

Supplementary material for HeavyWater and SimplexWater: Watermarking Low-Entropy Text Distributions

Table of contents:

- **A** Additional Information on Related Works.
 - **A.1** Additional Information on Related Work.
 - **A.2** Instantiation of Existing Watermarks as Score, Distribution and Randomness Design.
 - **A.3** Discussion on Randomness Efficiency.
- **B** Proofs for Theorems from Sections 3 & 4.
 - **B.1** Proof of Proposition 1.
 - **B.2** Proof of Theorem 1.
 - **B.3** Proof of Theorem 2.
 - **B.4** Proof of Theorem 3.
 - **B.5** Proof of Theorem 4.
- **C** Additional Information and Implementation Details for HeavyWater and SimplexWater.
 - **C.1** Low-Entropy Distributions in LLMs.
 - **C.2** Theoretical Effect of Different Tails of Distributions.
 - **C.3** Q-ary Code.
 - **C.4** Implementation Details.
 - **C.5** Information on Optimal Transport and Sinkhorn’s Algorithm.
- **D** Additional Numerical Results and Ablation Study.
 - **D.1** Ablation Study.
 - **D.2** Impact of Non-i.i.d. Side Information Generation.
 - **D.3** Experiment: Robustness to Textual Attacks.
 - **D.4** Computational Overhead.
 - **D.5** Experiment: Alternative Text Generation Metrics.
 - **D.6** Alternative Detection Metric.
 - **D.7** Additional Detection–Distortion Trade-off Results.

A Additional Information on Related Works

A.1 Additional Information on Related Work

Optimization Framework Several optimization frameworks have been proposed for watermark analysis. [16, 25, 26, 52] adopts a hypothesis-testing framework for analyzing the statistical power of watermarking schemes. [52] goes beyond the vanilla threshold test to determine the optimal detection rule by solving a minimax optimization program. The authors of [25] consider an optimization under an additional constraint of controlled false-positive error. The watermarking scheme follows by learning a coupling that spreads the LLM distribution into an auxiliary sequence of variables (with alphabet size greater than m). While the proposed scheme is the optimal solution for the considered optimization, the scheme requires access to the (proxy of) LLM logits on both generation and detection ends. The authors of [53] consider a multi-objective optimization — maximization of the green list bias while minimizing the log-perplexity. Building on the hypothesis testing framework, we optimize over classes of score functions, by observing that the detection power of a watermark boils down to the separation of expected scores between the null and alternative hypotheses.

Hashing Schemes in LLM Watermarking Current LLM watermarking schemes derive their per-token pseudorandom seed through five recurring hashing patterns. **LeftHash** hashes the immediately preceding token and feeds the digest to a keyed Pseudorandom function (PRF), yielding a light-weight, self-synchronising seed that survives single-token edits [15]. **Window-hash** generalises this by hashing the last h tokens, expanding the key-space and hindering brute-force list enumeration at the cost of higher edit sensitivity [7, 15]. **SelfHash** (sometimes dubbed right-hand hashing”) appends the *candidate* token to the left context before hashing, so the seed depends on both history and the token being scored; this hardens the scheme against key-extraction attacks and is used in multi-bit systems such as MPAC [31]. Orthogonally, **counter-based keystreams** drop context altogether and set the seed to $\text{PRF}(K, \text{position})$, a strategy adopted by inverse-transform and Gumbel watermarks to preserve the original LM distribution in expectation [13, 17]. Finally, **adaptive sliding-window hashing**—popularised by *SynthID-Text*—hashes the last $H = 4$ tokens together with the secret key and *skips watermarking whenever that exact window has appeared before* (K -sequence repeated context masking”), thereby avoiding repetition artefacts while retaining the robustness benefits of a short window [11]. Semantic extensions build on these primitives: SemaMark quantises sentence embeddings [32], while Semantic-Invariant Watermarking uses a learned contextual encoder [54]. Collectively, these hashing families balance secrecy, robustness to editing or paraphrasing, and computational overhead, offering a rich design space for practical watermark deployments. Both SimplexWater and HeavyWater are agnostic to and can be applied on top of any hashing scheme. We provide the detection gain over Red-Green using various hashing schemes in Fig. [3b].

Additional Distortion-Free Watermarks An array of recent works propose distortion-free watermarks that preserve the original LLM distribution [13, 16, 17, 20, 48, 55–57]. In addition to the ones that we have introduced in Section [1], [56] constructs undetectable watermarks using one-way functions, which is a cryptography-inspired technique. [57] proposes a watermark using error-correcting codes that leverages double-symmetric binary channels to obtain the watermarked distribution. [48] designs a distortion-free watermark based on multiple draws from a black-box LLM, which involves fixing a score function, drawing multiple tokens in each step, and outputting the highest-score token.

A.2 Instantiation of Existing Watermarks As Score, Distribution and Randomness Design

Existing watermarks can be represented under the proposed outlook of side information, score function and watermarked distribution from Section [2]. We next demonstrate that by instantiating several popular schemes through our lenses.

The **Red-Green watermark** [15] randomly partitions \mathcal{X} into a green list and a red list. Here, S is a set of m binary random variables, representing the random list assignment. The function f has binary outputs, which are used to increase the probability of green list tokens through exponential tilting of P_X .

The **SynthID** watermark [11] employs tournament sampling: a tournament between a set of N^m token candidates along m -layers with N competing token groups. Each tournament is performed given a sample of shared randomness (denoted r [11]). On the ℓ th layer the winners are taken to

Watermark	# Bits
Red-Green [15]	m
Inverse Transform [13]	mF
Gumbel [17]	mF
SimplexWater (ours)	$\log(m)$
HeavyWater (ours)	$\log(k)$

Table A.1: Amount of random bits generated per step in popular watermarks. m is the vocabulary size, F is the floating point precision and k is the side information alphabet size.

be the token with highest score $f_\ell(x, s)$ within each N -sized set. The value of the score function f is obtained from the side information for each token candidate. Different domains of f induce a different construction of the function (See [11] Appendix E) for more information). The watermarked distribution $P_{X|S=s}$ is obtained with a closed form in specific cases of (f, N, m) and can be directly used instead of the instantiating a tournament. This form of watermarking is termed *vectorized tournament sampling*, in which the tournament is not applied, but the induced conditional distribution is employed instead. In this paper we consider vectorized tournament sampling with binary-valued scores, as this is the formulation given in [11] with a closed form, and the one that was utilized in their proposed experiments.

In the **Gumbel watermark** [17], the side information consists of m i.i.d., uniform random variables on $[0, 1]$, each one corresponding to a single element in the vocabulary $x \in \mathcal{X}$. The score function f is obtained by assigning each $x \in \mathcal{X}$ with a value $l_x = -\frac{1}{P_X(x)} \log(u_x)$, where u_x the uniform variable corresponding to x and $P_X(x)$ is the probability of $x \in \mathcal{X}$ under the unwatermarked model. The watermarked distribution is then given by assigning probability 1 to $\arg\max_{x \in \mathcal{X}} l_x(x)$ and probability 0 to the rest of $x \in \mathcal{X}$ (i.e., a singleton distribution).

In the **Correlated Channel** watermark [16], the side information corresponds to a partition of \mathcal{X} into $k \geq 2$ lists and a single shared uniform variables S' that is uniformly distributed on $[1 : k]$. The score function is then an indicator of the matching between the additional variable realization and the token random assignment, i.e., $f(x, s) = \mathbf{1}(B(x) = s)$, where $B(x) \in [1 : k]$ is the assignment of x into one of the k lists. The watermarked distribution is then given in closed form by solving the maximum coupling problem.

A.3 Discussion on Randomness Efficiency

Given a hashing procedure that determines the random number generator seed value, we sample the side information $S \sim P_S$ over \mathcal{S} . The size of \mathcal{S} determines the amount of side information we are required to sample. As described in Appendix A.2, each watermarks corresponds to a different size fo \mathcal{S} . We interpret that as *randomness efficiency*. That is, the bigger \mathcal{S} is, the more bits of randomness we are required to extract from the random seed. We argue that a byproduct of our method is randomness efficiency, i.e., SimplexWater and HeavyWater require less random bits to sampled from the random seed compared to existing schemes. To that end, we compare with several popular watermarks (see Table A.1 for a summary).

The Red-Green watermark [15] corresponds to sampling a single bit for each element in \mathcal{X} , whose value determines the token's list assignment (red or green), thus resulting in a total of m bits. The inverse transform watermark [13] requires sampling a single uniform and sampling a random permutation of $[1 : m]$. Thus, it asymptotically requires $F \log(m!)$ bits, where F is the resolution of the floating point representation used to sample the sampled uniform variable in bits (e.g. 32 for float32). For the Gumbel watermark, we sample a uniform variable for each $x \in \mathcal{X}$, resulting in a total of mF bits.

In SimplexWater, the side information size is $|\mathcal{S}| = m - 1$. To that end, we required $\log(m)$ bits to sample a single $s \in [1 : m - 1]$. Furthermore, HeavyWater is not constrained to a specific size of $|\mathcal{S}| = k$, and in the proposed experiments we take $k = 1024$ which is significantly smaller than both m and 2^F . The resulting amount of bits to be sampled from the random seed is $\log(1024) = 10$ bits. Consequently, we observe that HeavyWater is the most randomness-efficient watermark across considered schemes, while also being the best-performing watermark across considered experiments (see Section 5).

Minimizing the number of random bits extracted from a random number generator directly reduces computational and energy overhead in watermark embedding as less information is to be stored on the GPU. This eases implementation in resource-constrained hardware by lowering entropy demands and memory usage. Additionally, it limits side-channel leakage by shrinking an adversary's observable output [58], which may lead to more secure watermarking. We leave an extensive study and quantification of the benefits of randomness efficiency to future work.

B Proofs for Theorems from Section 3 and 4

We prove Proposition 1 Theorem 1 Theorem 2 Theorem 3 and Theorem 4

B.1 Proof of Proposition 1

In this section, we prove Proposition 1 which connects the watermark design problem to a coding theoretic problem when the score function class is chosen to be binary.

Proposition 1 (restated): Let $\lambda \in [\frac{1}{2}, 1)$. For $f \in \mathcal{F}_{\text{bin}}$, define the vector $f_i = [f(i, 1), \dots, f(i, k)] \in \{0, 1\}^k$ for each $i \in \mathcal{X}$. Then,

$$D_{\text{gap}}(m, k, \lambda, \mathcal{F}_{\text{bin}}) = \max_{f \in \mathcal{F}_{\text{bin}}} \min_{i, j \in \mathcal{X}, i \neq j} \frac{(1 - \lambda)d_H(f_i, f_j)}{k}, \quad (\text{B.9})$$

where $d_H(a, b) = \sum_{i=1}^k \mathbf{1}_{\{a_i \neq b_i\}}$ denotes the Hamming distance between $a, b \in \{0, 1\}^k$ and $\mathbf{1}_{\{\cdot\}}$ is the indicator function.

Proof of Prop. 1. Recall that we consider $P_S = \text{Unif}[1:k]$. For any P_X , let $\Psi(P_X) = \max_{P_{XS}} (\mathbb{E}_{P_{XS}} [f(X, S)] - \mathbb{E}_{P_X P_S} [f(X, S)])$.

Claim 1. $\arg\min_{P \in \mathcal{P}_\lambda} \Psi(P) \in \mathcal{P}_{\text{spike}, \lambda}$, where

$$\mathcal{P}_{\text{spike}, \lambda} = \{P \in \Delta_m \mid \{P_X(x_1), P_X(x_2), \dots, P_X(x_m)\} = \{\lambda, 1 - \lambda, 0, 0, \dots, 0\}\}, \quad (\text{B.10})$$

where $P_X(x_i)$ denotes the x_i th element of the P_X probability vector with (x_1, \dots, x_m) representing any permutation of $(1, \dots, m)$. Δ_m is the m -dimensional probability simplex.

Proof of Claim 1. To prove Claim 1 we first show that $\Psi(P_X)$ is concave in P_X . By definition, for any given P_X and uniform P_S , we have,

$$\Psi(P_X) = \max_{P_{XS}} (\mathbb{E}_{P_{XS}} [f(X, S)] - \mathbb{E}_{P_X P_S} [f(X, S)]).$$

Define P_{XS}^* as the optimal joint distribution that achieves the maximum for a given P_X . Then, we have:

$$\Psi(P_X) = \mathbb{E}_{P_{XS}^*} [f(X, S)] - \mathbb{E}_{P_X P_S} [f(X, S)].$$

Now, consider the mixture $P_X^\theta = \theta P_X^{(1)} + (1 - \theta) P_X^{(2)}$ for some $\theta \in [0, 1]$ and $P_X^{(1)}, P_X^{(2)} \in \Delta_m$. Given $P_{XS}^{(1)*}$ and $P_{XS}^{(2)*}$, the maximizing couplings of $\Psi(P_X^{(1)})$ and $\Psi(P_X^{(2)})$ respectively, we define a mixed joint distribution: $P_{XS}^{\theta*} = \theta P_{XS}^{(1)*} + (1 - \theta) P_{XS}^{(2)*}$. Note that,

$$\sum_s P_{XS}^{\theta*} = \theta \sum_s P_{XS}^{(1)*} + (1 - \theta) \sum_s P_{XS}^{(2)*} \quad (\text{B.11})$$

$$= \theta P_X^{(1)} + (1 - \theta) P_X^{(2)} \quad (\text{as } P_{XS}^{(i)*} \text{ is the optimal coupling of } P_X^{(i)} \text{ and } P_S.) \quad (\text{B.12})$$

$$= P_X^{(\theta)} \quad (\text{B.13})$$

$$\sum_x P_{XS}^{\theta*} = \theta \sum_x P_{XS}^{(1)*} + (1 - \theta) \sum_x P_{XS}^{(2)*} \quad (\text{B.14})$$

$$= \frac{\theta}{k} + (1 - \theta) P_X^{(2)} \quad (\text{as } P_{XS}^{(i)*} \text{ is the optimal coupling of } P_X^{(i)} \text{ and } P_S.) \quad (\text{B.15})$$

$$= \frac{1}{k} \quad (\text{B.16})$$

which shows that $P_{XS}^{\theta*}$ is a valid coupling of $P_X^{(\theta)}$ and uniform P_S . Using the linearity of the expectation operation, the expectation under the mixed distribution $P_{XS}^{\theta*}$ is:

$$\mathbb{E}_{P_{XS}^{\theta*}}[f(X, S)] = \theta \mathbb{E}_{P_X^{(1)*}}[f(X, S)] + (1 - \theta) \mathbb{E}_{P_X^{(2)*}}[f(X, S)].$$

Similarly, for the independent case:

$$\mathbb{E}_{P_X^\theta P_S}[f(X, S)] = \theta \mathbb{E}_{P_X^{(1)} P_S}[f(X, S)] + (1 - \theta) \mathbb{E}_{P_X^{(2)} P_S}[f(X, S)].$$

1040 Since $\Psi(P_X^\theta)$ by definition takes the maximum coupling among all P_{XS} , it upper bounds the value
1041 attained by the specific mixture $P_{XS}^{\theta*}$. Thus,

$$\begin{aligned} \Psi(P_X^\theta) &\geq \mathbb{E}_{P_{XS}^{\theta*}}[f(X, S)] - \mathbb{E}_{P_X^\theta P_S}[f(X, S)] \\ &= \theta \mathbb{E}_{P_X^{(1)*}}[f(X, S)] + (1 - \theta) \mathbb{E}_{P_X^{(2)*}}[f(X, S)] - \theta \mathbb{E}_{P_X^{(1)} P_S}[f(X, S)] \\ &\quad - (1 - \theta) \mathbb{E}_{P_X^{(2)} P_S}[f(X, S)] \\ &= \theta \Psi(P_X^{(1)}) + (1 - \theta) \Psi(P_X^{(2)}), \end{aligned}$$

1042 which proves concavity.

1043 We now show that for any $\lambda \in [\frac{1}{2}, 1]$, the minimizer of $\Psi(P_X)$ lies in the set $\mathcal{P}_{\text{spike}, \lambda}$, where

$$\mathcal{P}_{\text{spike}, \lambda} = \{P \in \Delta_m \mid \{P_X(x_1), P_X(x_2), \dots, P_X(x_m)\} = \{\lambda, 1 - \lambda, 0, 0, \dots, 0\}\}, \quad (\text{B.17})$$

1044 Recall that $\mathcal{P}_\lambda = \{P \in \Delta_m, \|P\|_\infty \leq \lambda\}$. This is a convex set and the extreme points of this set are
1045 precisely the spike distributions in $\mathcal{P}_{\text{spike}, \lambda}$, which correspond to permutations of $(\lambda, 1 - \lambda, 0, \dots, 0)$.
1046 Since $\Psi(P_X)$ is concave, its minimum over the convex set \mathcal{P}_λ occurs at an extreme point of \mathcal{P}_λ .
1047 Hence, the minimizer of $\Psi(P_X)$ belongs to $\mathcal{P}_{\text{spike}, \lambda}$, which completes the proof. \square

1048 Using Claim 1, we prove Prop. 1 as follows. Let $p^* \in \mathcal{P}_{\text{spike}, \lambda}$ be the minimizer in the minimization
1049 of $D_{\text{gap}}(m, k, \lambda, \mathcal{F}_{\text{bin}})$ in (1), and let (i, j) be its non-zero entries, i.e., $p_i^* = \lambda$ and $p_j^* = 1 - \lambda$. Under
1050 such p^* and uniform P_S , any coupling can be written as

$$P_{XS}(x, s) = \begin{cases} \lambda Q_{X|S}(s \mid i) & \text{if } x = i, \\ (1 - \lambda) Q_{X|S}(s \mid j) & \text{if } x = j \\ 0 & \text{if } x \neq i, j \end{cases}$$

1051 where $Q_{S|X} = P_{XS}/P_X$ denotes the conditional distribution of shared randomness S given the
1052 selected token X . The coupling constraint for each $s \in \mathcal{S}$ becomes

$$\lambda Q_{S|X}(s \mid i) + (1 - \lambda) Q_{S|X}(s \mid j) = \frac{1}{k},$$

To that end, we represent such coupling as

$$Q_{S|X}(s \mid i) = a_s, \quad Q_{S|X}(s \mid j) = \frac{\frac{1}{k} - \lambda a_s}{1 - \lambda}$$

1053 for some set of parameters $(a_s)_{s \in \mathcal{S}}$, such that $a_s \in [0, \frac{1}{k\lambda}]$. Under this construction, considering the
1054 worst case (i, j) pair of non-zero P_X indices, we have,

$$\begin{aligned} D_{\text{gap}}(m, k, \lambda, \mathcal{F}_{\text{bin}}) &= \max_{f: \mathcal{X} \times \mathcal{S} \rightarrow [0, 1]} \min_{i \neq j} \max_{Q_{X|S}} \sum_{s=1}^k \lambda Q_{S|X}(s \mid i) f(i, s) + (1 - \lambda) Q_{S|X}(s \mid j) f(j, s) \\ &\quad - \left(\frac{\lambda}{k} f(i, s) + \frac{1 - \lambda}{k} f(j, s) \right) \end{aligned} \quad (\text{B.18})$$

$$\begin{aligned} &= \max_{f: \mathcal{X} \times \mathcal{S} \rightarrow [0, 1]} \min_{i \neq j} \max_{a_s \in [0, \frac{1}{k\lambda}]} \sum_{s=1}^k \lambda a_s f(i, s) + (1 - \lambda) \left(\frac{\frac{1}{k} - \lambda a_s}{1 - \lambda} \right) f(j, s) \\ &\quad - \left(\frac{\lambda}{k} f(i, s) + \frac{1 - \lambda}{k} f(j, s) \right) \end{aligned} \quad (\text{B.19})$$

$$= \max_{f: \mathcal{X} \times \mathcal{S} \rightarrow [0, 1]} \min_{i \neq j} \max_{a_s \in [0, \frac{1}{k\lambda}]} \sum_{s=1}^k \lambda \left(a_s - \frac{1}{k} \right) (f(i, s) - f(j, s)) \quad (\text{B.20})$$

1055 The design of the coupling P_{XS} boils down to choosing $\{a_s\}_{s \in \mathcal{S}}$ such that (B.20) is maximized.
 1056 Moreover, since $\sum_{s=1}^k Q_{S|X}(s|x) = 1$ for $x = i$ and $x = j$, we have,

$$\sum_{s=1}^k Q_{S|X}(s|x) = 1 \implies \sum_{s=1}^k a_s = 1 \quad (\text{B.21})$$

1057 For a given f and any two indices (i, j) , we characterize optimal a_s as follows. Let $m_+^{i,j} = |s : f(i, s) - f(j, s) = 1|$ and $m_-^{i,j} = |s : f(i, s) - f(j, s) = -1|$. Assume that the score functions f
 1058 satisfy $m_+^{i,j} \leq k\lambda$ and $m_-^{i,j} + m_+^{i,j} < k$. These assumptions are needed to eliminate trivial edge cases
 1059 for which the inner minimization in (B.20) is zero, which leads to zero detection (see Appendix B.1.1).
 1060 The inner-most maximum in (B.20) is obtained when:
 1061

$$a_s = \begin{cases} \frac{1}{k\lambda}, & s : f(i, s) - f(j, s) = 1 \\ 0, & s : f(i, s) - f(j, s) = -1 \\ \frac{1 - \frac{1}{k\lambda} m_+^{i,j}}{k - m_+^{i,j} - m_-^{i,j}}, & s : f(i, s) - f(j, s) = 0 \end{cases} \quad (\text{B.22})$$

1062 The optimal values of a_s are obtained by allocating the maximum probability mass to the highest
 1063 scores $f(i, s) - f(j, s)$ in (B.20). Continuing from (B.20), we have,

$$\begin{aligned} D_{\text{gap}}(m, k, \lambda, \mathcal{F}_{\text{bin}}) &= \max_{f: \mathcal{X} \times \mathcal{S} \mapsto [0,1]} \min_{i \neq j} \frac{1}{k} \sum_{s: f_i - f_j = -1} \lambda |f(i, s) - f(j, s)| \\ &\quad + \frac{1}{k} \sum_{s: f_i - f_j = 1} (1 - \lambda) |f(i, s) - f(j, s)| \end{aligned} \quad (\text{B.23})$$

$$= \max_{f: \mathcal{X} \times \mathcal{S} \mapsto [0,1]} \min_{i \neq j} \frac{1}{k} (\lambda m_-^{i,j} + (1 - \lambda) m_+^{i,j}) \quad (\text{B.24})$$

1064 Let $d_{ij} = |s : f(i, s) \neq f(j, s)|$. Then, $m_-^{i,j} + m_+^{i,j} = d_{ij}$. Let $m_-^{i,j} = \beta d_{ij}$ and $m_+^{i,j} = (1 - \beta) d_{ij}$
 1065 for some $\beta \in [0, 1]$. Continuing from (B.24), we have,

$$D_{\text{gap}}(m, k, \lambda, \mathcal{F}_{\text{bin}}) = \max_{f: \mathcal{X} \times \mathcal{S} \mapsto [0,1]} \min_{i \neq j} \min_{\beta \in [0,1]} \frac{1}{k} (\lambda \beta + (1 - \lambda)(1 - \beta)) d_{i,j} \quad (\text{B.25})$$

$$= \max_{f: \mathcal{X} \times \mathcal{S} \mapsto [0,1]} \min_{i \neq j} \min_{\beta \in [0,1]} \frac{1}{k} (\beta(2\lambda - 1) + (1 - \lambda)) d_{i,j} \quad (\text{B.26})$$

$$= \max_{f: \mathcal{X} \times \mathcal{S} \mapsto [0,1]} \min_{i \neq j} \frac{d_{i,j}}{k} (1 - \lambda) \quad (\text{B.27})$$

1066 since the inner minimum is achieved when $\beta = 0$, as $\lambda \geq \frac{1}{2}$. The proof is completed as d_{ij} is the
 1067 Hamming distance between $f(i, \cdot)$ and $f(j, \cdot)$.
 1068 □

1069 B.1.1 Edge Cases

1070 In Appendix B.1 we assumed that the score functions f satisfy $m_+^{i,j} \leq k\lambda$ and $m_+^{i,j} + m_-^{i,j} < k$. In
 1071 this section, we show the consequences of not satisfying these constraints.

1072 **Case 1: $m_+^{i,j} > k\lambda$:** In this case, we obtain the optimal values of a_s as in Appendix B.1 by assigning
 1073 the probability masses to the high score cases as follows. Let $\mathcal{J} \subset m_+^{i,j}$, $|\mathcal{J}| = k\lambda$ be any subset
 1074 of $k\lambda$ values of s , each satisfying $f(i, s) - f(j, s) = 1$. In this case, similar to case 1, the inner
 1075 maximum in (B.20) is obtained when:

$$a_s = \begin{cases} \frac{1}{k\lambda}, & s \in \mathcal{J} \\ 0, & s \notin \mathcal{J} \end{cases} \quad (\text{B.28})$$

1076 Continuing from (B.20), we have,

$$D_{\text{gap}}(m, k, \lambda, \mathcal{F}_{\text{bin}}) = \max_{f: \mathcal{X} \times \mathcal{S} \rightarrow [0,1]} \min_{i \neq j} \lambda(1 - \lambda) - \frac{\lambda(m_+^{i,j} - k\lambda)}{k} + \frac{\lambda m_-^{i,j}}{k} \quad (\text{B.29})$$

$$= \max_{f: \mathcal{X} \times \mathcal{S} \rightarrow [0,1]} \min_{i \neq j} \lambda + \frac{\lambda}{k} (m_-^{i,j} - m_+^{i,j}) \quad (\text{B.30})$$

$$= \max_{f: \mathcal{X} \times \mathcal{S} \rightarrow [0,1]} \min_{i \neq j} \min_{\beta \in [0,1]} \lambda + \frac{\lambda}{k} (2\beta - 1) d_{ij} \quad (\text{B.31})$$

$$= \max_{f: \mathcal{X} \times \mathcal{S} \rightarrow [0,1]} \min_{i \neq j} \lambda \left(1 - \frac{1}{k} d_{ij} \right) \quad (\text{B.32})$$

$$= 0 \quad (\text{B.33})$$

1077 where the last equality follows from the fact that the inner minimum is achieved when $d_{ij} = k$, i.e.,
 1078 $f(i, \cdot)$ is the all zeros vector and $f(j, \cdot)$ is the all ones vector. The score functions f that result in
 1079 such cases are uninteresting as the detection can not be improved.

1080 **Case 2:** $m_+^{i,j} \leq k\lambda$ and $m_+^{i,j} + m_-^{i,j} = k$: In this case, we obtain the optimal values of a_s as in
 1081 Appendix B.1 by assigning the probability masses to the high score cases as follows. Note that in
 1082 this case $|s : f(i, s) - f(j, s) = 0| = 0$. Therefore,

$$a_s = \begin{cases} \frac{1}{k\lambda}, & s : f(i, s) - f(j, s) = 1 \\ \frac{1 - \frac{1}{k\lambda} m_+^{i,j}}{k - m_+^{i,j}}, & s : f(i, s) - f(j, s) = -1 \end{cases} \quad (\text{B.34})$$

1083 Continuing from (B.24), we have,

$$D_{\text{gap}}(m, k, \lambda, \mathcal{F}_{\text{bin}}) = \max_{f: \mathcal{X} \times \mathcal{S} \rightarrow [0,1]} \min_{i \neq j} \frac{(1 - \lambda) m_+^{ij}}{k} - \lambda \left(\frac{1}{k} - \frac{1 - \frac{1}{k\lambda} m_+^{ij}}{k - m_+^{ij}} \right) (k - m_+^{ij}) \quad (\text{B.35})$$

$$= \max_{f: \mathcal{X} \times \mathcal{S} \rightarrow [0,1]} \min_{i \neq j} \frac{2m_+^{ij}}{k} (1 - \lambda) \quad (\text{B.36})$$

$$= 0 \quad (\text{B.37})$$

1084 where the last equality follows from the fact that the inner minimum is achieved when $m_+^{ij} = 0$, i.e.,
 1085 $f(i, \cdot)$ is the all zeros vector and $f(j, \cdot)$ is the all ones vector. The score functions f that result in
 1086 such cases are uninteresting as the detection can not be improved.

1087 B.2 Proof of Theorem 1

1088 In this section, we derive an upper bound for the detection gap in (1) using the Plotkin bound from
 1089 coding theory [14].

1090 **Theorem 1 (restated):** Consider the class of binary score functions \mathcal{F}_{bin} and uniform P_S . Then, for
 1091 any $\lambda \in [\frac{1}{2}, 1)$, the maximum detection gap can be bounded as

$$D_{\text{gap}}(m, k, \lambda, \mathcal{F}_{\text{bin}}) \leq \frac{m(1 - \lambda)}{2(m - 1)} \quad (\text{B.38})$$

1092 *Proof of Thm. 1* Thm. 1 follows directly from the Plotkin bound [14], which provides an upper
 1093 bound on the minimum normalized Hamming distance between any two codewords, considering any
 1094 code construction.

$$D_{\text{gap}}(m, k, \lambda, \mathcal{F}_{\text{bin}}) = \max_{f \in \mathcal{F}_{\text{bin}}} \min_{i, j \in \mathcal{X}, i \neq j} \frac{(1 - \lambda) d_H(f_i, f_j)}{k} \leq (1 - \lambda) \frac{m}{2(m - 1)}. \quad (\text{B.39})$$

1095

□

1096 B.3 Proof of Theorem 2

1097 In this section, we show that SimplexWater achieves the upper bound in (B.39).

1098 Theorem 2 (restated): For any $\lambda \in [\frac{1}{2}, 1)$ the maximum detection gap upper bound (4) is attained by
1099 SimplexWater.

1100 *Proof.* SimplexWater uses the Simplex code construction in Def. 1 as the score function. The
1101 simplex code achieves the Plotkin bound [14], i.e.,

$$\min_{i \neq j} \frac{d_H(f_{\text{sim}}(i, \cdot), f_{\text{sim}}(j, \cdot))}{k} = \frac{m}{2(m-1)} \quad (\text{B.40})$$

1102 Therefore,

$$D_{\text{gap}}(m, k, \lambda, \mathcal{F}_{\text{bin}}) \geq \min_{i, j \in \mathcal{X}, i \neq j} \frac{(1-\lambda)d_H(f_{\text{sim}}(i, \cdot), f_{\text{sim}}(j, \cdot))}{k} = (1-\lambda) \frac{m}{2(m-1)}. \quad (\text{B.41})$$

1103 Considering the upper and lower bounds in (B.39) and (B.41), we have,

$$D_{\text{gap}}(m, k, \lambda, \mathcal{F}_{\text{bin}}) = (1-\lambda) \frac{m}{2(m-1)}, \quad (\text{B.42})$$

1104 which is achieved by SimplexWater. \square

1105 B.4 Proof of Theorem 3

1106 In this section, we establish that the Gumbel watermark scheme [17] can be understood within our
1107 optimal transport framework, i.e., we prove Theorem 3, restated below.

1108 **Theorem 5** (Gumbel Watermark as OT). *When the score random variables $f(x, s)$, are sampled i.i.d.
1109 from $\text{Gumbel}(0, 1)$, the solution to the OT problem in (2) converges to the Gumbel watermark [17]
1110 as $|\mathcal{S}| = k \rightarrow \infty$.*

1111 To prove this, we first write down the Kantorovich dual of our optimal transport formulation and
1112 identify the dual potentials $\{\alpha_x, \beta_s\}$ that generate the arg-max coupling. We then analyze the
1113 behavior of these potentials in the limit $k \rightarrow \infty$, showing that the resulting arg-max rule converges
1114 to the classical Gumbel-Max sampling procedure. A final concentration argument ensures that
1115 the random coupling concentrates around its expectation, thereby establishing that the OT-derived
1116 sampler coincides with the Gumbel watermarking scheme.

1117 To understand this connection, we first review the Gumbel watermarking scheme. The Gumbel-Max
1118 trick states that sampling from a softmax distribution can be equivalently expressed as:

$$y_t = \arg \max_{y \in \mathcal{V}} \frac{u_t(y)}{T} + G_t(y) \quad (\text{B.43})$$

1119 where $u_t(y)$ are the logits, T is the temperature parameter, and $G_t(y) \sim \text{Gumbel}(0, 1)$ independently
1120 for each token position t and vocabulary element y . A $\text{Gumbel}(0, 1)$ random variable can be generated
1121 via:

$$G_t(y) = -\log(-\log(r_t(y))) \quad (\text{B.44})$$

1122 where $r_t(y) \sim \text{Uniform}([0, 1])$.

1123 In the Gumbel watermarking scheme, the uniform random variables are replaced with pseudo-random
1124 values:

$$r_t(y) \sim \text{Uniform}([0, 1]) \text{ (in unwatermarked model)} \quad (\text{B.45})$$

$$r_t(y) = F_{y_{t-m:t-1}, \mathbf{k}}(y) \text{ (in watermarked model)} \quad (\text{B.46})$$

1125 Here, $F_{y_{t-m:t-1}, \mathbf{k}}(y)$ uses a secret key \mathbf{k} and previously generated tokens $y_{t-m:t-1}$ to deterministi-
1126 cally generate values that appear random without knowledge of \mathbf{k} .

1127 To establish the connection to our OT framework (2), we analyze the dual formulation of the optimal
1128 transport problem. In this formulation, we seek to minimize $\sum_x P_X(x) \alpha_x + \frac{1}{k} \sum_s \beta_s$ subject to
1129 $\alpha_x + \beta_s \geq f(x, s)$. The optimal values for these dual variables satisfy specific conditions that

link to the Gumbel-Max construction. The key insight comes from the optimality condition in our framework:

$$P_X(i) = \mathbb{P}(\arg \max_j [z(j, s) - \alpha_j^*] = i) \quad (\text{B.47})$$

This can be directly mapped to the Gumbel-Max trick by identifying that $z(j, s)$ corresponds to the Gumbel noise $G_t(y)$ and α_j^* corresponds to $-\frac{u_t(y)}{T}$ (the negative normalized logits). With these identifications, the Gumbel-Max sampling expression:

$$y_t = \arg \max_{y \in \mathcal{Y}} \frac{u_t(y)}{T} + G_t(y) = \arg \max_{y \in \mathcal{Y}} [G_t(y) - (-\frac{u_t(y)}{T})] \quad (\text{B.48})$$

Takes exactly the same form as our framework's expression $\arg \max_j [z(j, s) - \alpha_j^*]$. Further, the probability that this argmax equals i is precisely $P_X(i)$ in our framework. Therefore, when $f(j, s) \sim \text{Gumbel}(0, 1)$, we will show below that our optimal transport solution exactly recovers the Gumbel watermarking scheme.

The detailed proof proceeds by analyzing the convergence of the discretized dual problem to its continuous limit as $k \rightarrow \infty$. The optimality conditions in the limit confirm that our OT framework generalizes the Gumbel watermark as a special case when scores are drawn from the Gumbel distribution. We start by defining a general optimal transport problem and its Kantorovich duality.

Definition 1 (Optimal Transport Problem). *Given probability measures μ and ν on spaces \mathcal{X} and \mathcal{Y} respectively, and a cost function $c : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, the optimal transport problem seeks to find a coupling π (a joint distribution with marginals μ and ν) that minimizes the expected cost:*

$$\inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) d\pi(x, y) \quad (\text{B.49})$$

where $\Pi(\mu, \nu)$ is the set of all probability measures on $\mathcal{X} \times \mathcal{Y}$ with marginals μ and ν .

The Kantorovich duality is a fundamental result that provides an equivalent formulation of this problem. For more details, see [59] Chapter 5].

Theorem 1 (Kantorovich Duality). *The optimal transport problem is equivalent to:*

$$\sup_{(\varphi, \psi) \in \Phi_c} \left\{ \int_{\mathcal{X}} \varphi(x) d\mu(x) + \int_{\mathcal{Y}} \psi(y) d\nu(y) \right\} \quad (\text{B.50})$$

where $\Phi_c = \{(\varphi, \psi) : \varphi(x) + \psi(y) \leq c(x, y)\}$ is the set of functions satisfying the c -inequality constraint.

This duality relationship (i.) transforms a complex constrained optimization problem over probability measures into an optimization over functions, (ii.) provides a way to certify optimality through complementary slackness conditions and (iii.) enables us to analyze the convergence properties of OT problems through the convergence of dual objective functions. In many applications, the Kantorovich duality is rewritten with the constraint reversed (potential functions sum to $\geq c$), transforming the problem into a minimization rather than maximization. We adopt this convention in our formulation below.

Primal Formulation. Let us consider an optimal transport problem described by the inner maximization of (1) – equation (2). Remember that given a pair (P_X, f) , the inner maximization in (1) amounts to an OT problem between P_X and P_S , which is set to be uniform on $[1 : k]$. There, the score function f can be equivalently denoted as the $(m \times k)$ -dimensional OT cost matrix C , which is defined as $C_{x', s'} = -f(x', s')$ for $(x', s') \in [1 : m] \times [1 : k]$. This matrix is only generated once.

Now, let P_X be a probability distribution over a finite set $\{1, 2, \dots, m\}$ and P_S be a uniform distribution with mass $1/k$ at each point in $\{1, 2, \dots, k\}$. We have a cost function $f(x, s)$ with zero mean and unit variance, i.e., $\mathbb{E}_P[z] = 0$ and $\mathbb{E}_P[z^2] = 1$. The inner optimization in (1) aims to find a coupling P_{XS} that maximizes:

$$C_{P,k}^* = \max_{P_{XS}} \sum_{x,s} P_{XS}(x, s) \cdot f(x, s) \quad (\text{B.51})$$

1168

$$\text{s.t. } \sum_s P_{XS}(x, s) = P_X(x) \quad \forall x \quad (\text{B.52})$$

$$\sum_x P_{XS}(x, s) = \frac{1}{k} \quad \forall s \quad (\text{B.53})$$

$$P_{XS}(x, s) \geq 0 \quad \forall x, s \quad (\text{B.54})$$

1169 **Dual Formulation.** In order to analyze this primal optimal transport problem we can look at the dual
1170 by using a Kantorovich duality argument. The equivalent dual problem is given by:

$$C_{D,k}^* = \min_{\alpha_x, \beta_s} \sum_x P_X(x) \alpha_x + \sum_s \frac{1}{k} \beta_s \quad (\text{B.55})$$

1171

$$\text{Subject to: } \alpha_x + \beta_s \geq f(x, s) \quad \forall x, s \quad (\text{B.56})$$

1172 In this dual formulation, α_x and β_s are the Kantorovich potentials (i.e., Lagrange multipliers) enforcing
1173 the marginal constraints $\sum_s P_{XS}(x, s) = P_X(x)$ and $\sum_x P_{XS}(x, s) = \frac{1}{k}$ respectively. For any
1174 fixed values of α_x , we need to determine the optimal values of β_s that minimize the dual objective
1175 function. Since we aim to minimize the objective and the coefficients of β_s are positive ($\frac{1}{k} > 0$), we
1176 want to make each β_s as small as possible while still satisfying the constraints. For a given s , the
1177 constraint becomes:

$$\beta_s \geq f(x, s) - \alpha_x \quad \forall x \quad (\text{B.57})$$

1178 This means that β_s must be at least as large as $f(x, s) - \alpha_x$ for every value of x . To ensure all
1179 constraints are satisfied while keeping β_s minimal, we set:

$$\beta_s = \max_{x'} [z(x', s) - \alpha_{x'}] \quad (\text{B.58})$$

1180 This is the smallest value of β_s that satisfies all constraints for a given s . Any smaller value would
1181 violate at least one constraint, and any larger value would unnecessarily increase the objective
1182 function. Substituting this expression back into the dual objective yields:

$$C_{D,k}^* = \min_{\alpha_x} \sum_{x=1}^m P_X(x) \alpha_x + \frac{1}{k} \sum_{s=1}^k \max_{x'} [z(x', s) - \alpha_{x'}] \quad (\text{B.59})$$

1183 This reformulation reduces the dual problem to an unconstrained minimization over α_x only.

1184 To establish convergence of the dual problem—which is the key to embedding the Gumbel watermark-
1185 ing scheme in our framework—we will first recall a few fundamental concepts from variational analysis
1186 and convex optimization.

1187 **Definition 2** (Epi-convergence). *A sequence of functions $f_k : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\pm\infty\}$ is said to epi-
1188 converge to a function $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\pm\infty\}$ if the following two conditions hold:*

1189 (i) *For every $x \in \mathbb{R}^n$ and every sequence $x_k \rightarrow x$, $\liminf_{k \rightarrow \infty} f_k(x_k) \geq f(x)$.*

1190 (ii) *For every $x \in \mathbb{R}^n$, there exists a sequence $x_k \rightarrow x$ such that $\limsup_{k \rightarrow \infty} f_k(x_k) \leq f(x)$.*

1191 Epi-convergence is particularly important in optimization because it guarantees that minimizers and
1192 minimal values converge appropriately. For more details on epi-convergence, see [60].

1193 **Definition 3** (Equi-lower semicontinuity). *A family of functions $\{f_k\}$ is equi-lower semicontinuous
1194 if for every $x \in \mathbb{R}^n$ and every $\varepsilon > 0$, there exists a neighborhood V of x such that*

$$\inf_{y \in V} f_k(y) > f_k(x) - \varepsilon \quad (\text{B.60})$$

1195 *for all k .*

1196 The following result connects pointwise convergence with epi-convergence for convex functions:

1197 **Theorem 2** ([61] Theorem 2]). If $\{f_k\}$ is a sequence of convex, continuous, and equi-lower semi-
 1198 continuous functions that converge pointwise to a function f on \mathbb{R}^n , then $\{f_k\}$ epi-converges to
 1199 f .

1200 As a first step toward applying our framework to the Gumbel watermarking scheme, we now
 1201 characterize how the optimal transport cost behaves in the limit $k \rightarrow \infty$.

1202 **Theorem 3.** As $k \rightarrow \infty$, the optimal value of the discrete problem converges to the expected value
 1203 problem:

$$C_{D,k}^* \rightarrow C_D^* = \min_{x \in \mathbb{R}^m} p^T x + \mathbb{E}[\max_i (z_i - x_i)] \quad (\text{B.61})$$

1204 where $p_i = P_X(i)$ and z_i are random variables with the distribution matching the problem's cost
 1205 function.

1206 *Proof of Theorem 3* Let us rewrite the objective function in vector notation:

$$f_k(x) = p^T x + \frac{1}{k} \sum_{j=1}^k \max_i [\bar{z}_j^i - x_i] \quad (\text{B.62})$$

1207 where $\bar{z}_j = [z_{1,j}, \dots, z_{m,j}]$ and $x = [\alpha_1, \dots, \alpha_m]$.

1208 We need to establish that the functions f_k are continuous and convex. Note that:

- 1209 1. The term $p^T x$ is linear and therefore continuous and convex.
- 1210 2. The function $g_j(x) = \max_i [\bar{z}_j^i - x_i]$ is continuous for each j because:
 - 1211 (a) The functions $h_i(x) = \bar{z}_j^i - x_i$ are continuous for each i .
 - 1212 (b) The maximum of a finite number of continuous functions is continuous. It is also
 - 1213 convex as the maximum of linear functions.
- 1214 3. The sum $\frac{1}{k} \sum_{j=1}^k g_j(x)$ is continuous as a linear combination of continuous functions and
- 1215 convex as a positive linear combination of convex functions.

1216 As $k \rightarrow \infty$, by the Law of Large Numbers, for any fixed x , the sample average converges to the
 1217 expected value:

$$\frac{1}{k} \sum_{j=1}^k \max_i [\bar{z}_j^i - x_i] \rightarrow \mathbb{E}[\max_i (z_i - x_i)] \quad (\text{B.63})$$

1218 Therefore,

$$f_k(x) \rightarrow f(x) = p^T x + \mathbb{E}[\max_i (z_i - x_i)] \quad (\text{B.64})$$

1219 This establishes pointwise convergence of f_k to f . Next, we need to prove that the family of functions
 1220 $\{f_k\}$ is equi-lower semicontinuous.

1221 **Lemma B.1.** The family of functions $\{f_k\}$ defined above is equi-lower semicontinuous under mild
 1222 assumptions on the boundedness of the data points \bar{z}_j .

1223 *Proof.* First, observe that each f_k is continuous (and hence lower semicontinuous). For any $x \in \mathbb{R}^n$
 1224 and $\varepsilon > 0$, consider the neighborhood

$$V = \{y : \|y - x\|_\infty < \varepsilon/2\} \quad (\text{B.65})$$

1225 For any $y \in V$ and any j, i , we have

$$\bar{z}_j^i - y_i > \bar{z}_j^i - x_i - \varepsilon/2 \quad (\text{B.66})$$

1226 Therefore,

$$\max_i [\bar{z}_j^i - y_i] \geq \max_i [\bar{z}_j^i - x_i - \varepsilon/2] = \max_i [\bar{z}_j^i - x_i] - \varepsilon/2 \quad (\text{B.67})$$

1227 This implies

$$\frac{1}{k} \sum_{j=1}^k \max_i [\bar{z}_j^i - y_i] \geq \frac{1}{k} \sum_{j=1}^k \max_i [\bar{z}_j^i - x_i] - \varepsilon/2 \quad (\text{B.68})$$

1228 Combined with the linear term, we get

$$f_k(y) = p^T y + \frac{1}{k} \sum_{j=1}^k \max_i [\bar{z}_j^i - y_i] \quad (\text{B.69})$$

$$\geq p^T x - \|p\|_1 \cdot \varepsilon/2 + \frac{1}{k} \sum_{j=1}^k \max_i [\bar{z}_j^i - x_i] - \varepsilon/2 \quad (\text{B.70})$$

$$= f_k(x) - \|p\|_1 \cdot \varepsilon/2 - \varepsilon/2 \quad (\text{B.71})$$

1229 By choosing ε small enough, we ensure that $\|p\|_1 \cdot \varepsilon/2 + \varepsilon/2 < \varepsilon$, which gives us

$$\inf_{y \in V} f_k(y) > f_k(x) - \varepsilon \quad (\text{B.72})$$

1230 This holds for all k , establishing the equi-lower semicontinuity of the family $\{f_k\}$. \square

1231 Since we have established that the functions f_k are convex, continuous, and equi-lower semicontinuous, and that they converge pointwise to f , we can apply Lemma 2 to conclude that f_k epi-converges to f . By the fundamental properties of epi-convergence, if f_k epi-converges to f , then $\min f_k \rightarrow \min f$. Also, every limit point of minimizers of f_k is a minimizer of f . This establishes that $C_{D,k}^* \rightarrow C_D^*$ as $k \rightarrow \infty$. \square

1236 **Optimality Conditions and Characterization.** Now, we analyze the optimality conditions for the
1237 dual problem. Consider our objective function:

$$f_k(\alpha) = \sum_{i=1}^m P_X(i) \alpha_i + \frac{1}{k} \sum_{s=1}^k \max_j [f(j, s) - \alpha_j] \quad (\text{B.73})$$

1238 To find the derivative with respect to α_i , we note that the derivative of the first term is simply $P_X(i)$.
1239 For the second term, we need to understand how the max function behaves. At points where a unique
1240 index j achieves the maximum value of $f(j, s) - \alpha_j$, the derivative is:

$$\frac{\partial}{\partial \alpha_i} \max_j [f(j, s) - \alpha_j] = \begin{cases} -1 & \text{if } i = \operatorname{argmax}_j [f(j, s) - \alpha_j] \\ 0 & \text{otherwise} \end{cases} \quad (\text{B.74})$$

1241 The negative sign appears because α_i has a negative coefficient in the expression inside the
1242 max. For simplicity, we often write the optimality condition using the indicator function $\mathbb{1}[i \in$
1243 $\operatorname{argmax}_j [f(j, s) - \alpha_j]]$, which equals 1 when i is in the argmax set and 0 otherwise.

$$\frac{\partial}{\partial \alpha_i} \max_j [f(j, s) - \alpha_j] = -\mathbb{1}[i = \operatorname{argmax}_j [f(j, s) - \alpha_j]] \quad (\text{B.75})$$

1244 If there are ties (multiple indices achieve the maximum), then the max function is not differentiable.
1245 Instead, we use the concept of subdifferential. A valid subgradient can be written as $-\mathbb{1}[i \in$
1246 $\operatorname{argmax}_j [f(j, s) - \alpha_j]] \cdot w_i$, where $w_i \geq 0$ are weights such that $\sum_{i \in \operatorname{argmax}} w_i = 1$.

1247 The derivative (or subgradient) of the entire objective function with respect to α_i is:

$$\frac{\partial f_k}{\partial \alpha_i} = P_X(i) - \frac{1}{k} \sum_{s=1}^k \mathbb{1}[i \in \operatorname{argmax}_j [f(j, s) - \alpha_j]] \quad (\text{B.76})$$

1248 Then, for the optimal dual variables α^* , the first-order optimality condition gives:

$$\frac{\partial f_k}{\partial \alpha_i}(\alpha^*) = P_X(i) - \frac{1}{k} \sum_{s=1}^k \mathbb{1}[i \in \operatorname{argmax}_j [f(j, s) - \alpha_j^*]] = 0 \quad (\text{B.77})$$

1249 We interpret this optimality condition as follows: the probability $P_X(i)$ equals the empirical probability
1250 that i is in the argmax set of $f(j, s) - \alpha_j^*$ across all samples s . Rearranging, we have:

$$\frac{1}{k} \sum_{s=1}^k \mathbb{1}[i \in \operatorname{argmax}_j [f(j, s) - \alpha_j^*]] = P_X(i) \quad (\text{B.78})$$

1251 To fully understand the emergence of the Gumbel-Max connection, we must carefully analyze how
1252 the discrete optimality condition converges to its continuous counterpart as $k \rightarrow \infty$. The key step is
1253 understanding how the empirical average on Equation [B.78](#) becomes the probability statement:

$$\Pr[\operatorname{argmax}_j (z_j - \alpha_j^*) = i] = P_X(i) \quad (\text{B.79})$$

1254 In other words, it remains to show that

$$\frac{1}{k} \sum_{s=1}^k \mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_j^*]\} \rightarrow \mathbb{E}[\mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_j^*]\}]. \quad (\text{B.80})$$

1255 In order to show this convergence, let α_k^* denote the optimal dual variables for the problem with k
1256 samples. From the epi-convergence of f_k to f , we know that $\alpha_k^* \rightarrow \alpha^*$ as $k \rightarrow \infty$.

1257 For any fixed α , the indicator functions $\mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_j]\}$ are independent and identically
1258 distributed across s , since $f(j, s)$ are sampled independently. Therefore, by the Strong Law of Large
1259 Numbers:

$$\frac{1}{k} \sum_{s=1}^k \mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_j]\} \xrightarrow{a.s.} \mathbb{E}[\mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_j]\}] \quad (\text{B.81})$$

1260 To address the case where α is replaced by α_k^* which depends on the samples, we decompose:

$$\left| \frac{1}{k} \sum_{s=1}^k \mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_{k,j}^*]\} - \mathbb{E}[\mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_j^*]\}] \right| \quad (\text{B.82})$$

$$\leq \left| \frac{1}{k} \sum_{s=1}^k \mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_{k,j}^*]\} - \frac{1}{k} \sum_{s=1}^k \mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_j^*]\} \right| \quad (\text{B.83})$$

$$+ \left| \frac{1}{k} \sum_{s=1}^k \mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_j^*]\} - \mathbb{E}[\mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_j^*]\}] \right| \quad (\text{B.84})$$

1261 The second term converges to zero by the Strong Law of Large Numbers. For the first term, we
1262 exploit the fact that the argmax function is stable under small perturbations except at ties, which
1263 occur with probability zero for continuous distributions of z .

1264 Specifically, let $\Delta_k = \|\alpha_k^* - \alpha^*\|_\infty$. For any realization of $f(j, s)$, the argmax changes only if the
1265 perturbation Δ_k exceeds the minimum gap between the maximum value and the second-largest value.
1266 Let $G_s = \min_{j \neq j^*} [z(j^*, s) - \alpha_{j^*}^*] - [f(j, s) - \alpha_j^*]$, where $j^* = \operatorname{argmax}_j [f(j, s) - \alpha_j^*]$. Then:

$$\mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_{k,j}^*]\} \neq \mathbb{1}\{i \in \operatorname{argmax}_j [f(j, s) - \alpha_j^*]\} \implies \Delta_k > G_s \quad (\text{B.85})$$

1267 Since $\alpha_k^* \rightarrow \alpha^*$, we have $\Delta_k \rightarrow 0$ as $k \rightarrow \infty$. The probability $\mathbb{P}(\Delta_k > G_s)$ converges to zero
1268 because $G_s > 0$ almost surely for continuous distributions. By dominated convergence, the first term
1269 also converges to zero.

1270 Consequently:

$$\frac{1}{k} \sum_{s=1}^k \mathbb{1}\{i \in \arg \max_j [f(j, s) - \alpha_{k,j}^*]\} \xrightarrow{P} \mathbb{E}[\mathbb{1}\{i \in \arg \max_j [f(j, s) - \alpha_j^*]\}] \quad (\text{B.86})$$

1271 Combined with our optimality condition in equation (B.78), we obtain:

$$P_X(i) = \mathbb{E}[\mathbb{1}\{i \in \arg \max_j [f(j, s) - \alpha_j^*]\}] = \mathbb{P}(\arg \max_j [f(j, s) - \alpha_j^*] = i) \quad (\text{B.87})$$

1272 In other words, the random mapping $S \mapsto \arg \max_j [f(j, s) - \alpha_j^*]$ reproduces the original law P_X
 1273 exactly. This fact is precisely what Theorem 3 asserts: the Gumbel-Max procedure constitutes the
 1274 optimal-transport coupling between P_X and the side-information mechanism.

1275 **Connection to Gumbel Watermarking.** The Gumbel watermarking scheme described above can
 1276 be directly interpreted within our optimal transport framework by noticing that this same arg-max
 1277 coupling is exactly what underlies the Gumbel-Max trick: adding Gumbel noise $G_t(j)$ to (negative)
 1278 logits $-\frac{u_t(y)}{T}$ and taking argmax samples from the softmax. In our formulation, the cost function
 1279 $f(j, s)$ corresponds to the Gumbel noise $G_t(j)$, while the dual variables α_j correspond to $-\frac{u_t(j)}{T}$.
 1280 The optimality condition

$$P_X(i) = \mathbb{P}(\arg \max_j [f(j, s) - \alpha_j^*] = i) \quad (\text{B.88})$$

1281 is precisely the Gumbel-Max trick, which states that sampling from a softmax distribution is equivalent
 1282 to adding Gumbel noise to logits and taking the argmax. Under the watermarking process, the
 1283 Gumbel scheme replaces the uniform variables $r_t(y)$ with pseudo-random values $F_{y_{t-m:t-1}, k}(y)$
 1284 that depend on the secret key and previously generated tokens. This creates a coupling between the
 1285 original token distribution and the side information, exactly as prescribed in our optimal transport
 1286 approach. Thus, the Gumbel watermarking scheme represents a specific instantiation of our general
 1287 optimal transport framework, where the coupling is designed to preserve the original distribution in
 1288 expectation while maximizing detectability through heavy-tailed score distributions.

1289 B.5 Proof of the Detection Gap, Theorem 4.

1290 In Section 4 we introduced the HeavyWater scheme by generalizing the scores beyond the binary
 1291 case. Since $f(x, s)$ is random, $D_{\text{gap}}^{[P_F]}(m, k, \lambda)$ given by

$$D_{\text{gap}}^{[P_F]}(m, k, \lambda) = \min_{P_X \in \mathcal{P}_\lambda} \max_{P_{X,S}} (\mathbb{E}_{P_{X,S}} [f(X, S)] - \mathbb{E}_{P_X P_S} [f(X, S)]) \quad (\text{B.89})$$

1292 is also a random variable. Now, we will show that we can go beyond Theorem 3 improving on
 1293 watermarking scheme like the Gumbel one, and prove Theorem 4 that connects the asymptotic
 1294 detection gap with the quantiles of the distribution of the score difference $\Delta = f(x, s) - f(x', s')$.

1295 **Theorem 4** (Detection Gap, asymptotic randomness). *Let $\lambda \in [\frac{1}{2}, 1)$, and consider the score
 1296 difference random variable $\Delta = f(x, s) - f(x', s')$ for some $(x, s) \neq (x', s')$, where $f(x, s)$ and
 1297 $f(x', s')$ are sampled i.i.d. from P_F . Let the cumulative distribution function of Δ be F , and let
 1298 $Q = F^{-1}$ be its inverse. Then,*

$$\lim_{k \rightarrow \infty} D_{\text{gap}}^{[P_F]}(m, k, \lambda) = \int_{1-\lambda}^1 Q(u) du. \quad (\text{B.90})$$

⁵To see this precisely, substitute the j th logit as $-\alpha_j^*$ on the RHS of (B.88), and simplify the RHS using the same steps as in Appendix B.1 of [24]. This shows that (B.88) holds when $-\alpha_j^* = j$ th logit. Which means that the Gumbel watermark is a special case of our construction in sec. 4 (which boils down to (B.88)), with $f(x, s)$ specifically chosen has Gumbel(0, 1).

1299 *Proof.* For a given k , $\max_{P_{XS}} \mathbb{E}_{P_{XS}}^{[k]} [f(X, S)]$ is given by

$$\max_{P_{XS}} \mathbb{E}_{P_{XS}}^{[k]} [f(X, S)] \quad (\text{B.91})$$

$$= \frac{1}{k} \sum_{s=1}^r f(1, s) + \left(\lambda - \frac{r}{k}\right) f(1, r+1) + \left(\frac{r+1}{k} - \lambda\right) f(2, r+1) + \frac{1}{k} \sum_{s=r+2}^k f(2, s) \quad (\text{B.92})$$

$$= \frac{1}{k} \sum_{s=1}^k f(2, s) + \left(\lambda - \frac{r}{k}\right) \Delta_{r+1} + \frac{1}{k} \sum_{s=1}^r \Delta_s \quad (\text{B.93})$$

1300 For $\mathbb{E}_{P_X P_S}^{[k]} [f(X, S)]$, we have

$$\mathbb{E}_{P_X P_S}^{[k]} [f(X, S)] = \frac{\lambda}{k} \sum_{s=1}^k f(1, s) + \frac{1-\lambda}{k} \sum_{s=1}^k f(2, s). \quad (\text{B.94})$$

1301 Therefore, the difference is given by

$$\max_{P_{XS}} \mathbb{E}_{P_{XS}}^{[k]} [f(X, S)] - \mathbb{E}_{P_X P_S}^{[k]} [f(X, S)] \quad (\text{B.95})$$

$$= \frac{1}{k} \sum_{s=1}^k f(2, s) + \left(\lambda - \frac{r}{k}\right) \Delta_{r+1} + \frac{1}{k} \sum_{s=1}^r \Delta_s - \frac{\lambda}{k} \sum_{s=1}^k f(1, s) - \frac{1-\lambda}{k} \sum_{s=1}^k f(2, s) \quad (\text{B.96})$$

$$= \frac{\lambda}{k} \sum_{s=1}^k f(2, s) + \left(\lambda - \frac{r}{k}\right) \Delta_{r+1} + \frac{1}{k} \sum_{s=1}^r \Delta_s - \frac{\lambda}{k} \sum_{s=1}^k f(1, s) \quad (\text{B.97})$$

1302 Let us define the following terms that we will analyze precisely:

$$\bar{f}_{1,k} = \frac{1}{k} \sum_{s=1}^k f(1, s), \quad \bar{f}_{2,k} = \frac{1}{k} \sum_{s=1}^k f(2, s) \quad \text{and} \quad I_k = \frac{1}{k} \sum_{s=1}^r \Delta_s. \quad (\text{B.98})$$

1303 With these definitions, we can rewrite the detection gap as:

$$\max_{P_{XS}} \mathbb{E}_{P_{XS}}^{[k]} [f(X, S)] - \mathbb{E}_{P_X P_S}^{[k]} [f(X, S)] = \lambda \bar{f}_{2,k} + \left(\lambda - \frac{r}{k}\right) \Delta_{r+1} + I_k - \lambda \bar{f}_{1,k} \quad (\text{B.99})$$

1304 To establish the exact limit, we need to analyze each term with greater precision.

1305 **Exact characterization of $\lambda(\bar{f}_{2,k} - \bar{f}_{1,k})$:** Both $f(1, s)$ and $f(2, s)$ are i.i.d. sub-exponential random
1306 variables with parameters (ν^2, b) . By the strong law of large numbers, we have almost surely:

$$\lim_{k \rightarrow \infty} \bar{f}_{1,k} = \mathbb{E}[f(1, 1)] \quad \text{and} \quad \lim_{k \rightarrow \infty} \bar{f}_{2,k} = \mathbb{E}[f(2, 1)] \quad (\text{B.100})$$

1307 Since $f(x, s)$ are i.i.d. with zero mean, we have $\mathbb{E}[f(1, 1)] = \mathbb{E}[f(2, 1)] = 0$. Therefore, almost
1308 surely:

$$\lim_{k \rightarrow \infty} \lambda(\bar{f}_{2,k} - \bar{f}_{1,k}) = 0 \quad (\text{B.101})$$

1309 **Exact characterization of $(\lambda - \frac{r}{k}) \Delta_{r+1}$:** By definition of $r = \lfloor \lambda k \rfloor$, we have $0 \leq \lambda - \frac{r}{k} < \frac{1}{k}$.
1310 Since Δ_{r+1} is a sub-exponential random variable with parameters $(4\nu^2, 2b)$, it is almost surely finite.
1311 Combining this with the above inequality:

$$\lim_{k \rightarrow \infty} \left(\lambda - \frac{r}{k}\right) \Delta_{r+1} = 0 \quad \text{almost surely.} \quad (\text{B.102})$$

1312 **Exact characterization of I_k :** For this term, we use order statistics. Let $\Delta_{(1)} \leq \Delta_{(2)} \leq \dots \leq \Delta_{(k)}$
1313 denote the ordered values for the difference of scores Δ_s . Then:

$$I_k = \frac{1}{k} \sum_{j=k-r+1}^k \Delta_{(j)}. \quad (\text{B.103})$$

1314 Let F be the CDF of Δ_s and F_k be the empirical CDF given by

$$F_k(x) := \frac{1}{k} \sum_{s=1}^k \mathbf{1}\{\Delta_s \leq x\}, \quad (\text{B.104})$$

1315 By the Glivenko-Cantelli theorem, we have:

$$\sup_{x \in \mathbb{R}} |F_k(x) - F(x)| \leq \eta_k \quad \text{where } \eta_k \rightarrow 0 \text{ almost surely as } k \rightarrow \infty \quad (\text{B.105})$$

1316 For each order statistic $\Delta_{(j)}$, the empirical CDF by definition gives $F_k(\Delta_{(j)}) = \frac{j}{k}$. The bound from
1317 Glivenko-Cantelli gives:

$$\frac{j}{k} - \eta_k \leq F(\Delta_{(j)}) \leq \frac{j}{k} + \eta_k. \quad (\text{B.106})$$

1318 Let $Q = F^{-1}$ be the quantile function. By definition of the quantile function, if $p \leq F(x)$ then
1319 $Q(p) \leq x$, and if $F(x) \leq q$ then $x \leq Q(q)$. Applying these relationships:

$$Q\left(\frac{j}{k} - \eta_k\right) \leq \Delta_{(j)} \leq Q\left(\frac{j}{k} + \eta_k\right) \quad (\text{B.107})$$

1320 Now we perform a change of index. We want to rewrite the sum in I_k which uses index j ranging
1321 from $k - r + 1$ to k . Let's set $j = k - i + 1$, so i ranges from 1 to r . This gives:

$$\frac{j}{k} = \frac{k - i + 1}{k} = 1 - \frac{i - 1}{k} \quad (\text{B.108})$$

1322 Substituting this into our bounds on $\Delta_{(j)}$:

$$Q\left(1 - \frac{i - 1}{k} - \eta_k\right) \leq \Delta_{(k-i+1)} \leq Q\left(1 - \frac{i - 1}{k} + \eta_k\right) \quad (\text{B.109})$$

1323 Summing over i from 1 to r and dividing by k :

$$\frac{1}{k} \sum_{i=1}^r Q\left(1 - \frac{i - 1}{k} - \eta_k\right) \leq \frac{1}{k} \sum_{i=1}^r \Delta_{(k-i+1)} \quad (\text{B.110})$$

$$= \frac{1}{k} \sum_{j=k-r+1}^k \Delta_{(j)} \quad (\text{B.111})$$

$$= I_k \quad (\text{B.112})$$

$$\leq \frac{1}{k} \sum_{i=1}^r Q\left(1 - \frac{i - 1}{k} + \eta_k\right) \quad (\text{B.113})$$

1324 As $k \rightarrow \infty$, we know that $\eta_k \rightarrow 0$ almost surely by the Glivenko-Cantelli theorem. The points
1325 $u_i := 1 - \frac{i-1}{k}$ for $i = 1, \dots, r$ where $r = \lfloor \lambda k \rfloor$, form a partition of the interval $[1 - \lambda, 1]$ as follows:

$$u_1 = 1, u_2 = 1 - \frac{1}{k}, u_3 = 1 - \frac{2}{k}, \dots, u_r = 1 - \frac{r-1}{k} \approx 1 - \lambda. \quad (\text{B.114})$$

1326 As $k \rightarrow \infty$, the number of points $r = \lfloor \lambda k \rfloor$ also increases, and the distance between adjacent points
1327 $\frac{1}{k} \rightarrow 0$. Therefore, these points $\{u_i\}_{i=1}^r$ form an increasingly fine partition of the interval $[1 - \lambda, 1]$.
1328 For any fixed $u \in [1 - \lambda, 1]$, as $k \rightarrow \infty$, there exists a sequence of indices i_k such that $u_{i_k} \rightarrow u$.
1329 Specifically, we can take $i_k = \lceil k(1 - u) + 1 \rceil$, which ensures $u_{i_k} \rightarrow u$ as $k \rightarrow \infty$. Our bounds for
1330 I_k can be written as:

$$\frac{1}{k} \sum_{i=1}^r Q(u_i - \eta_k) \leq I_k \leq \frac{1}{k} \sum_{i=1}^r Q(u_i + \eta_k) \quad (\text{B.115})$$

1331 Since Q is non-decreasing, it is bounded on the compact interval $[1 - \lambda - \beta, 1 + \beta]$ for some $\beta > 0$.
 1332 Let

$$M := \sup_{t \in [1 - \lambda - \beta, 1 + \beta]} |Q(t)| < \infty, \quad (\text{B.116})$$

1333 then $|Q(u_i - \eta_k)| \leq M$ and $|Q(u_i + \eta_k)| \leq M$ for all i and sufficiently large k .

1334 The lower sum $\frac{1}{k} \sum_{i=1}^r Q(u_i - \eta_k)$ is a perturbed lower Riemann sum for the integral $\int_{1-\lambda}^1 Q(u) du$,
 1335 and similarly the upper sum is a perturbed upper Riemann sum. As $k \rightarrow \infty$, two things happen
 1336 simultaneously, namely, (i) the mesh width $\frac{1}{k} \rightarrow 0$, so the Riemann sums converge to the integral and
 1337 (ii) the perturbation $\eta_k \rightarrow 0$, so $Q(u_i \pm \eta_k) \rightarrow Q(u_i)$. By the Dominated Convergence Theorem, we
 1338 have

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^r Q(u_i - \eta_k) = \int_{1-\lambda}^1 Q(u) du \quad (\text{B.117})$$

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^r Q(u_i + \eta_k) = \int_{1-\lambda}^1 Q(u) du \quad (\text{B.118})$$

1339 Since I_k is bounded between these two quantities that converge to the same limit, we have

$$\lim_{k \rightarrow \infty} I_k = \int_{1-\lambda}^1 Q(u) du \quad \text{almost surely.} \quad (\text{B.119})$$

1340 **Combining all terms:** From our asymptotic analysis of each term, we have:

$$\lim_{k \rightarrow \infty} \max_{P_{XS}} \mathbb{E}_{P_{XS}}^{[k]} [f(X, S)] - \mathbb{E}_{P_X P_S}^{[k]} [f(X, S)] = \lim_{k \rightarrow \infty} I_k = \int_{1-\lambda}^1 Q(u) du \quad \text{almost surely.} \quad (\text{B.120})$$

1341 Therefore:

$$\lim_{k \rightarrow \infty} D_{\text{gap}}^{[P_F]}(m, k, \lambda) = \int_{1-\lambda}^1 Q(u) du. \quad (\text{B.121})$$

1342 □

1343 Theorem 4 implies that distributions with heavier tails imply larger values of the integral $\int_{1-\lambda}^1 Q(u) du$,
 1344 which in turn imply higher detection gaps. Indeed, fix λ . We say that a distribution F_2 is (*right-*
 1345 *heavier-tailed*) than F_1 when $1 - F_2(x) \leq 1 - F_1(x)$ for all large x , equivalently $Q_2(u) \geq Q_1(u)$
 1346 for every u in a neighbourhood of 1. In particular, for all $u \in [1 - \lambda, 1]$:

$$Q_2(u) \geq Q_1(u)$$

1347 which implies:

$$\int_{1-\lambda}^1 Q_2(u) du \geq \int_{1-\lambda}^1 Q_1(u) du$$

1348 Therefore, keeping the same confidence levels:

$$\text{heavier tail} \implies \text{larger } \int_{1-\lambda}^1 Q(u) du \implies \text{larger guaranteed detection gap.}$$

1349 The asymptotic detection gap, given by $\int_{1-\lambda}^1 Q(u) du$, represents the average value of the quantile
 1350 function over the upper λ fraction of the distribution. This integral captures how much signal can
 1351 be extracted from the tail of the score differences. Distributions whose upper tail places more mass
 1352 far from zero (resulting in larger values of the integral $\int_{1-\lambda}^1 Q(u) du$) directly increase the detection
 1353 capability of the watermark. This explains why the choice of score distribution fundamentally impacts
 1354 watermark detectability. Consequently, among distributions with the same mean and variance, the
 1355 heavier-tailed ones yield strictly higher guaranteed detection rates. Therefore, if we constrain
 1356 ourselves to the ensemble of sub-exponential probability distributions, by choosing something with a
 1357 heavier tail than Gumbel, e.g., Log-Normal, we can achieve a better detection scheme, as we show in
 1358 our experiments, cf. Figure 1

1359 **Remark 1.** By using Bernstein’s inequality for sub-exponential variables and Dvoretzky–Kiefer–Wolfowitz inequality instead of Glivenko–Cantelli, it is possible to show a non-asymptotic
 1360 version of the theorem above:
 1361

1362 **Theorem 5** (Detection gap, formal). Let $0 < \lambda < 1$ be fixed, write $r = \lfloor \lambda k \rfloor$ and define

$$I_k = \frac{1}{k} \sum_{s=1}^r \Delta_s, \quad \bar{f}_{1,k} = \frac{1}{k} \sum_{s=1}^k f(1, s), \quad \bar{f}_{2,k} = \frac{1}{k} \sum_{s=1}^k f(2, s),$$

1363 where $\Delta_s = f(1, s) - f(2, s)$. For any confidence levels $\delta, \delta', \delta^\dagger \in (0, 1)$ set

$$\varepsilon_k(\delta) = \sqrt{\frac{2\nu^2 \log(1/\delta)}{k}} + \frac{b \log(1/\delta)}{k}, \quad t_*(\delta') = \max\left\{2\nu \sqrt{\log \frac{1}{\delta'}}, 2b \log \frac{1}{\delta'}\right\},$$

1364

$$\eta_k(\delta^\dagger) = \sqrt{\frac{\log(2/\delta^\dagger)}{2k}}.$$

1365 Then, with probability at least $1 - \delta - \delta' - \delta^\dagger$,

$$\max_{P_{XS}} \mathbb{E}_{P_{XS}}^{[k]}[f(X, S)] - \mathbb{E}_{P_X P_S}^{[k]}[f(X, S)] \geq \frac{1}{k} \sum_{i=1}^r Q\left(1 - \frac{i-1}{k} - \eta_k(\delta^\dagger)\right) - \left[2\varepsilon_k(\delta) + \frac{t_*(\delta')}{k}\right],$$

(B.122)

1366 where $Q = F^{-1}$ is the quantile function of Δ_s , and $\mathbb{E}^{[k]}$ denotes an expectation where the side
 1367 information alphabet is of size k .

1368 C Additional Information and Implementation Details

1369 C.1 Low-Entropy Distributions in LLMs

1370 To motivate the low-entropy regime, we compute summary statistics of token distributions of popular
 1371 open-weight LLMs. We consider the densities of three statistics: infinity norm (connects directly to
 1372 min-entropy⁶), entropy, and L-2 norm, all calculated on the next token prediction along a collection
 1373 of responses.

1374 We observe that 90% LLM token distributions fall into the low-entropy regime we consider with
 1375 infinity norm greater than $1/2$, i.e. $\max_x P(x) \geq \frac{1}{2}$, across the three open LLMs (Llama2-7B[45],
 1376 Llama3-8B[62], Mistral-7B) and two on popular prompt-generation datasets (Q&A tasks from
 1377 Finance-QA[43] and coding tasks from LCC[44]). We show the histogram and CDF plots in Fig.
 1378 D.14 and D.15

1379 C.2 Theoretical effect of different tails of distributions

1380 To further improve the detection performance, we go beyond binary score functions to explore the
 1381 flexibility in the design space that continuous score distributions offer. Motivated by the Gumbel
 1382 watermark [17], we observe that we can significantly improve detection by using continuous score
 1383 distributions, particularly those with heavy tails. Recall that our minimax formulation considers the
 1384 low-entropy regime ($\lambda \in [1/2, 1]$), where the worst-case token distribution P_X has only two non-zero
 1385 elements with values $\{\lambda, 1 - \lambda\}$. Working with this distribution, consider score matrices where each
 1386 entry $f(x, s)$ is sampled independently from a distribution P_F with zero mean and unit variance.
 1387 We additionally assume that f (and hence every $\Delta_s = f(1, s) - f(2, s)$) is *sub-exponential* with
 1388 parameters (ν^2, b) , i.e., $\mathbb{E}[e^{\lambda f}] \leq \exp(\frac{\nu^2 \lambda^2}{2})$ for all $|\lambda| \leq 1/b$. Many distributions, such as Gamma,
 1389 Gaussian, and Lognormal satisfy this property. This formulation leads to Theorem 4 which formally
 1390 characterizes the achievable maximum detection gap for any P_F for large k , in terms of quantile tail
 1391 integrals of various candidate distributions as its score distribution P_F . We visualize the result in Fig.
 1392 C.5

1393 In Fig. C.5 we present the quantile tail integrals of four different distributions: Lognormal, Gamma,
 1394 Gaussian, and Gumbel. A higher value on the y-axis (quantile tail integral) indicates a greater

⁶An infinity norm of λ , i.e. $\max_s P(x) < \lambda$, translates directly to min-entropy constraint of $-\log \lambda$.

1395 detection gap under the low-entropy regime, which translates to a greater probability of detection
 1396 under adversarial token distributions. In theory and as seen from Figure C.5 drawing score functions
 1397 i.i.d from either the Lognormal or Gamma distribution outperforms that from Gumbel. Recall that
 1398 the significance of adopting the Gumbel score function is that we have established its equivalence
 1399 with the Gumbel watermark by [17] in Theorem 3. Although the Gamma distribution maximizes the
 1400 detection gap in the worst-case regime, we observe that choosing P_F to be lognormal achieves the
 1401 highest detection accuracy in practice, which is what we eventually adopt for HeavyWater.

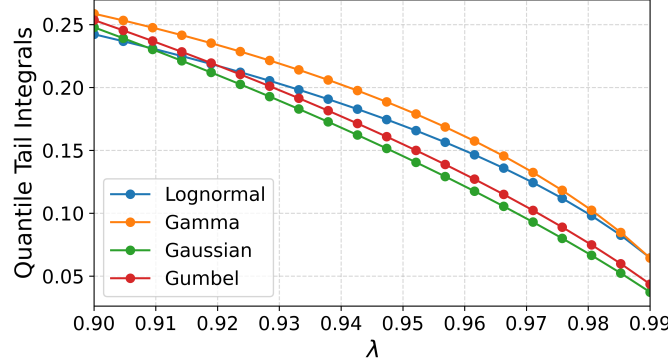


Figure C.5: Tail integrals of different *score difference distributions*: Higher the tail integral, better the detection.

1402 C.3 Q-ary Code

1403 We provide a discussion of the direct extension of SimplexWater to go beyond binary-valued scores.
 1404 Recall that SimplexWater uses the binary Simplex Code as its score function. For a given prime
 1405 field-size q , a Q-ary SimplexWater adopts the corresponding Q-ary Simplex Code [14], which we
 1406 define next. Besides the score function, the algorithm for Q-ary SimplexWater is identical to the
 1407 binary case, which we have provided in the main text.

1408 Given an alphabet of size m and field size $q > 2$, the size of a Q-ary codeword is q to the power
 1409 of the ceiling of $\log m$ with base q , i.e. $n = q^{\lceil \log_q m \rceil}$. For any $x, s \in [0 : n - 1]$, let $\text{qary}(x)$,
 1410 $\text{qary}(s)$ denote their Q-ary representations respectively using n bits. A Q-ary simplex code $f_{\text{sim}} : [0 : n - 1] \times [1 : n - 1] \rightarrow [0 : q - 1]$ is characterized by

$$f_{\text{sim}}(x, s) \triangleq \text{dot}(\text{qary}(x), \text{qary}(s)), \quad (\text{C.123})$$

1412 where $\text{dot}(\text{qary}(x), \text{qary}(s)) \triangleq \sum_{i=1}^n \text{qary}(x)_i \cdot \text{qary}(s)_i$ and $\text{qary}(v)_i$ denotes the i th bit in the
 1413 Q-ary representation of v .

1414 There are two main limitations of Q-ary SimplexWater, which motivated us to explore a continuous
 1415 score function directly and ultimately led to HeavyWater. The first limitation is that we do not have
 1416 optimality guarantees for the Q-ary code. Recall that to maximize watermark detection, the optimal
 1417 code maximizes the L_1 distance. Simplex Code achieves the Plotkin bound, which maximizes the
 1418 pair-wise Hamming distance between codewords. In the binary case, maximizing the Hamming
 1419 distance and L_1 distance are equivalent, where $1_{i,j} = |i - j|$; in the Q-ary case, this equivalence
 1420 doesn't hold. Hence, we do not have an optimality guarantee in detection for Q-ary SimplexWater.
 1421 The second limitation is that the size of Q-ary codewords is potentially very large, leading to memory
 1422 issues in actual implementation on a GPU. Recall that in the binary case, we have $n = m - 1$. In the
 1423 Q-ary case, however, the use of the ceiling function when converting m to base- q artificially inflates
 1424 n , often far beyond the actual vocabulary size. For example, with $q = 7$ and a vocabulary size of
 1425 $m = 100,000$, we compute $\lceil \log_7(100,000) \rceil \approx \lceil 5.92 \rceil = 6$, which corresponds to $n = 7^6 = 117,649$
 1426 codewords—more than 18% larger than the original vocabulary. Furthermore, the required alphabet
 1427 size to implement a Q-ary code grows with q .

1428 C.4 Implementation Details

1429 In this section we provide additional implementation details for SimplexWater and HeavyWater.
 1430 Our code implementation employs the code from the two benchmark papers, namely WaterBench⁷
 1431 [41] and MarkMyWords⁸ [42].

1432 **Score Matrix Instantiation** We sample the score matrix once during the initialization stage of
 1433 HeavyWater generation and it has shape (m, k) . Each row maps a vocabulary token to a cost vector
 1434 over the side information space \mathcal{S} . This matrix is stored as a `torch.Tensor` and reused across
 1435 watermarking calls. During generation, the score matrix is used as the cost matrix for Sinkhorn-based
 1436 optimal transport computations.

1437 **Normalization of f** As described before, for each element in the vocabulary, k samples for the
 1438 score f are drawn from a heavy-tailed distribution (log-normal in the experiment). To ensure
 1439 numerical stability and suitability for optimal transport computations, each row of the score matrix
 1440 (size `vocab_size * k`) is normalized to have zero mean and unit variance.

1441 **Top- p Filtering** To reduce the computational cost of watermarking, top- p filtering is applied prior
 1442 to watermarking. Identical to the common definition of top- p , we take the minimal set of tokens
 1443 whose cumulative softmax probability exceeds a threshold (e.g., $p = 0.999$). The watermarking
 1444 algorithm is then restricted to this filtered subset. We emphasize that, in the considered experiments,
 1445 top- p filtering was applied to all considered watermarks to maintain consistency in the experimental
 1446 setting.

1447 **Detection Algorithm** We outline in Algorithm 2 a standard watermark detection algorithm that
 1448 employs a threshold-test with a score matrix F . Given the sampled token x and side information s ,
 1449 the corresponding score is $F_{x,s} = f(x, s)$, which is obtained in a similar fashion to its construction in
 1450 generation: If SimplexWater is used, then it is obtained from the Simplex code, and if HeavyWater
 1451 it is randomly sampled using the shared secret key as the initial seed. This maintains the generation
 1452 of the same cost matrix F on both ends of generation and detection. A watermark is detected if the
 1453 sum of scores exceeds a certain predetermined threshold.

Algorithm 2 Detection using a threshold-test with a score matrix

```

1: Input: Token sequence  $x^n$ , side information  $s$ , seed, score matrix  $F \in \mathbb{R}^{m \times k}$ 
2: Outputs:  $p$ -value based detection outcome
3: for  $t = 1$  to  $T$  do
4:   Compute score  $\phi_t := F_{x_t, s}$ 
5: end for
6:  $Z \leftarrow \sum_{t=1}^T \phi_t$ 
7: if  $Z > \tau$  then return "Watermark Detected"
8: else return "No Watermark"

```

1454 **Fresh Randomness Generation** Fresh randomness is crucial to ensure side information is sampled
 1455 independently from previous tokens to avoid seed collision. Our implementation supports several
 1456 seeding strategies. In majority of our experiment, we use the 'fresh' strategy, which generates
 1457 a unique seed for each token by incrementing a counter and combining it with the shared secret
 1458 key. This allows both ends to share the same seed that dynamically changes, but is independent of
 1459 previously generated tokens. Our implementation of HeavyWater and SimplexWater also allows
 1460 for various forms of temporal or token-based encoding strategies, such as sliding-window hashing.

1461 C.5 Information on Optimal Transport and Sinkhorn's Algorithm

In this section, we provide preliminary information on the considered OT problem and its solution through Sinkhorn's algorithm. Let $p \in \Delta_m$ and $q \in \Delta_k$ be two probability vectors. For a given cost

⁷<https://github.com/THU-KEG/WaterBench>

⁸<https://github.com/wagner-group/MarkMyWords>

matrix $C \in \mathbb{R}^{m \times k}$, the OT between p and q is given by

$$\text{OT}(p, q) = \min_{P \in \Pi_{p, q}} \sum_{i=1}^m \sum_{j=1}^k C_{i, j} P_{i, j},$$

where $\Pi_{p, q}$ is the set of joint distributions with marginals p and q , which we call *couplings*. Efficiently solving the optimization $\text{OT}(p, q)$ is generally considered challenging [34]. To that end, a common approach to obtain a solution efficiently is through entropic regularization. An entropic OT (EOT) with parameter ϵ is given by

$$\text{OT}_\epsilon(p, q) = \min_{P \in \Pi_{p, q}} \left(\sum_{i=1}^m \sum_{j=1}^k C_{i, j} P_{i, j} - \epsilon H(P) \right),$$

1462 where $H(P) \triangleq -\sum_{i, j} P_{i, j} \log(P_{i, j})$ and C is the OT cost. The EOT is an ϵ -strongly convex problem,
 1463 which implies its fast convergence to the *unique* optimal solution. However, the EOT provides an
 1464 approximate solution which converges to the unregularized solution with rate $O(\epsilon \log(1/\epsilon))$.

One of the main reasons EOT has gained its popularity is due to Sinkhorn’s algorithm [63], which is a matrix scaling algorithm that has found its application to solve the dual formulation of the EOT problem [19]. Sinkhorn’s algorithm looks for a pair of vectors u, v that obtain the equality

$$P^* = \text{diag}(u) K \text{diag}(v),$$

1465 where P^* is the EOT solution and $K = \exp(-C/\epsilon)$ is called the *Gibbs Kernel*. Consequently,
 1466 Sinkhorn’s algorithm follows from a simple iterative procedure of alternately updating u and v . The
 1467 steps of Sinkhorn’s algorithm are given in Algorithm 3. Having solved Sinkhorn’s algorithm, we
 1468 obtain the optimal EOT coupling. When using Sinkhorn’s algorithm, the stopping criteria is often
 1469 regarding the marginalization of the current coupling against the corresponding marginal, i.e., we
 1470 check whether $\|u \odot (Kv) - p\|_1 \leq \delta$ where $\triangleq \sum_{i=1}^n |u_i (Kv)_i - p_i|$ and $\delta > 0$ is some threshold.
 1471 In our implementation, we solve Sinkhorn’s algorithm using the Python optimal transport package
 1472 [64].

Algorithm 3 Sinkhorn’s Algorithm for Entropic OT

```

1: Input: Marginals  $P_X \in \Delta_m, P_S \in \Delta_k$ , cost matrix  $C \in \mathbb{R}^{m \times k}$  regularization parameter  $\epsilon > 0$ ,
   Threshold  $\delta > 0$ .
2: Output: Optimal coupling  $P \in \mathbb{R}_{\geq 0}^{n \times m}$ 
3: Calculate kernel  $K \leftarrow \exp(-C/\epsilon)$ 
4:  $u \leftarrow \mathbf{1}_m, v \leftarrow \mathbf{1}_k$ 
5: while  $\|u \odot (Kv) - P_X\|_1 > \delta$  do
6:    $u \leftarrow a/(Kv)$  ▷ element-wise division
7:    $v \leftarrow b/(K^\top u)$ 
8: end while
9:  $P \leftarrow \text{diag}(u) K \text{diag}(v)$ 
10: return  $P$ 

```

D Additional Numerical Results and Ablation Study

D.1 Ablation Study

1475 We perform an ablation study to investigate the effect of various hyperparameters set in
 1476 SimplexWater and HeavyWater. The ablation study is performed in a curated subset of prompts
 1477 from the Finance-QA dataset [43] considered in Section 5.

1478 **Sinkhorn Algorithm Parameters.** We study the effect of Sinkhorn’s algorithm’s parameter on the
 1479 performance of the proposed watermarking scheme. We note that, while the Sinkhorn’s algorithm is
 1480 set with a predetermined maximum iterations parameters, in the considered experiments the algorithm
 1481 runs until convergence. We study this effect through three cases.

1. We analyze the effect of Sinkhorn’s algorithm’s regularization parameter ϵ on the overall runtime. While lower values of ϵ provide solution that are closer to the underlying OT solution, often a smaller value of ϵ required more time for convergence of the algorithm. As seen from Figure D.6a as expected, smaller ϵ increase overall runtime. In our experiments we chose $\epsilon = 0.05$ which resulted in overall satisfactory performance, while incurring mild runtime overhead.
2. We analyze the effect of the error threshold on runtime. The lower the error threshold, the higher the accuracy in the solution and the higher the overall algorithm runtime. This is indeed the case, and the effect on the watermarking procedure runtime is visualized in Figure D.6b
3. We analyzed the effect of the error threshold on the watermarked distribution cross entropy (see Section 5 for definition). As seen from Figure D.6c while a lower threshold results with a higher runtime, the improvement on the cross entropy, which is a proxy for textual quality, becomes negligible from some point. In our experiments, we chose a threshold value of 10^{-5} , which demonstrated the best performance between runtime, cross entropy and detection.

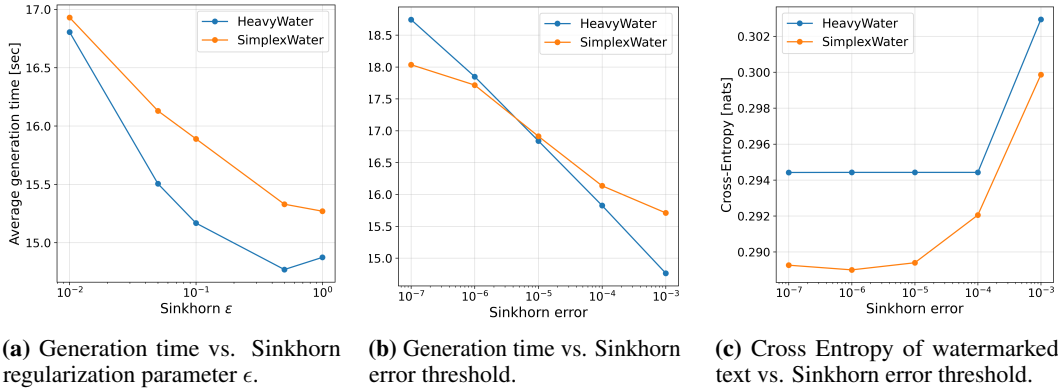


Figure D.6: Effect of Sinkhorn Algorithm’s parameters on Runtime and distortion.

D.2 Impact of Non i.i.d. Side Information Generation

As we previously mentioned, most of the considered experiments in Section 5 operate under a ‘fresh randomness’ scheme, in which we try to replicate independence between the random side information and the LLM net token distribution. However, in practice various hashing scheme are employed, often with the purpose of increasing the overall watermarking scheme’s robustness to attacks. Such hashing schemes aggregate previous tokens (using some sliding window with context size h) and a shared secret key $r \in \mathbb{N}$. We are interested in verifying that indeed, robustness-driven seed generation scheme do not degrade the performance of our methods.

To that end, in this section we test the effect of various popular hashing schemes in the performance of our watermarks. As both SimplexWater and HeavyWater follow the same watermarking algorithm we anticipate them to demonstrate similar dependence on the hashing scheme. We therefore prioritize an extensive study on a single watermark - SimplexWater. For a given sliding window size h , we consider the following seed generation functions:

1. min-hash, which takes the minimum over token-ids and multiplies it with the secret key, i.e.

$$\text{seed} = \min(x_{t-1}, \dots, x_{t-h}) \cdot r.$$
2. sum-hash, which takes the sum of the token-ids and multiplies it with the secret key, i.e.

$$\text{seed} = \text{sum}(x_{t-1}, \dots, x_{t-h}) \cdot r.$$
3. prod-hash, which takes the product of the token-ids and multiplies it with the secret key, i.e.

$$\text{seed} = \text{prod}(x_{t-1}, \dots, x_{t-h}) \cdot r.$$
4. Markov-1 scheme, which considers $h = 1$.

1518 We present performance across the aforementioned schemes, considering several values of h . As
 1519 seen from Figure D.8 the change of seed generation scheme is does have a significant effect on the
 1520 overall detection-distortion tradeoff. Furthermore, as emphasize in Figure D.7 the size of the sliding
 1521 window also results in a negligible effect on the watermark performance.

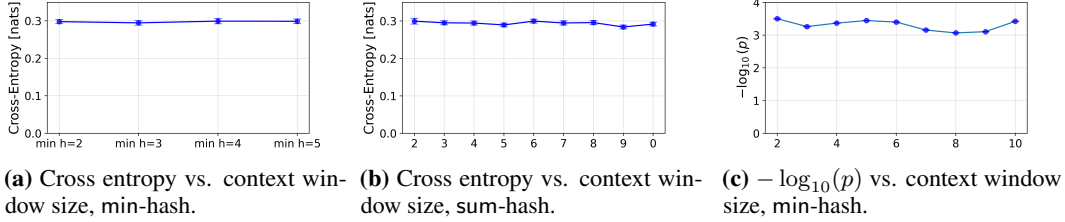


Figure D.7: Seed Ablation: The effect of the context window is negligible on the performance of SimplexWater.

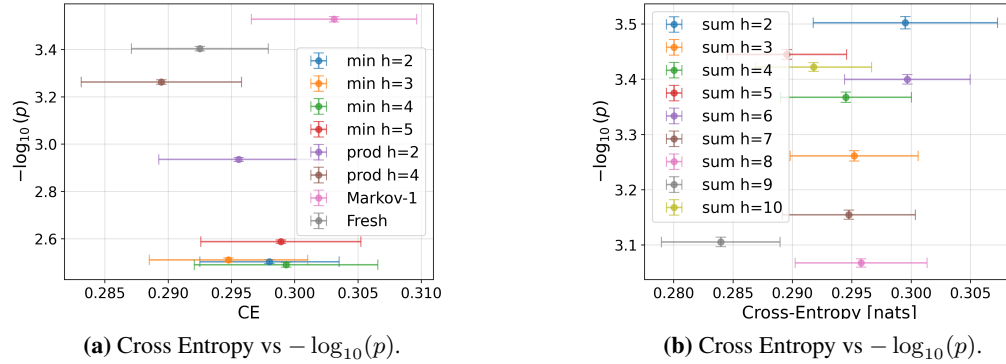


Figure D.8: Seed Ablation visualized in the detection-distortion plane. It is visible that the performance of our watermark is consistent across an array of hashing scheme and context window sizes.

1522 D.3 Experiment: Robustness To Textual Attacks

1523 The watermarks in this paper are obtained by optimizing a problem that encodes the tradeoff between
 1524 detection and distortion under worst case distribution. To that end, the proposed watermarks are
 1525 not theoretically optimized for robustness guarantees. However, robustness is often a byproduct of
 1526 the considered randomness generation scheme, as text edit attacks mainly effect the context from
 1527 which the seed is generated. A discrepancy in the seed results in a discrepancy in the shared side
 1528 information sample s . However, regardless of the seed generation scheme, one has to choose a score
 1529 function and a watermarked distribution design.

1530 In this section, we show that, while not optimized for robustness directly, SimplexWater and
 1531 HeavyWater demonstrate competitive performance in terms on robustness to common textual edit
 1532 attacks. We consider the setting from the watermarking benchmark MarkMyWords [42]. We
 1533 compare our performance with the Red-Green watermark and the Gumbel watermark. We choose
 1534 the value of δ for the Red-Green watermark such that its cross-entropy distortion is comparable with
 1535 SimplexWater, HeavyWater and Gumbel ($\delta = 1$).

1536 We consider three attacks:

- 1537 1. A Lowercase attack, in which all the characters are replaced with their lowercase version.
- 1538 2. A Misspelling attack, in which words are replaced with a predetermined misspelled version.
 1539 Each word is misspelled with probability 0.1.
- 1540 3. A Typo attack, in which, each character is replaced with its neighbor in the QWERTY
 1541 keyboard. A character is replaced with probability 0.05.

As seen in Figure D.9, our schemes demonstrate strong robustness under the considered attacks, resulting in the highest detection capabilities in 3 out of 4 cases and competitive detection power in the 4th. This implies that, even though SimplexWater and HeavyWater are not designed to maximize robustness, the resulting schemes show competitive resilience to common text edit attacks.

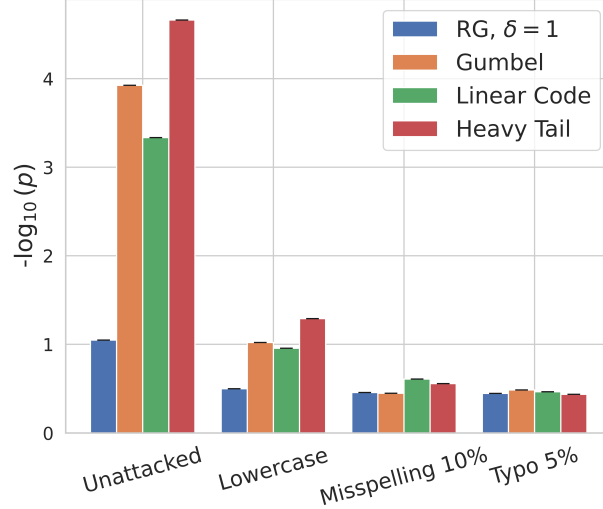


Figure D.9: Robustness to attacks — HeavyWater demonstrates equal or superior detection performance, as measured by $-\log_{10}(p)$, across a variety of attacks involving edits to generated outputs.

D.4 Computational Overhead

We analyze the computational overhead induced by the considered watermarking scheme. Theoretically, Sinkhorn’s algorithm has an iteration computational complexity of $O(km)$ for token vocabulary of size $|\mathcal{X}| = m$ and side information of alphabet $|\mathcal{S}| = k$ due to its vector-matrix operations. In practice, watermarking is a single step within the entire next token generation pipeline.

We analyze the computational overhead induced by applying SimplexWater and HeavyWater. Figure D.10 shows the overhead of watermarking in a few common watermarks - Red-Green [15], Gumbel [17], Inverse-transform [13], SynthID [11] and our watermarks. It can be seen that The Gumbel, Inverse transform and Red-Green watermarks induce a computational overhead of $\sim 10\%$, while SimplexWater, HeavyWater and SynthID induce an overhead of $\sim 30\%$. While this overhead is not negligible, our methods demonstrate superior performance over considered methods. However, replacing a ‘fast’, yet ‘weaker’ watermark with ours boils down to a difference in $\sim 20\%$ increase in generation time. We consider an implementation of the SynthID through vectorized tournament sampling with a binary score function and 15 tournament layers, which is the method reported in the main text experiments. As we previously mentioned, we consider top- p sampling with $p=0.999$. We note that, in many text generation schemes, lower top- p values, which accelerate Sinkhorn’s algorithm’s runtime, thus further closing the computational gap.

D.5 Experiment: Alternative Text Generation Metrics

This paper focused on distortion as the proxy for textual quality. This is a common practice in watermarking (e.g. [11, 13, 16, 17, 24, 25]). Distortion is measured by the discrepancy between the token distribution P_X and the expected watermarked distribution $\mathbb{E}_S[P_{X|S}]$. In practice, distortion and textual quality are often measured with some perplexity-based measure (e.g. cross-entropy in this paper). However, as explored in the WaterBench benchmark [41], such measures are not guaranteed to faithfully represent degradation in textual quality.

To that end, WaterBench proposed an array of alternative generation metrics, whose purpose us to evaluate the quality of generated watermarked text, and are tailored for specific text generation tasks. We consider 4 datasets from the WaterBench benchmark [41]:

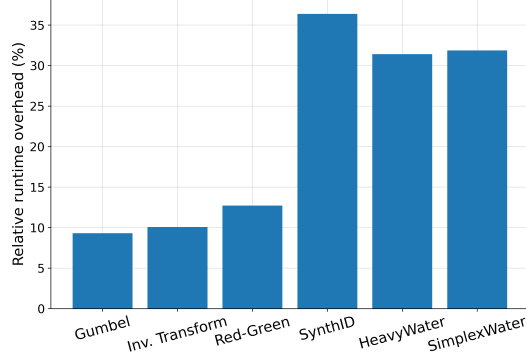


Figure D.10: Computational overhead over unwatermarked text generation, Llama2-7b.

1. Longform QA [51]: A dataset of 200 long questions-answer generation prompts. The considered generation metric is the ROGUE-L score.
2. Knowledge memorization: A closed-ended entity-probing benchmark drawn from KoLA [65], consisting of 200 triplets sampled at varying frequencies from Wikipedia the test an LLM’s factual recall. The considered generation metric is the F1 score as it is a factual knowledge dataset.
3. Knowledge understanding [66]: A dataset of 200 questions that demonstrate the LLM’s understanding of various concepts. The considered generation metric is the F1 score as it is a factual knowledge dataset.
4. Multi-news summarization: A collection of 200 long news clusters, coupled with summarization prompts. The score here is the ROGUE-L score.

As seen in Table D.2 our watermarks maintain competitive performance in the considered set of textual generation tasks, even under alternative text generation evaluation metrics.

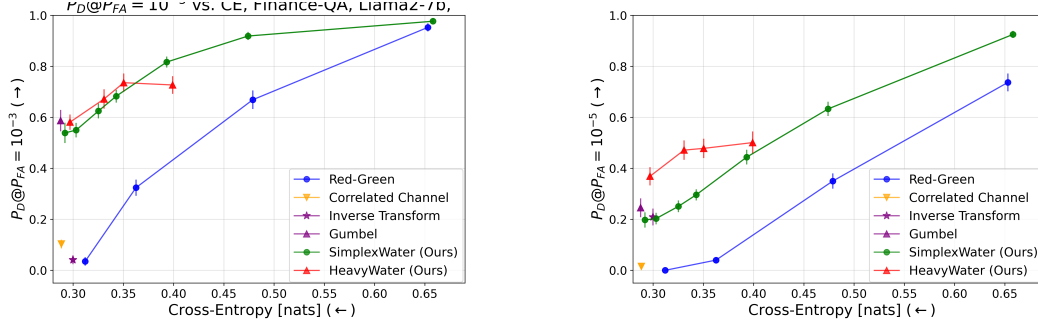
D.6 Alternative Detection Metric

In this section we provide results on an additional detection metric. We consider the detection probability under a false-alarm (FA) constraint. As we consider watermarks from which p -values can be calculated, we can impose such a FA constraint. For a given set of responses obtained from a dataset of prompts, we are interested in calculating an estimate of the detection probability at some FA constraint, given a set of p -values, each calculated for one of the responses. We obtain an estimate of the detection probability at a given FA constraint by taking the ratio of responses whose p -value is lower than the proposed p -value threshold, over the total number of responses.

We provide results on the FinanceQA dataset using Llama2-7b. We consider several FA values and visualize the resulting tradeoff curves in Figures D.11 and D.12. To obtain error-bars, we consider the following bootstrapping technique: Out of the 200 responses, we randomly sample 200 subsets with 150 responses and calculate the corresponding metric. From the set of 150 results we provide error-bars, considering the average value and standard deviation. It can be seen that the trends presented in Figures 1 and 3a are preserved under the considered detection metric.

D.7 Additional Detection-Distortion Tradeoff Results

We provide results that explore the detection-distortion tradeoff, in addition to ones presented in Fig. 1 and Fig. 3a. We run three models (Llama2-7b, Llama3-8b, Mistral-7b) on two tasks (Q&A and coding). We employ the popular Q&A dataset, FinanceQA, and code-completion dataset LCC. Fig. 1 shows the result for Llama2-7b on Q&A, while Fig. 3a shows the result for Mistral-7B on coding. In Fig. D.13 we present this tradeoff over the remaining datasets and LLMs.



(a) P_D at $P_{FA} = 10^{-3}$

(b) P_D at $P_{FA} = 10^{-5}$

Figure D.11: Detection probability at a given false alarm constraint.

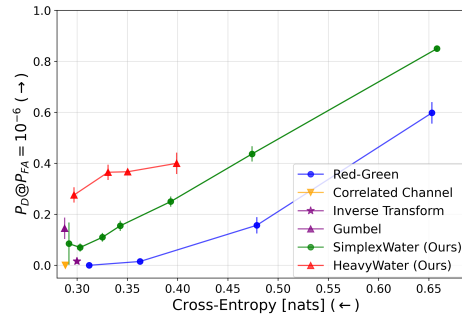


Figure D.12: P_D at $P_{FA} = 10^{-6}$

Table D.2: Performance of Watermarking Methods across Four Datasets

Dataset	Watermark	Gen. Metric \uparrow	% Drop in GM \downarrow	$-\log_{10} p \uparrow$
Longform	Gumbel	21.20	-0.856	8.006
	HeavyWater	21.48	-2.188	8.089
	Simplex	21.90	-4.186	4.985
	Inv. Tr.	21.27	-1.189	3.687
	RG, $\delta = 1$	21.25	-1.094	1.456
	RG, $\delta = 3$	21.19	-0.809	7.078
Memorization	Gumbel	5.66	-2.536	1.085
	HeavyWater	5.73	-3.804	1.605
	Simplex	5.71	-3.442	0.977
	Inv. Tr.	5.38	2.536	0.792
	RG, $\delta = 1$	5.35	3.080	0.482
	RG, $\delta = 3$	5.82	5.435	0.912
Understanding	Gumbel	33.42	-9.574	0.396
	HeavyWater	32.59	-6.852	0.308
	Simplex	31.50	-3.279	0.920
	Inv. Tr.	27.93	8.426	1.045
	RG, $\delta = 1$	32.96	8.066	0.184
	RG, $\delta = 3$	33.83	10.918	0.300
MultiNews	Gumbel	25.69	2.579	3.172
	HeavyWater	25.67	2.655	3.491
	Simplex	25.86	1.934	2.701
	Inv. Tr.	25.74	2.389	1.586
	RG, $\delta = 1$	25.85	1.940	0.963
	RG, $\delta = 3$	25.74	2.389	3.781

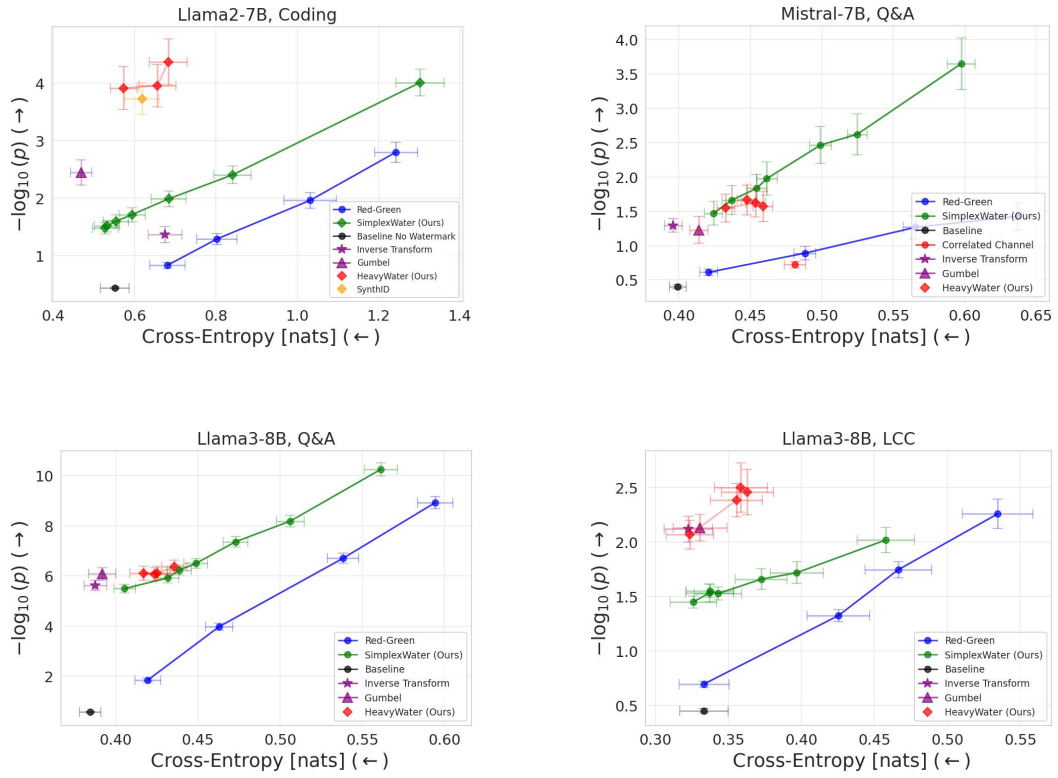
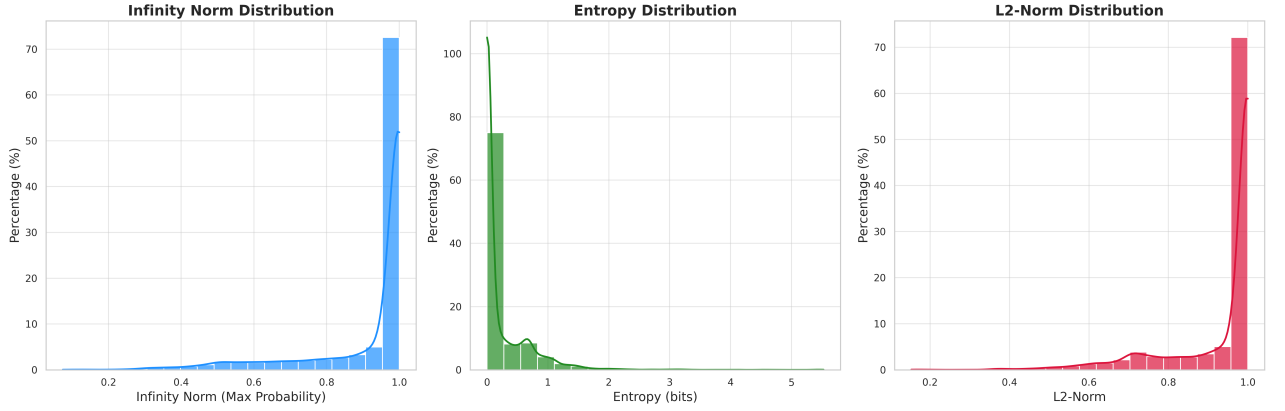
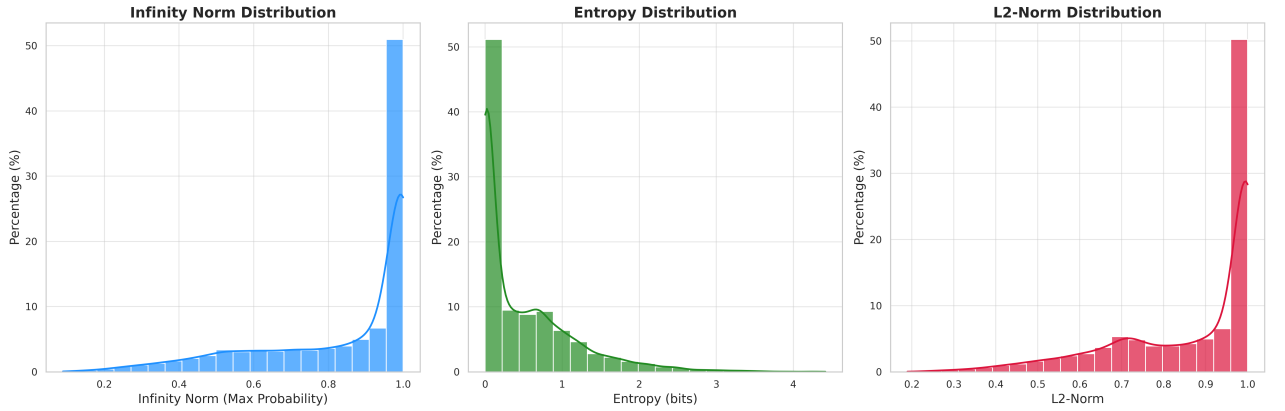


Figure D.13: Detection-distortion tradeoffs on multiple models and tasks.

Distribution of Probability Metrics for llama2-7b-chat-4k Token Predictions



Distribution of Probability Metrics for llama3-8b Token Predictions



Distribution of Probability Metrics for mistral-7b Token Predictions

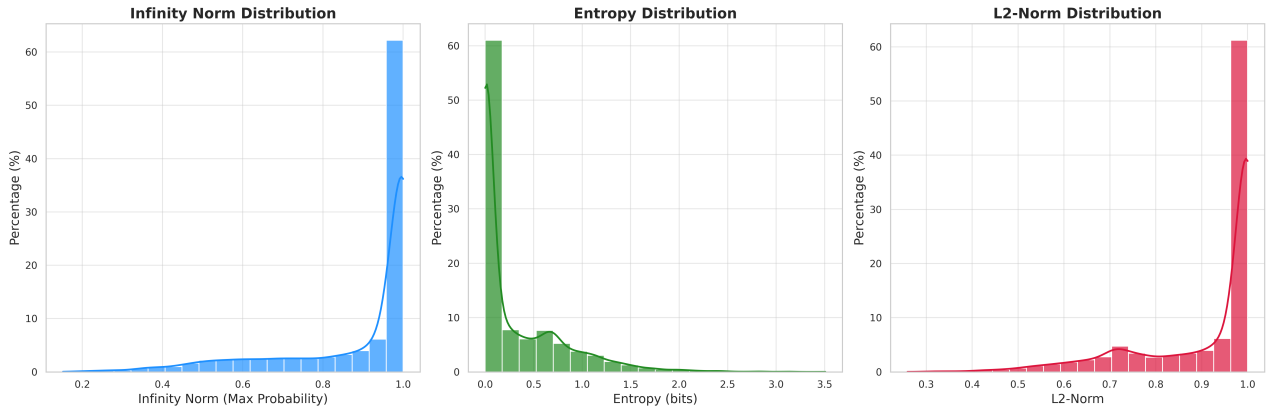


Figure D.14: Histograms of statistics of token distributions on Q&A dataset. 90% token distributions fall into the low-entropy regime with infinity norm greater than $1/2$, i.e. $\max_x P(x) \geq \frac{1}{2}$.

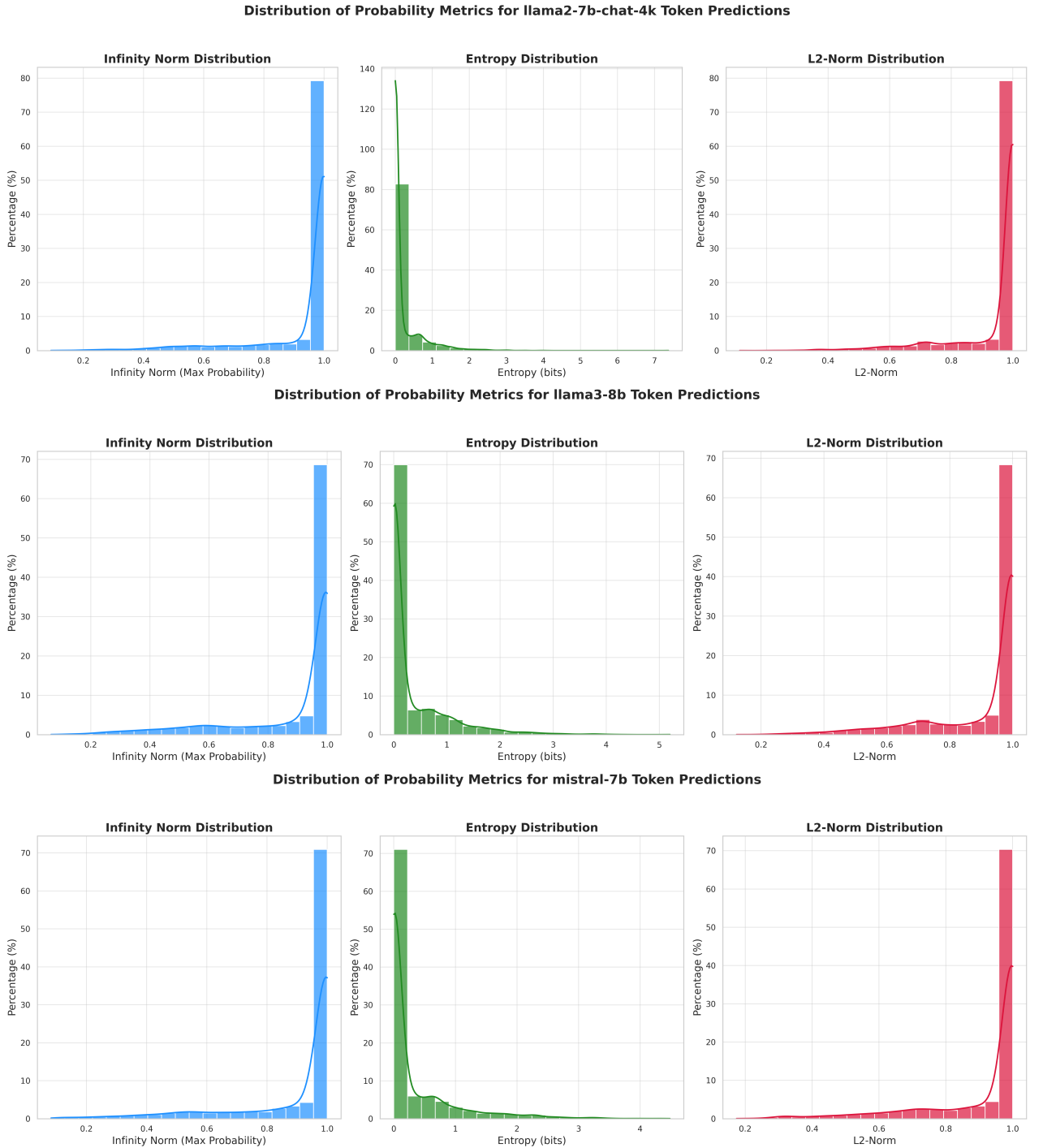


Figure D.15: Histograms of statistics of token distributions on `coding` dataset. 93% token distributions fall into the low-entropy regime with infinity norm greater than $1/2$, i.e. $\max_x P(x) \geq \frac{1}{2}$.