Wu}$ is close to Gaussian over the vocabulary, with a few dozen tokens beyond 4 standard deviations; detection reads w_{y_t} of the emitted token, so the controller buys signal by promoting exactly those tokens, and on Llama-3.1-8B with GSM8K the most-promoted token at each step has its probability multiplied by 5.5×10^8 on average. The promoted tokens are off-distribution, the model falls into repetition, and repetition has low perplexity. Suppressing this after the fact does not help. Clipping $\mathbf{w}$, stopping the watermark when the text starts to repeat, and capping the amplification each remove the accuracy loss and each raise the output needed for detection, to as much as 845,259 tokens, because the signal and the damage are produced by the same promoted tokens. Binarizing the direction removes the cause. Every token then carries $|w_v| = 1$, the signal is spread across the vocabulary, amplification is bounded by $e^{2\Delta_t}$, and the closed forms behind the certificates are unchanged. The second failure concerns detection. The closed-form null holds when keys are drawn per text, as in standard evaluation, but a provider signs everything with one key, and under a fixed key the null drifts by an amount that does not shrink with length. The drift inflates the false positive rate by $288\times$ and $778\times$ on two models; one scalar, estimated from the same probe, reduces the inflation to 5–$7\times$.

Contributions.

1. **Certificates for any watermark.** From one 30-prompt probe, a split-conformal procedure issues two pre-deployment certificates with finite-sample coverage, validated on HIDDENMARK, KGW and SynthID: the output length needed for 95% detection (88% realized at 90% nominal, within a factor of 2.0 across a 996-fold range) and a lower bound on accuracy change for qualified models. Their width matches the noise of measuring the outcome itself.

2. **Detection under a deployed key.** We show the closed-form null drifts under a single fixed key by an amount independent of length, and correct it with one precomputed scalar, reducing false-positive inflation from up to $778\times$ to 5–$7\times$.
3. **Design principle.** We identify token amplification as the mechanism by which watermarking destroys task accuracy while leaving perplexity intact, show that post-hoc suppression removes detection with it, and derive a spread-spectrum direction that bounds amplification without giving up the closed forms.
4. **Parity.** Under a common harness on GSM8K, HIDDENMARK sits on the accuracy–detection trade-off alongside KGW and SynthID on a qualified model, while adding an enforced budget and a fixed-key-valid null that neither offers. On a model that fails qualification it falls behind KGW.

2 RELATED WORK

Logit-perturbation watermarks. KGW (Kirchenbauer et al., 2023a) adds δ to the logits of a green list derived from a hash of the preceding token; the realized quality cost depends on model, task and δ and is measured after generation. Follow-up work studies its reliability under attack (Kirchenbauer et al., 2023b), fixes the partition (Zhao et al., 2023), gates on entropy (Lee et al., 2024), and adapts its strength (Liu & Bu, 2024). Unbiased reweighting (Hu et al., 2023; Wu et al., 2024) preserves the expected next-token distribution but still exposes hyperparameters whose effect on a given deployment is measured post hoc. None of these accepts a quality budget as an input or states its detection cost in advance.

Distortion-free watermarks. Exponential-minimum and inverse-transform sampling (Aaronson, 2022; Kuditipudi et al., 2023; Christ et al., 2024) and tournament sampling (Dathathri et al., 2024) keep the marginal distribution of each token unchanged when averaged over keys. They settle the quality half of our standard by construction. Their detection power depends on sampling entropy and is established by running the scheme; we find that on a low-entropy reasoning task SynthID needs two orders of magnitude more output than logit-based schemes to reach the same detection rate.

Evaluating watermark quality. Perplexity is the customary quality metric, and recent evaluations show it does not predict downstream task accuracy under watermarking (Rieff et al., 2026; Hastuti et al., 2025); they recommend measuring task accuracy directly. We agree with the diagnosis and ask for more than measurement: a bound on task accuracy that is available before deployment.

Hidden-state interventions. The LM head's linearity underlies the logit lens (nostalgebraist, 2020; Belrose et al., 2023) and activation steering (Turner et al., 2023; Li et al., 2023; Zou et al., 2023). We use it for a narrower purpose: to make the effect of a watermark on each next-token distribution a closed-form function of one known vector, which is what a certificate needs.

3 THE PREDICTABILITY STANDARD

Let κ be the secret key, p_t the model's next-token distribution at step t, and q_t the watermarked one. A deployer fixes an accuracy tolerance ϵ, a target detection rate $1-\beta$ and a false positive level α before generation.

Axiom 1 (Quality Predictability)**.** *The system accepts a per-token budget that it enforces during generation, and before deployment issues a bound L such that the change in task accuracy satisfies $\Delta\mathrm{acc} \geq L$ with probability at least $1-\gamma$, where γ is stated.*

Axiom 2 (Detection Predictability)**.** *The detection statistic Z_n has a closed-form null F_0 such that $\Pr_{H_0}[Z_n > F_0^{-1}(1-\alpha)] \leq \alpha$* for a single key held fixed across all texts*, and before deployment the system issues an interval that contains, with stated probability, the output length needed to reach detection rate $1-\beta$.*

Axiom 3 (Deployment Reliability)**.** *A statistic computed from plain generation, with no watermarked run, decides whether a model can carry the certificates of Axioms 1 and 2.*

Axiom 1 asks for a bound on task accuracy rather than on a divergence. A per-token divergence budget remains useful as the controller's input, but §6.1 shows it does not bound accuracy on its

own: an output can meet a small divergence budget and lose 28 points. Axiom 2 names the fixed key because that is how watermarks are deployed, and §5.2 shows it changes the null.

Where prior methods stand. KGW, Unigram and SWEET take a logit bias as input and report the resulting cost after generation; their binomial nulls are closed-form, but none ships with either certificate or accepts a budget it enforces. Distortion-free schemes meet the quality half by construction and ship with no detection certificate; SynthID has no closed-form null. None provides a qualification statistic. The probe-based certificates of §4.4 can be issued for any of them (§6.3); what the axioms additionally ask for, an enforced budget and a null valid under a deployed key, is what HIDDENMARK is designed to supply.

4 METHOD

4.1 A KNOWN DIRECTION IN VOCABULARY SPACE

At step t the model produces $\mathbf{h}_t \in \mathbb{R}^d$, logits $\boldsymbol{\ell}_t = \mathbf{W}\mathbf{h}_t$, and $p_t = \mathrm{softmax}(\boldsymbol{\ell}_t/T)$. Once per model we select the $K{=}256$ hidden coordinates with largest LM-head column norm, $\mathcal{S}$, and form $\mathbf{w}_{\mathrm{raw}} = \sum_{j\in\mathcal{S}} \mathbf{W}_{:,j}$. The watermarked logits are

$$\widetilde{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t + \Delta_t\, s_t\, \mathbf{w}, \qquad s_t = \mathrm{PRF}(\kappa, t) \in \{\pm 1\}, \tag{1}$$

with $\Delta_t \geq 0$ set by the controller. The sign depends only on the key and the position, never on the prefix. For the Gaussian-shaped choice $\mathbf{w} = (\mathbf{w}_{\mathrm{raw}} - \mu)/\sigma$ this is exactly a hidden-state shift $\mathbf{h}_t \mapsto \mathbf{h}_t + (\Delta_t/\sigma)s_t\mathbf{m}$, by linearity of $\mathbf{W}$.

Spread-spectrum direction. We use instead the binarized direction

$$w_v = \mathrm{sign}\big(w_{\mathrm{raw},v} - \mathrm{median}(\mathbf{w}_{\mathrm{raw}})\big) \in \{\pm 1\}, \tag{2}$$

which splits the vocabulary in half by the LM-head projection and gives every token $|w_v| = 1$. Structurally this is a fixed green list, as in Unigram (Zhao et al., 2023), whose bias flips sign with the keyed position; the sign flip is what makes the expected shift zero over positions and the null closed-form under a centred statistic. It is no longer a hidden-state direction, so we add it to the logits as in Eq. 1. Nothing in the certificates depends on $\mathbf{w}$ lying in the column space of $\mathbf{W}$: they need only that the logit shift be $\Delta_t s_t \mathbf{w}$ for a known, fixed, bounded $\mathbf{w}$. §5.1 gives the reason for the choice and §6.1 measures it.

4.2 CONTROLLER

The deployer sets a budget cert, which fixes a total cross-entropy allowance $B = N_{\max}\log(\mathrm{cert})$ for a generation of at most $N_{\max}$ tokens. Three mechanisms act at each step. (1) *Entropy gate*: if $H(p_t) < 0.5$ nats, $\Delta_t = 0$. (2) *Per-token search*: otherwise the step's allotment is the unspent allowance divided by the expected number of remaining ungated steps, scaled by a factor in $[0.3, 0.9]$ that grows as the running detection statistic approaches its target; Δ_t is the largest value in $[0, \Delta_{\max}]$ whose expected overhead $\mathbb{E}_{q_t}[-\log p_t(y)] - H(p_t)$ stays within the allotment, found by bisection on the already-computed p_t. (3) *Hard cap*: once the realized overhead $\sum_t(-\log p_t(y_t) - H(p_t))$ over watermarked steps reaches B, $\Delta_t = 0$ for the rest of the text.

4.3 DETECTION

Given tokens $y_{1:n}$, the key, $\mathbf{w}$, and one scalar c (below), the verifier computes

$$\bar{Z}_n = \frac{\sum_{t=1}^{n} s_t\,(w_{y_t} - c)}{\sqrt{\sum_{t=1}^{n}(w_{y_t} - c)^2}} \tag{3}$$

and flags the text when $\bar{Z}_n > z_{1-\alpha}$. It needs no model and no prompt. The centring constant $c = \mathbb{E}_{\bar{p}}[w]$ is estimated once, from the plain generations of the probe below, and shipped with $\mathbf{w}$. With $c = 0$ this is the statistic of prior hidden-state watermarking; §5.2 shows why c is needed under a deployed key.

4.4 CERTIFICATES FROM A PROBE

Before deployment the provider runs the watermark on $N{=}30$ prompts from the target distribution, alongside plain generation on the same prompts. Two statistics come out of the probe: the length inflation $\iota = \bar{n}_{\mathrm{wm}}/\bar{n}_{\mathrm{plain}}$ and the per-text statistics $\{\bar{Z}^{(i)}\}$. Pooling k texts under Eq. 3 and Proposition 3 gives the smallest pooled length $\hat{n}_{95}$ at which $\bar{Z}$ exceeds the null's zero-false-positive threshold in 95% of resamples. We then calibrate two split-conformal predictors on a held-out collection of deployments (model, task, configuration): one maps $\log \hat{n}_{95}$ from the probe to the full-deployment value, the other maps $\log \iota$ to the change in task accuracy. For a new deployment each returns an interval with finite-sample coverage $\geq 1-\gamma$ under exchangeability. The accuracy certificate is issued only if the model passes the qualification statistic of §4.5 and the probe's inflation lies below a gate; otherwise the deployer is told to change the configuration.

4.5 QUALIFICATION

Axiom 3 asks for a statistic, computed from plain generation alone, that says whether a model can carry the certificates. From the probe's plain generations over prompts $\mathcal{P}$ we form the pooled token distribution $\bar{p} = \mathbb{E}_t[p_t]$ and

$$\rho_W = \frac{\mathbb{E}_t\big[\mathrm{Var}_{p_t}(\tilde{w})\big]}{\mathrm{Var}_{\bar{p}}(\tilde{w})} \in [0,1], \qquad \tilde{w} = \mathbf{w} - \mathbb{E}_{\bar{p}}[\mathbf{w}], \tag{4}$$

the share of the variance of $\tilde{w}$ that lies within steps rather than between them (the bound is the law of total variance). The expected per-step signal is proportional to $\mathrm{Var}_{p_t}(\tilde{w})$, so a small ρ_W means the signal is concentrated in few steps and detection varies strongly across prompts. We qualify a model when $\rho_W \geq 0.985$. The centring constant c of Eq. 3 is the same $\mathbb{E}_{\bar{p}}[\mathbf{w}]$, computed from the same probe.

5 THEORY

Theorem 1 (Exact per-token divergence)**.** *With $\delta_v = \Delta_t s_t w_v$, the per-token divergence is $D_{\mathrm{KL}}(p_t \| q_t) = \log \mathbb{E}_{p_t}[e^{\delta}] - \mathbb{E}_{p_t}[\delta]$, computable in $O(V)$ from p_t for any $\mathbf{w}$.*

5.1 AMPLIFICATION

Proposition 1 (Amplification of a single token)**.** *The largest factor by which the watermark raises any token's probability at step t is*

$$A_t = \max_v \frac{q_t(v)}{p_t(v)} = \frac{e^{\Delta_t \max_v s_t w_v}}{\mathbb{E}_{p_t}[e^{\Delta_t s_t w}]}.$$

For a partition direction $w \in \{\pm 1\}^V$, $A_t \leq e^{2\Delta_t}$. For a greenlist with bias δ and green mass $g = p_t(G)$, $A_t = e^{\delta}/(1-g+ge^{\delta}) \leq 1/g$. For a z-scored direction whose largest entry is M standard deviations, A_t can reach $e^{\Delta_t M}/\mathbb{E}_{p_t}[e^{\Delta_t s_t w}]$ and is unbounded in M.

Proof sketch. $q_t(v)/p_t(v) = e^{\delta_v}/\mathbb{E}_{p_t}[e^{\delta}]$; the maximum is at the largest δ_v. For $w \in \{\pm 1\}$, $\delta_v \leq \Delta_t$ and $\mathbb{E}_{p_t}[e^{\delta}] \geq e^{-\Delta_t}$. The greenlist case is direct. □

The detection statistic reads w_{y_t} of the emitted token. Under a Gaussian-shaped $\mathbf{w}$ most of the attainable signal sits in a few tokens with large $|w_v|$, so the controller buys signal by driving A_t up for exactly those tokens. A partition direction spreads the signal over half the vocabulary and bounds A_t by the step's own strength. The divergence of Theorem 1 is an average over p_t; A_t is a maximum over the vocabulary, and the two can differ by orders of magnitude.

5.2 THE NULL UNDER A DEPLOYED KEY

Proposition 2 (Fixed-key drift)**.** *Fix the key, so that $s_{1:n}$ is a fixed sign sequence, and let the text be unwatermarked. With $c = 0$, $\mathbb{E}[\sum_t s_t w_{y_t}] = \sum_t s_t \, \mathbb{E}_{p_t}[w]$. When $\mathbb{E}_{p_t}[w] \approx \mu_w \neq 0$ for most t,*

the mean of $\bar{Z}_n$ is approximately $\mu_w \sum_t s_t / \sqrt{n\,\overline{w^2}}$, which is $O(1)$ and does not shrink as n grows. Centring at $c = \mu_w$ removes this term.

Proof sketch. Linearity of expectation, followed by $\sum_t s_t = O(\sqrt{n})$ for a pseudorandom sign sequence, so the ratio to the $\sqrt{n}$ normalization is constant in n. □

Averaging over keys, as evaluation protocols do when they draw a fresh key per text, makes $\sum_t s_t$ symmetric around zero and hides the drift; a deployed key does not. The drift matters because $\mathbb{E}_{\bar{p}}[w]$ is far from zero in practice: a z-scored direction is centred on the uniform distribution over the vocabulary, a median-split direction is balanced over the vocabulary, and the model's output distribution is neither.

Theorem 2 (Tail bound)**.** *Under H_0, conditionally on $y_{1:n}$ with the signs Rademacher and independent of the tokens,* $\Pr[\bar{Z}_n > z] \le e^{-z^2/2}$.

Theorem 2 is the usual Hoeffding bound and holds over the randomness of the key. Proposition 2 concerns the complementary situation a provider faces, one key and many texts, and is the reason Eq. 3 carries c.

Proposition 3 (Concatenation)**.** *Because s_t depends only on the key and the position, texts generated on disjoint position ranges under one key can be pooled, and the pooled numerator and denominator in Eq. 3 are sums of the per-text ones.*

6 Experiments

Setup. Models: AceReason-Nemotron-7B and Llama-3.1-Nemotron-Nano-8B, which pass the qualification statistic ($\rho_W \ge 0.985$), and Llama-3.1-8B-Instruct, which does not ($\rho_W = 0.970$) and serves as a stress case. Tasks: GSM8K (exact match), IFEval (strict instruction following), MMLU-Pro (multiple choice) and HumanEval (pass@1, executed). Every run pairs each prompt with a watermarked and a plain generation through the same decoding loop and seed; accuracy differences are tested with the exact McNemar test and reported with paired bootstrap intervals. Sampling temperature is 0.9 and the generation cap is raised until fewer than 30% of generations reach it; runs above that rate are excluded, because truncation clips both arms and biases the difference. Baselines use MarkLLM (Pan et al., 2024) under the same prompts, scorer and end-of-sequence set. Detection is compared by the pooled output needed to reach 95% detection at the zero-false-positive threshold of each run's own plain texts, one rule for every method.

6.1 The cost is invisible to the quality metric

With a Gaussian direction, Llama-3.1-8B on MMLU-Pro loses 28.3 points ($p < 10^{-4}$, $n = 120$) while its realized per-token cross-entropy overhead is -0.107 nats. The format-compliance rate falls from 79% to 19%; watermarked outputs loop, and the tokens they repeat sit at the 100th percentile of $|w_v|$ (frequency-weighted mean at the 89th percentile against the 78th for plain text). On GSM8K under the same configuration (Table 1), the per-step amplification of Proposition 1 averages 5.5×10^8 while the average divergence stays small, the gap between a maximum and a mean that the proposition predicts. Across 43 (model, task, configuration) cells that record a realized divergence, the divergence explains none of the variance in accuracy change out of sample (-6.8%, worse than predicting the mean), and its rank correlation with the change is weak and not significant (Spearman -0.279, $p = 0.07$).

Table 1 shows what happens when the damage is suppressed after the fact. Clipping $\mathbf{w}$, stopping the watermark on repetition, and capping A_t at 100, 10 or 2 each remove the accuracy loss and each raise the output needed for detection, up to 845,259 tokens. The signal and the damage are carried by the same promoted tokens, so removing one removes the other. The partition direction removes the cause: amplification falls from 5.5×10^8 to 4.1×10^3, the loss is no longer significant, and detection needs 3,905 tokens, fewer than any suppression and comparable to the unconstrained run.

Table 1: Suppressing amplification after the fact removes detection with it (Llama-3.1-8B, GSM8K, cert 1.4, n=100). Rows 2–6 each reduce the damage and raise the output needed for detection together. The partition direction bounds amplification by construction and keeps detection at the level of the unconstrained run. Mean A_t is the per-step $\max_v q_t(v)/p_t(v)$ averaged over watermarked steps; for the first row it is measured on an independent n=100 repeat of the same configuration (Δacc -20.0). Tokens to 95% TPR uses the rule of Table 2.

Configuration	Δacc (pp)	mean A_t	length	Tokens to 95% TPR
Gaussian $\mathbf{w}$, no intervention	-21.0^*	5.5×10^8	3.12×	3,012
clip $\lvert w_v \rvert \leq 1.5$	-6.0	–	0.91×	4,256
repetition guard	-3.0	–	1.26×	4,619
cap $A_t \leq 100$	-2.0	88.2	1.11×	6,777
cap $A_t \leq 10$	$+3.0$	9.8	1.06×	16,811
cap $A_t \leq 2$	$+1.0$	2.0	0.94×	845,259
Partition $\mathbf{w}$ (Eq. 2)	-5.0	4.1×10^3	0.94×	3,905

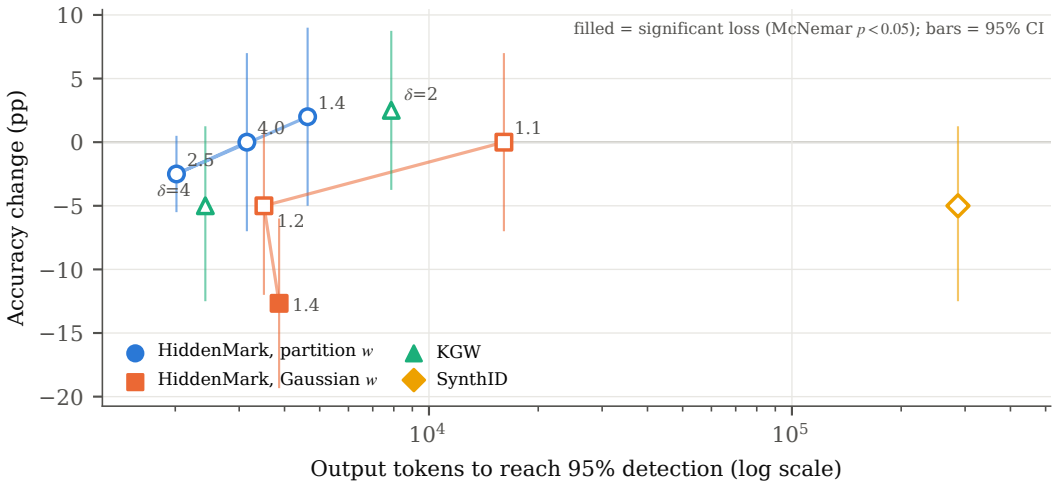

Figure 1: Accuracy–detection trade-off on AceReason-7B / GSM8K under one harness. Each point is a configuration; lines join HIDDENMARK budgets in increasing order. Filled markers mark a significant accuracy loss; bars are 95% bootstrap intervals. The partition direction occupies the region KGW reaches, with far less loss than the Gaussian direction at the same budget; SynthID keeps accuracy at two orders of magnitude more output.

6.2 PARITY WITH EXISTING WATERMARKS

Table 2 and Figure 1 place HIDDENMARK against KGW and SynthID on AceReason-7B. On AceReason-7B the partition direction's accuracy change is not significant at any budget we test; at cert 2.5 with $n = 400$ it is -2.5 points (95% CI $[-5.5, +0.5]$, $p = 0.13$). Raising the budget from 1.4 to 2.5 reduces the output needed for 95% detection from 4,631 to 2,015 tokens, fewer than KGW at δ=4 (2,420 tokens at -5.0 points); raising it further to 4.0 gives no additional gain (3,150 tokens, $+0.0$ points). SynthID loses no significant accuracy but needs 286,750 tokens on this low-entropy task. The Gaussian direction at the same budget loses 12.7 points. On this model HIDDENMARK at cert 2.5 needs less output than the strongest KGW setting and loses fewer points; the accuracy difference between the two is within sampling noise, so we claim parity, not superiority. What it adds is Section 6.3. On Llama-3.1-8B, the model our qualification statistic rejects, parity does not hold: KGW at δ=2 detects with less output and no loss (Table 4).

6.3 CERTIFICATES

We validate both certificates by leave-one-out over 48 deployment cells spanning three models, four tasks, and KGW, SynthID and HIDDENMARK configurations, including the suppression ablations of

Table 2: AceReason-7B on GSM8K, all methods under one harness: same prompts, scorer, sampling temperature and paired plain generations. Δacc is watermarked minus plain accuracy; * marks McNemar $p < 0.05$. The last column is the pooled output needed to reach 95% detection at the zero-false-positive threshold of each run's own plain texts, the same rule for every method. Generation caps are 4,096 tokens for the partition rows and 2,048 otherwise; at most 16% of generations reach the cap in any row.

Method	n	Δacc (pp)	95% CI	Tokens to 95% TPR
KGW, δ=2	80	$+2.5$	$[-3.8, +8.8]$	7,873
KGW, δ=4	80	-5.0	$[-12.5, +1.2]$	2,420
SynthID	80	-5.0	$[-12.5, +1.2]$	286,750
HIDDENMARK, Gaussian $\mathbf{w}$, cert 1.4	150	-12.7^*	$[-19.3, -6.0]$	3,865
HIDDENMARK, partition $\mathbf{w}$, cert 1.4	100	$+2.0$	$[-5.0, +9.0]$	4,631
HIDDENMARK, partition $\mathbf{w}$, cert 2.5	400	-2.5	$[-5.5, +0.5]$	2,015
HIDDENMARK, partition $\mathbf{w}$, cert 4.0	100	$+0.0$	$[-7.0, +7.0]$	3,150

Table 3: Leave-one-out validation of the pre-deployment certificates. Coverage is the fraction of held-out deployments whose measured value falls inside the certificate issued from their 30-prompt probe. Detection intervals are multiplicative on the output length; measured values span 848 to 845,259 tokens. Worst is the largest loss observed among deployments in scope.

Certificate	Scope	Cells	Nominal	Realized	Bound / width
Output length for 95% TPR	all	48	80%	77%	$\times 1.66$ each side
Output length for 95% TPR	all	48	90%	88%	$\times 1.97$ each side
Accuracy change	all	48	90%	92%	≥ -25.0 pp
Accuracy change	$\iota \leq 1.2$	21	90%	90%	≥ -9.0 pp, worst -23.3
Accuracy change	ρ_W and $\iota \leq 1.2$	12	90%	92%	≥ -9.0 pp, worst -9.0

Table 1. The set is uneven: 39 cells are GSM8K and 27 are on the disqualified Llama-3.1-8B. Each cell is certified by a predictor calibrated on the other 47, and we count how often its measured value falls inside (Table 3; Figure 2 in Appendix B).

The detection certificate is informative: at 90% nominal it covers 88% of held-out deployments, within one standard error of nominal at 48 cells, and pins the required output to a factor of 2.0 over a 996-fold range. It transfers: calibrating without any deployment of the held-out (model, task) pair keeps coverage at 88% ($\times 2.01$), and without any deployment of the held-out model at 90% ($\times 2.55$). The accuracy certificate covers 92% of held-out deployments at 90% nominal. Across all deployments the bound is loose (≥ -25.0 points). Gating on the probe's inflation ($\iota \leq 1.2$) tightens it to -9.0 points, but one deployment inside the gate still loses 23.3 points: Llama-3.1-8B on IFEval, with inflation 1.06, a failure the probe cannot see. The qualification statistic removes it. Among deployments that pass both $\rho_W \geq 0.985$ and the inflation gate, the bound is the same -9.0 points and no held-out deployment falls below it; with 11 calibration cells a 90% one-sided bound is the smallest calibration value, so this certificate is only as informative as the calibration set is large, and its 12 cells come from three (model, task) pairs, too few to test transfer to an unseen pair. The two statistics do different work: the probe sets the bound, and ρ_W removes the tail the probe is blind to.

The width is the measurement noise. A certificate cannot be sharper than the outcome it certifies can be measured. Splitting each deployment into two halves and estimating the required output on each (with an analytic Gaussian estimator, to avoid resampling noise) shows that two independent 100-prompt evaluations of the same deployment disagree by a factor of $\times 1.51$ at one standard deviation, the same as the certificate's error ($\times 1.97$ at 90%, $\times 1.51$ at one standard deviation). For accuracy, the 12 qualified deployments inside the gate spread with standard deviation 4.1 points, while the paired sampling error of a single 100-prompt evaluation is about 3.4; the spread of the true change across deployments is therefore about 2.3 points, around a mean of -2.6. The accuracy bound is set by how precisely a 100-prompt evaluation measures accuracy, not by the probe, and tightens with the size of the calibration evaluations.

6.4 Detection under a deployed key

We generate plain text, then score it under 64 to 96 fixed keys and record each key's mean statistic. With a Gaussian direction and no centring, the per-key mean has standard deviation 0.506 on AceReason and 0.722 on Llama, and the worst key raises the false positive rate at $\tau = z_{1-10^{-5}}$ to 2.9×10^{-3} and 7.8×10^{-3}, factors of 288 and 778 over nominal. Centring with c estimated from the 30-prompt probe reduces the spread to 0.147 and 0.169 and the worst-key inflation to 5 and 7. The partition direction shows the same drift: $c = -0.43$, and under the deployment protocol (one key, c from the first 30 plain texts, evaluation on the rest) the uncentred null mean is $+0.29$ and the centred one $+0.06$, with no false positives at τ.

7 Discussion and Limitations

What the certificates promise. Both certificates are conformal: they hold with the stated probability for a deployment exchangeable with those used to calibrate them. A deployer working on a task unlike any in the calibration set gets no guarantee from them, and the accuracy bound is only as tight as the calibration set is dense in the relevant region; ours has 12 qualified deployments inside the inflation gate. The qualification threshold $\rho_W \geq 0.985$ was set on the three models we evaluate and should be recalibrated as models are added. The budget caps the realized overhead of a whole generation ($N_{\max} \log \mathrm{cert}$), not of each token, so a short text can spend more per token than $\log \mathrm{cert}$.

Scope. Baseline comparisons cover GSM8K on two models, one qualified and one not; parity is claimed for the qualified one. On OpenThinker3-7B the watermarked arm reaches the 8,192-token cap in 56–85% of generations against 6–35% for plain generation, so its runs fail our validity rule; its probe inflation (1.47–2.03) lies above the gate, so the procedure would have refused an accuracy certificate for it. HumanEval runs are retained for one model only, as the others fail the truncation rule.

Robustness and security. Position-indexed signs make the statistic robust to substitutions and to deletions with position recovery, and vulnerable to insertion, reordering and paraphrase, which desynchronize the index; we do not evaluate attacks in this paper. A fixed key also leaks: averaging w_{y_t} at each position over many texts signed with one key estimates s_t, which enables removal and spoofing, the analogue of watermark stealing for greenlist schemes (Jovanović et al., 2024). Rotating keys breaks this but reintroduces the per-key drift of §5.2, which centring then has to absorb per key.

8 Conclusion

A watermark can be deployed against a stated accuracy tolerance and detection target when its per-token effect is known in closed form. The probe-based certificates apply to any watermark and are as tight as the outcome can be measured; HiddenMark matches existing watermarks on a qualified model and adds a budget enforced during generation and a null that holds under the deployed key.

AI use statement

We used generative AI tools extensively in this work: to design and run experiments, to write and debug the experiment and analysis code, to analyze results, to generate tables and figures from result files, to search for and verify related work and citation metadata, and to draft and edit the text. All numbers in the paper are produced by code from logged experiment outputs rather than transcribed by hand; citation metadata was checked against the publishers' records. The authors reviewed the experimental design, the analyses and the text, and take responsibility for the final content of this work, including all claims.

Ethics statement

Watermarking supports attribution of machine-generated text and can be misused to track users or to make false attributions. Our contribution makes the false positive rate of a deployed watermark computable and shows that the standard closed-form threshold can underestimate it by orders of

magnitude under a fixed key; we report the correction so that deployments do not overstate their reliability. Our experiments use public models and benchmarks and involve no human subjects.

Reproducibility statement

Section 4 and Appendix C specify the encoder, controller, detector, certificate procedure and every hyperparameter. Appendix A gives the proofs. Every accuracy comparison is paired by prompt and seed and reported with McNemar tests and bootstrap intervals, and runs that exceed a truncation threshold are excluded by a rule stated in advance. Code, generation logs and the scripts that produce every table and figure from those logs will be released.

References

Scott Aaronson. My AI safety lecture for UT effective altruism. *Shtetl-Optimized Blog*, 2022.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *Proceedings of the 37th Annual Conference on Learning Theory (COLT)*, Proceedings of Machine Learning Research, 2024. URL https://proceedings.mlr.press/v247/christ24a.html.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024. URL https://www.nature.com/articles/s41586-024-08025-4.

Rochana Prih Hastuti, Rian Adam Rajagede, Mansour Al Ghanim, Mengxin Zheng, and Qian Lou. Factuality beyond coherence: Evaluating LLM watermarking methods for medical texts. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15129–15147, 2025.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. *arXiv preprint arXiv:2310.10669*, 2023.

Nikola Jovanović, Robin Staab, and Martin Vechev. Watermark stealing in large language models. In *International Conference on Machine Learning (ICML)*, 2024.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning (ICML)*, 2023a.

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. *arXiv preprint arXiv:2306.04634*, 2023b. ICLR 2024 version.

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

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

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, 2023.

Yepeng Liu and Yuheng Bu. Adaptive text watermark for large language models. *arXiv preprint arXiv:2401.13927*, 2024.

nostalgebraist. Interpreting GPT: the logit lens. *LessWrong*, 2020.

Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. MarkLLM: An open-source toolkit for llm watermarking. In *EMNLP 2024 Demo*, 2024. URL https://aclanthology.org/2024.emnlp-demo.7/.

Melanie Rieff, Robin Staab, Thibaud Gloaguen, Stefan Hegselmann, and Martin Vechev. Marking the wrong symptoms: Evaluating LLM watermarks in medical texts. *arXiv preprint arXiv:2607.20462*, 2026.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

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

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. *arXiv:2306.17439*, 2023. URL https://arxiv.org/abs/2306.17439.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A PROOFS

Theorem 1. With $\delta_v = \Delta_t s_t w_v$ and $q_t(v) = p_t(v)e^{\delta_v}/Z$, $Z = \mathbb{E}_{p_t}[e^{\delta}]$,

$$D_{\mathrm{KL}}(p_t \| q_t) = \sum_v p_t(v) \log \frac{p_t(v)}{q_t(v)} = \sum_v p_t(v)\big(\log Z - \delta_v\big) = \log \mathbb{E}_{p_t}[e^{\delta}] - \mathbb{E}_{p_t}[\delta].$$

Both expectations are single passes over the already-computed p_t. No property of $\mathbf{w}$ beyond being fixed and known is used.

Proposition 1. From the same expression, $q_t(v)/p_t(v) = e^{\delta_v}/Z$, which is maximized at $v^\star = \arg\max_v \delta_v$. For $w \in \{\pm 1\}^V$, $\delta_{v^\star} \le \Delta_t$, and since every $\delta_v \ge -\Delta_t$, $Z \ge e^{-\Delta_t}$, so $A_t \le e^{2\Delta_t}$. For a greenlist with bias δ and green mass g, $Z = 1 - g + ge^{\delta}$ and the green tokens attain the maximum, giving $A_t = e^{\delta}/(1 - g + ge^{\delta})$, increasing in δ with limit $1/g$. For a z-scored direction with largest entry M, $\delta_{v^\star} = \Delta_t M$ when $s_t = +1$, and nothing bounds A_t except Δ_t and M.

Proposition 2. Condition on the key, so $s_{1:n}$ is fixed, and on no watermark, so $y_t \sim p_t$ independently of s_t. The numerator of Eq. 3 with $c = 0$ has mean $\sum_t s_t \mathbb{E}_{p_t}[w]$. Writing $\mathbb{E}_{p_t}[w] = \mu_w + \varepsilon_t$, the mean is $\mu_w \sum_t s_t + \sum_t s_t \varepsilon_t$. For a pseudorandom sign sequence $\sum_t s_t = O(\sqrt{n})$, while the denominator is $\Theta(\sqrt{n})$, so the first term contributes $O(1)$ to the mean of $\bar{Z}_n$ for every n. Replacing w by $w - c$ with $c = \mu_w$ removes it; the residual $\sum_t s_t \varepsilon_t$ has mean zero over the key and is what the centred spread in Section 6.4 measures.

Theorem 2. Conditionally on $y_{1:n}$, the summands $s_t(w_{y_t} - c)$ are independent, symmetric and bounded by $|w_{y_t} - c|$; Hoeffding's inequality gives $\Pr[\sum_t s_t(w_{y_t} - c) > z\sqrt{\sum_t (w_{y_t} - c)^2}] \le e^{-z^2/2}$.

B CERTIFICATES

Probe. For a deployment (model, task distribution, configuration) the provider draws 30 prompts, generates each twice through the same decoding loop and seed, once watermarked and once plain, and records lengths and per-text statistics. No labels are used.

Detection certificate. From the probe's per-text statistics and lengths we compute $\hat{n}_{95}$: resample k probe texts with replacement, pool them by Proposition 3, and find the smallest k at which the pooled statistic exceeds the largest pooled plain statistic in 95% of resamples; $\hat{n}_{95}$ is the mean pooled length at that k. A split-conformal regressor from $\log \hat{n}_{95}$ (probe) to $\log n_{95}$ (full deployment), fitted on calibration deployments, returns the interval $\log \hat{n}_{95} \pm q_{1-\gamma}$ where $q_{1-\gamma}$ is the $\lceil (m+1)(1-\gamma) \rceil$-th smallest absolute calibration residual among m deployments.

Accuracy certificate. The same construction with predictor $\log \iota$ and target the accuracy change, one-sided: the certificate is the $\lfloor (m+1)\gamma \rfloor$-th smallest calibration accuracy change among deployments that pass the gate. It is issued only when $\rho_W \ge 0.985$ and $\iota \le 1.2$.

Coverage. Both procedures are split conformal, so coverage holds at level $1 - \gamma$ for a deployment exchangeable with the calibration set. Table 3 reports leave-one-out coverage.

C EXPERIMENTAL DETAILS

Models. AceReason-Nemotron-7B and Llama-3.1-Nemotron-Nano-8B-v1 ($\rho_W = 0.9901, 0.9889$); Llama-3.1-8B-Instruct ($\rho_W = 0.9703$). bf16 or fp16, single A100 80GB, HuggingFace Transformers.

Decoding. Temperature 0.9, no top-p. Both arms of a pair share prompt, seed and decoding loop; the plain arm is the same loop with $\Delta_t = 0$. End-of-sequence set is the union of the tokenizer and generation-config terminators, identical for HIDDENMARK and baselines.

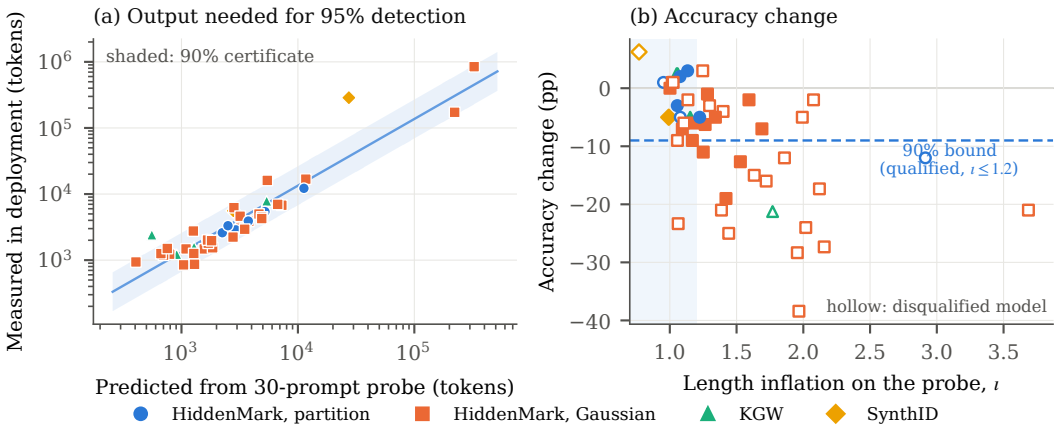

Figure 2: Pre-deployment certificates against deployment outcomes, 48 cells. (a) Output length needed for 95% detection, predicted from a 30-prompt probe versus measured on the full run; the band is the 90% certificate ($\times 1.97$ each side). (b) Accuracy change against the probe's length inflation ι. Filled markers pass the qualification $\rho_W \geq 0.985$; inside the shaded gate $\iota \leq 1.2$ every qualified deployment lies on or above the 90% bound.

Table 4: Llama-3.1-8B-Instruct on GSM8K, same harness and columns as Table 2. This model fails the qualification statistic ($\rho_W = 0.970$) and receives no accuracy certificate; it is shown as a stress case.

Method	n	Δacc (pp)	95% CI	Tokens to 95% TPR
KGW, δ=2	80	+2.5	[−7.5, +12.5]	1,582
KGW, δ=4	80	−21.3*	[−32.5, −10.0]	1,226
SynthID	80	+6.2	[−1.2, +15.0]	5,584
HIDDENMARK, Gaussian **w**, cert 1.4	100	−21.0*	[−31.0, −11.0]	3,012
HIDDENMARK, partition **w**, cert 1.4	100	−5.0	[−13.0, +2.0]	3,905
HIDDENMARK, partition **w**, cert 2.5	100	−12.0*	[−23.0, −2.0]	2,872

Scoring. Reasoning traces are stripped at `</think>` before scoring. GSM8K: last number after "The answer is". MMLU-Pro: last option letter after "The answer is". IFEval: strict instruction-following verifier of lm-evaluation-harness. HumanEval: extracted code executed against the reference tests with a 15 s limit. An unparseable answer is scored incorrect and the parse rate is reported.

Validity. A run is excluded when more than 30% of its generations reach the token cap. Excluded: all OpenThinker3-7B runs (watermarked generations reach the 8,192-token cap in 56–85% of cases, plain in 6–35%), HumanEval for three models, and several MMLU-Pro runs. Truncation clips both arms and biases the accuracy difference in either direction, so excluded runs are not reported as evidence.

Hyperparameters. $K = 256$, $H_{\text{gate}} = 0.5$, $\Delta_{\max} = 5$, ten bisection steps, detection target $\bar{Z} = 6$ for the controller's scaling factor, per-step realized overhead clipped to $[-5, 5]$ nats. $N_{\max}$ is the generation cap: 4,096 for the partition runs, 2,048 for the baselines and the AceReason Gaussian run. Budgets are stated per configuration. KGW $\gamma = 0.5$ with $\delta \in \{2, 4\}$; SynthID with MarkLLM defaults.

Temperature. The search evaluates a shift of Δ_t on the tempered logits ℓ_t/T, while Eq. 1 applies it before temperature, so the realized strength is Δ_t/T ($1.11\Delta_t$ at $T = 0.9$) and the per-step search slightly underestimates the overhead. The hard cap acts on the realized overhead and is unaffected, and Theorem 1 and Proposition 1 hold with Δ_t/T in place of Δ_t.