WOSPEC: WATERMARKED ONLINE SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) increasingly rely on speculative decoding for efficient inference and on watermarking for output provenance. Online speculative decoding improves efficiency by training the draft model on the queries it serves, but when the watermark is embedded through the draft, every draft update can change the watermark carried by the generated text. Therefore, we show that to avoid this problem, *drafter-invariant* speculative decoding is needed, and in such an online setting the generated text is the watermarked text for every draft, including drafts trained on earlier watermarked outputs. In contrast, the watermark strength of watermarked speculative sampling with the standard acceptance rate depends on the draft, which was unstable. We propose WOSpec, a watermarked online speculative decoding framework built on watermarked Gumbel-max list sampling, which naturally has a drafter-invariance property, achieves desriable sampling efficiency, and maintains stable watermark detectability during online training. On Llama-2-7B-chat and Vicuna-13B, the watermark strength of our WOSpec remains unchanged under online training and matches that of a recently proposed watermark-maintaining sampling scheme, while WOSpec accepts 25–28% more tokens per verification step in terms of sampling efficiency.

1 INTRODUCTION

Large language models are increasingly served with two additions: a watermark for provenance (Kirchenbauer et al., 2023) and speculative decoding for speed. Distortion-free watermarks drive token sampling with pseudorandom numbers derived from a secret key, so that the output distribution is unchanged while a detector holding the key can test the text for the watermark signal (Aaronson & Kirchner, 2023; Kuditipudi et al., 2023; Hu et al., 2024; Dathathri et al., 2024). Speculative decoding lets a small draft model propose tokens that the target model verifies in one forward pass, again without changing the output distribution; its speed depends on how often the proposals are accepted (Leviathan et al., 2023; Chen et al., 2023).

The two do not combine freely. Hu & Huang (2024) show that, within their reweighting framework, a single-draft method cannot keep both the full watermark strength and the maximal acceptance rate: watermarking the draft token and verifying it by ordinary rejection sampling keeps the acceptance rate but weakens the watermark, whereas driving the draft and the target with the same pseudorandom numbers keeps the watermark but lowers the acceptance rate. He et al. (2026) characterize this trade-off. Both analyses fix the draft model.

In deployment, however, the draft is not fixed. Online speculative decoding trains the draft on the traffic it serves (Liu et al., 2024), and OnlineSPEC casts this draft–verify–update loop as online learning (Qian et al., 2026), so the service runs a sequence of updated drafts. Online training thus adds a further dimension to the trade-off: how do the watermark strength and efficiency of a method change as its draft model is updated. In the Maintain Sampling Efficiency decoder (MSE) of Hu & Huang (2024), a watermarked token is committed only when it is accepted, so the watermark strength of the generated text changes with the draft. The Poisson Functional Representation decoder (PFR) of concurrent work (Liu et al., 2026b) shares the pseudorandom numbers between the draft and the target and keeps the watermark fixed, but coupling a single draft sequence in this way lowers its acceptance rate. The same work extends PFR to lists of drafts; these methods are analysed for a fixed

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.

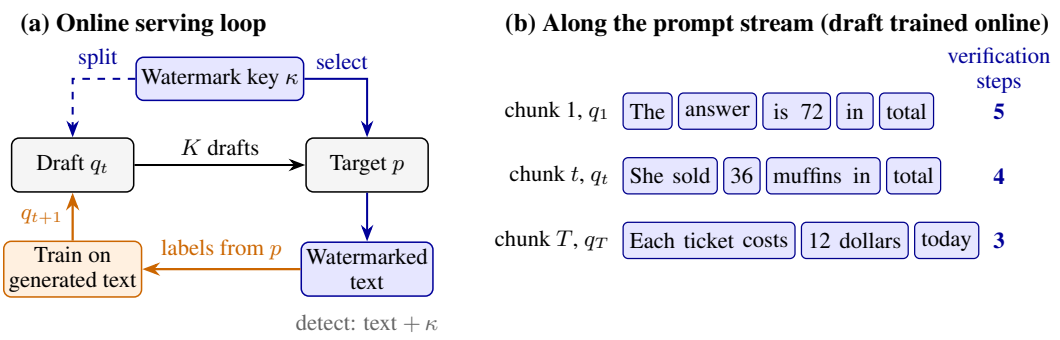

Figure 1: **Online training changes how fast the watermarked text is committed, not the text.** (a) In each chunk, the watermark key κ selects the target token and, through a split of the same pseudorandom numbers, gives the draft random numbers of the K draft sequences; after the chunk, the draft is trained on the generated responses, with labels from a target forward pass;the detector uses the text (prompt and response) and κ. (b) Prompts arrive as a stream and are served by the current draft. Whichever draft serves a prompt, the committed response is the one that Gumbel-max watermarked generation without a draft would produce for it (Proposition 2), so the watermark does not change as the draft is updated. Each box holds the tokens committed by one verification step; a better draft needs fewer steps for six tokens (illustrative).

draft. We ask: *can a watermarked speculative decoding method keep its watermark fixed across draft updates while its efficiency improves with online training?*

The property we need is drafter invariance (Daliri et al., 2025; Rowan et al., 2025): the committed tokens are determined by the target and the random numbers alone, so the draft affects only how many tokens each verification step commits. If these random numbers are those of the watermark, the committed text is a watermarked text for every draft, however the draft was trained. Gumbel-max list sampling (GLS) (Rowan et al., 2025) is a coupling with a conditional form of drafter invariance that recovers the acceptance rate by verifying K draft sequences. Its target, however, samples with the minimum of fresh draft random numbers, which a the watermark detector cannot reconstruct. We reverse the construction: watermarked GLS (WMGLS) starts from the pseudorandom numbers of the Gumbel-max watermark and splits them into K sets of draft random numbers using the order statistics of exponential random variables, so that the drafts keep the law of list sampling while the target samples the watermarked token. WOSPEC serves each chunk of the prompt stream with WMGLS and trains the draft on the generated responses (Figure 1).

Our contributions are as follows.

- **Stable watermarks under online draft updates.** We formulate watermarked online speculative decoding as tracking watermark strength and efficiency along the online stream, a dimension of the strength–efficiency trade-off not covered by fixed-draft analyses (Section 3).

- **Single-draft methods along the stream.** For the Gumbel-max watermark, we show that MSE is not drafter invariant and that its per-token watermark strength depends on the draft and in the worst case falls $\frac{3}{4}D_{\mathrm{TV}}(p, q)$ short of that of the watermark, whereas PFRis drafter invariant (Liu et al., 2026b), but has a lower acceptance rate (Section 4.1).

- **Watermarking a drafter-invariant coupling.** We construct WMGLS by deriving the draft random numbers of conditional GLS from the pseudorandom numbers. It commits the Gumbel-max watermarked text for any drafts, including drafts trained on its own output, and keeps the proposal law of conditional GLS for drafts independent of its random numbers (Section 4.3).

- **Empirical validation.** On Llama-2-7B-chat with three workloads and on Vicuna-13B, WMGLS has the same watermark strength as PFR while accepting 25–28% more tokens per verification step; in a frozen-draft ablation, online training raises this number by 21–31% (Section 5).

2 Related Work

Watermarking language models. Watermarks for language models drive token sampling with pseudorandom numbers derived from a secret key, either by reweighting the next-token distribution (Kirchenbauer et al., 2023) or, in distortion-free schemes, by sampling with the pseudorandom numbers so that the output distribution is unchanged on average over the key (Aaronson & Kirchner, 2023; Kuditipudi et al., 2023; Hu et al., 2024; Dathathri et al., 2024). Detection is a hypothesis test (Li et al., 2025) whose power is limited on low-entropy text (Tsur et al., 2025; Giboulot & Furon, 2024). We use the Gumbel-max watermark and its detector unchanged (Section 3).

Watermarking with speculative decoding. Hu & Huang (2024) study unbiased watermarks within speculative sampling and show that, within their reweighting framework, a single-draft method cannot keep both the full watermark strength and the maximal acceptance rate. He et al. (2026) quantify watermark strength and improve the trade-off by making the acceptance decision pseudorandom. In concurrent work, Liu et al. (2026b) embed the watermark in the coupling itself with Poisson processes generated from the key, for one draft (PFR) and for lists of drafts (MPFR), and show a stopping-time form of drafter invariance, which conditions on the number of tokens committed in a step. These works analyse a given draft model; our focus is a draft trained online on the watermarked text it helps generate.

Multi-draft speculative decoding and drafter invariance. Multi-draft methods verify several candidate continuations per verification step to raise the number of committed tokens (Sun et al., 2023; Miao et al., 2024; Khisti et al., 2025; Hu et al., 2025; Thomas & Pal, 2026); most resample from residual distributions, so the committed token depends on the drafts. Coupling without communication avoids this dependence: Daliri et al. (2025) couple a single draft to the target through shared Gumbel-max random numbers, and Rowan et al. (2025) extend the coupling to lists of draft sequences with Gumbel-max list sampling and its conditional variant for block decoding (see also Li & Anantharam, 2021). WMGLS derives the draft random numbers of conditional list sampling from the pseudorandom numbers of a watermark.

Online draft adaptation. Draft models are usually trained offline and fixed during serving, for example as lightweight heads on target features (Li et al., 2024; 2026) or with acceptance-oriented losses (Samarin et al., 2026). Online speculative decoding instead distils the target into the draft on the positions where the draft errs (Liu et al., 2024), and OnlineSPEC casts the draft–verify–update loop as online learning with an ensemble of learners (Qian et al., 2026); online methods select drafts or tune speculative parameters (Hou et al., 2025; Liu et al., 2026a). These methods do not consider watermarking; we adopt the ensemble update of OnlineSPEC unchanged.

3 Problem Formulation

Online speculative decoding. Let $\mathcal{W}$ be the vocabulary, x a prompt, and $h = (x, y_{<i})$ the history before output token y_i. The target model defines $p(\cdot \mid h)$ and is fixed. The draft model has parameters θ_t while serving chunk t and defines $q_{\theta_t}(\cdot \mid h)$. Following Qian et al. (2026), prompts arrive in chunks $\mathcal{B}_1, \mathcal{B}_2, \ldots$, and the system alternates serving and learning:

$$\mathcal{R}_t = \mathrm{Serve}(p, q_{\theta_t}, \kappa, \mathcal{B}_t), \qquad \theta_{t+1} = \mathrm{Update}(\theta_t, \mathcal{R}_t), \tag{1}$$

where $\mathcal{R}_t$ are the responses served for chunk t and κ is the watermark key. The draft is trained on the responses it served, so the responses of chunk t never inform the draft that served them.

Speculative decoding steps. Within Serve, each step starts at a committed history h. The draft proposes K sequences of L tokens. Sequences that share a prefix share a node u of a draft tree with history h_u (h followed by that prefix), and the active set $\mathcal{S}(u) \subseteq [K]$ contains the sequences that reach u. We write $p_u = p(\cdot \mid h_u)$ for the target and $q_u^{(k)}$ for the draft distribution of sequence k at u (in our experiments, $q_u^{(k)} = q_{\theta_t}(\cdot \mid h_u)$ for all k). The target evaluates p_u at all nodes in one forward pass, and the verifier commits tokens along one path of the tree, ending with a correction token at the first mismatch or a bonus token at a leaf.

Gumbel-max watermark. The Gumbel-max watermark (Aaronson & Kirchner, 2023) computes pseudorandom numbers $U^{\kappa}_{c,v} = \mathrm{PRF}(\kappa, c, v)$ and $S^{\kappa}_{c,v} = -\log U^{\kappa}_{c,v}$ for $v \in \mathcal{W}$ from a secret key κ and the watermark context c (the last three tokens of the history). It commits $Y^{\kappa}_c = Z_{p_c}(S^{\kappa}_c)$, where p_c is the target distribution at the current history and Z_r is the Gumbel-max trick in exponential form (Li & El Gamal, 2018; Rowan et al., 2025):

$$Z_r(S) := \arg\min_{v \in \mathcal{W}} \frac{S_v}{r(v)} \sim r \quad \text{if } S_v \overset{\text{i.i.d.}}{\sim} \mathrm{Exp}(1), \qquad \text{so} \qquad Y^{\kappa}_c = \arg\min_{v \in \mathcal{W}} \frac{S^{\kappa}_{c,v}}{p_c(v)} \tag{2}$$

As is standard (Zhao et al., 2025; Li et al., 2025), we treat the $U^{\kappa}_{c,v}$ over a random key as independent $\mathrm{Unif}(0,1)$ variables, and likewise all other pseudorandom values used below (Appendix A.1), so $Y^{\kappa}_c \sim p_c$ and the watermark is distortion-free; given κ, Y^{κ}_c is a deterministic function of p_c and c. We write $W_{\kappa}(p)$ for watermarked generation that applies Eq. (2) autoregressively to the target, without a draft. The detector scores each token y by $U^{\kappa}_{c,y}$, which is $\mathrm{Unif}(0,1)$ for text generated without the key, and needs only the prompt, the response and the key (Appendix E). As in Hu et al. (2024); Hu & Huang (2024), a context repeated within a response uses fallback random numbers and is not scored (Appendix A.1).

Problem: watermark and efficiency along the online stream. For a fixed draft, watermark strength and efficiency trade off (Hu & Huang, 2024; He et al., 2026); under online training, the draft changes after every chunk of Eq. (1), so a watermarked speculative decoding method serves the stream with a sequence of updated drafts, and two quantities evolve along it: the watermark strength of the generated text (the scores $U^{\kappa}_{c,y}$ of its tokens) and the efficiency (the accepted tokens per verification step). We study how they evolve for the Gumbel-max watermark and ask two questions. *(Q1)* How does the watermark strength of a watermarked speculative decoding method depend on the draft, and hence evolve under online training? *(Q2)* Can a decoder keep the watermark strength fixed at that of $W_{\kappa}(p)$ for every updated draft, so that one detector serves the whole stream, while its efficiency improves with online training and exceeds that of single-draft decoding? Section 4.1 answers (Q1) for single-draft methods, Sections 4.2 and 4.3 construct a decoder for (Q2), and Section 5 measures both quantities over the full online stream.

4 Watermarked Online Speculative Decoding

4.1 Drafter Invariance and Single-Draft Methods

Definition 1 (Drafter invariance; Daliri et al., 2025; Rowan et al., 2025, Definition 2)**.** *A speculative decoding method is* drafter invariant *if, for every prompt and every realization of its random numbers R (the pseudorandom numbers, the fallback random numbers and any auxiliary random numbers), the committed response is the same for every choice of the drafts (and of K and L); the drafts determine only how many tokens each verification step commits.*

Watermarked draft, standard verification (MSE). The Maintain Sampling Efficiency decoder (MSE) of Hu & Huang (2024, Section 5.2) samples the draft token $X = Z_q(S^{\kappa}_c)$ from the watermarked draft distribution and then runs standard speculative sampling with the original p and q: it accepts X with probability $\min\{1, p(X)/q(X)\}$ and otherwise samples $[p-q]_+$ without the key.

Shared randomness (PFR). The Poisson Functional Representation decoder (PFR) of Liu et al. (2026b) lets the draft and the target share the pseudorandom numbers; for discrete distributions it is the coupling of Daliri et al. (2025) with $S = S^{\kappa}_c$ (Liu et al., 2026b, Appendix A.1). The draft proposes $X = Z_q(S^{\kappa}_c)$, the target computes $Y = Z_p(S^{\kappa}_c) = Y^{\kappa}_c$, and X is accepted exactly when $X = Y$. At every new context, PFR coincides with the strength-maintaining decoder of Hu & Huang (2024) (Lemma 3).

Proposition 1 (Single-draft methods)**.** *Consider a position with a new context c at which a draft token is verified, with target p and a draft q independent of the decoder's random numbers, and let $D = D_{\mathrm{TV}}(p,q)$ and $\Delta(p) = \sum_v p_v(1-p_v)/(2(1+p_v))$. Let $U_Y = U^{\kappa}_{c,Y}$ be the score of the committed token Y; without the watermark, $\mathbb{E}[U_Y] = \frac{1}{2}$. Then:*

(i) MSE *accepts X with probability $1-D$, and $\max\{0, \Delta(p) - \frac{3}{4}D\} \le \mathbb{E}[U_Y] - \frac{1}{2} \le \Delta(p)$, with equality on the right if $q = p$.* MSE *is not drafter invariant.*

(ii) PFR *accepts X with probability between $(1-D)/(1+D)$ and $1-D$, and $\mathbb{E}[U_Y] - \frac{1}{2} = \Delta(p)$.* PFR *is drafter invariant, and its committed response is the text of $W_\kappa(p)$ for every draft.*

Here $\Delta(p)$ is the watermark strength of $W_\kappa(p)$ at this position (Proposition 4); the proof is in Appendix B.3. Part (ii) restates known results (Daliri et al., 2025; Liu et al., 2026b) except the statement on watermark strength. The two methods sit at the two ends of the trade-off of Hu & Huang (2024), and Proposition 1 shows how each end depends on the draft. MSE keeps the acceptance rate of speculative sampling, but its watermark strength moves with q: a rejected position is resampled without the key, and an accepted token is watermarked under q rather than p. Online updates therefore change its watermark strength; the $\frac{3}{4}D$ gap is a worst case. PFR keeps the watermark strength $\Delta(p)$ for every draft, so online training can change only its acceptance, which is lower than that of MSE by at most a factor $1 + D$: coupling a single draft without communication costs acceptance. Gumbel-max list sampling recovers it with several draft sequences and keeps a conditional form of drafter invariance; WMGLS watermarks it and makes it drafter invariant in the sense of Definition 1.

4.2 Gumbel-Max List Sampling

Gumbel-max list sampling (GLS) (Rowan et al., 2025) extends the coupling of Daliri et al. (2025) to K drafts. At a position with target p and draft distributions $q^{(1)}, \dots, q^{(K)}$, it draws independent draft random numbers $S^{(1)}, \dots, S^{(K)}$, each distributed as S in Eq. (2), and sets

$$X^{(k)} = Z_{q^{(k)}}\big(S^{(k)}\big), \quad k \in [K], \qquad Y = Z_p(M), \quad M_v := \min_{k \in [K]} S_v^{(k)}. \tag{3}$$

Each $X^{(k)}$ follows $q^{(k)}$, and since $M_v \overset{\text{i.i.d.}}{\sim} \mathrm{Exp}(K)$ and the common rate cancels in the $\arg\min$, $Y \sim p$; a proposal is accepted if $Y \in \{X^{(1)}, \dots, X^{(K)}\}$. Conditional GLS, used for block decoding, applies Eq. (3) at every node u of the draft tree, with fresh random numbers $S_u^{(k)}$ for the active drafts $k \in \mathcal{S}(u)$ and the minimum over $\mathcal{S}(u)$. At each node, verification commits the target token and moves to the child labelled by it; if there is none, that token is the correction token, or the bonus token at a leaf (Algorithm 2 in Appendix C). Its target token is thus determined by the minimum of fresh draft random numbers, which the watermark detector cannot reconstruct.

4.3 Watermarked GLS (WMGLS)

A direct way to watermark GLS is to generate each set of draft random numbers from the key. This does not preserve the Gumbel-max watermark, because the target token of Eq. (3) still depends on the minimum over the active drafts, which changes with the drafts and which the detector cannot recompute. The key idea of WMGLS is to reverse the construction: we fix the minimum to the pseudorandom numbers of the watermark, up to a constant factor, so that the target token is the watermarked token Y_c^κ, and generate the draft random numbers from this minimum so that they keep their law in GLS, Lemma 1 makes this possible

Lemma 1 (Order statistics of exponentials)**.** *Let $S^{(1)}, \dots, S^{(J)}$ be independent unit-rate exponentials, M their minimum, O the index attaining it, and $R^{(k)} = S^{(k)} - M$ for $k \neq O$. Then $M \sim \mathrm{Exp}(J)$, O is uniform on $\{1, \dots, J\}$, the $R^{(k)}$ are independent unit-rate exponentials, and M, O and $(R^{(k)})_{k \neq O}$ are mutually independent. Conversely, drawing M, O and $(R^{(k)})_{k \neq O}$ independently from these laws and setting $S^{(O)} = M$, $S^{(k)} = M + R^{(k)}$ yields J i.i.d.* $\mathrm{Exp}(1)$ *variables.*

Construction. At every node u with context c and $J = |\mathcal{S}(u)|$ active drafts, WMGLS applies the converse part of Lemma 1 with the scaled pseudorandom numbers as the minimum:

$$M_v = \frac{S_{c,v}^\kappa}{J}, \qquad S_{u,v}^{(k)} = \begin{cases} M_v, & k = O_v, \\ M_v + R_v^{(k)}, & k \neq O_v, \end{cases} \tag{4}$$

where, independently for every token v, O_v is uniform on $\mathcal{S}(u)$ and $R_v^{(k)} \sim \mathrm{Exp}(1)$. We call O and R the auxiliary random numbers; they are generated from an auxiliary key, the node and its active drafts, and the detector never uses them. The factor $1/J$ gives M_v the rate J of the minimum in GLS. Each active draft proposes $X^{(k)} = Z_{q_u^{(k)}}(S_u^{(k)})$, and the target token is

$$Y(u) = \arg\min_{v \in \mathcal{W}} \frac{S_{c,v}^\kappa}{p_u(v)}. \tag{5}$$

Algorithm 1: Watermarked GLS decoding

Input : prompt $x_{1:n_0}$, maximum length N (prompt plus new tokens), lookahead L, K draft distributions, target p, key κ

Output : continuation $x_{n_0+1:n}$

$n \leftarrow n_0$

while $n < N$ and no EOS has been generated **do**

  root u_0 with path $x_{1:n}$ and active drafts $\mathcal{S}(u_0) \leftarrow \{1, \dots, K\}$

  for $d = 1$ **to** $\min\{L,\ N - n\}$ **do** `// grow the draft tree`

    foreach *node* u *at depth* $d-1$ *with* $\mathcal{S}(u) \neq \emptyset$ **do**

      $S^{\kappa}_{c(u)} \leftarrow$ pseudorandom (or, if repeated, fallback) numbers at u `// Eq. (2); watermark`

      $\{S^{(k)}_u\}_{k \in \mathcal{S}(u)} \leftarrow$ split of $S^{\kappa}_{c(u)}$ by Eq. (4) `// auxiliary key`

      $X^{(k)} \leftarrow \arg\min_v S^{(k)}_{u,v} / q^{(k)}_u(v)$ for each $k \in \mathcal{S}(u)$

      add one child of u per distinct proposal; its active drafts are those that proposed it

  score every node of the tree with one target forward pass

  $u \leftarrow u_0$;

  $Y \leftarrow \arg\min_v S^{\kappa}_{c(u),v} / p_u(v)$ `// =` $Y^{\kappa}_{c(u)}$`; no draft enters`

  while u has a child u' labelled Y and $Y \neq$ EOS **do**

    $x_{n+1} \leftarrow Y$;

    $n \leftarrow n + 1$;

    $u \leftarrow u'$;

    $Y \leftarrow \arg\min_v S^{\kappa}_{c(u),v} / p_u(v)$

  if $n < N$ **then**

    $x_{n+1} \leftarrow Y$;

    $n \leftarrow n + 1$ `// correction or bonus`

return $x_{n_0+1:n}$

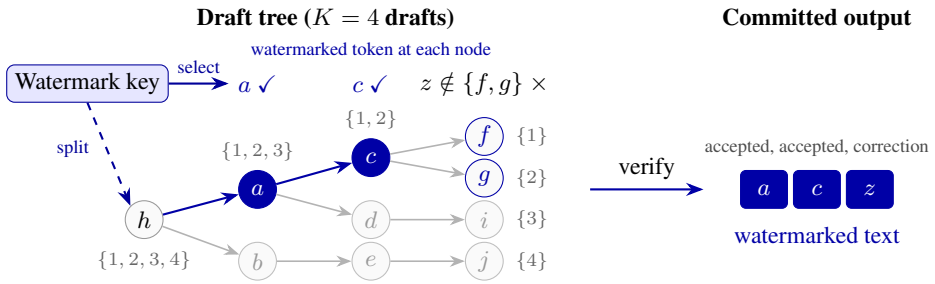

Figure 2: **In WMGLS, the watermark key selects every committed token; the drafts decide only how far verification proceeds.** One verification step from the committed history h with four draft sequences (a, c, f), (a, c, g), (a, d, i), (b, e, j); the sets list the drafts that pass through each node. At every node, the key gives each candidate token a pseudorandom number. *Split*: for each token, one randomly chosen draft among those at the node receives this number divided by the number of such drafts, and the others receive that value plus an independent random amount, so every draft still samples as in list sampling. *Select*: the committed token is chosen from the pseudorandom numbers and the target probabilities alone, without looking at the drafts. Verification follows the drafts while they contain the selected token: a and c are accepted, and z, which no draft proposed, is committed as the correction.

The drafts thus see the draft random numbers of Eq. (4), whereas the target sees only the pseudorandom numbers of the watermark. At a repeated context, the fallback random numbers of the history replace S^{κ}_c in Eqs. (4) and (5). With a single draft, $S^{(1)}_u = S^{\kappa}_c$ and WMGLS reduces to PFR. Algorithm 1 gives the decoding loop, and Figure 2 illustrates one step.

Proposition 2 (Watermarked GLS)**.** *For Algorithm 1, the following hold.*

(i) (Acceptance of GLS.) *If the drafts are independent of the random numbers of the decoder, the proposals of Algorithm 1 at every node, given the path to it, have the same law as in conditional GLS, so the number of tokens committed per verification step has the law of GLS.*

(ii) (Watermarked text.) *For every key κ, number of drafts K, lookahead L and draft,* WMGLS *commits Y_c^κ at every position with a new context c. It is drafter invariant, and its committed response is the text of $W_\kappa(p)$ with the same fallback; the drafts determine only how many tokens each verification step commits.*

Proof sketch. Appendix A gives the proofs. Part (i) follows from the converse part of Lemma 1, applied to independent drafts. Part (ii) rests on three observations. *(a)* The target token $Y(u)$ depends only on the key, the context and p_u, not on the drafts or the active set. *(b)* $Y(u)$ is also the target token of conditional GLS for these draft random numbers: draft O_v attains the minimum and $R_v^{(k)} \geq 0$, so $\min_{k \in \mathcal{S}(u)} S_{u,v}^{(k)} = S_{c,v}^{\kappa}/J$, and the factor $1/J$ does not change the $\arg\min$. WMGLS therefore verifies exactly as conditional GLS does. *(c)* Verification commits only target tokens, along a path of committed histories: it commits $Y(u)$, continues at the child labelled $Y(u)$ if an active draft proposed it, and otherwise ends the step. By induction over positions, every committed token is that of $W_\kappa(p)$. Part (ii) holds for every realization of the random numbers and needs no independence. By (b), WMGLS also keeps the active-set filtering of conditional GLS, whereas the invariant variant of Rowan et al. (2025, Proposition 6) takes the minimum over all K drafts and loses acceptance. Table 3 in Appendix C.2 compares the constructions.

Implication for online learning. In WOSPEC, the draft that serves chunk $t+1$ is trained on the responses of chunks $1, \ldots, t$, which were generated with the key; the draft therefore depends on κ and is no longer independent of the random numbers. Part (ii) does not need this independence: Definition 1 fixes the realization of R and allows any drafts, including drafts that depend on R through earlier responses. In principle, the generated responses therefore equal the output of $W_\kappa(p)$ on the same prompts for any online update rule, and every detector that uses only the prompts, the responses and κ returns the same result (Appendix A.5). Part (i) does not carry over: once the draft depends on the key, its proposals need not follow the law of conditional GLS, so the efficiency, unlike the text, is not guaranteed to equal that of unwatermarked GLS (Appendix A.1). The draft is trained towards the target distribution at the visited histories (Appendix D), so the key enters it only through which histories are visited; its weights nevertheless depend on κ.

5 EXPERIMENTS

Our experiments address the two questions of Section 3 in three parts. First, we trace the watermark strength and the efficiency of each method along the online stream (Section 5.2, Figure 3). Second, we compare WMGLS with the single-draft methods and unwatermarked GLS over the full online stream, on Llama-2-7B-chat and on the larger Vicuna-13B-v1.3 (Table 1). Third, we isolate the contribution of online training with a frozen draft (Table 2).

5.1 SETUP

Models and workloads. We use Vicuna-7B-v1.3 and Llama-2-7B-chat (Touvron et al., 2023) as targets, and Vicuna-13B-v1.3 as a larger target. Each target is paired with an EAGLE-3 draft head (Li et al., 2026) trained from scratch. Following Liu et al. (2024); Qian et al. (2026), we use GSM8K (Cobbe et al., 2021), CodeSearchNet-Python (Husain et al., 2019) and Finance-Alpaca (Taori et al., 2023). For every target and workload, we train one draft head on 800 warm-up examples and share it across all methods and seeds; 4,000 disjoint prompts then arrive in 100 chunks of 40, and the draft is updated after every chunk with the ensemble recipe of Qian et al. (2026) (Appendix D). Each method trains on its own generated responses. We run three seeds, each fixing the stream order, the training randomness and the watermark key. Unless stated otherwise, we report Llama-2-7B-chat; the corresponding results on Vicuna-7B-v1.3 are in Appendix F, and per-window results in Appendix H.

Methods. All methods are speculative decoding methods on the same serving stack and share the target, the initial draft head and the online protocol; WOSPEC is WMGLS under this protocol. MSE

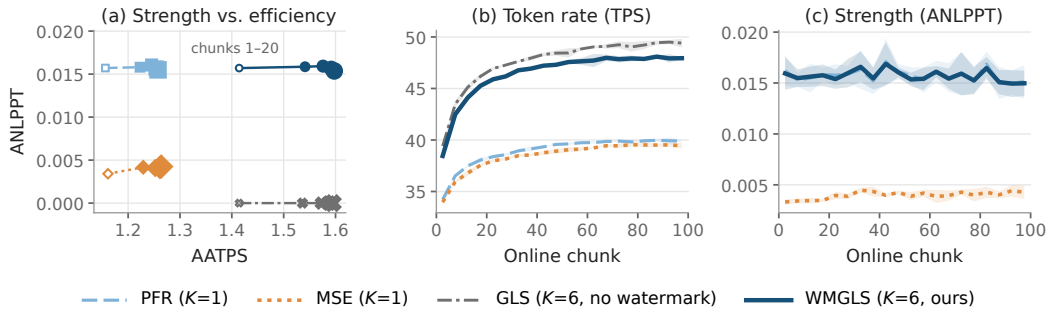

Figure 3: **Online training raises the efficiency of WMGLS while its watermark strength stays fixed** (Llama-2-7B-chat; mean over the three workloads and three seeds). (a) Watermark strength (ANLPPT) against accepted tokens per verification step (AATPS); each marker is a 20-chunk window, from chunks 1–20 (open) to 81–100 (largest). WMGLS and PFR move right at constant strength, and WMGLS reaches the AATPS of unwatermarked GLS (drawn at zero strength); MSE gains efficiency but keeps less than a third of the watermark strength of PFR. (b) The token rate of WMGLS stays above that of both single-draft methods, and its lead over PFR grows with online training. (c) The ANLPPT of WMGLS stays level as the draft is updated and coincides with that of PFR, whose curve lies under it ; that of MSE is lower and varies with the draft. In (b,c), bins of five chunks with $\pm$ one s.d. across seeds; the vertical axes do not start at zero.

Table 1: **Efficiency and watermark strength over the full online stream**; mean over three seeds of 100 chunks (standard deviations in Tables 7 and 10). PFR and MSE verify one draft sequence per step, GLS and WMGLS six ($L = 5$); GLS is unwatermarked. Vicuna-7B-v1.3 is reported in Table 8.

	Llama-2-7B-chat									Vicuna-13B-v1.3		
	GSM8K			CodeSearchNet-Python			Finance-Alpaca			GSM8K		
Method	AATPS	TPS	ANLPPT	AATPS	TPS	ANLPPT	AATPS	TPS	ANLPPT	AATPS	TPS	ANLPPT
PFR	1.265	39.77	0.008	1.224	38.95	0.018	1.192	38.08	0.021	1.220	32.45	0.025
MSE	1.275	37.96	0.004	1.227	39.15	0.004	1.199	38.22	0.004	1.228	32.63	0.005
GLS	1.594	48.71	N/A	1.540	47.53	N/A	1.489	46.55	N/A	1.551	38.18	N/A
WMGLS	1.595	47.82	0.008	1.548	46.62	0.018	1.490	44.91	0.021	1.557	37.56	0.025

and PFR are the two single-draft methods of Section 4.1: MSE attains the acceptance rate of standard speculative sampling, and PFR, which is drafter invariant, carries the full watermark strength of $W_\kappa(p)$; they are the references for efficiency and for strength among watermarked methods. GLS is unwatermarked conditional list sampling with the same number of drafts and lookahead as WMGLS, the closest unwatermarked reference. PFR and MSE use one draft sequence, GLS and WMGLS use $K = 6$, and all use lookahead $L = 5$; these values lie within the ranges used for GLS by Rowan et al. (2025) and are fixed across workloads. Appendix F reports a sweep over K. As in Rowan et al. (2025), we compare speculative decoding methods only; we do not run the concurrent MPFR (Liu et al., 2026b), whose AATPS is close to that of GLS. Sampling uses temperature 1 over the full vocabulary, at most 1,024 new tokens, and one request at a time on one A100-40GB GPU (Appendix D).

Metrics. Following Hu & Huang (2024); He et al. (2026), we measure efficiency by the average number of accepted tokens per verification step (AATPS), counting the correction or bonus token, and by the token rate in tokens per second (TPS), both computed per response and averaged over responses as in Qian et al. (2026); TPS excludes training time. Since TPS also depends on the implementation and the hardware (Appendix C.3), AATPS is our primary measure of efficiency. Following Hu & Huang (2024), we measure watermark strength by the average negative log p-value per token (ANLPPT) on the first 128 output tokens of each response of at least that length, skipping repeated contexts; the p-value is a Chernoff bound under the null hypothesis (Appendix E). Detection rates and log perplexity are in Appendix F.

Table 2: **Frozen versus online draft** for WMGLS on Vicuna-7B-v1.3; mean over the same two seeds for both rows (standard deviations in Table 9).

	GSM8K			CodeSearchNet-Python			Finance-Alpaca		
Draft	AATPS	TPS	ANLPPT	AATPS	TPS	ANLPPT	AATPS	TPS	ANLPPT
Frozen	1.275	37.82	0.039	1.128	33.88	0.066	1.108	33.34	0.065
Online	1.545	45.54	0.039	1.476	43.66	0.066	1.425	42.56	0.065

5.2 Watermark strength during online training

The watermark strength of WMGLS stays fixed. The ANLPPT of WMGLS remains between 0.0154 and 0.0159 across the five 20-chunk windows and closely matches that of PFR (Figure 3(c)). This is consistent with the invariance result in Section 4.3; finite-precision differences between the implementations are examined in Appendix A.5. For MSE, ANLPPT increases from 0.0034 in the first window to 0.0042 in the last, but remains below a third of that of PFR.

Online training raises the efficiency of every method. Averaged over the three workloads, the AATPS of WMGLS rises from 1.41 in the first 20-chunk window to 1.60 in the last, with about two thirds of the gain within the first 40 chunks (Figure 3(a)). Its token rate rises by 13%, and those of PFR and MSE by 9–10% (Figure 3(b)). The lead of WMGLS over PFR in token rate thus grows from 16% to 20%, and its AATPS stays within 0.006 of unwatermarked GLS in every window.

5.3 Comparison over the full online stream

Against the single-draft methods. On all three workloads of Table 1, WMGLS and PFR have the same ANLPPT to three decimals, while MSE has an ANLPPT of 0.004. The AATPS of WMGLS is 25–26% higher than that of PFR, and its token rate is 18–20% higher than that of PFR and 17–26% higher than that of MSE, with six draft sequences per verification step against one. Relative to GLS, which uses the same K and L without a key, WMGLS is within 0.008 AATPS and 4% in TPS.

Contribution of online training. To measure how much efficiency the online updates add, we run WMGLS on Vicuna-7B (two seeds) with the draft frozen at its warm-up weights and everything else unchanged (Table 2). Online updates raise the AATPS by 0.27–0.35 (21–31%) and the token rate by 20–29% on all three workloads, while ANLPPT is identical.

Larger target. The same pattern holds on Vicuna-13B-v1.3 with GSM8K (Table 1, right): WMGLS and PFR have the same ANLPPT (0.025), and the AATPS and token rate of WMGLS are 28% and 16% higher. Its AATPS is within 0.006 of that of GLS, and its token rate within 2%.

Limitations. Our experiments cover the Gumbel-max watermark on two 7B targets, one 13B target and one serving system. Watermark strength on the low-entropy Llama-2-7B-chat is low for every Gumbel-max watermarked method, including PFR (Proposition 4).Gumbel-max watermarks preserve the output distribution in expectation over the key rather than for a fixed key (Hu et al., 2024; Wu et al., 2026); with a chat template, statistics such as the response length can therefore vary with the key (Appendix D).

6 Conclusion

Online training adds a further dimension to the trade-off between watermark strength and efficiency. For the Gumbel-max watermark, WOSpec keeps the watermarked text fixed under online updates while verifying K draft sequences: on Llama-2-7B-chat and Vicuna-13B, WOSpec matches the watermark strength of PFR while accepting 25–28% more tokens per verification step.

AI USE STATEMENT

In this work, we used generative AI tools as writing and coding assistants: to edit the text, check notation and reported numbers, discuss and check proofs, implement and debug parts of the code, and help analyze results. The authors reviewed all AI-assisted content and take full responsibility for the paper.

ETHICS STATEMENT

This work aims to make provenance tracing of LLM-generated text compatible with efficient, continually adapted serving. It involves no human subjects and uses public datasets and openly released models under their licenses. Watermark detection can produce false positives and can be weakened by editing or paraphrasing, so it should support, not replace, human judgement about provenance.

REPRODUCIBILITY STATEMENT

Section 4 and Appendix C specify the decoder, including the construction of the random numbers, the rule for repeated contexts and the boundary tokens; Appendices A and B prove all stated results. Appendix D lists the data splits and the warm-up and online training settings and describes the role of seeds and keys, and Appendix E gives the detector and all metric definitions. Appendix H reports every 20-chunk window of every experiment. We will release the serving back-ends, the online-training driver, the detector, and the scripts that produce all tables and figures from the saved per-response records.

REFERENCES

Scott Aaronson and H Kirchner. Watermarking of large language models. In *Large Language Models and Transformers Workshop at Simons Institute for the Theory of Computing*, 2023.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Majid Daliri, Christopher Musco, and Ananda Theertha Suresh. Coupling without communication and drafter-invariant speculative decoding. In *2025 IEEE International Symposium on Information Theory (ISIT)*, pp. 1–6. IEEE, 2025.

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.

Jiayi Fu, Xuandong Zhao, Ruihan Yang, Yuansen Zhang, Jiangjie Chen, and Yanghua Xiao. Gumbelsoft: Diversified language model watermarking via the gumbelmax-trick. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5791–5808, 2024.

Eva Giboulot and Teddy Furon. Watermax: breaking the LLM watermark detectability-robustness-quality trade-off. *Advances in Neural Information Processing Systems*, 37:18848–18881, 2024.

Thibaud Gloaguen, Nikola Jovanović, Robin Staab, and Martin Vechev. Black-box detection of language model watermarks. In *International Conference on Learning Representations*, volume 2025, pp. 77539–77562, 2025.

Weiqing He, Xiang Li, Li Shen, Weijie Su, and Qi Long. Improving the trade-off between watermark strength and speculative sampling efficiency for language models. *arXiv preprint arXiv:2602.01428*, 2026.

Yunlong Hou, Fengzhuo Zhang, Cunxiao Du, Xuan Zhang, Jiachun Pan, Tianyu Pang, Chao Du, Vincent YF Tan, and Zhuoran Yang. Banditspec: Adaptive speculative decoding via bandit algorithms. In *International Conference on Machine Learning (ICML)*, 2025.

Zhengmian Hu and Heng Huang. Inevitable trade-off between watermark strength and speculative sampling efficiency for language models. *Advances in Neural Information Processing Systems*, 37: 55370–55402, 2024.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 45408–45436, 2024.

Zhengmian Hu, Tong Zheng, Vignesh Viswanathan, Ziyi Chen, Ryan Rossi, Yihan Wu, Dinesh Manocha, and Heng Huang. Towards optimal multi-draft speculative decoding. In *International Conference on Learning Representations*, volume 2025, pp. 3181–3203, 2025.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. Codesearchnet challenge: Evaluating the state of semantic code search. *arXiv preprint arXiv:1909.09436*, 2019.

Ashish Khisti, MohammadReza Ebrahimi, Hassan Dbouk, Arash Behboodi, Roland Memisevic, and Christos Louizos. Multi-draft speculative sampling: Canonical decomposition and theoretical limits. In *International Conference on Learning Representations*, volume 2025, pp. 1593–1633, 2025.

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

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

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International conference on machine learning*, pp. 19274–19286. PMLR, 2023.

Cheuk Ting Li and Venkat Anantharam. A unified framework for one-shot achievability via the Poisson matching lemma. *IEEE Transactions on Information Theory*, 67(5):2624–2651, 2021.

Cheuk Ting Li and Abbas El Gamal. Strong functional representation lemma and applications to coding theorems. *IEEE Transactions on Information Theory*, 64(11):6967–6978, 2018.

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.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *Advances in Neural Information Processing Systems*, 38:136737–136756, 2026.

Hongyi Liu, Jiaji Huang, Zhen Jia, Youngsuk Park, and Yu-Xiang Wang. Not-a-bandit: Provably no-regret drafter selection in speculative decoding for llms. In *International Conference on Learning Representations*, volume 2026, pp. 19046–19072, 2026a.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Alvin Cheung, Zhijie Deng, Ion Stoica, and Hao Zhang. Online speculative decoding. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Yanxiao Liu, Sicheng Wan, Zhan Gao, and Deniz Gündüz. Watermarkable multi-draft speculative sampling via poisson processes, 2026b. arXiv preprint.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, et al. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pp. 932–949, 2024.

Yu-Yang Qian, Hao-Cong Wu, Yichao Fu, Hao Zhang, and Peng Zhao. When drafts evolve: Speculative decoding meets online learning. *arXiv preprint arXiv:2603.12617*, 2026.

Joseph Rowan, Buu Phan, and Ashish Khisti. List-level distribution coupling with applications to speculative decoding and lossy compression. *Advances in Neural Information Processing Systems*, 38, 2025.

Alexander Samarin, Sergei Krutikov, Anton Shevtsov, Sergei Skvortsov, Filipp Fisin, and Alexander Golubev. Lk losses: Direct acceptance rate optimization for speculative decoding. *arXiv preprint arXiv:2602.23881*, 2026.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. Spectr: Fast speculative decoding via optimal transport. *Advances in Neural Information Processing Systems*, 36:30222–30242, 2023.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

Rahul Thomas and Arka Pal. Global resolution: Optimal multi-draft speculative sampling via convex optimization. In *International Conference on Learning Representations*, volume 2026, pp. 81274–81307, 2026.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Dor Tsur, Carol Xuan Long, Claudio Mayrink Verdun, Sajani Vithana, Hsiang Hsu, Richard Chen, Haim H. Permuter, and Flavio P. Calmon. Heavywater and simplexwater: Distortion-free LLM watermarks for low-entropy distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Yihan Wu, Xuehao Cui, Ruibo Chen, and Heng Huang. Analyzing and evaluating unbiased language model watermark. In *International Conference on Learning Representations*, volume 2026, pp. 142418–142433, 2026.

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

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody H Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37:62557–62583, 2024.

A PROOFS

In this section we prove the results of Section 4. For clarity, each result is restated before its proof. We first fix the setting (Appendix A.1), then prove Lemma 1 (Appendix A.2) and a pointwise identity used throughout (Appendix A.3), then prove part (ii) of Proposition 2 and its extension to the online process (Appendices A.4 and A.5), and finally part (i) and the distortion-freeness of a response (Appendices A.6 and A.7). Proposition 1 is proved in Appendix B.3.

A.1 SETTING

Consider one response to a prompt x and write $h_i = (x, y_{<i})$ for the history before output position i. For a history $h = h_i$, let $\mathcal{C}(h) = \{c(h_j) : j < i\}$ be the contexts of its earlier output positions; the context $c(h)$ is *new* if $c(h) \notin \mathcal{C}(h)$ and *repeated* otherwise. A node u of the draft tree has history h_u, the committed history followed by the draft tokens on the path to u, so the same definition applies to nodes. The *random numbers* used at a history h are

$$N_{h,v} = \begin{cases} S^{\kappa}_{c(h),v}, & c(h) \text{ new}, \\ F_{h,v}, & c(h) \text{ repeated}, \end{cases} \qquad v \in \mathcal{W}, \tag{6}$$

where $S^{\kappa}_{c,v}$ is the pseudorandom number of Eq. (2) and $F_{h,v} = -\log U'_{h,v}$ is a fallback random number obtained from the auxiliary key, the full history h and its output position; it does not depend on K, the active set, the draft, or the step. WMGLS splits $N_{h_u,\cdot}$ at node u by Eq. (4), with $S^{\kappa}_{c,v}$ replaced by $N_{h_u,v}$. Gumbel-max watermarked generation $W_\kappa(p)$ with the same fallback commits $z_i = \arg\min_v N_{h_i,v}/p(v \mid h_i)$. The detector scores only positions with a new context, so each vector of pseudorandom numbers contributes at most once per response. Every $\arg\min$ over $\mathcal{W}$ breaks ties by a fixed order of $\mathcal{W}$, the same in all decoders, so the statements below hold for every realization of the random numbers; under Assumption 1 ties have probability zero. We use the convention $s/0 = +\infty$ for $s > 0$, so a token of probability zero is never selected by an exponential race, and sums over v may be restricted to the support of the distribution involved.

The main text states the following two conditions in words: pseudorandom values are treated as truly random (Section 3), and the drafts are independent of the random numbers of the decoder (Propositions 1 and 2(i)). The proofs use them in this form.

Assumption 1 (Truly random numbers)**.** *All pseudorandom values are independent* $\mathrm{Unif}(0,1)$ *variables. Hence the variables* $S^{\kappa}_{c,v}$ *over distinct* (c,v)*, the variables* $F_{h,v}$ *over distinct* (h,v)*, and the auxiliary random numbers* $(O_v, R_v^{(k)})$ *at distinct nodes are mutually independent, with* $S^{\kappa}_{c,v}, F_{h,v} \sim \mathrm{Exp}(1)$*, and they are independent of the prompt.*

Assumption 2 (Draft independent of the randomness)**.** *The draft parameters are independent of all pseudorandom variables used by the decoders: the pseudorandom numbers, the fallback random numbers, the auxiliary random numbers* $(O_v, R_v^{(k)})$*, and the auxiliary acceptance and resampling randomness of* MSE*.*

Assumption 1 is the standard idealization of a pseudorandom function, and it is used only within a response. Assumption 2 holds, for example, for a draft that is not trained on text generated with the key κ. It can fail in the online loop, because the same key is reused across requests and the draft is trained on earlier watermarked responses. For instance, let $p = (\frac{1}{2}, \frac{1}{2})$ at a context c, let an earlier response have committed the watermarked token $w = \arg\min_v S^{\kappa}_{c,v}/p_v$, and let the updated draft put $q_w = 0.9$. The single-draft coupling of PFR then proposes w whenever $S^{\kappa}_{c,w} < 9\, S^{\kappa}_{c,v}$ for the other token v, which always holds because $S^{\kappa}_{c,w} < S^{\kappa}_{c,v}$, so it matches with probability 1; with random numbers independent of the draft the matching probability is $\Pr[S_w < S_v] + \Pr[S_w > 9S_v] = 0.5 + 0.1 = 0.6$. Part (ii) of Proposition 2 and its online implication (Appendix A.5) do not use Assumption 2; part (i) and the analysis of Appendix B do.

A.2 PROOF OF LEMMA 1

Lemma 1. *Let* $S^{(1)}, \dots, S^{(J)}$ *be independent* $\mathrm{Exp}(1)$ *variables,* $M = \min_k S^{(k)}$*,* $O = \arg\min_k S^{(k)}$ *and* $R^{(k)} = S^{(k)} - M$ *for* $k \neq O$*. Then* $M \sim \mathrm{Exp}(J)$*,* O *is uniform on* $\{1, \dots, J\}$*,*

the $R^{(k)}$ are independent $\mathrm{Exp}(1)$*, and $(M, O, (R^{(k)})_{k\neq O})$ are mutually independent. Conversely, if these objects are drawn independently from these laws, then $S^{(O)} = M$ and $S^{(k)} = M + R^{(k)}$ $(k \neq O)$ are independent* $\mathrm{Exp}(1)$ *variables.*

Proof. For $o \in \{1, \dots, J\}$ let $\Omega_o = \{s \in (0,\infty)^J : s_o < s_k \text{ for all } k \neq o\}$. The sets $\Omega_1, \dots, \Omega_J$ are disjoint and cover $(0,\infty)^J$ up to a Lebesgue-null set. The map

$$\Phi_o : (0,\infty) \times (0,\infty)^{J-1} \to \Omega_o, \qquad (m, (r_k)_{k\neq o}) \mapsto s, \quad s_o = m, \;\; s_k = m + r_k \; (k \neq o), \tag{7}$$

is a bijection whose Jacobian matrix is triangular with unit diagonal, so $|\det D\Phi_o| = 1$. For $s = \Phi_o(m, r)$ the joint density of J independent $\mathrm{Exp}(1)$ variables factorizes as

$$\prod_{k=1}^{J} e^{-s_k} \;=\; e^{-Jm} \prod_{k\neq o} e^{-r_k} \;=\; \underbrace{Je^{-Jm}}_{M\sim\mathrm{Exp}(J)} \cdot \underbrace{\tfrac{1}{J}}_{O\sim\mathrm{Unif}} \cdot \prod_{k\neq o} \underbrace{e^{-r_k}}_{R^{(k)}\sim\mathrm{Exp}(1)} . \tag{8}$$

The right-hand side is the joint density of $(M, O, (R^{(k)})_{k\neq O})$ under the product law in the statement. Since $\Phi = (\Phi_o)_o$ is a bijection with unit Jacobian between the two spaces (up to null sets), Eq. (8) shows that Φ maps the product law onto the law of J independent $\mathrm{Exp}(1)$ variables, and Φ^{-1} maps it back. The first map is the converse statement and the second is the forward statement. Finally, $\Phi_o^{-1}(s)$ recovers $M = s_o$, $O = o$ and $R^{(k)} = s_k - s_o$, which are exactly the minimum, the minimizing index and the excesses. □

A.3 A POINTWISE IDENTITY

Lemma 2 (Minimum identity)**.** *At every node u with $J = |\mathcal{S}(u)|$ and for every realization of the random numbers, the draft random numbers of Eq.* (4) *satisfy*

$$\min_{k\in\mathcal{S}(u)} S^{(k)}_{u,v} = \frac{N_{h_u,v}}{J} \quad (v \in \mathcal{W}), \qquad \textit{hence} \qquad \arg\min_v \frac{\min_k S^{(k)}_{u,v}}{p_u(v)} = \arg\min_v \frac{N_{h_u,v}}{p_u(v)}. \tag{9}$$

Proof. For each v, the draft O_v has $S^{(O_v)}_{u,v} = N_{h_u,v}/J$, and every other active draft has $S^{(k)}_{u,v} = N_{h_u,v}/J + R^{(k)}_v \geq N_{h_u,v}/J$ because $R^{(k)}_v \geq 0$. This gives the first identity. Multiplying every ratio $\min_k S^{(k)}_{u,v}/p_u(v)$ by the same constant $J > 0$ does not change the minimizer, which gives the second. □

A.4 PROOF OF PROPOSITION 2(II)

Proposition 2 (part (ii))**.** WMGLS *is drafter invariant, and for every draft its committed response is the text of $W_\kappa(p)$ with the same fallback: for every key κ, number of drafts K, lookahead L and draft, the token committed at every position with a new context c is Y^κ_c.*

Proof. Fix the prompt x, the key, every realization of the fallback random numbers and the auxiliary random numbers, and the drafts, K and L used in every step; they may differ from step to step. The text of $W_\kappa(p)$ with the same fallback is the sequence

$$z_i = \arg\min_v \frac{N_{h^z_i,v}}{p(v \mid h^z_i)}, \qquad h^z_i = (x, z_{<i}), \tag{10}$$

stopped at the first EOS or after N tokens. Let $y_1, y_2, \dots$ be the tokens committed by Algorithm 1 and $h_i = (x, y_{<i})$.

Step 1: the target token at a node. At every node u, Algorithm 1 computes $Y(u) = \arg\min_v N_{h_u,v}/p(v \mid h_u)$, which is Eq. (5) with the fallback random numbers at a repeated context. It is a function of h_u, the key and the fallback variables only. By Lemma 2, it is also the target token of conditional GLS for the draft random numbers at u.

Step 2: every committed token is a target token at a committed history. Consider a step that starts after n committed tokens, at the root u_0 with $h_{u_0} = h_{n+1}$. Verification visits nodes $u_0, u_1, \dots, u_m$,

where u_{j+1} is the child of u_j labelled $Y(u_j)$, and stops at the first u_m without such a child. It commits $Y(u_0), \ldots, Y(u_{m-1})$ as accepted tokens and then $Y(u_m)$ as the correction or bonus token if the budget allows. The child of u_j labelled $Y(u_j)$ has history $(h_{u_j}, Y(u_j))$, so by induction on j, $h_{u_j} = (h_{u_0}, Y(u_0), \ldots, Y(u_{j-1})) = h_{n+j+1}$. Hence $y_{n+j+1} = Y(u_j)$, and by Step 1, for every committed position i,

$$y_i = \arg\min_v \frac{N_{h_i,v}}{p(v \mid h_i)}. \tag{11}$$

The drafts enter only through m, the number of tokens the step commits.

Step 3: induction over positions. If $y_{<i} = z_{<i}$, then $h_i = h_i^z$, and Eqs. (10) and (11) give $y_i = z_i$. Both sequences stop at the first EOS or after N tokens, which are determined by the committed tokens, so $y = z$. At a position with a new context, $N_{h_i,\cdot} = S^{\kappa}_{c(h_i),\cdot}$ and $y_i = Y^{\kappa}_{c(h_i)}$.

The committed text is therefore a function of x, p, κ and the fallback variables only; it depends neither on the drafts, K and L of any step nor on the auxiliary random numbers. This is Definition 1 with R the pseudorandom, fallback and auxiliary random numbers, and the committed response for every draft is the text of $W_\kappa(p)$. For a fixed draft q, which does not depend on these variables, the law of the committed text over the fallback variables is therefore that of $W_\kappa(p)$. For drafts that depend on the random numbers, for example through training on earlier responses, conditioning on the draft may carry information about the fallback variables; the relevant statement is then the pathwise identity $y = z$ (Appendix A.5). The proof uses neither Assumption 1 nor Assumption 2, and treats the target probabilities as computed without numerical error. □

A.5 Drafts updated online

This subsection proves the statement in Section 4.3. Fix the prompt stream, the target, the key κ and the fallback randomness, and let $\mathcal{A}$ be any procedure that chooses the draft, K and L before each step, possibly depending on all earlier responses and hence on κ. Then, if both processes use the same target probabilities, $Y^{\mathcal{A}}_{1:T} = Y^{W}_{1:T}$, the responses of $W_\kappa(p)$ with the same fallback, and $D(Y^{\mathcal{A}}_{1:T}, \kappa) = D(Y^{W}_{1:T}, \kappa)$ for every detector D that uses only the prompts, the responses and the key.

Proof. Order all committed positions of all requests by the time at which they are committed, and induct over this order. Suppose that all responses before the current one, and the first $i-1$ tokens of the current response, coincide under $\mathcal{A}$ and under $W_\kappa(p)$. The current prompt is the same in both processes because the prompt stream is fixed, so both have the same history h_i. Whatever draft, K and L the procedure $\mathcal{A}$ selects on the basis of the past, Steps 1 and 2 of the proof of Proposition 2(ii) apply to the current step, because they hold for every draft, K and L: the next committed token is $\arg\min_v N_{h_i,v}/p(v \mid h_i)$ by Eq. (11), which is also the token committed by $W_\kappa(p)$ by Eq. (10). The draft only determines the step in which this token is committed. Both processes end the response at the same position, since EOS and the budget depend only on the committed tokens. This completes the induction, so $Y^{\mathcal{A}}_{1:T} = Y^{W}_{1:T}$. The detection statement follows because D is a function of the prompts, $Y_{1:T}$ and κ. The argument uses neither Assumption 1 nor Assumption 2; it holds for every realization of the key and of the fallback random numbers, including when $\mathcal{A}$ depends on them through earlier responses. □

The proof compares the two processes position by position, so it requires both processes to use the same target distributions $p(\cdot \mid h)$. With floating-point computation, the probabilities of the target evaluated inside a draft tree and in a single-sequence pass can differ slightly, and when two tokens have nearly equal values of $N_{h,v}/p(v \mid h)$, the $\arg\min$ may then be resolved differently; from that position on the two responses diverge, while each still follows the rule of Proposition 2(ii) at every new context.

In our runs this happens at a small fraction of positions. Pairing the responses of PFR and WMGLS that share the key and the prompt stream, the first divergence occurs at 0.2–0.6% of the compared positions (0.3–0.4% for Vicuna-13B, and 0.2–0.3% between the frozen and the online draft of WMGLS). To measure the cause precisely, we recomputed the target probabilities in FP32 at 192 sampled divergence positions (Llama-2-7B-chat and Vicuna-7B-v1.3, GSM8K). The two committed tokens have nearly equal values of $\log(N_{h,v}/p(v \mid h))$, which differ by 0.05 in the median, against

3.7–10 between the best and the second-best token at other positions. The recomputation reproduces the committed token at 560 of 562 positions (99.6%) before the divergence.

A.6 Proof of Proposition 2(i)

Proposition 2 (part (i)). *Under Assumptions 1 and 2, at every node, given the path to it, the proposals of Algorithm 1 have the same law as those of conditional GLS; in particular the number of tokens committed per verification step has the law of GLS.*

Proof. Fix a node u in a step that starts at the committed history h_{i_0}, and let $\mathcal{F}_u$ be the σ-algebra generated by the prompt, the draft parameters, the random numbers $N_{h_j,\cdot}$ at the committed positions $j < i_0$, and all random numbers used at the strict ancestors of u, including their auxiliary random numbers. By Eq. (11), the committed history h_{i_0} is a function of the prompt and of these $N_{h_j,\cdot}$ only; random numbers used at unverified branches of earlier steps does not enter it. The path to u, the active set $\mathcal{S}(u)$, $J = |\mathcal{S}(u)|$, and the distributions p_u and $q_u^{(k)}$ are $\mathcal{F}_u$-measurable. The random numbers $N_{h_u,\cdot}$ are indexed either by the context $c(h_u)$, which is new and hence differs from the context of every committed position and every ancestor, or by the history h_u, which differs from all of their histories. The auxiliary random numbers at u are indexed by u. By Assumption 1 these variables are independent of the random numbers in $\mathcal{F}_u$ and of the prompt (the indices of all these variables are $\mathcal{F}_u$-measurable, and it suffices to argue on each of the finitely many events on which they take fixed values), and by Assumption 2 they are independent of the draft parameters; hence they are independent of $\mathcal{F}_u$. Conditionally on $\mathcal{F}_u$,

$$M_v = \frac{N_{h_u,v}}{J} \sim \mathrm{Exp}(J), \qquad O_v \sim \mathrm{Unif}(\mathcal{S}(u)), \qquad R_v^{(k)} \sim \mathrm{Exp}(1), \tag{12}$$

all mutually independent and independent across v. By the converse part of Lemma 1, the array $(S_{u,v}^{(k)})_{k\in\mathcal{S}(u),v\in\mathcal{W}}$ is, conditionally on $\mathcal{F}_u$, a family of independent $\mathrm{Exp}(1)$ variables: the law of the fresh random numbers that conditional GLS draws at u (Algorithm 2).

Both algorithms compute the proposals and the target token at u by the same measurable function of this array. The proposals are $X^{(k)} = \arg\min_v S_{u,v}^{(k)}/q_u^{(k)}(v)$ in both. The target token is $\arg\min_v \min_k S_{u,v}^{(k)}/p_u(v)$ in conditional GLS and $\arg\min_v N_{h_u,v}/p_u(v)$ in WMGLS, which coincide by Lemma 2. Hence the conditional joint law of $\big((X^{(k)})_{k\in\mathcal{S}(u)}, Y_u\big)$ given $\mathcal{F}_u$ is the same in both algorithms, and so is the conditional probability of the acceptance event $\{Y_u \in \{X^{(k)}\}_k\}$. Applying this node by node along the verified path gives the same law for the number of tokens committed per step. □

A.7 Distortion-freeness

Corollary 1. *Under Assumption 1, averaged over the key, each response committed by* WMGLS *is distributed as autoregressive sampling from p: for every position i and history h of that response, $\Pr[y_i = a \mid h_i = h] = p(a \mid h)$ for all $a \in \mathcal{W}$.*

Proof. By Appendix A.4, $y_{<i} = z_{<i}$, and z_j depends only on $N_{h_j,\cdot}$ and $p(\cdot \mid h_j)$. Hence the event $\{h_i = h\}$ is measurable with respect to $\mathcal{G}_h = \sigma\big(N_{h_j,\cdot} : j < i\big)$, the random numbers at the earlier committed positions of h; pseudorandom numbers evaluated only at unverified branches do not enter. If $c(h)$ is new, then $c(h) \neq c(h_j)$ for all $j < i$, so $N_{h,\cdot} = S^{\kappa}_{c(h),\cdot}$ is independent of $\mathcal{G}_h$ by Assumption 1. If $c(h)$ is repeated, $N_{h,\cdot} = F_{h,\cdot}$ is indexed by a new history and is again independent of $\mathcal{G}_h$. In both cases $(N_{h,v})_v$ are independent $\mathrm{Exp}(1)$ variables given $\{h_i = h\}$, and by Eq. (2),

$$\Pr[y_i = a \mid h_i = h] = \Pr\Big[\arg\min_v \frac{N_{h,v}}{p(v \mid h)} = a\Big] = p(a \mid h). \tag{13}$$

Chaining these conditional laws over i gives the claim. The draft does not enter, so Assumption 2 is not needed. □

The corollary concerns one response. Because the key is reused across requests, different responses are not independent samples; this is the same situation as for Gumbel-max watermarked generation,

whose text WMGLS reproduces (Appendix A.5); Hu et al. (2024, Appendix B.3) discuss how a context history shared across requests restores independence.

B Further Analysis

This section analyzes the acceptance of WMGLS as a function of the number of draft sequences and the watermark strength carried by a committed token, and proves Proposition 1 on the single-draft methods. Throughout, we consider a single node with target distribution p, draft distribution q (all draft sequences at a node share the same history and hence the same draft distribution), truly random numbers that are independent of the draft (Assumptions 1 and 2), and auxiliary random numbers for acceptance and residual sampling that are independent of the pseudorandom numbers. All statements are about a single position; they are not claims about the closed online loop, where Assumption 2 may fail.

B.1 Acceptance of WMGLS

Proposition 3 (Acceptance of WMGLS)**.** *At a node with J active draft sequences, the probability that the target token is proposed by some active draft satisfies*

$$\Pr\big[Y \in \{X^{(1)}, \dots, X^{(J)}\}\big] \;\geq\; \beta_J(p,q) := \sum_{j \in \mathcal{W}} \frac{J}{A_j + (J-1)B_j}, \tag{14}$$

where $A_j = \sum_{i \in \mathcal{W}} \max\{p_i/p_j,\, q_i/q_j\}$ and $B_j = 1/p_j$; here j ranges over the support of p, and a ratio $0/0$ is read as 0. Moreover, $\beta_J(p,q)$ is non-decreasing in J, and if $q_v > 0$ whenever $p_v > 0$, then $\beta_J(p,q) \to 1$ as $J \to \infty$.

Proof. By Proposition 2(i), the joint law of the proposals and the target token at the node equals that of conditional GLS with J drafts, so the list matching lemma of Rowan et al. (2025, Theorem 1), applied with target p and proposal distribution q, gives Eq. (14); note that $\sum_i p_i/p_j = 1/p_j = B_j$. For monotonicity, each summand $f_j(J) = J/(A_j + (J-1)B_j)$ satisfies, as a function of a real $J \geq 1$,

$$f_j'(J) = \frac{A_j + (J-1)B_j - JB_j}{\big(A_j + (J-1)B_j\big)^2} = \frac{A_j - B_j}{\big(A_j + (J-1)B_j\big)^2} \;\geq\; 0, \tag{15}$$

because $A_j \geq \sum_i p_i/p_j = B_j$. Finally, if $q_j > 0$ then $A_j < \infty$ and $f_j(J) \to 1/B_j = p_j$ as $J \to \infty$; under the support condition this holds for every j with $p_j > 0$, so $\beta_J(p,q) \to \sum_j p_j = 1$. (If $q_j = 0 < p_j$, then $A_j = \infty$ and the j-th term vanishes for all J.) □

The bound itself is the list matching lemma of Rowan et al. (2025); what Proposition 3 adds is that it transfers to WMGLS and that it is monotone in J. With $J = 1$ the node is a PFR node, and Eq. (14) lower-bounds its agreement probability. Additional draft sequences can only increase this guarantee. Monotonicity is shown for the lower bound, not for the realized acceptance or token rate, which also depend on the active sets deeper in the tree and on the cost of drafting. Because the committed token does not depend on J (Proposition 2(ii)), this gain in acceptance comes without any change in the watermark.

B.2 Watermark strength of a committed token

At a new context, PFR, WMGLS and $W_\kappa(p)$ all commit $Y = \arg\min_v S^\kappa_{c,v}/p(v)$ (Proposition 2(ii)). The detector scores $U_Y = \exp(-S^\kappa_{c,Y})$, which is uniform on $(0,1)$ under the null hypothesis. The following proposition gives its law under the watermark; it is the Gumbel-max pivot studied by Li et al. (2025), restated in the notation of our detector.

Proposition 4 (Per-token watermark strength)**.** *Conditionally on $Y = y$ (with $p_y > 0$), the score satisfies $U_Y \sim \mathrm{Beta}(1/p_y, 1)$, so $\mathbb{E}[U_Y \mid Y = y] = 1/(1+p_y)$. Consequently the expected excess over the null mean is*

$$\Delta(p) := \mathbb{E}[U_Y] - \tfrac{1}{2} = \sum_{y \in \mathcal{W}} \frac{p_y(1-p_y)}{2(1+p_y)}, \qquad \tfrac{1}{4}\Big(1 - \sum_y p_y^2\Big) \;\leq\; \Delta(p) \;\leq\; \tfrac{1}{2}\Big(1 - \sum_y p_y^2\Big). \tag{16}$$

Proof. Let $T_v = S^{\kappa}_{c,v}/p_v$. The T_v are independent with $T_v \sim \mathrm{Exp}(p_v)$, so their minimum $T = \min_v T_v$ is $\mathrm{Exp}(\sum_v p_v) = \mathrm{Exp}(1)$ and is independent of the minimizing index Y, which satisfies $\Pr[Y = y] = p_y$. Since $S^{\kappa}_{c,Y} = p_Y T$, we have $U_Y = e^{-p_Y T}$ and, for $u \in (0, 1)$,

$$\Pr[U_Y \le u \mid Y = y] = \Pr\Big[T \ge \frac{-\log u}{p_y}\Big] = u^{1/p_y}, \tag{17}$$

which is the distribution function of $\mathrm{Beta}(1/p_y, 1)$, with mean $(1/p_y)/(1/p_y + 1) = 1/(1 + p_y)$. Averaging over Y and using $\sum_y p_y = 1$,

$$\Delta(p) = \sum_y p_y \Big(\frac{1}{1+p_y} - \frac{1}{2}\Big) = \sum_y \frac{p_y(1-p_y)}{2(1+p_y)}. \tag{18}$$

The bounds follow from $1 \le 1 + p_y \le 2$ and $\sum_y p_y(1 - p_y) = 1 - \sum_y p_y^2$. □

The watermark strength of a token is thus governed by the collision entropy of the target: it vanishes as p concentrates on one token and is independent of the draft. The detector turns the mean score into ANLPPT through the Chernoff exponent of Eq. (29). Writing $\log \mathbb{E}[e^{\lambda U}] = \lambda/2 + \lambda^2/24 + O(\lambda^4)$ for a uniform U (mean $1/2$, variance $1/12$), a second-order expansion of the per-token Chernoff exponent gives

$$\mathrm{ANLPPT} = \sup_{\lambda \ge 0}\Big\{\lambda \bar{U} - \log \frac{e^{\lambda} - 1}{\lambda}\Big\} = 6\Big(\bar{U} - \tfrac{1}{2}\Big)^2 + O\Big(\big(\bar{U} - \tfrac{1}{2}\big)^4\Big) \tag{19}$$

for $\bar{U} \ge 1/2$. Since a single fresh position, taken on its own, has a score whose mean excess over $\frac{1}{2}$ is $\Delta(p_c)$, heuristically ANLPPT decreases quadratically as the target becomes more deterministic. We do not claim an exact identity for the expectation of $\bar{U}$: the number and the selection of scored positions (repeated contexts, the length requirement) depend on the generated text. This is only a heuristic: ANLPPT is a nonlinear function of $\bar{U}$, it is averaged over responses of different lengths, and a mean score does not by itself determine the detection rate at a finite length. It is consistent with the much weaker watermark strength of every watermarked method on Llama-2-7B-chat, whose responses have a far lower LPPL than those of Vicuna-7B (Tables 7 and 8).

B.3 Proof of Proposition 1

Part (i). MSE is not drafter invariant: let $p = (\frac{1}{2}, \frac{1}{2})$ at a new context, and fix a realization of R with $0.02\, S^{\kappa}_{c,2} < S^{\kappa}_{c,1} < S^{\kappa}_{c,2}$ and an auxiliary acceptance uniform $B < \frac{1}{2}$. With the draft $q = p$, the watermarked draft token is $X = 1$, which is accepted with probability one, so MSE commits token 1. With the draft $q = (0.01, 0.99)$, we have $S^{\kappa}_{c,1}/0.01 > 2S^{\kappa}_{c,2} > S^{\kappa}_{c,2}/0.99$, so $X = 2$, and $B < \frac{1}{2} < \min\{1, p_2/q_2\}$, so MSE commits token 2. The committed token thus depends on the draft for a fixed R, and this event has positive probability. The acceptance and strength statements follow from Propositions 5 and 6 below and from the decomposition and gap bound proved after them.

Part (ii). PFR is Algorithm 1 with $K = 1$: with one active draft, Eq. (4) gives $S^{(1)}_{u,v} = S^{\kappa}_{c,v}$. By Proposition 2(ii), PFR is therefore drafter invariant, and its committed response is the text of $W_{\kappa}(p)$ for every draft. Under Assumptions 1 and 2, $X = Z_q(S^{\kappa}_c)$ and $Y = Z_p(S^{\kappa}_c)$ are computed from the same random numbers with $X \sim q$ and $Y \sim p$, so $\Pr[X = Y] \ge (1 - D)/(1 + D)$ by Daliri et al. (2025). For any coupling of $X \sim q$ and $Y \sim p$, $\Pr[X = Y] = \sum_v \Pr[X = Y = v] \le \sum_v \min\{q_v, p_v\} = 1 - D$. The score of Y^{κ}_c is given by Proposition 4.

Proposition 5 (MSE under a fixed key)**.** *Let $X = \arg\min_v S^{\kappa}_{c,v}/q(v)$ be the watermarked draft token and $a(x) = \min\{1, p(x)/q(x)\}$. Conditionally on the key, the token committed by* MSE *has law*

$$\mathcal{L}(Y \mid \kappa) = a(X)\, \delta_X + \big(1 - a(X)\big)\, r, \qquad r(v) = \frac{[p(v) - q(v)]_+}{D_{\mathrm{TV}}(p, q)}. \tag{20}$$

Averaged over the key, $Y \sim p$, and the committed token is the watermarked draft token with probability $1 - D_{\mathrm{TV}}(p, q)$. If $D_{\mathrm{TV}}(p, q) = 0$, then $a(X) = 1$ almost surely, no rejection occurs, and the residual r is not needed.

Proof. Given the key, X is fixed; MSE accepts it with probability $a(X)$ using an auxiliary coin and otherwise samples from the residual r with auxiliary random numbers, which gives Eq. (20). Under truly random numbers $X \sim q$ by Eq. (2), so the probability of accepting the watermarked token is $\sum_x q(x) \min\{1, p(x)/q(x)\} = \sum_x \min\{p(x), q(x)\} = 1 - D_{\mathrm{TV}}(p, q)$, and

$$\Pr[Y = v] = \min\{p(v), q(v)\} + D_{\mathrm{TV}}(p, q)\, r(v) = \min\{p(v), q(v)\} + [p(v) - q(v)]_+ = p(v). \quad (21)$$

□

Proposition 5 shows that both the fraction $1 - D_{\mathrm{TV}}(p, q)$ of watermarked tokens and the watermarked token itself, which is selected under q rather than p, depend on the draft. The next proposition quantifies the effect on the score used by our detector. The acceptance rule of MSE is that of Hu & Huang (2024); the proposition analyzes its Gumbel-max instance.

Proposition 6 (Expected score of MSE)**.** *Here the* score *of a committed token Y at a new context c is the pseudorandom number $U_Y = U^{\kappa}_{c,Y} = \exp(-S^{\kappa}_{c,Y})$ that the detector recomputes for it; the detector averages these scores over the scored positions into $\bar{U}$ (Appendix E), and $\mathbb{E}[U_Y] = \frac{1}{2}$ under the null hypothesis. Let $\mu_{\mathrm{PFR}}(p) = \mathbb{E}[U_Y]$ for the token committed by* PFR *(equivalently* WMGLS *or $W_\kappa(p)$), and $\mu_{\mathrm{MSE}}(p, q) = \mathbb{E}[U_Y]$ for the token committed by* MSE*. Then*

$$\mu_{\mathrm{PFR}}(p) = \sum_v \frac{p_v}{1 + p_v}, \qquad \mu_{\mathrm{MSE}}(p, q) = \sum_v \frac{p_v + \min\{p_v, q_v\}}{2(1 + q_v)}, \quad (22)$$

and $\frac{1}{2} \le \mu_{\mathrm{MSE}}(p, q) \le \mu_{\mathrm{PFR}}(p)$, with equality on the right if $q = p$.

Proof. The formula for μ_{PFR} is Proposition 4. For MSE, let $T_v = S^{\kappa}_{c,v}/q_v \sim \mathrm{Exp}(q_v)$ and $X = \arg\min_v T_v$, so that $\Pr[X = x] = q_x$, $T = \min_v T_v \sim \mathrm{Exp}(1)$ is independent of X, and, by the memoryless property, conditionally on $X = x$ the overshoots $T_y - T$ ($y \neq x$, $q_y > 0$) are independent $\mathrm{Exp}(q_y)$ variables independent of T. Hence, given $X = x$,

$$U_x = e^{-q_x T}, \qquad U_y = e^{-q_y T}\, e^{-q_y (T_y - T)} \quad (y \neq x), \quad (23)$$

where $q_y(T_y - T) \sim \mathrm{Exp}(1)$ is independent of T. Taking expectations, $\mathbb{E}[U_x \mid X = x] = 1/(1 + q_x)$ and $\mathbb{E}[U_y \mid X = x] = 1/(2(1 + q_y))$ for $y \neq x$. For a token y with $q_y = 0$, $T_y = +\infty$ and $S^{\kappa}_{c,y}$ is independent of X, so $\mathbb{E}[U_y \mid X = x] = \frac{1}{2}$, which agrees with the same formula. The acceptance coin and the residual sample are independent of the pseudorandom numbers given X. On acceptance the committed token is X, and acceptance occurs jointly with $X = x$ with probability $q_x \min\{1, p_x/q_x\} = \min\{p_x, q_x\}$; this part contributes $\sum_x \min\{p_x, q_x\}/(1 + q_x)$. Rejection is possible only when $p_X < q_X$, and the residual is supported on $\{y : p_y > q_y\}$, so on rejection the committed token y differs from X. Since $\Pr[\text{reject}] = D_{\mathrm{TV}}(p, q)$ and $r_y = [p_y - q_y]_+ / D_{\mathrm{TV}}(p, q)$, the rejection part contributes $\sum_y [p_y - q_y]_+ / (2(1 + q_y))$. Adding both parts and using $2\min\{p_y, q_y\} + [p_y - q_y]_+ = p_y + \min\{p_y, q_y\}$ gives Eq. (22).

For the lower bound, write $m_v = \min\{p_v, q_v\}$. Then

$$2\mu_{\mathrm{MSE}}(p, q) - 1 = \sum_v \frac{p_v + m_v}{1 + q_v} - \sum_v p_v = \sum_v \frac{m_v - p_v q_v}{1 + q_v} \;\ge\; 0, \quad (24)$$

because $p_v q_v \le \min\{p_v, q_v\}$ for $p_v, q_v \in [0, 1]$. For the upper bound we compare the two sums term by term. If $q_v \ge p_v$, the v-th term of μ_{MSE} is $p_v/(1 + q_v) \le p_v/(1 + p_v)$. If $q_v < p_v$, it is $g(q_v)$ with $g(t) = (p_v + t)/(2(1 + t))$, and $g'(t) = (1 - p_v)/(2(1 + t)^2) \ge 0$, so $g(q_v) \le g(p_v) = p_v/(1 + p_v)$. Summing gives $\mu_{\mathrm{MSE}} \le \mu_{\mathrm{PFR}}$, with equality when $q = p$. □

Decomposition and gap. Write $m_v = \min\{p_v, q_v\}$, so that $[p_v - q_v]_+ = p_v - m_v$. The proof of the lower bound gives $\mu_{\mathrm{MSE}} - \frac{1}{2} = \sum_v (m_v - p_v q_v)/(2(1 + q_v))$, and $m_v - p_v q_v = m_v(1 - q_v) - (p_v - m_v)\, q_v$, so

$$\mu_{\mathrm{MSE}} - \tfrac{1}{2} = \underbrace{\sum_v \frac{\min\{p_v, q_v\}\,(1 - q_v)}{2(1 + q_v)}}_{\text{accepted tokens}} - \underbrace{\sum_v \frac{[p_v - q_v]_+\, q_v}{2(1 + q_v)}}_{\text{rejected tokens}}. \quad (25)$$

In the proof of the formula, the first term collects the accepted tokens and the second the rejected ones, whose conditional mean score $1/(2(1+q_y))$ is at most $\frac{1}{2}$. For the gap, since $\mu_{\text{PFR}} - \frac{1}{2} = \Delta(p)$, compare the v-th terms $d_v = p_v/(1+p_v) - (p_v+m_v)/(2(1+q_v))$ of $\mu_{\text{PFR}} - \mu_{\text{MSE}}$. If $q_v \geq p_v$, then $m_v = p_v$ and

$$d_v = \frac{p_v(q_v - p_v)}{(1+p_v)(1+q_v)} \leq \frac{p_v}{(1+p_v)^2}\,(q_v - p_v) \leq \tfrac{1}{4}\,(q_v - p_v), \tag{26}$$

because $t/(1+t)^2 \leq \frac{1}{4}$ for $t \in [0,1]$. If $q_v < p_v$, then $m_v = q_v$ and

$$d_v = \frac{(p_v - q_v)(1-p_v)}{2(1+p_v)(1+q_v)} \leq \tfrac{1}{2}\,(p_v - q_v). \tag{27}$$

Summing and using $\sum_v [q_v - p_v]_+ = \sum_v [p_v - q_v]_+ = D$ gives $\mu_{\text{PFR}} - \mu_{\text{MSE}} \leq \frac{1}{4}D + \frac{1}{2}D = \frac{3}{4}D$. Together with Proposition 6, this places $\mu_{\text{MSE}} - \frac{1}{2}$ in $[\max\{0, \Delta(p) - \frac{3}{4}D\},\ \Delta(p)]$. The constant $\frac{3}{4}$ cannot be improved in general: for $p = (1-\varepsilon, \varepsilon)$ and $q = (1, 0)$ the gap divided by $D = \varepsilon$ tends to $\frac{3}{4}$ as $\varepsilon \to 0$. Equality $\mu_{\text{MSE}} = \mu_{\text{PFR}}$ holds if $q = p$, but also in degenerate cases, for example when p is a point mass.

These statements compare expected scores at a single position with random numbers that are independent of the draft. They are not statements about detection rates at a finite length, and they do not extend without further conditions to the closed online loop with a reused key. In the experiments, the ANLPPT of MSE rises slightly over the stream while remaining far below that of PFR (Figure 3); the committed text of PFR and WMGLS does not depend on the draft (Proposition 2).

C Algorithms and Implementation

Algorithm 2 is the unwatermarked conditional GLS decoder of Section 4.2.

C.1 Single-draft decoders

Algorithm 3 gives one step of the single-draft methods. Both propose the watermarked draft tokens and differ only in verification.

At a repeated context the pseudorandom numbers are replaced by the fallback random numbers of Appendix A.1. Hu & Huang (2024, Algorithm 1) write their decoders in a two-reweight framework: at a new context, a reweighting function maps a distribution r to a watermarked one, $R_\kappa(r)$; in both decoders the draft token is sampled from $R_\kappa(q)$, and verification applies speculative rejection to a pair (P, Q): with an auxiliary uniform $B \in [0,1)$, the draft token x is accepted if $B < \min\{1, P(x)/Q(x)\}$, and otherwise a token is sampled from $[P-Q]_+ / \sum_z [P(z) - Q(z)]_+$; after a fully accepted draft sequence, a bonus token is sampled from P at the next position. MSE verifies with the original pair $(P, Q) = (p, q)$, and their Maintain Watermark Strength decoder (MWS) with the reweighted pair $(P, Q) = (R_\kappa(p), R_\kappa(q))$ (Hu & Huang, 2024, Section 5). Under the Gumbel watermark, $R_\kappa(r)$ is the point mass on $Z_r(S_c^\kappa) = \arg\min_v S_{c,v}^\kappa / r(v)$, which is distributed as r over the key by Eq. (2).

Lemma 3 (PFR and MWS)**.** *At every position with a new context, for every realization of the pseudorandom numbers and of the auxiliary uniform B, MWS under the Gumbel watermark accepts the watermarked draft token if and only if* PFR *does, and commits the same token Y_c^κ, including the bonus token.*

Proof. Write $Y = Y_c^\kappa = Z_p(S_c^\kappa)$. Under the Gumbel reweighting, $Q = \delta_X$ with $X = Z_q(S_c^\kappa)$ and $P = \delta_Y$, so the draft token is X and $\min\{1, P(X)/Q(X)\} = \min\{1, \delta_Y(X)\} = \mathbb{1}[X = Y]$. If $X = Y$, the condition $B < 1$ holds and $X = Y$ is accepted. If $X \neq Y$, the condition $B < 0$ fails and X is rejected; the residual is $[\delta_Y - \delta_X]_+ = \delta_Y$, which has mass one, so the committed token is Y. In both cases the acceptance event is $\{X = Y\}$ and the committed token is Y, which are the rules of the PFR branch of Algorithm 3. The bonus token is drawn from $R_\kappa(p_{L+1}) = \delta_{Z_{p_{L+1}}(S^\kappa_{c_{L+1}})}$, the watermarked target token, as in PFR. □

The MSE branch of Algorithm 3 is the rule of Hu & Huang (2024) with $(P, Q) = (p, q)$ and the watermarked draft token X. In our implementation of MSE, a fully accepted sequence receives the

Algorithm 2: Conditional GLS decoding (Rowan et al., 2025), unwatermarked

Input : prompt $x_{1:n_0}$, maximum length N (prompt plus new tokens), block length L, K draft distributions $q^{(1)}, \dots, q^{(K)}$, target p
Output : continuation $x_{n_0+1:n}$

$n \leftarrow n_0$
while $n < N$ and $x_n \neq \texttt{EOS}$ **do**
  root u_0 with path $x_{1:n}$ and active drafts $\mathcal{S}(u_0) \leftarrow \{1, \dots, K\}$
  for $d = 1$ **to** $\min\{L,\ N - n\}$ **do** `// grow the draft tree`
    foreach *node* u *at depth* $d-1$ *with* $\mathcal{S}(u) \neq \emptyset$ **do**
      draw fresh random numbers $S^{(k)}_{u,v} \sim \mathrm{Exp}(1)$ for $k \in \mathcal{S}(u)$, $v \in \mathcal{W}$
      $X^{(k)} \leftarrow \arg\min_v S^{(k)}_{u,v} / q^{(k)}_u(v)$ for each $k \in \mathcal{S}(u)$ `// Eq.` (2)
      add one child of u per distinct proposal; its active drafts are those that proposed it
  draw fresh random numbers $S^{(k)}_{u,v} \sim \mathrm{Exp}(1)$ for every leaf u at the last depth and $k \in \mathcal{S}(u)$ `// for the bonus token`
  score every node of the tree with one target forward pass
  $u \leftarrow u_0$;
  $Y \leftarrow \arg\min_v \big(\min_{k \in \mathcal{S}(u)} S^{(k)}_{u,v}\big) / p_u(v)$ `// Eq.` (3)
  while u has a child u' labelled Y and $Y \neq \texttt{EOS}$ **do**
    $x_{n+1} \leftarrow Y$;
    $n \leftarrow n + 1$;
    $u \leftarrow u'$;
    $Y \leftarrow \arg\min_v \big(\min_{k \in \mathcal{S}(u)} S^{(k)}_{u,v}\big) / p_u(v)$
  if $n < N$ **then**
    $x_{n+1} \leftarrow Y$;
    $n \leftarrow n + 1$ `// correction, or bonus token at a leaf`
return $x_{n_0+1:n}$

Algorithm 3: One step of the single-draft methods PFR and MSE

Input : committed history h, lookahead L, draft q, target p, key κ
Output : tokens committed in this step

for $i = 1$ **to** L **do**
  $h_i \leftarrow h\, x_{1:i-1}$;
  $c_i \leftarrow$ context of h_i
  $x_i \leftarrow \arg\min_v S^{\kappa}_{c_i,v} / q(v \mid h_i)$ `// watermarked draft token`
compute $p_i = p(\cdot \mid h_i)$ for $i = 1, \dots, L+1$ in one target forward pass, with $h_{L+1} = h\, x_{1:L}$
for $i = 1$ **to** L **do**
  if decoder is PFR **then**
    $y_i \leftarrow \arg\min_v S^{\kappa}_{c_i,v} / p_i(v)$
    if $x_i \neq y_i$ **then return** $(x_{1:i-1}, y_i)$
  else
    draw an auxiliary $B \sim \mathrm{Unif}(0,1)$
    if $B > \min\{1, p_i(x_i)/q(x_i \mid h_i)\}$ **then**
      draw $y_i \propto [p_i - q(\cdot \mid h_i)]_+$ with auxiliary random numbers
      return $(x_{1:i-1}, y_i)$
return $(x_{1:L},\ \arg\min_v S^{\kappa}_{c_{L+1},v} / p_{L+1}(v))$ `// bonus token`

watermarked token $Y^{\kappa}_{c_{L+1}}$ as bonus; compared with an unwatermarked bonus token, this can only increase its expected score (Proposition 4).

C.2 Relation to other constructions

Table 3 compares WMGLS with the constructions it builds on or is closest to. Conditional GLS obtains its acceptance from the active-set filtering, but its target token is the minimum of fresh draft random numbers, so it carries no watermark. Taking the minimum over all K drafts makes GLS drafter invariant at a cost in acceptance (Rowan et al., 2025, Proposition 6), and the target token is still not a watermark. PFR commits the Gumbel-max watermarked token but couples a single draft. The concurrent MPFR of Liu et al. (2026b) couples several drafts through keyed Poisson processes and satisfies their stopping-time form of drafter invariance. WMGLS keeps the proposal law of conditional GLS exactly (Proposition 2(i)) and commits the Gumbel-max watermarked token for every draft, including drafts that depend on the key (Proposition 2(ii)).

Table 3: **Constructions related to WMGLS.** Drafter invariance refers to Definition 1 unless stated otherwise; $D = D_{\mathrm{TV}}(p, q)$.

Construction	Drafter invariance	Acceptance	Watermark
Conditional GLS (Rowan et al., 2025, Alg. 2)	conditional form only	active-set filtering over K drafts	none (fresh draft random numbers)
GLS, minimum over all K (Rowan et al., 2025, Prop. 6)	yes	lower than conditional GLS	none
MSE (Hu & Huang, 2024)	no (Proposition 1)	$1 - D$, one draft	strength depends on the draft
PFR (Liu et al., 2026b)	yes	one draft, between $\frac{1-D}{1+D}$ and $1 - D$	Gumbel-max, full strength
MPFR (Liu et al., 2026b), concurrent	stopping-time form	several drafts	keyed Poisson coupling
WMGLS (ours)	yes	that of conditional GLS (Proposition 2(i))	Gumbel-max, full strength

C.3 Implementation

GLS draws fresh random numbers for every node, as in Algorithm 2, and WMGLS derives the draft random numbers by Eq. (4), as in Algorithm 1. All decoders share the target, the draft head and the online protocol. MSE falls back to p_i if the residual mass is below 10^{-12}.

Proposals come from an EAGLE-3 draft head (Li et al., 2026); the draft tree is the union of the sampled draft sequences and is scored by the target in one forward pass with tree attention in SGLang (Zheng et al., 2024). The auxiliary random numbers are indexed by the output position, a hash of the full history, and the ordered active draft indices, so that branches sharing a three-token suffix receive different auxiliary random numbers; they are never an input to detection. PFR, WMGLS and all 13B watermarked runs use a counter-based pseudorandom generator; MSE uses PCG64 on Vicuna-7B GSM8K and CodeSearchNet-Python and on Llama-2-7B-chat GSM8K, and the counter generator elsewhere; GLS uses the PyTorch GPU generator. Measured token rates include these generator differences. Target and draft run in BF16, with probability and random-number computations in FP32.

D Experimental Details

Data and prompts. For each workload we form disjoint warm-up (800 examples) and online (4,000 prompts) splits after Unicode normalization and deduplication, without filtering on generated text. The user turn is the GSM8K question, the CodeSearchNet docstring prefixed by "Please generate code based on the following doc:", or the Finance-Alpaca instruction, wrapped in the chat template of the target (Table 4) for warm-up, serving and training. Prompts are admitted if the templated prompt, 1,024 new tokens and the speculative budget fit the 2,048-token context. Each method trains only on its own generated responses.

Table 4: **Chat templates.** `{x}` is the user turn; the response is generated after the template.

Target	Prompt format
Vicuna-7B-v1.3, Vicuna-13B-v1.3	`A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions. USER: {x} ASSISTANT:`
Llama-2-7B-chat	`[INST] {x} [/INST]` (no system prompt)

Warm-up and online training. For every target and workload, a one-layer EAGLE-3 head is trained from scratch on the 800 warm-up examples (one epoch, batch size 4, learning rate 10^{-4}) and shared by all methods and seeds; the Vicuna-13B head (hidden size 5,120) is warmed up on the same GSM8K examples. Online updates follow the ensemble recipe of Qian et al. (2026) (Table 5). After chunk t has been served, the target is evaluated on the generated responses, and each of three learners $a \in \{1, 2, 3\}$ starts from the current head θ_t and minimizes the EAGLE-3 objective, a cross-entropy between the target's next-token distribution and the draft over seven rollout steps weighted by 0.8^j, with AdamW (learning rate η_a, no weight decay, gradient clipping at 1) for two epochs. With $d_{t,a}$ the learner's mean training loss, the next head is

$$\theta_{t+1} = \sum_{a=1}^{3} \omega_{t,a}\theta_{t,a}, \qquad \omega_{t,a} = \frac{\exp(-d_{t,a}/0.2)}{\sum_{b=1}^{3} \exp(-d_{t,b}/0.2)}. \tag{28}$$

Optimizer states are not carried across chunks and earlier chunks are not replayed. A chunk is always served by the head trained on the previous chunks.

Table 5: **Serving and online settings.**

Component	Setting
Online stream	100 chunks of 40 prompts; update after every chunk.
Learners	Three; learning rates 2×10^{-4}, 10^{-4}, 4×10^{-4}; two epochs; batch size 2; merged by Eq. (28).
Speculation	Lookahead $L = 5$; one draft sequence (PFR, MSE) or $K = 6$ (GLS, WMGLS).
Sampling	Temperature 1; full vocabulary; at most 1,024 new tokens; 2,048-token context; BF16.
Watermark	Pseudorandom numbers from the key and the last three tokens; repeated contexts skipped by the detector.
Timing	One request at a time; training and offline scoring excluded.

Seeds. Each of the three seeds fixes the order of the online stream, the training randomness, the request sampling seeds and the watermark key, which are shared by all methods of that seed. The frozen-draft ablation uses two of the three seeds.

Key sensitivity of the opening tokens. Our experiments keep the history of contexts within each response, as in Hu & Huang (2024) and He et al. (2026). Under a fixed key, the same context then receives the same pseudorandom numbers in every response: identical prompts receive identical responses (Fu et al., 2024), responses to a repeated prompt do not jointly follow the model distribution (Wu et al., 2026), and the resulting lack of diversity can be detected from outside (Gloaguen et al., 2025). With a chat template, the context of the first output position is the end of the template, which is the same for every prompt, so under a fixed key the first positions of all responses use the same pseudorandom numbers. Averaged over keys the output distribution is unchanged, but a single key can favour one opening for all prompts. Gumbel-max watermarked generation $W_\kappa(p)$ with the same key commits the same openings (Appendix A.5), so the effect comes from the watermark and not from the speculative decoding method. On Llama-2-7B-chat, and on Vicuna-7B-v1.3 with Finance-Alpaca, the mean response length of every seed is within 4% of that of GLS. With the key of seed 0 on Vicuna-7B-v1.3, the responses of PFR and WMGLS on GSM8K and CodeSearchNet-Python are about a third shorter than those of GLS (113 against 175 and 280 against 412 tokens on average), which raises their AATPS averaged over responses. Weighted by tokens rather than by responses,

the AATPS of WMGLS on these two workloads is 1.531 ± 0.010 and 1.438 ± 0.007, within 0.02 of that of GLS (1.522 and 1.420).

Masking template contexts. Hu et al. (2024) avoid reusing a context across generations by sharing the history of contexts across the requests of a batch. To limit the effect above, we apply a restricted form of this shared history that contains only the contexts overlapping the chat template: the first three output positions are treated as repeated contexts, use the fallback random numbers, which are derived from the auxiliary key and the full history including the prompt, and are skipped by the detector. The random numbers remain independent of the draft, so drafter invariance is unaffected, and each response has three fewer scored tokens. With the key of seed 0 on Vicuna-7B-v1.3 (Table 6), the response length and the AATPS of WMGLS are then close to those of GLS, and the ANLPPT and the detection rate do not decrease. The masking only concerns contexts shared through the template; under a fixed key, identical prompts still receive identical responses. We did not use it in the other experiments.

Table 6: **Masking template contexts** (Vicuna-7B-v1.3, key of seed 0; GSM8K and CodeSearchNet-Python, chunks 1–20). Length is the mean number of output tokens per response. GLS is unwatermarked.

Workload	Method	Length	AATPS	TPS	ANLPPT	Det. (%)
GSM8K	GLS	168.8	1.450	44.18	N/A	N/A
	WMGLS	111.2	1.616	46.77	0.037	30.8
	WMGLS, masked	173.9	1.445	42.66	0.037	31.8
Code	GLS	396.9	1.354	41.46	N/A	N/A
	WMGLS	285.9	1.552	45.20	0.046	45.2
	WMGLS, masked	407.5	1.363	40.67	0.055	48.5

E Detection and Metrics

Detector. For each response we take the first $B = 128$ output tokens (special tokens removed) and keep the positions $\mathcal{I}$ whose three-token context has not occurred earlier in the response; initial contexts use the prompt tokens. For $n = |\mathcal{I}|$ and the mean pseudorandom number $\bar{U} = n^{-1} \sum_{i \in \mathcal{I}} U^{\kappa}_{c(h_i), y_i}$ of the observed tokens, the p-value is the Chernoff bound for the mean of n independent uniforms,

$$\widehat{p} = \exp\left[-n \sup_{\lambda \geq 0} \left\{ \lambda \bar{U} - \log \frac{e^{\lambda} - 1}{\lambda} \right\} \right], \qquad \mathrm{ANLPPT} = -\frac{\log \widehat{p}}{n}, \tag{29}$$

which follows from Markov's inequality with the moment generating function $(e^{\lambda} - 1)/\lambda$ of a uniform variable; $\widehat{p} = 1$ if $\bar{U} \leq 1/2$. A response is detected if $\widehat{p} \leq 0.01$. Under the null hypothesis that the text is independent of the key, and with ideal pseudorandom numbers (Assumption 1), the scores at distinct contexts are independent uniforms, so the false-positive rate at this threshold is at most 1%. Responses shorter than B tokens are excluded from the watermark metrics and included in all others.

Metrics. For response r with N_r committed tokens, V_r target verification steps and end-to-end serving latency T_r, we average N_r/V_r (AATPS) and N_r/T_r (TPS) over the responses of a seed; committed corrections and bonus tokens count towards N_r. LPPL is the token-weighted negative log-likelihood of all returned output tokens under the target at temperature 1, conditioned on the prompt. ANLPPT and the detection rate are averaged over responses with at least 128 tokens. Each seed receives equal weight, and tables report the mean and sample standard deviation over seeds; windows are disjoint groups of 20 chunks.

F Additional Results

Table 7 gives the Llama-2-7B-chat block of Table 1 with standard deviations, detection rates and LPPL. Table 8 and Figure 4 repeat Table 1 and Figure 3 for Vicuna-7B-v1.3, and Tables 9 and 10

give Table 2 and the Vicuna-13B-v1.3 block of Table 1 with standard deviations. On Vicuna-7B-v1.3, in all three workloads, WMGLS and PFR have the same ANLPPT to three decimals and detection rates within 0.5 percentage points; the token rate of WMGLS is 17–22% higher than that of PFR and 16–31% higher than that of MSE. On GSM8K and CodeSearchNet-Python, the AATPS of PFR and WMGLS varies more across seeds than that of GLS (s.d. 0.07–0.14 against 0.001), and the mean AATPS of WMGLS exceeds that of GLS; Appendix D relates this variation to the key.

Table 7: **Efficiency and watermark strength over the full online stream** (Llama-2-7B-chat), including the detection rate (Det., the percentage of responses with $p \leq 0.01$) and LPPL; mean $\pm$ sample s.d. over three seeds of 100 chunks. GLS is unwatermarked.

Method	AATPS	TPS	ANLPPT	Det. (%)	LPPL
GSM8K					
PFR	$1.265_{\pm 0.002}$	$39.77_{\pm 0.18}$	$0.008_{\pm 0.002}$	$2.0_{\pm 1.2}$	$0.124_{\pm 0.015}$
MSE	$1.275_{\pm 0.008}$	$37.96_{\pm 0.23}$	$0.004_{\pm 0.001}$	$0.4_{\pm 0.2}$	$0.117_{\pm 0.000}$
GLS	$1.594_{\pm 0.001}$	$48.71_{\pm 0.43}$	N/A	N/A	$0.117_{\pm 0.001}$
WMGLS	$1.595_{\pm 0.002}$	$47.82_{\pm 0.53}$	$0.008_{\pm 0.002}$	$2.1_{\pm 1.4}$	$0.124_{\pm 0.015}$
CodeSearchNet-Python					
PFR	$1.224_{\pm 0.004}$	$38.95_{\pm 0.17}$	$0.018_{\pm 0.002}$	$11.8_{\pm 1.9}$	$0.216_{\pm 0.001}$
MSE	$1.227_{\pm 0.005}$	$39.15_{\pm 0.14}$	$0.004_{\pm 0.001}$	$0.4_{\pm 0.3}$	$0.214_{\pm 0.002}$
GLS	$1.540_{\pm 0.001}$	$47.53_{\pm 0.17}$	N/A	N/A	$0.210_{\pm 0.002}$
WMGLS	$1.548_{\pm 0.005}$	$46.62_{\pm 0.21}$	$0.018_{\pm 0.002}$	$11.8_{\pm 2.3}$	$0.215_{\pm 0.003}$
Finance-Alpaca					
PFR	$1.192_{\pm 0.000}$	$38.08_{\pm 0.09}$	$0.021_{\pm 0.003}$	$15.8_{\pm 4.3}$	$0.276_{\pm 0.003}$
MSE	$1.199_{\pm 0.000}$	$38.22_{\pm 0.09}$	$0.004_{\pm 0.001}$	$0.5_{\pm 0.4}$	$0.276_{\pm 0.003}$
GLS	$1.489_{\pm 0.000}$	$46.55_{\pm 0.17}$	N/A	N/A	$0.274_{\pm 0.001}$
WMGLS	$1.490_{\pm 0.001}$	$44.91_{\pm 0.21}$	$0.021_{\pm 0.003}$	$15.8_{\pm 3.8}$	$0.276_{\pm 0.003}$

Table 8: **Efficiency and watermark strength over the full online stream** (Vicuna-7B-v1.3); mean $\pm$ sample s.d. over three seeds of 100 chunks. GLS is unwatermarked.

Method	AATPS	TPS	ANLPPT	Det. (%)	LPPL
GSM8K					
PFR	$1.260_{\pm 0.072}$	$38.82_{\pm 1.91}$	$0.038_{\pm 0.003}$	$32.8_{\pm 2.4}$	$0.379_{\pm 0.001}$
MSE	$1.237_{\pm 0.001}$	$36.19_{\pm 0.40}$	$0.007_{\pm 0.001}$	$1.2_{\pm 0.5}$	$0.368_{\pm 0.009}$
GLS	$1.544_{\pm 0.001}$	$46.90_{\pm 0.28}$	N/A	N/A	$0.368_{\pm 0.003}$
WMGLS	$1.619_{\pm 0.129}$	$47.27_{\pm 3.00}$	$0.038_{\pm 0.003}$	$32.9_{\pm 2.8}$	$0.379_{\pm 0.001}$
CodeSearchNet-Python					
PFR	$1.221_{\pm 0.065}$	$38.17_{\pm 1.80}$	$0.060_{\pm 0.013}$	$57.1_{\pm 11.5}$	$0.519_{\pm 0.017}$
MSE	$1.212_{\pm 0.048}$	$35.63_{\pm 1.25}$	$0.007_{\pm 0.001}$	$1.7_{\pm 0.6}$	$0.552_{\pm 0.029}$
GLS	$1.455_{\pm 0.001}$	$44.50_{\pm 0.17}$	N/A	N/A	$0.526_{\pm 0.004}$
WMGLS	$1.557_{\pm 0.142}$	$45.72_{\pm 3.56}$	$0.060_{\pm 0.012}$	$57.0_{\pm 11.4}$	$0.521_{\pm 0.021}$
Finance-Alpaca					
PFR	$1.159_{\pm 0.006}$	$36.54_{\pm 0.40}$	$0.065_{\pm 0.004}$	$67.9_{\pm 4.2}$	$0.585_{\pm 0.017}$
MSE	$1.170_{\pm 0.005}$	$36.93_{\pm 0.17}$	$0.006_{\pm 0.001}$	$1.6_{\pm 0.3}$	$0.588_{\pm 0.015}$
GLS	$1.422_{\pm 0.001}$	$43.74_{\pm 0.27}$	N/A	N/A	$0.589_{\pm 0.002}$
WMGLS	$1.431_{\pm 0.011}$	$42.84_{\pm 0.56}$	$0.065_{\pm 0.005}$	$68.4_{\pm 4.8}$	$0.586_{\pm 0.017}$

Number of draft sequences. Table 11 varies K on one stream (Vicuna-7B-v1.3, GSM8K, stream and key of seed 0, 100 chunks, $L = 5$). The AATPS increases with K up to $K = 8$, while the ANLPPT and the detection rate do not change, as Proposition 2(ii) implies. We fix $K = 6$ in all other runs.

G Per-Workload Evolution

Figure 5 shows the curves of Figures 3 and 4 separately for each target and workload.

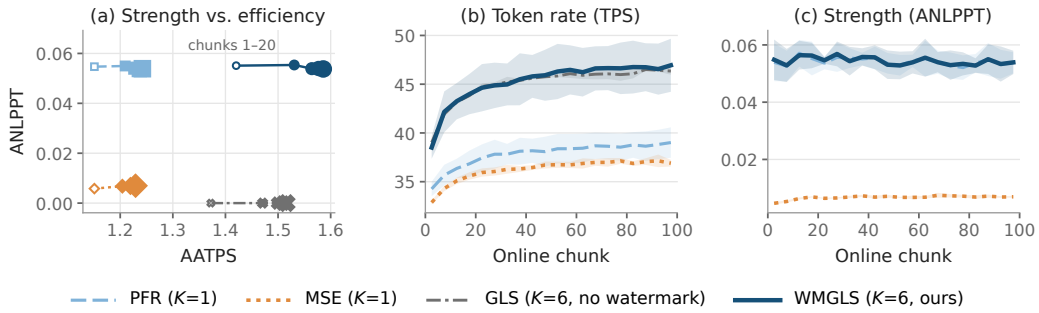

Figure 4: **Efficiency and watermark strength along the online stream** (Vicuna-7B-v1.3), as in Figure 3. In (c) the PFR curve lies under the WMGLS curve.

Table 9: **Frozen versus online draft for WMGLS** (Vicuna-7B-v1.3; full version of Table 2). Mean ± sample s.d. over two seeds, 100 chunks each; the online rows are the runs of the same two seeds.

WMGLS head	AATPS	TPS	ANLPPT	Det. (%)	LPPL
GSM8K					
Frozen	$1.275_{\pm 0.012}$	$37.82_{\pm 0.30}$	$0.039_{\pm 0.003}$	$34.1_{\pm 3.3}$	$0.377_{\pm 0.000}$
Online	$1.545_{\pm 0.006}$	$45.54_{\pm 0.27}$	$0.039_{\pm 0.003}$	$33.9_{\pm 3.2}$	$0.379_{\pm 0.001}$
CodeSearchNet-Python					
Frozen	$1.128_{\pm 0.003}$	$33.88_{\pm 0.01}$	$0.066_{\pm 0.007}$	$62.8_{\pm 7.4}$	$0.527_{\pm 0.017}$
Online	$1.476_{\pm 0.023}$	$43.66_{\pm 0.02}$	$0.066_{\pm 0.007}$	$62.8_{\pm 7.5}$	$0.531_{\pm 0.017}$
Finance-Alpaca					
Frozen	$1.108_{\pm 0.007}$	$33.34_{\pm 0.33}$	$0.065_{\pm 0.007}$	$67.0_{\pm 5.7}$	$0.588_{\pm 0.024}$
Online	$1.425_{\pm 0.001}$	$42.56_{\pm 0.35}$	$0.065_{\pm 0.007}$	$67.2_{\pm 6.2}$	$0.589_{\pm 0.022}$

H Twenty-Chunk Results

The following tables report every 20-chunk window of the runs in Tables 1, 2 and 8. Method order is PFR, MSE, GLS, WMGLS throughout.

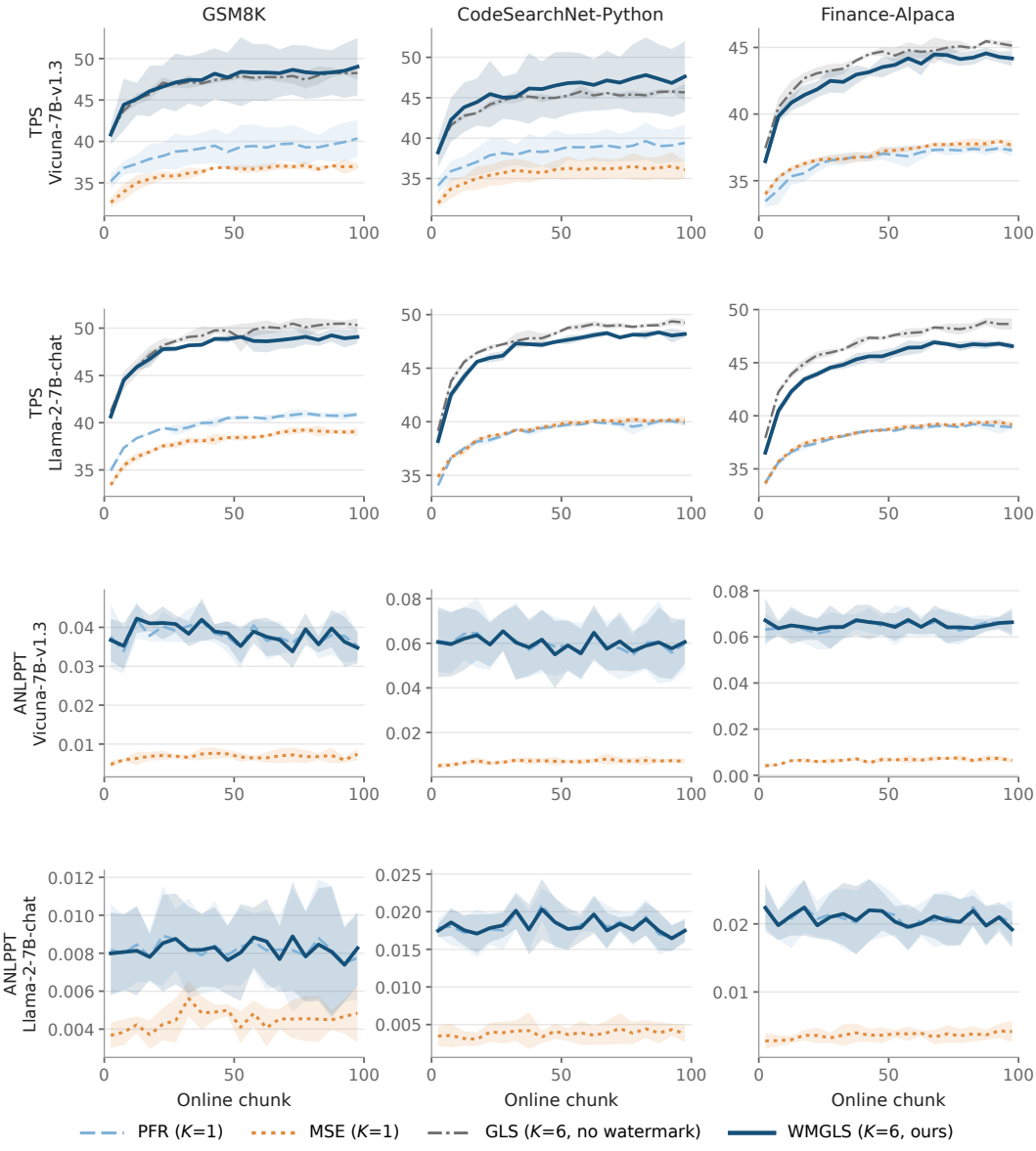

Figure 5: **Token rate (top two rows) and watermark strength (bottom two rows) over the online stream, per target and workload.** Lines are the mean over three seeds in bins of five chunks, and bands show ± one s.d. across seeds. In the ANLPPT rows the PFR curve largely lies under the WMGLS curve; GLS is unwatermarked and omitted.

Table 10: **Larger target** (Vicuna-13B-v1.3, GSM8K; Vicuna-13B-v1.3 block of Table 1); mean ± sample s.d. over three seeds of 100 chunks. LPPL is reported per window in Appendix H.

Method	AATPS	TPS	ANLPPT	Det. (%)
PFR	$1.220_{\pm 0.003}$	$32.45_{\pm 0.14}$	$0.025_{\pm 0.003}$	$17.8_{\pm 2.7}$
MSE	$1.228_{\pm 0.004}$	$32.63_{\pm 0.06}$	$0.005_{\pm 0.001}$	$0.6_{\pm 0.3}$
GLS	$1.551_{\pm 0.001}$	$38.18_{\pm 0.16}$	N/A	N/A
WMGLS	$1.557_{\pm 0.005}$	$37.56_{\pm 0.03}$	$0.025_{\pm 0.003}$	$17.5_{\pm 3.2}$

Table 11: **Number of draft sequences** (Vicuna-7B-v1.3, GSM8K, seed 0, 100 chunks). $K = 1$ is PFR; $K = 6$ is the WMGLS run of Table 8.

K	AATPS	TPS	ANLPPT	Det. (%)
1	1.343	41.02	0.036	30.7
2	1.502	44.68	0.036	30.8
4	1.673	49.12	0.036	31.3
6	1.768	50.72	0.036	30.9
8	1.840	52.01	0.036	31.2

Table 12: **Vicuna-7B-v1.3 / GSM8K: 20-chunk windows.** Mean ± sample s.d. over three seeds; 800 prompts per window and seed. Detection averages use responses with at least 128 tokens only.

Metric	Method	1–20	21–40	41–60	61–80	81–100
AATPS	PFR	$1.191_{\pm 0.042}$	$1.258_{\pm 0.071}$	$1.276_{\pm 0.080}$	$1.288_{\pm 0.083}$	$1.290_{\pm 0.082}$
	MSE	$1.180_{\pm 0.002}$	$1.233_{\pm 0.001}$	$1.253_{\pm 0.002}$	$1.260_{\pm 0.002}$	$1.261_{\pm 0.002}$
	GLS	$1.444_{\pm 0.005}$	$1.547_{\pm 0.002}$	$1.567_{\pm 0.007}$	$1.575_{\pm 0.003}$	$1.586_{\pm 0.006}$
	WMGLS	$1.506_{\pm 0.096}$	$1.616_{\pm 0.132}$	$1.649_{\pm 0.141}$	$1.662_{\pm 0.139}$	$1.662_{\pm 0.136}$
TPS	PFR	$36.79_{\pm 1.12}$	$38.77_{\pm 1.84}$	$39.24_{\pm 2.35}$	$39.48_{\pm 2.21}$	$39.80_{\pm 2.03}$
	MSE	$34.25_{\pm 0.74}$	$36.04_{\pm 0.49}$	$36.75_{\pm 0.22}$	$36.98_{\pm 0.20}$	$36.91_{\pm 0.41}$
	GLS	$43.91_{\pm 0.44}$	$47.03_{\pm 0.29}$	$47.64_{\pm 0.24}$	$47.72_{\pm 0.62}$	$48.19_{\pm 0.64}$
	WMGLS	$44.11_{\pm 2.33}$	$47.14_{\pm 3.05}$	$48.16_{\pm 3.41}$	$48.39_{\pm 3.12}$	$48.53_{\pm 3.10}$
ANLPPT	PFR	$0.038_{\pm 0.004}$	$0.040_{\pm 0.001}$	$0.038_{\pm 0.003}$	$0.037_{\pm 0.002}$	$0.037_{\pm 0.005}$
	MSE	$0.006_{\pm 0.001}$	$0.007_{\pm 0.001}$	$0.007_{\pm 0.001}$	$0.007_{\pm 0.002}$	$0.007_{\pm 0.001}$
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	$0.039_{\pm 0.004}$	$0.041_{\pm 0.001}$	$0.038_{\pm 0.003}$	$0.037_{\pm 0.003}$	$0.037_{\pm 0.004}$
Det. (%)	PFR	$33.0_{\pm 3.7}$	$35.1_{\pm 1.4}$	$31.8_{\pm 1.9}$	$32.2_{\pm 3.1}$	$32.1_{\pm 5.6}$
	MSE	$1.0_{\pm 0.7}$	$1.5_{\pm 0.9}$	$1.0_{\pm 0.3}$	$1.0_{\pm 0.2}$	$1.4_{\pm 0.6}$
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	$33.7_{\pm 5.1}$	$35.7_{\pm 1.0}$	$31.6_{\pm 1.9}$	$32.4_{\pm 5.0}$	$31.3_{\pm 3.9}$
LPPL	PFR	$0.376_{\pm 0.011}$	$0.377_{\pm 0.017}$	$0.380_{\pm 0.004}$	$0.381_{\pm 0.004}$	$0.380_{\pm 0.007}$
	MSE	$0.367_{\pm 0.006}$	$0.366_{\pm 0.015}$	$0.365_{\pm 0.012}$	$0.370_{\pm 0.015}$	$0.371_{\pm 0.005}$
	GLS	$0.367_{\pm 0.008}$	$0.369_{\pm 0.011}$	$0.377_{\pm 0.004}$	$0.363_{\pm 0.012}$	$0.365_{\pm 0.006}$
	WMGLS	$0.372_{\pm 0.010}$	$0.384_{\pm 0.020}$	$0.379_{\pm 0.007}$	$0.378_{\pm 0.007}$	$0.379_{\pm 0.012}$

Table 13: **Vicuna-7B-v1.3 / CodeSearchNet-Python: 20-chunk windows.** Mean ± sample s.d. over three seeds; 800 prompts per window and seed. Detection averages use responses with at least 128 tokens only.

Metric	Method	1–20	21–40	41–60	61–80	81–100
AATPS	PFR	1.153±0.037	1.217±0.065	1.236±0.065	1.243±0.075	1.256±0.086
	MSE	1.155±0.040	1.214±0.054	1.224±0.047	1.234±0.052	1.232±0.047
	GLS	1.355±0.002	1.451±0.009	1.481±0.002	1.493±0.002	1.494±0.003
	WMGLS	1.431±0.105	1.554±0.146	1.585±0.143	1.598±0.143	1.617±0.169
TPS	PFR	35.89±1.29	38.09±1.83	38.63±1.88	38.97±1.84	39.28±2.19
	MSE	33.78±1.01	35.74±1.39	36.05±1.12	36.31±1.35	36.28±1.40
	GLS	41.50±0.15	44.77±0.36	45.24±0.28	45.39±0.30	45.61±0.29
	WMGLS	42.19±2.63	45.43±3.50	46.57±3.75	47.00±3.56	47.39±4.38
ANLPPT	PFR	0.063±0.013	0.061±0.012	0.059±0.013	0.059±0.011	0.059±0.014
	MSE	0.006±0.001	0.007±0.001	0.007±0.001	0.008±0.002	0.007±0.001
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	0.061±0.014	0.061±0.012	0.058±0.012	0.060±0.011	0.059±0.012
Det. (%)	PFR	58.8±11.9	58.4±10.2	55.4±11.6	54.6±11.8	58.1±12.2
	MSE	1.3±0.5	1.6±0.4	1.8±0.3	2.0±1.3	1.9±0.7
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	59.1±12.4	58.2±11.1	54.9±11.9	55.1±11.1	57.8±10.9
LPPL	PFR	0.518±0.022	0.522±0.012	0.519±0.013	0.522±0.020	0.516±0.025
	MSE	0.551±0.020	0.549±0.037	0.536±0.023	0.554±0.036	0.568±0.032
	GLS	0.531±0.006	0.524±0.005	0.527±0.001	0.524±0.010	0.524±0.004
	WMGLS	0.513±0.029	0.527±0.024	0.512±0.022	0.528±0.014	0.525±0.017

Table 14: **Vicuna-7B-v1.3 / Finance-Alpaca: 20-chunk windows.** Mean ± sample s.d. over three seeds; 800 prompts per window and seed. Detection averages use responses with at least 128 tokens only.

Metric	Method	1–20	21–40	41–60	61–80	81–100
AATPS	PFR	1.109±0.005	1.154±0.007	1.171±0.006	1.178±0.006	1.181±0.008
	MSE	1.118±0.004	1.165±0.006	1.182±0.004	1.190±0.005	1.194±0.005
	GLS	1.317±0.002	1.414±0.003	1.449±0.001	1.461±0.003	1.469±0.002
	WMGLS	1.324±0.009	1.422±0.011	1.459±0.012	1.471±0.009	1.479±0.013
TPS	PFR	34.67±0.68	36.54±0.39	36.89±0.55	37.26±0.36	37.35±0.32
	MSE	35.36±0.14	36.65±0.28	37.17±0.13	37.68±0.17	37.79±0.27
	GLS	40.57±0.54	43.44±0.34	44.60±0.34	44.88±0.55	45.21±0.20
	WMGLS	39.64±0.54	42.43±0.69	43.64±0.54	44.20±0.57	44.32±0.46
ANLPPT	PFR	0.064±0.005	0.064±0.005	0.066±0.006	0.065±0.004	0.066±0.003
	MSE	0.005±0.001	0.006±0.001	0.006±0.001	0.007±0.001	0.007±0.001
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	0.065±0.007	0.065±0.006	0.066±0.006	0.065±0.004	0.065±0.002
Det. (%)	PFR	67.2±3.7	67.3±4.9	68.4±3.5	67.9±4.5	68.6±4.9
	MSE	0.9±0.2	1.6±0.3	1.6±0.3	2.2±0.9	1.9±0.1
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	68.0±4.5	67.3±5.4	69.4±4.8	69.1±5.4	68.0±4.8
LPPL	PFR	0.590±0.026	0.588±0.021	0.588±0.013	0.576±0.025	0.585±0.003
	MSE	0.592±0.023	0.586±0.009	0.589±0.016	0.586±0.013	0.589±0.014
	GLS	0.591±0.004	0.584±0.002	0.589±0.007	0.592±0.013	0.590±0.004
	WMGLS	0.588±0.028	0.587±0.015	0.584±0.013	0.582±0.025	0.587±0.006

Table 15: **Llama-2-7B-chat / GSM8K: 20-chunk windows.** Mean ± sample s.d. over three seeds; 800 prompts per window and seed. Detection averages use responses with at least 128 tokens only.

Metric	Method	1–20	21–40	41–60	61–80	81–100
AATPS	PFR	1.190±0.002	1.263±0.002	1.285±0.003	1.293±0.003	1.294±0.005
	MSE	1.197±0.005	1.272±0.008	1.297±0.010	1.304±0.007	1.303±0.008
	GLS	1.472±0.005	1.594±0.005	1.626±0.005	1.636±0.004	1.642±0.002
	WMGLS	1.471±0.008	1.596±0.004	1.627±0.003	1.638±0.001	1.641±0.004
TPS	PFR	37.38±0.10	39.54±0.28	40.40±0.20	40.73±0.33	40.79±0.27
	MSE	35.55±0.33	37.84±0.24	38.38±0.02	38.98±0.24	39.05±0.37
	GLS	44.67±0.79	48.76±0.46	49.58±0.64	50.17±0.64	50.39±0.53
	WMGLS	44.44±0.47	48.00±0.15	48.85±0.45	48.82±0.96	49.00±0.71
ANLPPT	PFR	0.008±0.002	0.008±0.002	0.008±0.002	0.008±0.002	0.008±0.003
	MSE	0.004±0.001	0.005±0.001	0.005±0.000	0.004±0.001	0.005±0.001
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	0.008±0.002	0.008±0.002	0.008±0.002	0.008±0.002	0.008±0.003
Det. (%)	PFR	1.7±0.9	1.9±0.8	2.3±1.3	2.4±1.9	1.7±1.3
	MSE	0.2±0.0	0.5±0.3	0.4±0.3	0.4±0.4	0.6±0.2
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	1.6±0.7	2.0±1.2	2.3±1.4	2.5±2.6	2.0±1.4
LPPL	PFR	0.124±0.014	0.124±0.015	0.126±0.019	0.123±0.017	0.123±0.013
	MSE	0.117±0.002	0.118±0.001	0.116±0.001	0.116±0.002	0.118±0.001
	GLS	0.116±0.001	0.120±0.000	0.117±0.002	0.117±0.001	0.116±0.001
	WMGLS	0.124±0.014	0.124±0.013	0.125±0.019	0.123±0.014	0.123±0.014

Table 16: **Llama-2-7B-chat / CodeSearchNet-Python: 20-chunk windows.** Mean ± sample s.d. over three seeds; 800 prompts per window and seed. Detection averages use responses with at least 128 tokens only.

Metric	Method	1–20	21–40	41–60	61–80	81–100
AATPS	PFR	1.155±0.003	1.221±0.004	1.242±0.006	1.251±0.005	1.254±0.003
	MSE	1.158±0.004	1.223±0.007	1.244±0.003	1.254±0.006	1.256±0.004
	GLS	1.418±0.003	1.537±0.002	1.571±0.003	1.585±0.004	1.589±0.007
	WMGLS	1.421±0.005	1.548±0.008	1.577±0.006	1.596±0.006	1.598±0.001
TPS	PFR	36.59±0.10	38.82±0.17	39.61±0.20	39.79±0.26	39.93±0.23
	MSE	36.78±0.24	39.00±0.07	39.75±0.20	40.11±0.25	40.12±0.26
	GLS	43.73±0.24	47.37±0.34	48.41±0.23	48.98±0.09	49.16±0.15
	WMGLS	42.63±0.32	46.67±0.26	47.53±0.27	48.09±0.14	48.17±0.37
ANLPPT	PFR	0.018±0.002	0.018±0.002	0.019±0.002	0.018±0.002	0.018±0.001
	MSE	0.003±0.001	0.004±0.002	0.004±0.001	0.004±0.001	0.004±0.001
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	0.018±0.002	0.018±0.002	0.019±0.002	0.018±0.002	0.018±0.001
Det. (%)	PFR	11.8±3.4	11.5±1.0	12.5±2.5	12.4±1.7	11.2±1.7
	MSE	0.1±0.1	0.6±0.3	0.3±0.4	0.6±0.5	0.6±0.3
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	11.7±2.8	12.1±1.9	12.0±3.6	12.5±2.3	10.7±1.3
LPPL	PFR	0.213±0.003	0.216±0.002	0.218±0.002	0.216±0.001	0.215±0.004
	MSE	0.213±0.002	0.215±0.003	0.213±0.003	0.215±0.005	0.214±0.002
	GLS	0.208±0.002	0.209±0.003	0.210±0.002	0.213±0.005	0.212±0.003
	WMGLS	0.213±0.005	0.215±0.003	0.219±0.003	0.216±0.001	0.214±0.003

Table 17: **Llama-2-7B-chat / Finance-Alpaca: 20-chunk windows.** Mean ± sample s.d. over three seeds; 800 prompts per window and seed. Detection averages use responses with at least 128 tokens only.

Metric	Method	1–20	21–40	41–60	61–80	81–100
AATPS	PFR	1.123±0.002	1.185±0.001	1.209±0.002	1.218±0.002	1.225±0.002
	MSE	1.129±0.001	1.193±0.001	1.216±0.001	1.226±0.003	1.232±0.003
	GLS	1.352±0.004	1.477±0.005	1.521±0.002	1.542±0.005	1.553±0.006
	WMGLS	1.350±0.004	1.480±0.003	1.524±0.001	1.541±0.004	1.553±0.006
TPS	PFR	35.75±0.02	37.92±0.02	38.69±0.06	38.97±0.14	39.07±0.35
	MSE	35.85±0.03	38.04±0.11	38.76±0.15	39.14±0.05	39.32±0.18
	GLS	42.25±0.25	46.19±0.26	47.52±0.08	48.15±0.08	48.63±0.41
	WMGLS	40.67±0.26	44.64±0.25	45.90±0.53	46.67±0.30	46.69±0.16
ANLPPT	PFR	0.021±0.003	0.021±0.004	0.021±0.003	0.021±0.003	0.021±0.003
	MSE	0.003±0.001	0.004±0.001	0.004±0.001	0.004±0.001	0.004±0.001
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	0.021±0.003	0.021±0.004	0.021±0.003	0.020±0.003	0.020±0.002
Det. (%)	PFR	16.6±4.6	15.3±5.0	16.3±3.9	15.4±4.2	15.6±3.6
	MSE	0.4±0.4	0.6±0.5	0.6±0.5	0.5±0.2	0.5±0.5
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	16.7±4.8	15.7±4.6	16.0±4.1	15.3±2.6	15.1±2.8
LPPL	PFR	0.276±0.005	0.277±0.001	0.274±0.004	0.278±0.005	0.277±0.001
	MSE	0.276±0.005	0.277±0.005	0.274±0.006	0.276±0.001	0.277±0.003
	GLS	0.273±0.002	0.278±0.002	0.273±0.001	0.273±0.002	0.274±0.006
	WMGLS	0.275±0.004	0.278±0.004	0.274±0.004	0.276±0.003	0.276±0.003

Table 18: **Vicuna-13B-v1.3 / GSM8K: 20-chunk windows.** Mean ± sample s.d. over three seeds; 800 prompts per window and seed. Detection averages use responses with at least 128 tokens only.

Metric	Method	1–20	21–40	41–60	61–80	81–100
AATPS	PFR	1.168±0.003	1.216±0.002	1.235±0.007	1.241±0.002	1.241±0.003
	MSE	1.174±0.006	1.223±0.005	1.242±0.002	1.248±0.003	1.252±0.004
	GLS	1.457±0.008	1.551±0.006	1.581±0.001	1.581±0.003	1.587±0.007
	WMGLS	1.464±0.006	1.555±0.004	1.583±0.011	1.593±0.004	1.591±0.005
TPS	PFR	31.02±0.04	32.53±0.07	32.91±0.16	32.89±0.34	32.92±0.23
	MSE	30.97±0.21	32.36±0.31	33.13±0.13	33.25±0.14	33.45±0.08
	GLS	35.78±0.40	38.08±0.31	38.94±0.17	38.98±0.16	39.15±0.07
	WMGLS	35.24±0.07	37.30±0.22	38.27±0.22	38.50±0.18	38.48±0.03
ANLPPT	PFR	0.025±0.004	0.025±0.003	0.025±0.003	0.024±0.002	0.024±0.003
	MSE	0.005±0.001	0.005±0.001	0.006±0.001	0.005±0.001	0.005±0.001
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	0.024±0.003	0.025±0.003	0.026±0.003	0.024±0.002	0.024±0.003
Det. (%)	PFR	18.2±4.4	18.0±2.6	19.4±3.6	17.7±2.3	15.9±1.7
	MSE	0.4±0.4	0.7±0.8	0.6±0.2	0.9±0.5	0.6±0.5
	GLS	N/A	N/A	N/A	N/A	N/A
	WMGLS	16.9±3.4	17.1±2.8	20.3±5.3	16.9±3.1	16.3±2.3
LPPL	PFR	0.267±0.008	0.266±0.003	0.267±0.005	0.264±0.011	0.262±0.009
	MSE	0.273±0.009	0.274±0.007	0.275±0.007	0.271±0.009	0.263±0.006
	GLS	0.261±0.006	0.265±0.005	0.266±0.004	0.265±0.008	0.266±0.007
	WMGLS	0.266±0.012	0.264±0.002	0.268±0.003	0.263±0.010	0.265±0.008

Table 19: **Frozen versus online WMGLS: Vicuna-7B-v1.3 / GSM8K.** Mean ± sample s.d. over two seeds, with 800 prompts per window and seed; the online rows are the runs of the same two seeds.

Metric	Head	1–20	21–40	41–60	61–80	81–100
AATPS	Frozen	$1.277_{\pm 0.012}$	$1.274_{\pm 0.018}$	$1.274_{\pm 0.012}$	$1.275_{\pm 0.010}$	$1.275_{\pm 0.010}$
	Online	$1.451_{\pm 0.011}$	$1.540_{\pm 0.010}$	$1.567_{\pm 0.005}$	$1.582_{\pm 0.001}$	$1.583_{\pm 0.002}$
TPS	Frozen	$37.77_{\pm 0.25}$	$37.83_{\pm 0.44}$	$37.87_{\pm 0.29}$	$37.78_{\pm 0.20}$	$37.86_{\pm 0.32}$
	Online	$42.78_{\pm 0.50}$	$45.39_{\pm 0.34}$	$46.19_{\pm 0.19}$	$46.59_{\pm 0.14}$	$46.74_{\pm 0.16}$
ANLPPT	Frozen	$0.040_{\pm 0.004}$	$0.040_{\pm 0.002}$	$0.039_{\pm 0.002}$	$0.038_{\pm 0.001}$	$0.038_{\pm 0.005}$
	Online	$0.040_{\pm 0.004}$	$0.040_{\pm 0.001}$	$0.039_{\pm 0.002}$	$0.038_{\pm 0.002}$	$0.038_{\pm 0.004}$
Det. (%)	Frozen	$34.3_{\pm 5.2}$	$35.1_{\pm 1.0}$	$33.0_{\pm 1.3}$	$35.6_{\pm 3.0}$	$32.4_{\pm 5.9}$
	Online	$35.1_{\pm 6.3}$	$35.2_{\pm 0.5}$	$32.4_{\pm 1.7}$	$35.0_{\pm 3.1}$	$31.9_{\pm 5.3}$
LPPL	Frozen	$0.376_{\pm 0.011}$	$0.371_{\pm 0.009}$	$0.378_{\pm 0.006}$	$0.374_{\pm 0.004}$	$0.385_{\pm 0.001}$
	Online	$0.378_{\pm 0.003}$	$0.373_{\pm 0.010}$	$0.382_{\pm 0.007}$	$0.374_{\pm 0.003}$	$0.386_{\pm 0.008}$

Table 20: **Frozen versus online WMGLS: Vicuna-7B-v1.3 / CodeSearchNet-Python.** Mean ± sample s.d. over two seeds, with 800 prompts per window and seed; the online rows are the runs of the same two seeds.

Metric	Head	1–20	21–40	41–60	61–80	81–100
AATPS	Frozen	$1.128_{\pm 0.004}$	$1.127_{\pm 0.001}$	$1.127_{\pm 0.004}$	$1.128_{\pm 0.003}$	$1.128_{\pm 0.002}$
	Online	$1.371_{\pm 0.020}$	$1.470_{\pm 0.018}$	$1.503_{\pm 0.027}$	$1.515_{\pm 0.020}$	$1.520_{\pm 0.030}$
TPS	Frozen	$33.97_{\pm 0.02}$	$33.96_{\pm 0.05}$	$33.89_{\pm 0.08}$	$33.85_{\pm 0.01}$	$33.73_{\pm 0.00}$
	Online	$40.69_{\pm 0.47}$	$43.41_{\pm 0.22}$	$44.40_{\pm 0.22}$	$44.94_{\pm 0.05}$	$44.87_{\pm 0.05}$
ANLPPT	Frozen	$0.069_{\pm 0.006}$	$0.066_{\pm 0.008}$	$0.064_{\pm 0.010}$	$0.066_{\pm 0.006}$	$0.066_{\pm 0.007}$
	Online	$0.069_{\pm 0.005}$	$0.067_{\pm 0.007}$	$0.064_{\pm 0.009}$	$0.066_{\pm 0.006}$	$0.066_{\pm 0.008}$
Det. (%)	Frozen	$65.7_{\pm 4.3}$	$62.9_{\pm 7.3}$	$61.0_{\pm 9.1}$	$60.3_{\pm 7.0}$	$64.0_{\pm 9.3}$
	Online	$66.0_{\pm 4.6}$	$63.7_{\pm 7.9}$	$60.8_{\pm 8.8}$	$60.8_{\pm 7.2}$	$62.8_{\pm 9.2}$
LPPL	Frozen	$0.537_{\pm 0.017}$	$0.529_{\pm 0.013}$	$0.522_{\pm 0.015}$	$0.528_{\pm 0.017}$	$0.521_{\pm 0.025}$
	Online	$0.529_{\pm 0.014}$	$0.539_{\pm 0.017}$	$0.520_{\pm 0.022}$	$0.535_{\pm 0.012}$	$0.531_{\pm 0.020}$

Table 21: **Frozen versus online WMGLS: Vicuna-7B-v1.3 / Finance-Alpaca.** Mean ± sample s.d. over two seeds, with 800 prompts per window and seed; the online rows are the runs of the same two seeds.

Metric	Head	1–20	21–40	41–60	61–80	81–100
AATPS	Frozen	$1.107_{\pm 0.005}$	$1.110_{\pm 0.009}$	$1.110_{\pm 0.008}$	$1.107_{\pm 0.006}$	$1.108_{\pm 0.004}$
	Online	$1.319_{\pm 0.001}$	$1.416_{\pm 0.003}$	$1.452_{\pm 0.002}$	$1.466_{\pm 0.002}$	$1.472_{\pm 0.003}$
TPS	Frozen	$33.31_{\pm 0.41}$	$33.32_{\pm 0.30}$	$33.32_{\pm 0.29}$	$33.36_{\pm 0.39}$	$33.36_{\pm 0.28}$
	Online	$39.37_{\pm 0.39}$	$42.08_{\pm 0.44}$	$43.37_{\pm 0.39}$	$43.89_{\pm 0.23}$	$44.08_{\pm 0.31}$
ANLPPT	Frozen	$0.065_{\pm 0.008}$	$0.064_{\pm 0.009}$	$0.066_{\pm 0.008}$	$0.064_{\pm 0.006}$	$0.065_{\pm 0.003}$
	Online	$0.065_{\pm 0.009}$	$0.064_{\pm 0.009}$	$0.066_{\pm 0.008}$	$0.065_{\pm 0.006}$	$0.065_{\pm 0.003}$
Det. (%)	Frozen	$67.4_{\pm 4.8}$	$65.8_{\pm 7.1}$	$68.0_{\pm 6.8}$	$67.4_{\pm 6.1}$	$66.1_{\pm 3.6}$
	Online	$66.5_{\pm 5.1}$	$66.4_{\pm 7.3}$	$69.2_{\pm 6.8}$	$67.8_{\pm 6.9}$	$66.1_{\pm 5.0}$
LPPL	Frozen	$0.589_{\pm 0.039}$	$0.592_{\pm 0.025}$	$0.584_{\pm 0.021}$	$0.585_{\pm 0.028}$	$0.589_{\pm 0.009}$
	Online	$0.591_{\pm 0.040}$	$0.593_{\pm 0.017}$	$0.584_{\pm 0.018}$	$0.588_{\pm 0.031}$	$0.590_{\pm 0.005}$