Inevitable Trade-off between Watermark Strength and Speculative Sampling Efficiency for Language Models

Zhengmian Hu, Heng Huang
Department of Computer Science
University of Maryland
College Park, MD 20742
huzhengmian@gmail.com,heng@umd.edu

Abstract

Large language models are probabilistic models, and the process of generating content is essentially sampling from the output distribution of the language model. Existing watermarking techniques inject watermarks into the generated content without altering the output quality. On the other hand, existing acceleration techniques, specifically speculative sampling, leverage a draft model to speed up the sampling process while preserving the output distribution. However, there is no known method to simultaneously accelerate the sampling process and inject watermarks into the generated content. In this paper, we investigate this direction and find that the integration of watermarking and acceleration is non-trivial. We prove a no-go theorem, which states that it is impossible to simultaneously maintain the highest watermark strength and the highest sampling efficiency. Furthermore, we propose two methods that maintain either the sampling efficiency or the watermark strength, but not both. Our work provides a rigorous theoretical foundation for understanding the inherent trade-off between watermark strength and sampling efficiency in accelerating the generation of watermarked tokens for large language models. We also conduct numerical experiments to validate our theoretical findings and demonstrate the effectiveness of the proposed methods.

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance in various natural language processing tasks, enabling a wide range of applications such as chatbots [23], content generation [17], code generation [6], and more. However, the high training and inference costs of LLMs pose significant challenges. The substantial computational resources along with the high latency during inference can negatively impact user experience and limit their potential applications.

To address the issue of high inference costs, speculative sampling [16, 5] has emerged as a promising approach. This technique leverages a smaller, faster draft model to generate candidate results, which are then validated and corrected by a larger, more accurate target model. Compared with other acceleration methods such as knowledge distillation, model quantization, and model pruning, the key advantage of speculative sampling is that it can significantly reduce inference latency without compromising the quality of the generated content.

In addition to the challenge of high inference costs, protecting the intellectual property rights of LLMs generated content has become increasingly important. Digital watermarking techniques [1, 13] have been proposed to embed watermark information into the generated content, enabling the tracking of model usage. Unbiased watermarking schemes [12] have been developed to ensure that the watermarking process does not affect the quality of the generated content.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

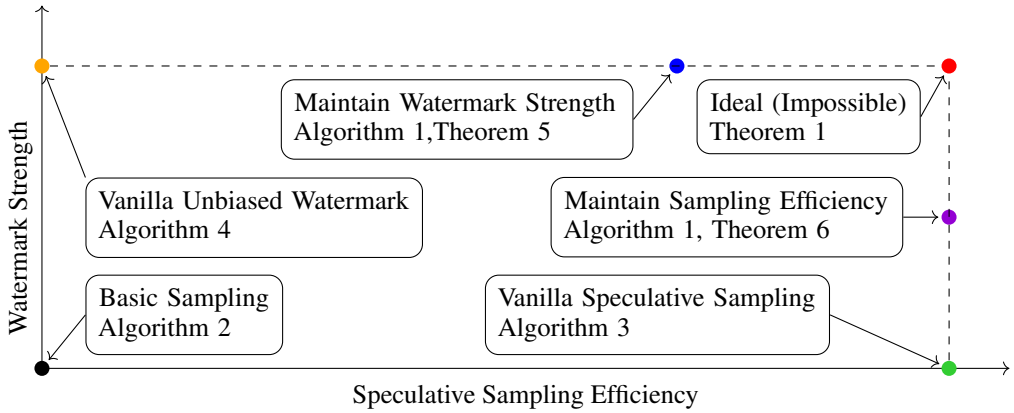


Figure 1: Taxonomy of watermarking and speculative sampling trade-offs in language models. The ideal case of maintaining both watermark strength and sampling efficiency is proven to be impossible by the no-go theorem. The proposed algorithms focus on maintaining either watermark strength or sampling efficiency.

A natural question arises: can we leverage speculative sampling to accelerate the generation of watermarked content? To address this question, we propose a general framework called the *two reweight framework*, which allows for the integration of unbiased watermarking and speculative sampling techniques while guaranteeing an unchanged output distribution. The main innovation of our framework lies in the simultaneous reweighting of both the target model and the draft model, which improves the sampling efficiency compared to naively applying speculative sampling to a watermarked target model.

To evaluate the effectiveness of our framework, we consider two key metrics: watermark strength and acceleration performance. A fundamental question is whether it is possible to achieve both strong watermarking and efficient speculative sampling simultaneously. Specifically, we aim to answer the following question:

> *Can we obtain the same watermark strength as in the case without acceleration while maintaining the same sampling efficiency as in the case without watermarking?*

Surprisingly, we got a negative answer to this question. We prove a no-go theorem, which states that under the *two reweight framework*, it is impossible to simultaneously maintain both the watermark strength and the sampling efficiency when the vocabulary size is greater than two. This result highlights the inherent trade-off between watermarking and acceleration in the context of large language models.

To better explore the trade-offs between these two objectives, we propose two practical algorithms within the *two reweight framework*. The first algorithm focuses on maintaining the watermark strength, while the second algorithm aims to maintain the sampling efficiency.

The main contributions of this paper are as follows:

- We propose the *two reweight framework*, a general framework that allows for the integration of unbiased watermarking and speculative sampling techniques while ensuring an unchanged output distribution.
- We prove a no-go theorem, which states that under the *two reweight framework*, it is impossible to simultaneously maintain both the watermark strength and the sampling efficiency when the vocabulary size is greater than two.
- We propose two practical algorithms within the *two reweight framework* that focus on maintaining either the watermark strength or the sampling efficiency, providing insights into the achievable trade-offs.

To the best of our knowledge, this work represents the first exploration of the intersection between unbiased watermarking and speculative sampling, introducing a novel framework, a significant no-go theorem, and pioneering practical algorithms.

2 Preliminary

In this section, we will introduce the basic concepts and notations used throughout the paper, and provide a brief overview of watermarking and speculative sampling techniques for large language models.

A language model defines a probability distribution over sequences of tokens from a vocabulary set Σ. It assigns a probability $P(x_{n+1}|x_1, x_2, ..., x_n)$ to the next token x_{n+1} given the context of previous tokens $x_1, x_2, ..., x_n$. We use Δ_Σ to denote the set of all possible probability distributions over the vocabulary Σ.

Following Hu et al. [12], we define a watermarking scheme as a tuple $(\mathcal{E}, P_E, R)$, where $\mathcal{E}$ is a set of watermark codes, P_E is a probability distribution over $\mathcal{E}$, and $R : \mathcal{E} \times \Delta_\Sigma \to \Delta_\Sigma$ is a reweighting function that maps a watermark code $E \in \mathcal{E}$ and a probability distribution $P \in \Delta_\Sigma$ to a watermarked distribution $R_E(P) \in \Delta_\Sigma$. We focus on unbiased watermarking schemes that satisfy $\mathbb{E}_{E\sim P_E}[R_E(P)] = P$ for all $P \in \Delta_\Sigma$, unless explicitly stated otherwise.

To generate a watermarked token x, we first compute a watermark code $E \sim P_E$ based on the context, and then sample the token from the watermarked distribution, i.e., $x \sim R_E(P)$. The entropy of the distribution P determines the maximum amount of watermark that can be injected. For a distribution P with high entropy, the divergence between the watermarked distribution $R_E(P)$ and the original distribution P can be larger, allowing for more watermark information to be injected.

The presence of the watermark can be detected by statistical tests. The pivotal quantity used in these tests is often referred to as the watermark score. A higher watermark score implies a more detectable watermark. The log likelihood ratio (LLR) is the most powerful score for detecting watermarks in the absence of any perturbations. However, in practice, more robust scores such as the maximin-LLR or likelihood-agnostic scores are often used. In this paper, we consider two specific watermark scores: the maximin-LLR score, which is described in detail in [12], and the U score, which is a likelihood-agnostic score that can be defined for both DeltaGumbel reweight and Gamma reweight schemes. The details of the U score, DeltaGumbel reweight and Gamma reweight are provided in Appendix D.

The P-value can be computed by considering the absence of a watermark as the null hypothesis. For a score S with a known moment-generating function (MGF), the P-value can be upper bounded using the Chernoff bound:

$$P_{\text{null}}(S \geq \hat{S}) \leq \min_{\lambda \geq 0} \mathbb{E}[e^{\lambda S}] \exp\Big(-\lambda \hat{S}\Big). \tag{1}$$

Speculative sampling [16, 5] is a technique for accelerating the generation of tokens from a target model P by leveraging a faster draft model Q. The key idea is to first sample a draft token $\tilde{x}$ from the draft model Q, and then accept or reject it based on the ratio of the target and draft probabilities. If the draft token is rejected, a new token is sampled from a residual distribution proportional to the difference between the target and draft probabilities. Formally, the speculative sampling process generates a token x as follows:

$$\mathcal{P}(x = j|\tilde{x} = i) = \begin{cases} \min(1, \frac{P(i)}{Q(i)}) & \text{if } i = j, \\ \frac{(1-\frac{P(i)}{Q(i)})_+(P(j)-Q(j))_+}{\sum_{z\in\Sigma}(Q(z)-P(z))_+} & \text{if } i \neq j, \end{cases} \tag{2}$$

where $(x)_+ = \max(0, x)$. The design of the speculative process ensures that the final distribution of the generated token x matches the target distribution P. The efficiency of speculative sampling can be measured by the overlap probability $\alpha(P, Q) = \sum_{t\in\Sigma} \min(P(t), Q(t))$, which is the probability of accepting the draft token in each step. The overlap probability is related to the total variation distance between P and Q by $\text{TV}(P, Q) = 1 - \alpha(P, Q)$. Such a speculative sampling process can be applied multiple times to generate and verify multiple draft tokens in one step.

Due to space limitations, we have moved the discussion of other related works to the Appendix A.

3 Two Reweight Framework for Accelerated Generation of Watermarked Tokens

In this section, we propose a novel framework called the *two reweight framework* for accelerating the generation of watermarked tokens based on speculative sampling techniques. The motivation behind this non-trivial framework is that naively applying speculative sampling to a watermarked target distribution $R_E(P)$ may significantly reduce the overlap probability $\alpha(R_E(P), Q)$ with the draft distribution Q, leading to a small sampling efficiency.

The key innovation of the *two reweight framework* is to apply a separate reweighting function R' to the draft distribution Q, using the same watermark code E as the one used for reweighting the target distribution. By doing so, we aim to increase the overlap probability between the watermarked target distribution $R_E(P)$ and the watermarked draft distribution $R'_E(Q)$, i.e., $\alpha(R_E(P), R'_E(Q))$, thus improving the sampling efficiency.

Formally, we define the watermarked draft distribution using another reweighting function $(\mathcal{E}, P_E, R')$, where $R' : \mathcal{E} \times \Delta_\Sigma \to \Delta_\Sigma$ is a function that maps a watermark code $E \in \mathcal{E}$ and a draft distribution $Q \in \Delta_\Sigma$ to a watermarked draft distribution $R'_E(Q) \in \Delta_\Sigma$. The framework itself does not require the watermarked draft distribution to be unbiased, i.e., $\mathbb{E}_{E\sim P_E}[R'_E(Q)] = Q$ for all $Q \in \Delta_\Sigma$. However, we will see later that this unbiasedness property naturally emerges when we require the final output distribution to be unbiased and aim to improve the sampling efficiency (Lemma 3).

To generate a watermarked token, we first sample a draft token $\tilde{x}$ from the watermarked draft distribution, i.e., $\tilde{x} \sim R'_E(Q)$ or equivalently $\mathcal{P}(\tilde{x} = i) = R'_E(Q)(i)$ for all $i \in \Sigma$. Then, we perform certain speculative sampling based on the draft token to obtain the generated token x. The speculative process is defined by a conditional probability distribution $A(j|i)$ for all $i, j \in \Sigma$, where $A(\cdot|i) \in \Delta_\Sigma$ for each i. The design of A can depend on the target distribution P, the draft distribution Q, and the watermark code E. The probability of generating a token $x = j$ given a draft token $\tilde{x} = i$ is given by $\mathcal{P}(x = j|\tilde{x} = i) = A(j|i)$.

The distribution of the generated token, which we call the generation distribution, can be computed as follows:

$$\mathcal{P}(x = j) = \sum_{i\in\Sigma} \mathcal{P}(x = j|\tilde{x} = i)\mathcal{P}(\tilde{x} = i) = \sum_{i\in\Sigma} A(j|i)R'_E(Q)(i) = (A \circ R'_E(Q))(j). \quad (3)$$

We denote the generation distribution by $\widehat{R}_E(P) = A \circ R'_E(Q)$.

To ensure that the *two reweight framework* produces an unbiased output distribution, we require that for all $P \in \Delta_\Sigma$:

$$\mathbb{E}_{E\sim P_E}[\widehat{R}_E(P)] = P. \quad (4)$$

4 No-go Theorem

Despite the potential of the *two reweight framework*, we present a no-go theorem that shows the impossibility of simultaneously maintaining the watermark strength and sampling efficiency when the vocabulary size is greater than two.

Theorem 1 (No-go Theorem)**.** *When the vocabulary size $|\Sigma| > 2$, there do not exist non-trivial reweighting functions $R : \mathcal{E} \times \Delta_\Sigma \to \Delta_\Sigma$ and $R' : \mathcal{E} \times \Delta_\Sigma \to \Delta_\Sigma$, and a speculative process $A(j|i)$ such that for all $P, Q \in \Delta_\Sigma$:*

1. *The watermark strength is maintained: $\widehat{R}_E(P) = R_E(P)$.*
2. *The sampling efficiency is maintained: $\alpha(P, Q) = \mathbb{E}_{E\sim P_E}[\sum_{i\in\Sigma} A(i|i)R'_E(Q)(i)]$.*

Remark 2 (Condition for maintaining the watermark strength)**.** *The condition $\widehat{R}_E(P) = R_E(P)$ in Theorem 1 ensures that the watermark strength is maintained by keeping the average watermark score unchanged, i.e.,*

$$\underbrace{\mathbb{E}_{E\sim P_E}\mathbb{E}_{t\sim R_E(P)}[Score(t, E)]}_{w:=} = \underbrace{\mathbb{E}_{E\sim P_E}\mathbb{E}_{t\sim \widehat{R}_E(P)}[Score(t, E)]}_{w':=}, \quad (5)$$

where $Score(t, E)$ is an arbitrary function that measures the watermark strength.

Strictly speaking, to ensure that the watermark strength remains unchanged, we only need to require $w = w'$, and the condition $\widehat{R}_E(P) = R_E(P)$ is a sufficient condition. However, due to the large design space of scoring functions, if we want to maintain $w = w'$ for every possible score, then $\widehat{R}_E(P) = R_E(P)$ becomes a necessary condition.

On the other hand, for a fixed scoring function and a specific R_E, it is possible that $\widehat{R}_E(P) \neq R_E(P)$ while $w = w'$ or even $w < w'$. In other words, the condition $\widehat{R}_E(P) = R_E(P)$ is not always necessary for maintaining the watermark strength for a specific scoring function and reweighting function.

The proof of the no-go theorem relies on the following two lemmas, which reveal the connections between maintaining the sampling efficiency, maintaining the watermark strength, and the properties of the reweighting functions.

Lemma 3 (Maintaining Sampling Efficiency Implies Unbiased Watermarked Draft Model)**.** *If for all $P, Q \in \Delta_\Sigma$, we have*

$$\alpha(P, Q) = \mathbb{E}_{E\sim P_E}\left[\sum_{i\in\Sigma} A(i|i)R'_E(Q)(i)\right],$$

then $\mathbb{E}_{E\sim P_E}[R'_E(Q)] = Q$ for all $Q \in \Delta_\Sigma$.

Lemma 4 (Maintaining Watermark Strength and Sampling Efficiency Implies Same Reweight Function)**.** *Under the two reweight framework, if for all $P, Q \in \Delta_\Sigma$, we have*

$$\alpha(P, Q) = \mathbb{E}_{E\sim P_E}\left[\sum_{i\in\Sigma} A(i|i)R'_E(Q)(i)\right], \qquad \widehat{R}_E(P) = R_E(P),$$

then $R'_E(Q) = R_E(Q)$ for all $Q \in \Delta_\Sigma$.

The proofs of these lemmas are deferred to Appendix B. With these lemmas, we can now prove the no-go theorem.

Proof of Theorem 1. According to Lemma 4, maintaining both the watermark strength and sampling efficiency under the *two reweight framework* implies that $R'_E(Q) = R_E(Q)$ for all $Q \in \Delta_\Sigma$. Therefore, we have

$$\alpha(P, Q) \leq \mathbb{E}_{E\sim P_E}[\alpha(R_E(P), R_E(Q))]. \tag{6}$$

To see this, note that

$$R_E(P)(i) = \widehat{R}_E(P)(i) = \sum_j A(i|j)R_E(Q)(j) \geq A(i|i)R_E(Q)(i),$$
$$R_E(Q)(i) \geq A(i|i)R_E(Q)(i),$$
$$A(i|i)R_E(Q)(i) \leq \min(R_E(Q)(i), R_E(P)(i)).$$

Summing over i, we get

$$\sum_i A(i|i)R_E(Q)(i) \leq \sum_i \min(R_E(Q)(i), R_E(P)(i)) = \alpha(R_E(Q), R_E(P)).$$

Taking the expectation over E, we obtain Equation (6).

Recall that $\alpha(P, Q) = 1 - \mathrm{TV}(P, Q)$, where $\mathrm{TV}(P, Q)$ denotes the total variation distance between P and Q. Therefore, Equation (6) is equivalent to

$$\mathrm{TV}(P, Q) \geq \mathbb{E}_{E\sim P_E}[\mathrm{TV}(R_E(P), R_E(Q))]. \tag{7}$$

Viewing $P, Q \in \Delta_\Sigma$ as n-dimensional vectors, where $n = |\Sigma|$, we can express the total variation distance as

$$2\,\mathrm{TV}(P, Q) = \max_{u\in[-1,1]^n} \langle u, P - Q\rangle, \tag{8}$$

where the maximum is attained at $u^*(P,Q) = \operatorname{sign}(P-Q)$. Using this expression, we have

$$\begin{aligned}\mathbb{E}_{E\sim P_E} 2\,\mathrm{TV}(R_E(P), R_E(Q)) &= \mathbb{E}_{E\sim P_E}\left[\max_{u\in[-1,1]^n} \langle u, R_E(P) - R_E(Q)\rangle\right] \\ &\geq \mathbb{E}_{E\sim P_E}\langle u^*(P,Q), R_E(P) - R_E(Q)\rangle \quad (9)\\ &= \langle u^*(P,Q), P-Q\rangle = 2\,\mathrm{TV}(P,Q). \quad (10)\end{aligned}$$

Combining Equations (7) and (10), we conclude that the equality in Equation (9) must hold, which is equivalent to

$$u^*(P,Q) \in \mathrm{Argmax}_{u\in[-1,1]^n}\langle u, R_E(P) - R_E(Q)\rangle, \quad (11)$$

almost surely for random E. This condition is equivalent to the following:

$$(P-Q)(i) = 0 \implies (R_E(P) - R_E(Q))(i) = 0, \quad (12)$$
$$(P-Q)(i) \geq 0 \implies (R_E(P) - R_E(Q))(i) \geq 0, \quad (13)$$
$$(P-Q)(i) \leq 0 \implies (R_E(P) - R_E(Q))(i) \leq 0, \quad (14)$$

almost surely for random E and for all $i \in \Sigma$.

Now, let us label the symbols in the vocabulary Σ as $i \in \{1,\dots,n\}$. For a distribution $P = (p_1, p_2, \dots, p_n)$, define

$$T_i(j) = \begin{cases} p_i & j = i, \\ 1-p_i & j = i \bmod n + 1, \\ 0 & \text{otherwise.}\end{cases}$$

For example, $T_1 = (p_1, 1-p_1, 0, \dots, 0)$, $T_2 = (0, p_2, 1-p_2, 0, \dots, 0)$, and $T_n = (1-p_n, 0, \dots, 0, p_n)$. Let functions $F_i(p_i) = R_E(T_i)(i)$. We claim that

$$R_E(P)(i) = F_i(p_i). \quad (15)$$

To see this, note that $(P-T_i)(i) = p_i - p_i = 0$, so by Equation (12), we have $(R_E(P) - R_E(T_i))(i) = 0$, which implies $R_E(P)(i) = R_E(T_i)(i) = F_i(p_i)$ almost surely.

Next, we show that the functions F_i satisfy the following properties:

$$F_i(0) = 0, \quad (16)$$
$$F_i(1) = 1, \quad (17)$$
$$F_i(p) \text{ is monotonically increasing in } p, \quad (18)$$
$$\sum_i p_i = 1 \implies \sum_i F_i(p_i) = 1. \quad (19)$$

To prove Equation (16), consider the case when $p_i = 0$. In this case, $T_i(j) = 1$ if $j = i \bmod n + 1$ and $T_i(j) = 0$ otherwise. To ensure the unbiasedness of the reweighting function, we must have $\mathbb{E}_{E\sim P_E}[R_E(T_i)] = T_i$, which implies $R_E(T_i) = T_i$ almost surely. Therefore, $R_E(T_i)(i) = T_i(i) = p_i = 0$, and thus $F_i(0) = 0$.

Similarly, to prove Equation (17), consider the case when $p_i = 1$. In this case, $T_i(j) = 1$ if $j = i$ and $T_i(j) = 0$ otherwise. Again, to ensure the unbiasedness of the reweighting function, we must have $\mathbb{E}_{E\sim P_E}[R_E(T_i)] = T_i$, which implies $R_E(T_i) = T_i$ almost surely. Therefore, $R_E(T_i)(i) = T_i(i) = p_i = 1$, and thus $F_i(1) = 1$.

To prove Equation (18), consider two values $p_i \geq p_i'$. Define T_i and T_i' as follows:

$$T_i(j) = \begin{cases} p_i & j = i, \\ 1-p_i & j = i \bmod n + 1, \\ 0 & \text{otherwise,}\end{cases} \qquad T_i'(j) = \begin{cases} p_i' & j = i, \\ 1-p_i' & j = i \bmod n + 1, \\ 0 & \text{otherwise.}\end{cases}$$

Since $p_i - p_i' = (T_i - T_i')(i) \geq 0$, by Equation (13), we have $F_i(p_i) - F_i(p_i') = (R_E(T_i) - R_E(T_i'))(i) \geq 0$, which proves the monotonicity of F_i.

To prove Equation (19), notice that due to Equation (15), we have $\sum_i F_i(p_i) = \sum_i R_E(P)(i) = 1$.

Finally, according to Lemma 8, the functions F_i satisfying Equations (16) to (19) must be the identity function, i.e., $F_i(p) = p$ for all $i \in \{1, 2, \dots, n\}$ and $p \in [0,1]$. Combining this with Equation (15),

we conclude that $R_E(P) = P$ almost surely for random E, which means that the reweighting function R_E is trivial.

Therefore, when the vocabulary size $|\Sigma| > 2$, it is impossible to simultaneously maintain the watermark strength and sampling efficiency using non-trivial reweighting functions under the *two reweight framework*. □

5 Algorithms for Maintaining Watermark Strength or Sampling Efficiency

Algorithm 1 Maintaining Watermark Strength or Sampling Efficiency

Given draft sequence length K, prompt $x_1, \ldots, x_n$, target model $P(\cdot|\cdot)$, draft model $Q(\cdot|\cdot)$, code history cch as a list of context code, context code function $cc : \Sigma^* \to C$, watermark code generation function $\hat{E} : C \times Z \to \mathcal{E}$, reweighting functions $R : \mathcal{E} \times \Delta_\Sigma \to \Delta_\Sigma$, and key for watermark $z \in Z$.
Initialize draft context code history $\widetilde{cch} \leftarrow cch$.
for $t = 1 : K + 1$ **do**
  Compute context code $c_t = cc(x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_{t-1})$, watermark code $E_t = \hat{E}(c_t, z)$.
  Check skipped $skipped_t = \begin{cases} \text{true} & c_t \text{ exists in } \widetilde{cch}, \\ \text{false} & c_t \text{ doesn't exists in } \widetilde{cch}. \end{cases}$ Set $\widetilde{cch} \leftarrow \widetilde{cch} + [c_t]$.
  if $t = K + 1$ **then** Exit for loop. **end if**
  Compute distribution $Q_t(\cdot) = Q(\cdot|x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_{t-1})$.
  Let $\mathfrak{Q}_t = \begin{cases} Q_t & skipped_t = \text{true}, \\ R_{E_t}(Q_t) & skipped_t = \text{false}. \end{cases}$ Sample draft token $\tilde{x}_t \sim \mathfrak{Q}_t$.
end for
for $t = 1 : K + 1$ **in parallel do**
  Compute distribution $P_t(\cdot) = P(\cdot|x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_{t-1})$.
  Let $\mathfrak{P}_t = \begin{cases} P_t & skipped_t = \text{true}, \\ R_{E_t}(P_t) & skipped_t = \text{false}. \end{cases}$
end for
Initialize empty output list: $out \leftarrow []$. Let $(\mathbb{P}_t, \mathbb{Q}_t) = \begin{cases} (\mathfrak{P}_t, \mathfrak{Q}_t) & \text{maintain watermark strength}, \\ (P_t, Q_t) & \text{maintain sampling efficiency}. \end{cases}$
for $t = 1 : K$ **do**
  Set $cch \leftarrow cch + [c_t]$. Sample $r \sim U[0, 1]$ from a uniform distribution.
  if $r < \min(1, \frac{\mathbb{P}_t(\tilde{x}_t)}{\mathbb{Q}_t(\tilde{x}_t)})$ **then** Set $out \leftarrow out + [\tilde{x}_t]$. **else**
    Sample $x_{n+t} \sim (\mathbb{P}_t - \mathbb{Q}_t)_+$. Set $out \leftarrow out + [x_{n+t}]$. Exit for loop.
  end if
end for
if $out = [\tilde{x}_1, \ldots, \tilde{x}_K]$ **then**
  Set $cch \leftarrow cch + [c_{K+1}]$. Sample $x_{n+K+1} \sim \mathfrak{P}_{K+1}$. Set $out \leftarrow out + [x_{n+K+1}]$.
end if
Return out as generated tokens, and cch as context code history.

In this section, we present two algorithms that aim to maintain either the watermark strength or the sampling efficiency under the *two reweight framework*. In light of the no-go theorem (Theorem 1), which precludes the simultaneous maintenance of watermark strength and sampling efficiency, these algorithms provide deeper insights into the trade-offs between the two objectives.

5.1 Maintaining Watermark Strength

To maintain the watermark strength, we choose the reweight function for draft distribution to be the same as the reweight function for the target distribution, i.e., $R'_E(Q) = R_E(Q)$. The speculative process is designed as follows:

$$A(j|i) = \begin{cases} \min(1, \frac{R_E(P)(i)}{R_E(Q)(i)}) & \text{if } i = j, \\ \frac{(1 - \frac{R_E(P)(i)}{R_E(Q)(i)})_+ (R_E(P)(j) - R_E(Q)(j))_+}{\sum_{z \in \Sigma} (R_E(Q)(z) - R_E(P)(z))_+} & \text{if } i \neq j. \end{cases} \tag{20}$$

Theorem 5 (Maintaining Watermark Strength)**.** *Under the two reweight framework, if $R'_E(Q) = R_E(Q)$ and the speculative process $A(j|i)$ is defined as in Equation* (20)*, then the watermark strength is maintained, i.e., $\widehat{R}_E(P) = R_E(P)$. Moreover, the generation distribution is unbiased, i.e., $\mathbb{E}_{E\sim P_E}[\widehat{R}_E(P)] = P$ for all $P \in \Delta_\Sigma$.*

Intuitively, this algorithm applies the same reweighting function R_E to both the draft distribution Q and the target distribution P, and then performs speculative sampling based on the reweighted distributions $R_E(Q)$ and $R_E(P)$ as draft and target distribution.

5.2 Maintaining Sampling Efficiency

To maintain the sampling efficiency, we again choose the reweight function for draft distribution to be the same as the reweight function for the target distribution, i.e., $R'_E(Q) = R_E(Q)$. However, the speculative process is designed differently:

$$A(j|i) = \begin{cases} \min(1, \frac{P(j)}{Q(i)}) & \text{if } i = j, \\ \frac{(1-\frac{P(i)}{Q(i)})_+(P(j)-Q(j))_+}{\sum_{z\in\Sigma}(Q(z)-P(z))_+} & \text{if } i \neq j. \end{cases} \tag{21}$$

Theorem 6 (Maintaining Sampling Efficiency)**.** *Under the two reweight framework, if $R'_E(Q) = R_E(Q)$ and the speculative process $A(j|i)$ is defined as in Equation* (21)*, then the sampling efficiency is maintained, i.e., $\alpha(P, Q) = \mathbb{E}_{E\sim P_E}[\sum_{i\in\Sigma} A(i|i)R'_E(Q)(i)]$. Moreover, the generation distribution is unbiased, i.e., $\mathbb{E}_{E\sim P_E}[\widehat{R}_E(P)] = P$ for all $P \in \Delta_\Sigma$.*

Intuitively, this algorithm generates a watermarked draft token using the watermarked draft distribution $R_E(Q)$, and then performs the standard speculative sampling process using the original distributions Q and P as draft and target distribution.

5.3 Algorithms

The pseudo code for the two methods described in the previous sections is provided in Algorithm 1. This pseudo code applies the methods in previous sections for multiple times in each step, and also considers the context code history to ensure unbiasedness for the whole sequence. For reference, similar pseudo code for basic sampling, vanilla speculative sampling and vanilla unbiased watermarking is provided in Algorithms 2 to 4.

Remark 7 (Context code history)**.** *According to [12], the context code history is crucial for ensuring the unbiasedness of the entire generated sequence. In both algorithms, all accepted draft tokens' context codes need to be preserved in the context code history. Additionally, when a draft token is rejected, its context code should also be preserved because the newly generated random token after rejection, i.e. x_{n+t}, is not independent of the rejected random draft token $\tilde{x}_t$. By preserving the right context code history, we ensures that not only the distribution of a single token, but also the distribution of the entire generated sequence is unbiased. During computing watermark score for detection, a context code history is also necessary so that each context code only contributes to the watermark score once.*

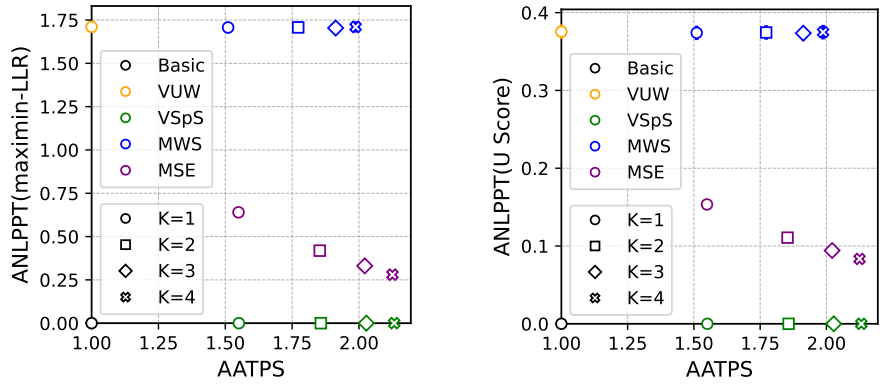
6 Experiments

To verify that Algorithm 1 can indeed maintain either the watermark strength or the sampling efficiency as claimed, we test different methods on a text summarization task on CNN_DAILYMAIL dataset [33, 10] using the Llama-7b model [42] as the target model and the Llama-68m model [25] as the draft model.

We measure the sampling efficiency by the number of accepted tokens in the out list of Algorithm 1, and report the Average Accepted Tokens Per Step (AATPS). A higher AATPS indicates a higher sampling efficiency.

To measure the watermark strength, we compute the log P-value. For likelihood-based scores, the computation follows the method in [12]. For likelihood-agnostic scores, we use U score with the Chernoff bound in Equation (1), where λ is optimized numerically. We test the watermark strength

(a) DeltaGumbel Reweight



(b) Gamma Reweight

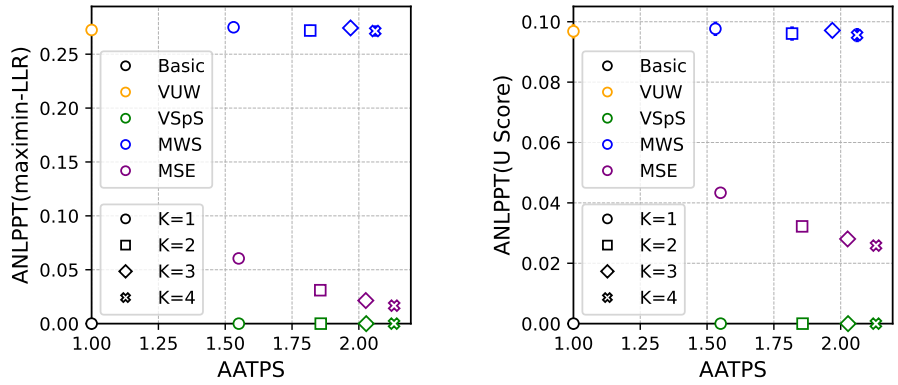


Figure 2: Comparison of different methods. The x-axis shows the Average Accepted Tokens Per Step (AATPS) as a measure of speculative sampling efficiency, while y-axis shows the Average Negative Log P-value Per Token (ANLPPT) as a measure of watermark strength. The P-value is computed based on either a likelihood-based test using the maximin-LLR score (left) or a likelihood-agnostic test using the U score (right). Watermarking is performed using either the DeltaGumbel reweight (top) or the Gamma reweight (bottom). Error bars represent 3σ confidence intervals[1].

for both the DeltaGumbel reweight and the Gamma reweight schemes. The Average Negative Log P-value Per Token (ANLPPT) is reported, with a higher value indicating a stronger watermark.

The results are shown in Figure 2. We compare the performance of Basic Sampling, Vanilla Unbiased Watermark (VUW), Vanilla Speculative Sampling (VSpS), Maintain Watermark Strength (MWS), and Maintain Sampling Efficiency (MSE).

We also measure the Per Token Time (PTT) in millisecond to evaluate the wall-time latency and verify that Algorithm 1 can indeed achieve acceleration compared to the vanilla unbiased watermark method. The Log Perplexity (LOGPPL) is computed to verify that all algorithms produce the same output distribution and do not affect the quality of the language model output. The raw data for these additional metrics are provided in Table 1 in the appendix due to space constraints.

We also conduct additional experiments using different models and tasks. In addition to the Llama-7b model, we test the Llama-13b model [42] as the target model, with Llama-68m [25] as the draft model. Besides the text summarization task, we also evaluate the methods on an open-ended text generation task. The results of these additional experiments are provided in Appendix H. The total computational cost for reproducing all the experiments in this paper is approximately 1200 A6000 GPU hours.

[1]The error bars for some methods are very small and may not be visible in the plot. The exact error bar can be found in Table 1.

The experimental results in Figure 2 and Appendix H support the following findings:

- Algorithm 1 can indeed maintain either the watermark strength or the sampling efficiency as claimed. The MWS method achieves the same watermark strength as the VUW method, while the MSE method achieves the same sampling efficiency as the VSpS method.
- Algorithm 1 can indeed accelerate the generation process compared to the vanilla unbiased watermark method. Both the MWS and MSE methods achieve lower PTT than the VUW method, as shown in Table 1.
- MWS method has only marginal sampling efficiency gap compared to VSpS, while maintain the watermark strength as VUW method, making it highly practical.
- All algorithms produce the same output distribution and do not affect the quality of the language model output, as evidenced by the similar LOGPPL values across all methods in Table 1.
- The above findings are consistent across different draft sequence length ($K = 1, 2, 3, 4$), different models (Llama-7b and Llama-13b), different tasks (text summarization and open-ended text generation), different reweight schemes (DeltaGumbel and Gamma), and different watermark detection methods (likelihood-based and likelihood-agnostic). Our extensive experiments validate the generality of the findings.

In summary, our experimental results validate the theoretical findings and demonstrate the effectiveness of the proposed Algorithm 1.

7 Conclusion

Our work provides a rigorous theoretical foundation for understanding the trade-off between watermark strength and sampling efficiency in the context of accelerated generation of watermarked tokens from large language models. We prove a no-go theorem, showing that non-trivial trade-offs are inevitable when the vocabulary size is greater than two. To explore these trade-offs, we design algorithms that prioritize either watermark strength or sampling efficiency. Our findings contribute to the development of methods for protecting the intellectual property of language models while leveraging the efficiency of speculative sampling techniques.

Acknowledgments

ZH and HH were partially supported by NSF IIS 2347592, 2347604, 2348159, 2348169, DBI 2405416, CCF 2348306, CNS 2347617.

References

[1] Scott Aaronson. My ai safety lecture for ut effective altruism. November 2022. URL `https://scottaaronson.blog/?p=6823`.

[2] Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding: 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings 4*, pages 185–200. Springer, 2001.

[3] Jack T Brassil, Steven Low, Nicholas F Maxemchuk, and Lawrence O'Gorman. Electronic marking and identification techniques to discourage document copying. *IEEE Journal on Selected Areas in Communications*, 13(8):1495–1504, 1995.

[4] Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*, 2024.

[5] 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.

[6] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[7] Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. *arXiv preprint arXiv:2306.09194*, 2023.

[8] Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. In *2023 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2023.

[9] Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D Lee, and Di He. Rest: Retrieval-based speculative decoding. *arXiv preprint arXiv:2311.08252*, 2023.

[10] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Twenty-eighth Conference on Neural Information Processing Systems*, pages 1693–1701, 2015.

[11] Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. Semstamp: A semantic watermark with paraphrastic robustness for text generation. *arXiv preprint arXiv:2310.03991*, 2023.

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

[13] 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*, pages 17061–17084. PMLR, 2023.

[14] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. *arXiv preprint arXiv:2306.04634*, 2023.

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

[16] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR, 2023.

[17] Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39, 2024.

[18] 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. *arXiv preprint arXiv:2404.01245*, 2024.

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

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

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

[22] Yixin Liu, Hongsheng Hu, Xuyun Zhang, and Lichao Sun. Watermarking text data on large language models for dataset copyright protection. *arXiv preprint arXiv:2305.13257*, 2023.

[23] Bei Luo, Raymond YK Lau, Chunping Li, and Yain-Whar Si. A critical review of state-of-the-art chatbot designs and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(1), 2022.

[24] Hasan Mesut Meral, Bülent Sankur, A Sumru Özsoy, Tunga Güngör, and Emre Sevinç. Natural language watermarking via morphosyntactic alterations. *Computer Speech & Language*, 23(1): 107–125, 2009.

[25] Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Rae Ying Yee Wong, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. Specinfer: Accelerating generative llm serving with speculative inference and token tree verification. *arXiv preprint arXiv:2305.09781*, 2023.

[26] Giovanni Monea, Armand Joulin, and Edouard Grave. Pass: Parallel speculative sampling. *arXiv preprint arXiv:2311.13581*, 2023.

[27] Travis Munyer and Xin Zhong. Deeptextmark: Deep learning based text watermarking for detection of large language model generated text. *arXiv preprint arXiv:2305.05773*, 2023.

[28] Jie Ou, Yueming Chen, and Wenhong Tian. Lossless acceleration of large language model via adaptive n-gram parallel decoding. *arXiv preprint arXiv:2404.08698*, 2024.

[29] Lip Yee Por, KokSheik Wong, and Kok Onn Chee. Unispach: A text-based data hiding method using unicode space characters. *Journal of Systems and Software*, 85(5):1075–1082, 2012.

[30] Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. A robust semantics-based watermark for large language model against paraphrasing. *arXiv preprint arXiv:2311.08721*, 2023.

[31] Stefano Giovanni Rizzo, Flavio Bertini, and Danilo Montesi. Content-preserving text watermarking through unicode homoglyph substitution. In *Proceedings of the 20th International Database Engineering & Applications Symposium*, pages 97–104, 2016.

[32] Ryoma Sato, Yuki Takezawa, Han Bao, Kenta Niwa, and Makoto Yamada. Embarrassingly simple text watermarks. *arXiv preprint arXiv:2310.08920*, 2023.

[33] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099.

[34] Benjamin Spector and Chris Re. Accelerating llm inference with staged speculative decoding. *arXiv preprint arXiv:2308.04623*, 2023.

[35] Hanshi Sun, Zhuoming Chen, Xinyu Yang, Yuandong Tian, and Beidi Chen. Triforce: Lossless acceleration of long sequence generation with hierarchical speculative decoding. *arXiv preprint arXiv:2404.11912*, 2024.

[36] Zhensu Sun, Xiaoning Du, Fu Song, Mingze Ni, and Li Li. Coprotector: Protect open-source code against unauthorized training usage with data poisoning. In *Proceedings of the ACM Web Conference 2022*, pages 652–660, 2022.

[37] Zhensu Sun, Xiaoning Du, Fu Song, and Li Li. Codemark: Imperceptible watermarking for code datasets against neural code completion models. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1561–1572, 2023.

[38] Ziteng Sun, Jae Hun Ro, Ahmad Beirami, and Ananda Theertha Suresh. Optimal block-level draft verification for accelerating speculative decoding. *arXiv preprint arXiv:2403.10444*, 2024.

[39] Ruixiang Tang, Qizhang Feng, Ninghao Liu, Fan Yang, and Xia Hu. Did you train on my dataset? towards public dataset protection with cleanlabel backdoor watermarking. *ACM SIGKDD Explorations Newsletter*, 25(1):43–53, 2023.

[40] Mercan Topkara, Umut Topkara, and Mikhail J Atallah. Words are not enough: sentence level natural language watermarking. In *Proceedings of the 4th ACM international workshop on Contents protection and security*, pages 37–46, 2006.

[41] Umut Topkara, Mercan Topkara, and Mikhail J Atallah. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In *Proceedings of the 8th workshop on Multimedia and security*, pages 164–174, 2006.

[42] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[43] Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. Towards codable text watermarking for large language models. *arXiv preprint arXiv:2307.15992*, 2023.

[44] Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. Dipmark: A stealthy, efficient and resilient watermark for large language models. *arXiv preprint arXiv:2310.07710*, 2023.

[45] Nan Yang, Tao Ge, Liang Wang, Binxing Jiao, Daxin Jiang, Linjun Yang, Rangan Majumder, and Furu Wei. Inference with reference: Lossless acceleration of large language models. *arXiv preprint arXiv:2304.04487*, 2023.

[46] Sen Yang, Shujian Huang, Xinyu Dai, and Jiajun Chen. Multi-candidate speculative decoding. *arXiv preprint arXiv:2401.06706*, 2024.

[47] Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. Tracing text provenance via context-aware lexical substitution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11613–11621, 2022.

[48] Xi Yang, Kejiang Chen, Weiming Zhang, Chang Liu, Yuang Qi, Jie Zhang, Han Fang, and Nenghai Yu. Watermarking text generated by black-box language models. *arXiv preprint arXiv:2305.08883*, 2023.

[49] KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. Robust multi-bit natural language watermarking through invariant features. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2092–2115, 2023.

[50] KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for language models. *arXiv preprint arXiv:2308.00221*, 2023.

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

[52] Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. Distillspec: Improving speculative decoding via knowledge distillation. *arXiv preprint arXiv:2310.08461*, 2023.

A Related Works

Our work lies at the intersection of two active research areas: speculative sampling for accelerated inference and watermarking techniques for language models.

A.1 Speculative Sampling for Accelerated Inference

In the domain of speculative sampling, a common approach is to use a smaller language model as the draft model [16, 5]. Efforts have been made to further increase the overlap between the draft and target models through distillation [52]. Other works focus on modifying the target model itself, such as adding "look ahead" tokens [26], introducing new heads to predict future tokens [4], reusing the computation of the large model to achieve a better latency-overlap trade-off for the draft model [19], or using the target model with partial key-value cache as the draft model [35]. Alternative approaches include using document retrieval [45, 9] or n-gram models [28] as the draft model.

When the draft sequence length is greater than one, vanilla speculative sampling is known to be suboptimal. Methods have been proposed to amend the verification process for the draft sequence, verifying the entire sequence at once instead of individual tokens, leading to a longer expected number of accepted tokens [38].

An extension of speculative sampling is to change the sequence input to a tree input. While typical language models take a sequence as input, which is a path in the symbol tree space, some works modify the input to be a tree with multiple branches. A single forward pass can then obtain probabilities on multiple branches, gathering more information to help accelerate decoding. This requires modifying the transformer implementation to change the causal attention to tree attention [46, 25, 4, 34]. Speculative sampling can also be used repeatedly, with an additional draft model to accelerate the draft model itself [34].

Our work is independent of the specific draft model used. While many recent advancements stem from faster and more accurate draft models, our method does not rely on any assumptions about the draft model. A better draft model can always be plugged in to provide faster acceleration.

The methods in Section 5 of the main text only consider the basic speculative sampling approach and do not take into account other variants such as verifying the entire sequence, tree verification, or multi-candidacy. However, our ideas can be extended to these variants and still maintain either the watermark strength or the sampling efficiency, as discussed in Appendix E.

A.2 Watermarking Techniques for Language Models

In the domain of watermarking for language models, various approaches have been explored. Some works attempt to edit existing text to embed watermarks [3, 29, 31, 32, 41, 27, 47–49, 2, 40, 24]. Others try to incorporate watermarks during the training phase [22, 39, 36, 37].

More closely related to our work is the direction of modifying the sampling stage to directly generate watermarked results. Since the pioneering works of Aaronson [1] and Kirchenbauer et al. [13], watermarking techniques have seen significant development.

To address the bias introduced by watermarking, researchers have proposed skipping watermarking on low-entropy tokens [21, 43] or accumulating entropy during the generation process and only adding a watermark when the accumulated entropy exceeds a threshold [7]. Hu et al. [12] introduced a framework that includes unbiased reweighting and context code history to ensure that the output distribution is strictly unbiased.

Subsequently, many variants have been proposed, including multi-bit watermarks [50, 8] and more robust watermarking schemes [30, 20, 14, 51, 15, 11]. Efforts have also been made to search for better watermark detection methods [44, 18].

Our work builds upon the unbiased watermarking framework of Hu et al. [12] and explores the trade-off between watermark strength and sampling efficiency when integrating watermarking with speculative sampling. To the best of our knowledge, this is the first work to investigate this intersection and provide theoretical insights and practical algorithms for navigating the inherent trade-offs.

B Proofs

Proof of Lemma 3. Let $P = Q$. Then we have

$$1 = \alpha(P, Q) = \mathbb{E}_{E\sim P_E}\left[\sum_{i\in\Sigma} A(i|i)R'_E(Q)(i)\right].$$

Note that for all $i \in \Sigma$, $A(i|i) \leq 1$, and thus

$$\sum_{i\in\Sigma} A(i|i)R'_E(Q)(i) \leq \sum_{i\in\Sigma} R'_E(Q)(i) = 1.$$

Therefore, we must have $A(i|i) = 1$ almost surely for random E and for all $i \in \Sigma$. Considering the unbiasedness requirement for the final output distribution, i.e.,

$$\forall P \in \Delta_\Sigma, \mathbb{E}_{E\sim P_E}[A \circ R'_E(Q)] = P, \tag{22}$$

we obtain $\mathbb{E}_{E\sim P_E}[R'_E(Q)] = P = Q$. □

Proof of Lemma 4. Let $P = Q$. Following the proof of Lemma 3, we have $A(i|i) = 1$ almost surely for random E and for all $i \in \Sigma$. Therefore,

$$\widehat{R}_E(P) = A \circ R'_E(Q) = R'_E(Q).$$

Since $\widehat{R}_E(P) = R_E(P)$, we conclude that $R'_E(Q) = R_E(Q)$. □

Lemma 8 (A Function Equation)**.** *Given n monotonically increasing functions $F_i : [0, 1] \to [0, 1]$ for $i \in \{1, 2, 3, \ldots, n\}$, i.e., $x \geq x' \implies F_i(x) \geq F_i(x')$, satisfying*

$$\forall i \in \{1, 2, 3, \ldots, n\}, F_i(0) = 0, F_i(1) = 1,$$
$$\sum_i x_i = 1 \implies \sum_i F_i(x_i) = 1,$$

we have $F_1(x) = F_i(x) = x$ for all $i \in \{1, 2, 3, \ldots, n\}$ and $x \in [0, 1]$.

Proof of Lemma 8. We first prove that $F_1(x) = F_2(x)$ for all $x \in [0, 1]$. Let $x_1 = 0$, $x_2 = 1 - x_3$, and $x_i = 0$ for all $i \geq 4$. We obtain

$$F_3(x_3) = 1 - F_2(1 - x_3).$$

Next, let $x_2 = 0$, $x_3 = 1 - x_1$, and $x_i = 0$ for all $i \geq 4$. This gives us

$$\begin{aligned} F_1(x_1) &= 1 - F_3(1 - x_1) \\ &= 1 - (1 - F_2(1 - (1 - x_1))) \\ &= F_2(x_1). \end{aligned}$$

Similarly, we can show that $F_1(x) = F_i(x)$ for all $i \in \{1, 2, 3, \ldots, n\}$ and $x \in [0, 1]$.

Next, we prove that for all $n \in \mathbb{N}$ and $b \leq 2^n$, $F_1(\frac{b}{2^n}) = \frac{b}{2^n}$. First, let $x_1 = \frac{1}{2}$, $x_2 = \frac{1}{2}$, and $x_i = 0$ for all $i \geq 3$. We obtain $F_1(\frac{1}{2}) = \frac{1}{2}$.

Assume that for some n, we have $F_1(\frac{b}{2^n}) = \frac{b}{2^n}$ for all $b \leq 2^n$. We will prove that $F_1(\frac{b}{2^{n+1}}) = \frac{b}{2^{n+1}}$ for all $b \leq 2^{n+1}$.

For $b \leq 2^n$, let $x_1 = \frac{b}{2^{n+1}}$, $x_2 = \frac{b}{2^{n+1}}$, $x_3 = 1 - \frac{b}{2^n}$, and $x_i = 0$ for all $i \geq 4$. We obtain $F_1(\frac{b}{2^{n+1}}) = \frac{b}{2^{n+1}}$.

For $2^n \leq b \leq 2^{n+1}$, let $x_1 = \frac{b}{2^{n+1}}$, $x_2 = 1 - \frac{b}{2^{n+1}}$, and $x_i = 0$ for all $i \geq 3$. We obtain $F_1(\frac{b}{2^{n+1}}) = \frac{b}{2^{n+1}}$.

By mathematical induction, we have $F_1(\frac{b}{2^n}) = \frac{b}{2^n}$ for all $n \in \mathbb{N}$ and $b \leq 2^n$.

Since F_1 is monotonically increasing, for all $x \in [0, 1]$ and $n \in \mathbb{N}$, we have

$$F_1(\frac{\lfloor x2^n \rfloor}{2^n}) \leq F_1(x) \leq F_1(\frac{\lceil x2^n \rceil}{2^n}).$$

Taking the limit as $n \to \infty$, we obtain

$$\lim_{n\to\infty} F_1(\frac{\lfloor x2^n \rfloor}{2^n}) = x,$$
$$\lim_{n\to\infty} F_1(\frac{\lceil x2^n \rceil}{2^n}) = x.$$

Therefore, $F_1(x) = x$ for all $x \in [0,1]$, and consequently, $F_i(x) = x$ for all $i \in \{1,2,3,\ldots,n\}$ and $x \in [0,1]$. □

Proof of Theorem 5. First, we have $\widehat{R}_E(P) = (A \circ R'_E)(Q) = (A \circ R_E)(Q)$. For any $j \in \Sigma$,

$$\begin{aligned}
\widehat{R}_E(P)(j) =& \sum_i A(j|i) R_E(Q)(i) \\
=& \min(R_E(Q)(j), R_E(P)(j)) \\
&+ \sum_{i\neq j} \frac{(R_E(Q)(i) - R_E(P)(i))_+ (R_E(P)(j) - R_E(Q)(j))_+}{\sum_{z\in\Sigma}(R_E(Q)(z) - R_E(P)(z))_+} \\
=& \min(R_E(Q)(j), R_E(P)(j)) + (R_E(P)(j) - R_E(Q)(j))_+ \\
=& R_E(P)(j).
\end{aligned}$$

Therefore, $\widehat{R}_E(P) = R_E(P)$, which means the watermark strength is maintained.

To prove the unbiasedness of the generation distribution, note that for all $P \in \Delta_\Sigma$,

$$\mathbb{E}_{E\sim P_E}[\widehat{R}_E(P)] = \mathbb{E}_{E\sim P_E}[R_E(P)] = P,$$

where the last equality follows from the unbiasedness of the reweighting function R_E. □

Proof of Theorem 6. To prove that the sampling efficiency is maintained, we have

$$\begin{aligned}
\mathbb{E}_{E\sim P_E}\left[\sum_{i\in\Sigma} A(i|i) R'_E(Q)(i)\right] &= \mathbb{E}_{E\sim P_E}\left[\sum_{i\in\Sigma} A(i|i) R_E(Q)(i)\right] \\
&= \sum_{i\in\Sigma} A(i|i)\mathbb{E}_{E\sim P_E}[R_E(Q)(i)] \\
&= \sum_{i\in\Sigma} A(i|i) Q(i) \\
&= \sum_{i\in\Sigma} \min(Q(i), P(i)) \\
&= \alpha(P,Q).
\end{aligned}$$

To prove the unbiasedness of the generation distribution, we have for any $j \in \Sigma$,

$$\begin{aligned}
\mathbb{E}_{E\sim P_E}[\widehat{R}_E(P)](j) &= \mathbb{E}_{E\sim P_E}\left[\sum_i A(j|i) R_E(Q)(i)\right] \\
&= \sum_i A(j|i)\mathbb{E}_{E\sim P_E}[R_E(Q)](i) \\
&= \sum_i A(j|i) Q(i) \\
&= \min(Q(j), P(j)) + \sum_{i\neq j} \frac{(Q(i) - P(i))_+ (P(j) - Q(j))_+}{\sum_{z\in\Sigma}(Q(z) - P(z))_+} \\
&= \min(Q(j), P(j)) + (P(j) - Q(j))_+ \\
&= P(j).
\end{aligned}$$

Therefore, $\mathbb{E}_{E\sim P_E}[\widehat{R}_E(P)] = P$ for all $P \in \Delta_\Sigma$, which means the generation distribution is unbiased. □

C Algorithms

Algorithm 2 Basic Sampling

Given generated sequence length K, prompt $x_1, \ldots, x_n$, target model $P(\cdot|\cdot)$.
Initialize empty output list: $out \leftarrow []$.
for $t = 1 : K$ **do**
  Compute distribution $P_t(\cdot) = P(\cdot|x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+t-1})$.
  Sample token $x_{n+t} \sim P_t$. Set $out \leftarrow out + [x_{n+t}]$.
end for
Return out as generated tokens.

Algorithm 3 Vanilla Speculative Sampling

Given draft sequence length K, prompt $x_1, \ldots, x_n$, target model $P(\cdot|\cdot)$, and draft model $Q(\cdot|\cdot)$.
for $t = 1 : K$ **do**
  Compute distribution $Q_t(\cdot) = Q(\cdot|x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_{t-1})$. Sample draft token $\tilde{x}_t \sim Q_t$.
end for
for $t = 1 : K + 1$ **in parallel do**
  Compute distribution $P_t(\cdot) = P(\cdot|x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_{t-1})$.
end for
Initialize empty output list: $out \leftarrow []$.
for $t = 1 : K$ **do**
  Sample $r \sim U[0, 1]$ from a uniform distribution.
  if $r < \min(1, \frac{P_t(\tilde{x}_t)}{Q_t(\tilde{x}_t)})$ **then** Set $out \leftarrow out + [\tilde{x}_t]$. **else**
    Sample $x_{n+t} \sim (P_t - Q_t)_+$. Set $out \leftarrow out + [x_{n+t}]$. Exit for loop.
  end if
end for
if $out = [\tilde{x}_1, \ldots, \tilde{x}_K]$ **then**
  Sample $x_{n+K+1} \sim P_{K+1}$. Set $out \leftarrow out + [x_{n+K+1}]$.
end if
Return out as generated tokens.

Algorithm 4 Vanilla Unbiased Watermark Method

Given generated sequence length K, prompt $x_1, \ldots, x_n$, target model $P(\cdot|\cdot)$, code history cch as a list of context code, context code function $cc : \Sigma^* \to C$, watermark code generation function $\hat{E} : C \times Z \to \mathcal{E}$, reweighting functions $R : \mathcal{E} \times \Delta_\Sigma \to \Delta_\Sigma$, and key for watermark $z \in Z$.
Initialize empty output list: $out \leftarrow []$.
for $t = 1 : K$ **do**
  Compute context code $c_t = cc(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+t-1})$, watermark code $E_t = \hat{E}(c_t, z)$.
  Check skipped $skipped_t = \begin{cases} \text{true} & c_t \text{ exists in } cch, \\ \text{false} & c_t \text{ doesn't exists in } cch. \end{cases}$ Set $cch \leftarrow cch + [c_t]$.
  Compute distribution $P_t(\cdot) = P(\cdot|x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+t-1})$.
  Let $\mathfrak{P}_t = \begin{cases} P_t & skipped_t = \text{true}, \\ R_{E_t}(P_t) & skipped_t = \text{false}. \end{cases}$ Sample token $x_{n+t} \sim \mathfrak{P}_t$. Set $out \leftarrow out + [x_{n+t}]$.
end for
Return out as generated tokens.

D U Score, DeltaGubel Reweight, and Gamma Reweight

In this section, we provide detailed definitions of the U score, DeltaGumbel reweight, and Gamma reweight schemes, which are used in the main text to compute the P-value for detecting watermarks.

D.1 DeltaGumbel Reweight

In the DeltaGumbel reweight scheme, the watermark code E is a list of $|\Sigma|$ independent and identically distributed standard Gumbel variables. The reweighting function is defined as:

$$R_E(P) := \delta_{a^*}, \qquad a^* := \text{argmax}_a\{\log P(a) \ + \ E(a)\} \tag{23}$$

where δ_{a^*} is the Dirac delta function centered at a^*.

The U score for the DeltaGumbel reweight is defined as:

$$U = \exp(-\exp(-E(x))) \in [0, 1]. \tag{24}$$

If there is no watermark added while generating x, in other word, if the token x is independent with E, then the random U is uniformly distribution in $[0, 1]$.

The logarithm of the moment-generating function (MGF) of the U score for the DeltaGumbel reweight is given by:

$$\log \mathbb{E}[\exp(\lambda U)] = -\log(\lambda) + \log\left(e^\lambda - 1\right). \tag{25}$$

D.2 Gamma Reweight

In the Gamma reweight scheme, the watermark code E is a random bijection from Σ to the set $\{0, 1, 2, \ldots, |\Sigma| - 1\}$. The reweighting function is defined as:

$$R_E(P)(t) := A_{E,P}(E(t)) - A_{E,P}(E(t) - 1), \tag{26}$$

$$A_{E,P}(i) := \max\left\{2\left(\sum_{a\in\Sigma} \mathbf{1}(E(a) \le i)P(a)\right) - 1, 0\right\}. \tag{27}$$

The U score for the Gamma reweight is defined as:

$$U = \frac{E(x) + \frac{1}{2}}{|\Sigma|} \in [0, 1]. \tag{28}$$

If there is no watermark added while generating x, in other word, if the token x is independent with E, then the random U is uniformly distribution in $\{\frac{\frac{1}{2}}{|\Sigma|}, \frac{\frac{3}{2}}{|\Sigma|}, \ldots, \frac{|\Sigma|-\frac{1}{2}}{|\Sigma|}\}$.

The logarithm of the moment-generating function (MGF) of the U score for the Gamma reweight is given by:

$$\log \mathbb{E}[\exp(\lambda U)] = -\log\left(2|\Sigma|\sinh(\frac{\lambda}{2|\Sigma|})\right) + \log\left(e^\lambda - 1\right). \tag{29}$$

Both the DeltaGumbel reweight and Gamma reweight schemes are unbiased [12], meaning that for any distribution $P \in \Delta_\Sigma$, we have:

$$\mathbb{E}_{E\sim P_E}[R_E(P)] = P. \tag{30}$$

The U scores defined for these reweight schemes are likelihood-agnostic, which means that they do not depend on the original distribution P. This property makes them possibly more robust to perturbations compared to likelihood-based scores such as the LLR score.

To compute the P-value for detecting watermarks using the U score, we can substitute the corresponding MGF into Equation (1).

E Extension to Variants of Speculative Sampling

The analysis and process presented in Section 5 focus on the basic speculative sampling approach, where a single draft token is sampled and then accepted or rejected. Algorithm 1 apply such process multiple times, accepting or rejecting tokens one by one, similar to vanilla speculative sampling.

Recent developments in speculative sampling have introduced various new techniques, such as verifying the entire sequence, tree verification, or multi-candidacy (see Appendix A for details).

While Algorithm 1 in Section 5 does not explicitly consider these variants, the underlying ideas can be directly extended to general speculative sampling approaches.

To maintain the watermark strength, the intuition is to apply the watermark to the draft tokens by sampling them from the watermarked draft model distribution, denoted as Q_w. **Then, the watermarked target distribution P_w is computed, and speculative sampling is performed, treating Q_w as the draft model and P_w as the target model.** Since speculative sampling ensures that the generated content follows the distribution of the target model, the final generated results will be drawn from the distribution P_w. Consequently, the watermark strength remains unchanged compared to directly sampling from P_w.

To maintain the sampling efficiency, the intuition is to apply the watermark to the draft tokens by sampling them from the watermarked draft model distribution Q_w. **Then, speculative sampling is performed, treating Q as the draft model and P as the target model.** Under the expectation of random watermark codes, the draft tokens follow the distribution Q. Therefore, the efficiency of speculative sampling remains unchanged compared to directly sampling draft tokens from Q.

These arguments do not depend on the specific form of speculative sampling, and do not assume the structure of the draft tokens and draft models, making the ideas presented in Section 5 applicable to various speculative sampling variants.

We acknowledge that experimental validations would be helpful to demonstrate the effectiveness of the extended methods in practice. However, due to implementation/computation cost and the focus of this paper on the foundational theory, we only present a high-level discussion and left out the empirical validations for extended methods.

F Broader Impacts

This paper presents work whose goal is to accelerate the generation speed of existing watermarking methods for large language models. There are several potential positive societal impacts of our work. By making watermarking techniques more practical and efficient, it may encourage their wider adoption. This can help protect the intellectual property rights.

However, there are also potential negative societal impacts to consider. Although our unbiased watermarking approach ensures the validity of the model outputs is not compromised, there is a risk that watermarking techniques could be abused. For example, unbiased watermarks are undetectable, which could enable tracking and surveillance, raising privacy concerns.

To mitigate potential negative impacts, it is important that watermarking techniques are used responsibly. This includes transparency about the use of watermarks, obtaining user consent where applicable, and putting safeguards in place to prevent misuse.

In conclusion, while our work on accelerating watermarking for language models has the potential to encourage wider adoption and protect intellectual property, it is important to carefully consider and address potential negative societal impacts to ensure the technology is used responsibly and ethically.

G Limitation

Our work makes significant contributions to the field of watermarking and speculative sampling for large language models, but it also has several limitations.

In terms of theoretical analysis, the no-go theorem assumes a specific *two reweight framework*. Although this framework is general, it is possible that other frameworks or methods may lead to different theoretical results. This paper represents the first exploration in this direction, and the *two reweight framework* is also the first attempt. Future work may discover more powerful frameworks that yield different insights.

Regarding experimental validation, we use relatively small language models and basic draft model. While our experiments verify the effectiveness of the theory, the acceleration ratio may not represent the state-of-the-art. Speculative sampling techniques have been developing rapidly in recent times. If combined with the latest advances, it should be possible to achieve even higher Average Accepted

Tokens Per Step (AATPS) and lower Per Token Time (PTT), though it is not directly related to our main contribution.

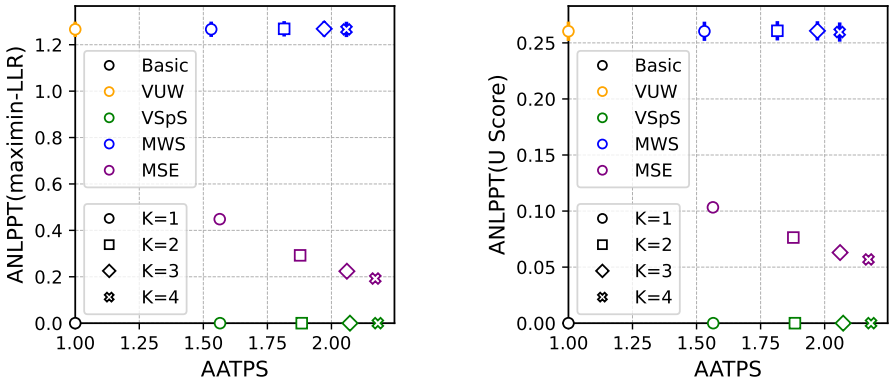
The choice of draft sequence length is critical in the deployment. Most of the existing methods select the optimal draft sequence length based on trial and error. This paper does not make contributions to determining the optimal draft sequence length. However, it should be noted that the maintain watermark strength (MWS) method proposed in this paper reduces the speculative efficiency, which also affects the selection of the optimal draft sequence length. Care should be taken in the deployment to optimally select this parameter.

If the maintain watermark strength (MWS) method is used, the inference speed will decrease slightly. This creates a side channel, and users may infer from this whether the backend service uses the MWS algorithm, which compromises the undetectability of the watermark.

Despite these limitations, our work provides valuable insights and advances the state-of-the-art in watermarking and speculative sampling techniques for large language models. We hope our findings will stimulate further research to realize the full potential of these techniques.

H Additional Experiment Results

(a) DeltaGumbel Reweight



(b) Gamma Reweight

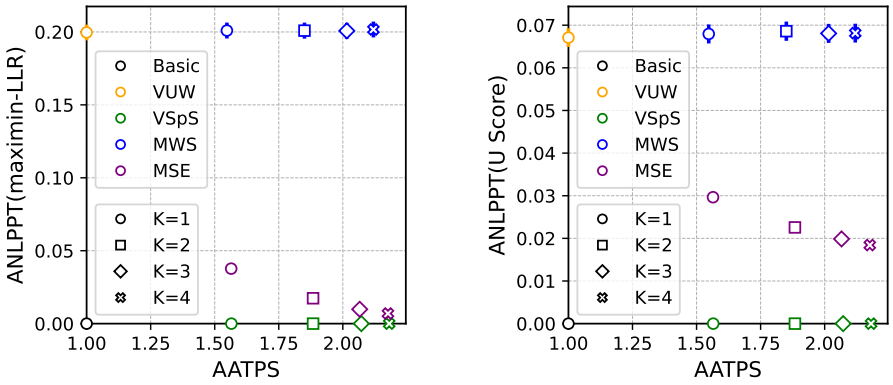
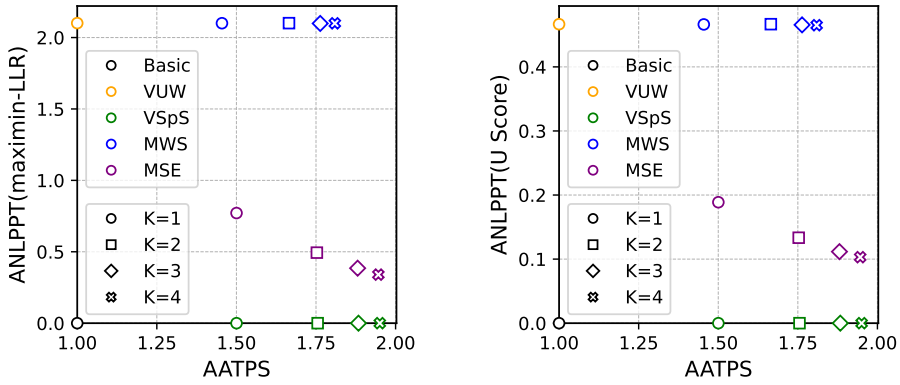


Figure 3: Text summarization task with LLaMa-13b model [42] as target model and LLaMa-68m model [25] as reference model.

(a) DeltaGumbel Reweight



(b) Gamma Reweight

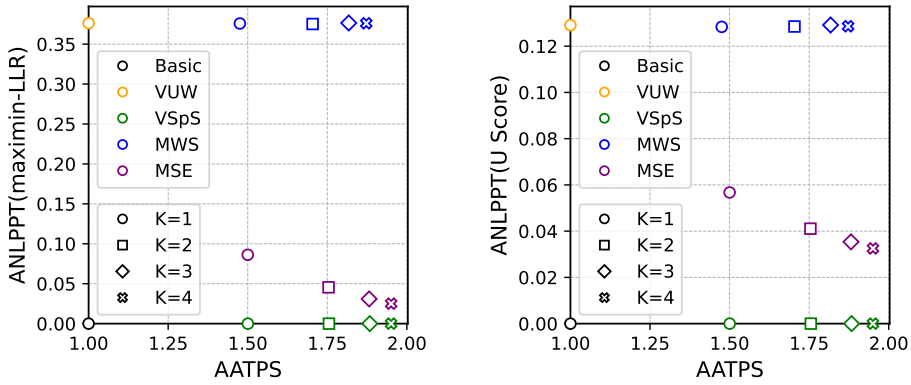
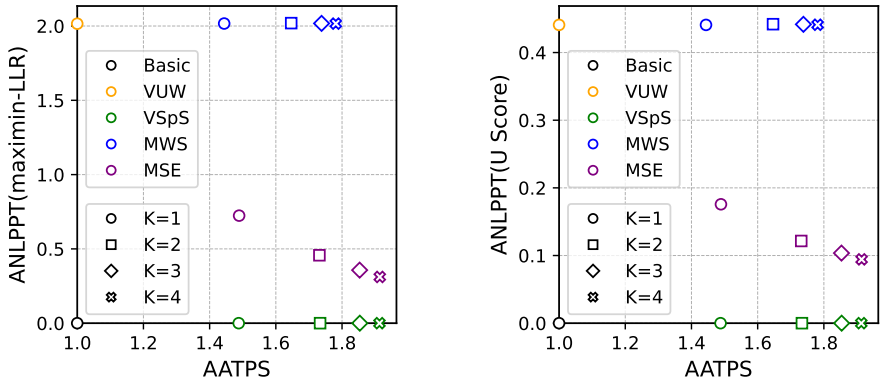


Figure 4: Open-ended text generation task with LLaMa-7b model [42] as target model and LLaMa-68m model [25] as reference model.

(a) DeltaGumbel Reweight



(b) Gamma Reweight

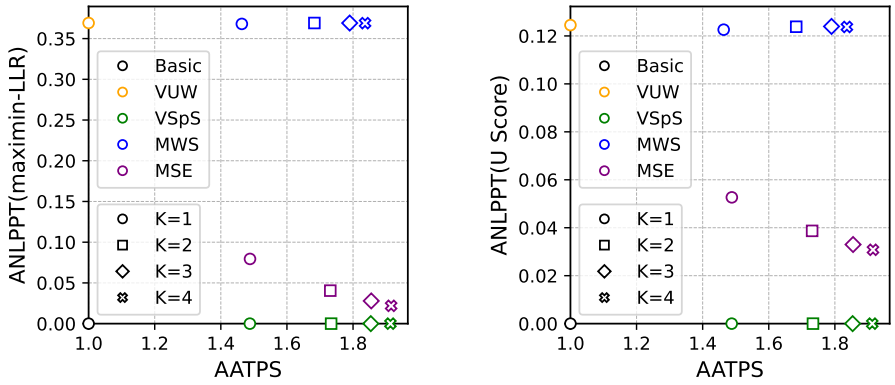


Figure 5: Open-ended text generation task with LLaMa-13b model [42] as target model and LLaMa-68m model [25] as reference model.

K	method	reweight	AATPS	PTT	LOGPPL
1	Basic	No Reweight	1.0 ± 0.0	29.58 ± 0.07	1.75 ± 0.03
1	VUW	DeltaGumbel	1.0 ± 0.0	30.40 ± 0.07	1.77 ± 0.03
1	VUW	Gamma	1.0 ± 0.0	32.81 ± 0.07	1.74 ± 0.03
1	VSpS	No Reweight	$\mathbf{1.5508 \pm 0.0017}$	21.64 ± 0.06	1.76 ± 0.03
1	MSE	DeltaGumbel	$\mathbf{1.5494 \pm 0.0017}$	22.58 ± 0.06	1.77 ± 0.03
1	MSE	Gamma	$\mathbf{1.5504 \pm 0.0017}$	25.15 ± 0.07	1.74 ± 0.03
1	MWS	DeltaGumbel	1.5105 ± 0.0017	23.24 ± 0.07	1.77 ± 0.03
1	MWS	Gamma	1.5312 ± 0.0017	25.57 ± 0.07	1.76 ± 0.03
2	VSpS	No Reweight	$\mathbf{1.857 \pm 0.003}$	19.41 ± 0.07	1.74 ± 0.03
2	MSE	DeltaGumbel	$\mathbf{1.853 \pm 0.003}$	20.46 ± 0.07	1.78 ± 0.03
2	MSE	Gamma	$\mathbf{1.856 \pm 0.003}$	23.42 ± 0.08	1.73 ± 0.03
2	MWS	DeltaGumbel	1.773 ± 0.003	21.50 ± 0.08	1.77 ± 0.03
2	MWS	Gamma	1.818 ± 0.003	25.29 ± 0.09	1.73 ± 0.03
3	VSpS	No Reweight	$\mathbf{2.028 \pm 0.004}$	19.03 ± 0.08	1.75 ± 0.03
3	MSE	DeltaGumbel	$\mathbf{2.022 \pm 0.004}$	20.24 ± 0.09	1.77 ± 0.03
3	MSE	Gamma	$\mathbf{2.026 \pm 0.004}$	23.98 ± 0.10	1.74 ± 0.03
3	MWS	DeltaGumbel	1.913 ± 0.004	21.53 ± 0.10	1.76 ± 0.03
3	MWS	Gamma	1.969 ± 0.004	25.52 ± 0.11	1.75 ± 0.03
4	VSpS	No Reweight	$\mathbf{2.132 \pm 0.005}$	19.27 ± 0.09	1.72 ± 0.03
4	MSE	DeltaGumbel	$\mathbf{2.125 \pm 0.005}$	20.79 ± 0.10	1.73 ± 0.03
4	MSE	Gamma	$\mathbf{2.132 \pm 0.005}$	25.28 ± 0.12	1.71 ± 0.03
4	MWS	DeltaGumbel	1.987 ± 0.005	22.20 ± 0.12	1.77 ± 0.03
4	MWS	Gamma	2.061 ± 0.005	27.14 ± 0.14	1.72 ± 0.03

K	method	reweight	ANLPPT(U Score)	ANLPPT(maximin-LLR)
1	Basic	No Reweight	0.0 ± 0.0	0.0 ± 0.0
1	VUW	DeltaGumbel	$\mathbf{0.376 \pm 0.009}$	$\mathbf{1.71 \pm 0.03}$
1	VUW	Gamma	$\mathbf{0.097 \pm 0.002}$	$\mathbf{0.272 \pm 0.005}$
1	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
1	MSE	DeltaGumbel	0.153 ± 0.004	0.640 ± 0.014
1	MSE	Gamma	0.0433 ± 0.0012	0.0605 ± 0.0019
1	MWS	DeltaGumbel	$\mathbf{0.374 \pm 0.009}$	$\mathbf{1.71 \pm 0.03}$
1	MWS	Gamma	$\mathbf{0.098 \pm 0.002}$	$\mathbf{0.275 \pm 0.005}$
2	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
2	MSE	DeltaGumbel	0.111 ± 0.003	0.419 ± 0.010
2	MSE	Gamma	0.0322 ± 0.0010	0.0310 ± 0.0014
2	MWS	DeltaGumbel	$\mathbf{0.374 \pm 0.009}$	$\mathbf{1.71 \pm 0.03}$
2	MWS	Gamma	$\mathbf{0.096 \pm 0.002}$	$\mathbf{0.272 \pm 0.005}$
3	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
3	MSE	DeltaGumbel	0.094 ± 0.003	0.331 ± 0.009
3	MSE	Gamma	0.0281 ± 0.0009	0.0214 ± 0.0012
3	MWS	DeltaGumbel	$\mathbf{0.374 \pm 0.009}$	$\mathbf{1.70 \pm 0.03}$
3	MWS	Gamma	$\mathbf{0.097 \pm 0.002}$	$\mathbf{0.274 \pm 0.005}$
4	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
4	MSE	DeltaGumbel	0.083 ± 0.002	0.280 ± 0.008
4	MSE	Gamma	0.0258 ± 0.0008	0.0167 ± 0.0011
4	MWS	DeltaGumbel	$\mathbf{0.375 \pm 0.009}$	$\mathbf{1.71 \pm 0.03}$
4	MWS	Gamma	$\mathbf{0.096 \pm 0.002}$	$\mathbf{0.271 \pm 0.005}$

Table 1: Text summarization task with LLaMa-7b model [42] as target model and LLaMa-68m model [25] as reference model.

K	method	reweight	AATPS	PTT	LOGPPL
1	Basic	No Reweight	1.0 ± 0.0	46.169 ± 0.012	1.27 ± 0.03
1	VUW	DeltaGumbel	1.0 ± 0.0	46.956 ± 0.012	1.31 ± 0.03
1	VUW	Gamma	1.0 ± 0.0	49.333 ± 0.017	1.28 ± 0.03
1	VSpS	No Reweight	$\mathbf{1.5651 \pm 0.0017}$	31.94 ± 0.06	1.24 ± 0.03
1	MSE	DeltaGumbel	$\mathbf{1.5639 \pm 0.0017}$	32.81 ± 0.06	1.30 ± 0.03
1	MSE	Gamma	$\mathbf{1.5643 \pm 0.0017}$	35.35 ± 0.06	1.26 ± 0.03
1	MWS	DeltaGumbel	1.5307 ± 0.0017	33.62 ± 0.07	1.31 ± 0.03
1	MWS	Gamma	1.5476 ± 0.0017	35.86 ± 0.07	1.29 ± 0.03
2	VSpS	No Reweight	$\mathbf{1.884 \pm 0.003}$	27.91 ± 0.09	1.29 ± 0.03
2	MSE	DeltaGumbel	$\mathbf{1.878 \pm 0.003}$	28.98 ± 0.09	1.34 ± 0.03
2	MSE	Gamma	$\mathbf{1.884 \pm 0.003}$	31.89 ± 0.10	1.30 ± 0.03
2	MWS	DeltaGumbel	1.815 ± 0.003	30.05 ± 0.11	1.32 ± 0.03
2	MWS	Gamma	1.850 ± 0.003	33.59 ± 0.11	1.29 ± 0.03
3	VSpS	No Reweight	$\mathbf{2.072 \pm 0.005}$	26.60 ± 0.11	1.24 ± 0.03
3	MSE	DeltaGumbel	$\mathbf{2.060 \pm 0.005}$	27.94 ± 0.12	1.31 ± 0.03
3	MSE	Gamma	$\mathbf{2.066 \pm 0.005}$	31.63 ± 0.13	1.28 ± 0.03
3	MWS	DeltaGumbel	1.972 ± 0.004	29.26 ± 0.13	1.32 ± 0.03
3	MWS	Gamma	2.016 ± 0.004	33.45 ± 0.15	1.30 ± 0.03
4	VSpS	No Reweight	$\mathbf{2.181 \pm 0.006}$	26.36 ± 0.12	1.27 ± 0.03
4	MSE	DeltaGumbel	$\mathbf{2.171 \pm 0.006}$	28.00 ± 0.13	1.30 ± 0.03
4	MSE	Gamma	$\mathbf{2.176 \pm 0.006}$	32.44 ± 0.15	1.28 ± 0.03
4	MWS	DeltaGumbel	2.059 ± 0.005	29.51 ± 0.16	1.31 ± 0.03
4	MWS	Gamma	2.119 ± 0.005	34.32 ± 0.17	1.29 ± 0.03

K	method	reweight	ANLPPT(U Score)	ANLPPT(maximin-LLR)
1	Basic	No Reweight	0.0 ± 0.0	0.0 ± 0.0
1	VUW	DeltaGumbel	$\mathbf{0.260 \pm 0.009}$	$\mathbf{1.27 \pm 0.03}$
1	VUW	Gamma	$\mathbf{0.067 \pm 0.002}$	$\mathbf{0.200 \pm 0.005}$
1	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
1	MSE	DeltaGumbel	0.103 ± 0.004	0.448 ± 0.014
1	MSE	Gamma	0.0297 ± 0.0011	0.0377 ± 0.0017
1	MWS	DeltaGumbel	$\mathbf{0.260 \pm 0.009}$	$\mathbf{1.27 \pm 0.03}$
1	MWS	Gamma	$\mathbf{0.068 \pm 0.002}$	$\mathbf{0.201 \pm 0.005}$
2	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
2	MSE	DeltaGumbel	0.076 ± 0.003	0.292 ± 0.010
2	MSE	Gamma	0.0226 ± 0.0009	0.0174 ± 0.0012
2	MWS	DeltaGumbel	$\mathbf{0.261 \pm 0.009}$	$\mathbf{1.27 \pm 0.03}$
2	MWS	Gamma	$\mathbf{0.069 \pm 0.002}$	$\mathbf{0.201 \pm 0.005}$
3	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
3	MSE	DeltaGumbel	0.063 ± 0.002	0.224 ± 0.008
3	MSE	Gamma	0.0199 ± 0.0008	0.0098 ± 0.0011
3	MWS	DeltaGumbel	$\mathbf{0.261 \pm 0.009}$	$\mathbf{1.27 \pm 0.03}$
3	MWS	Gamma	$\mathbf{0.068 \pm 0.002}$	$\mathbf{0.201 \pm 0.006}$
4	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
4	MSE	DeltaGumbel	0.057 ± 0.002	0.192 ± 0.007
4	MSE	Gamma	0.0184 ± 0.0007	0.0069 ± 0.0009
4	MWS	DeltaGumbel	$\mathbf{0.260 \pm 0.009}$	$\mathbf{1.27 \pm 0.03}$
4	MWS	Gamma	$\mathbf{0.068 \pm 0.002}$	$\mathbf{0.202 \pm 0.005}$

Table 2: Text summarization task with LLaMa-13b model [42] as target model and LLaMa-68m model [25] as reference model.

K	method	reweight	AATPS	PTT	LOGPPL
1	Basic	No Reweight	1.0 ± 0.0	30.20 ± 0.08	2.161 ± 0.014
1	VUW	DeltaGumbel	1.0 ± 0.0	31.06 ± 0.08	2.147 ± 0.014
1	VUW	Gamma	1.0 ± 0.0	33.34 ± 0.09	2.158 ± 0.014
1	VSpS	No Reweight	$\mathbf{1.5004 \pm 0.0016}$	22.69 ± 0.07	2.158 ± 0.014
1	MSE	DeltaGumbel	$\mathbf{1.5004 \pm 0.0016}$	23.52 ± 0.07	2.150 ± 0.014
1	MSE	Gamma	$\mathbf{1.5001 \pm 0.0016}$	26.30 ± 0.07	2.160 ± 0.014
1	MWS	DeltaGumbel	1.4546 ± 0.0016	24.63 ± 0.07	2.147 ± 0.014
1	MWS	Gamma	1.4753 ± 0.0016	26.67 ± 0.08	2.151 ± 0.014
2	VSpS	No Reweight	$\mathbf{1.755 \pm 0.003}$	21.00 ± 0.07	2.145 ± 0.014
2	MSE	DeltaGumbel	$\mathbf{1.753 \pm 0.003}$	21.93 ± 0.07	2.152 ± 0.014
2	MSE	Gamma	$\mathbf{1.754 \pm 0.003}$	25.00 ± 0.08	2.159 ± 0.014
2	MWS	DeltaGumbel	1.665 ± 0.003	23.11 ± 0.07	2.147 ± 0.014
2	MWS	Gamma	1.704 ± 0.003	27.28 ± 0.09	2.145 ± 0.014
3	VSpS	No Reweight	$\mathbf{1.883 \pm 0.004}$	20.57 ± 0.08	2.153 ± 0.014
3	MSE	DeltaGumbel	$\mathbf{1.881 \pm 0.004}$	22.08 ± 0.08	2.148 ± 0.014
3	MSE	Gamma	$\mathbf{1.882 \pm 0.004}$	25.97 ± 0.09	2.143 ± 0.014
3	MWS	DeltaGumbel	1.763 ± 0.004	23.71 ± 0.08	2.145 ± 0.014
3	MWS	Gamma	1.817 ± 0.004	28.23 ± 0.10	2.153 ± 0.014
4	VSpS	No Reweight	$\mathbf{1.950 \pm 0.005}$	21.35 ± 0.08	2.153 ± 0.014
4	MSE	DeltaGumbel	$\mathbf{1.946 \pm 0.005}$	23.10 ± 0.09	2.153 ± 0.014
4	MSE	Gamma	$\mathbf{1.951 \pm 0.005}$	28.10 ± 0.11	2.143 ± 0.014
4	MWS	DeltaGumbel	1.809 ± 0.004	24.65 ± 0.09	2.146 ± 0.014
4	MWS	Gamma	1.872 ± 0.004	30.24 ± 0.12	2.148 ± 0.014

K	method	reweight	ANLPPT(U Score)	ANLPPT(maximin-LLR)
1	Basic	No Reweight	0.0 ± 0.0	0.0 ± 0.0
1	VUW	DeltaGumbel	$\mathbf{0.467 \pm 0.005}$	$\mathbf{2.101 \pm 0.014}$
1	VUW	Gamma	$\mathbf{0.1291 \pm 0.0015}$	$\mathbf{0.3762 \pm 0.0018}$
1	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
1	MSE	DeltaGumbel	0.189 ± 0.003	0.771 ± 0.008
1	MSE	Gamma	0.0567 ± 0.0010	0.0862 ± 0.0014
1	MWS	DeltaGumbel	$\mathbf{0.466 \pm 0.005}$	$\mathbf{2.100 \pm 0.014}$
1	MWS	Gamma	$\mathbf{0.1283 \pm 0.0015}$	$\mathbf{0.3759 \pm 0.0018}$
2	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
2	MSE	DeltaGumbel	0.133 ± 0.002	0.494 ± 0.007
2	MSE	Gamma	0.0411 ± 0.0008	0.0455 ± 0.0011
2	MWS	DeltaGumbel	$\mathbf{0.467 \pm 0.005}$	$\mathbf{2.101 \pm 0.014}$
2	MWS	Gamma	$\mathbf{0.1285 \pm 0.0015}$	$\mathbf{0.3752 \pm 0.0017}$
3	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
3	MSE	DeltaGumbel	0.1116 ± 0.0019	0.385 ± 0.006
3	MSE	Gamma	0.0354 ± 0.0008	0.0309 ± 0.0010
3	MWS	DeltaGumbel	$\mathbf{0.466 \pm 0.005}$	$\mathbf{2.099 \pm 0.014}$
3	MWS	Gamma	$\mathbf{0.1292 \pm 0.0015}$	$\mathbf{0.3765 \pm 0.0017}$
4	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
4	MSE	DeltaGumbel	0.1030 ± 0.0018	0.340 ± 0.005
4	MSE	Gamma	0.0325 ± 0.0007	0.0250 ± 0.0009
4	MWS	DeltaGumbel	$\mathbf{0.465 \pm 0.005}$	$\mathbf{2.099 \pm 0.014}$
4	MWS	Gamma	$\mathbf{0.1286 \pm 0.0015}$	$\mathbf{0.3761 \pm 0.0017}$

Table 3: Open-ended text generation task with LLaMa-7b model [42] as target model and LLaMa-68m model [25] as reference model.

K	method	reweight	AATPS	PTT	LOGPPL
1	Basic	No Reweight	1.0 ± 0.0	44.982 ± 0.012	2.087 ± 0.014
1	VUW	DeltaGumbel	1.0 ± 0.0	45.766 ± 0.012	2.062 ± 0.013
1	VUW	Gamma	1.0 ± 0.0	48.030 ± 0.017	2.086 ± 0.014
1	VSpS	No Reweight	$\mathbf{1.4879 \pm 0.0016}$	32.55 ± 0.04	2.076 ± 0.013
1	MSE	DeltaGumbel	$\mathbf{1.4891 \pm 0.0016}$	33.40 ± 0.04	2.071 ± 0.013
1	MSE	Gamma	$\mathbf{1.4883 \pm 0.0016}$	35.78 ± 0.05	2.073 ± 0.014
1	MWS	DeltaGumbel	1.4439 ± 0.0016	34.52 ± 0.05	2.063 ± 0.013
1	MWS	Gamma	1.4636 ± 0.0016	36.55 ± 0.05	2.073 ± 0.013
2	VSpS	No Reweight	$\mathbf{1.734 \pm 0.003}$	29.32 ± 0.06	2.071 ± 0.013
2	MSE	DeltaGumbel	$\mathbf{1.732 \pm 0.003}$	30.39 ± 0.06	2.074 ± 0.014
2	MSE	Gamma	$\mathbf{1.731 \pm 0.003}$	33.50 ± 0.07	2.081 ± 0.014
2	MWS	DeltaGumbel	1.646 ± 0.003	32.07 ± 0.07	2.066 ± 0.013
2	MWS	Gamma	1.683 ± 0.003	35.64 ± 0.07	2.076 ± 0.013
3	VSpS	No Reweight	$\mathbf{1.854 \pm 0.004}$	28.76 ± 0.07	2.071 ± 0.014
3	MSE	DeltaGumbel	$\mathbf{1.853 \pm 0.004}$	30.05 ± 0.08	2.075 ± 0.014
3	MSE	Gamma	$\mathbf{1.855 \pm 0.004}$	33.97 ± 0.09	2.070 ± 0.013
3	MWS	DeltaGumbel	1.738 ± 0.004	32.15 ± 0.08	2.064 ± 0.013
3	MWS	Gamma	1.790 ± 0.004	36.37 ± 0.09	2.080 ± 0.014
4	VSpS	No Reweight	$\mathbf{1.913 \pm 0.004}$	29.05 ± 0.08	2.074 ± 0.014
4	MSE	DeltaGumbel	$\mathbf{1.914 \pm 0.004}$	30.69 ± 0.09	2.072 ± 0.013
4	MSE	Gamma	$\mathbf{1.915 \pm 0.004}$	35.47 ± 0.10	2.075 ± 0.014
4	MWS	DeltaGumbel	1.781 ± 0.004	33.09 ± 0.09	2.062 ± 0.013
4	MWS	Gamma	1.836 ± 0.004	38.21 ± 0.11	2.072 ± 0.013

K	method	reweight	ANLPPT(U Score)	ANLPPT(maximin-LLR)
1	Basic	No Reweight	0.0 ± 0.0	0.0 ± 0.0
1	VUW	DeltaGumbel	$\mathbf{0.441 \pm 0.005}$	$\mathbf{2.016 \pm 0.013}$
1	VUW	Gamma	$\mathbf{0.1245 \pm 0.0015}$	$\mathbf{0.3692 \pm 0.0017}$
1	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
1	MSE	DeltaGumbel	0.176 ± 0.003	0.724 ± 0.008
1	MSE	Gamma	0.0527 ± 0.0009	0.0794 ± 0.0013
1	MWS	DeltaGumbel	$\mathbf{0.441 \pm 0.005}$	$\mathbf{2.017 \pm 0.013}$
1	MWS	Gamma	$\mathbf{0.1226 \pm 0.0014}$	$\mathbf{0.3680 \pm 0.0017}$
2	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
2	MSE	DeltaGumbel	0.122 ± 0.002	0.457 ± 0.006
2	MSE	Gamma	0.0387 ± 0.0008	0.0405 ± 0.0011
2	MWS	DeltaGumbel	$\mathbf{0.442 \pm 0.005}$	$\mathbf{2.019 \pm 0.013}$
2	MWS	Gamma	$\mathbf{0.1238 \pm 0.0014}$	$\mathbf{0.3691 \pm 0.0017}$
3	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
3	MSE	DeltaGumbel	0.1036 ± 0.0018	0.357 ± 0.006
3	MSE	Gamma	0.0330 ± 0.0007	0.0279 ± 0.0010
3	MWS	DeltaGumbel	$\mathbf{0.442 \pm 0.005}$	$\mathbf{2.018 \pm 0.013}$
3	MWS	Gamma	$\mathbf{0.1240 \pm 0.0014}$	$\mathbf{0.3691 \pm 0.0017}$
4	VSpS	No Reweight	0.0 ± 0.0	0.0 ± 0.0
4	MSE	DeltaGumbel	0.0943 ± 0.0017	0.311 ± 0.005
4	MSE	Gamma	0.0308 ± 0.0007	0.0219 ± 0.0009
4	MWS	DeltaGumbel	$\mathbf{0.441 \pm 0.005}$	$\mathbf{2.016 \pm 0.013}$
4	MWS	Gamma	$\mathbf{0.1237 \pm 0.0014}$	$\mathbf{0.3689 \pm 0.0017}$

Table 4: Open-ended text generation task with LLaMa-13b model [42] as target model and LLaMa-68m model [25] as reference model.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We summarize the main contributions at the end of Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix G.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The conditions under which our theory holds are clearly stated in Theorems 1, 5 and 6 are clearly stated.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All code is provided in the supplementary.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All code is provided in the supplementary. Data can be obtained online.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The definition of algorithm and metrics used in the experiments are reported in the paper. Code is provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report 3σ interval in Figure 2 and Appendix H.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As reported in Section 6, the total computational cost for reproducing all the experiments in this paper is approximately 1200 A6000 GPU hours.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix F.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper doesn't release new data or model.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite See et al. [33], Hermann et al. [10] for CNN_DAILYMAIL dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No human subject is involved in this study.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subject is involved in this study.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.