Watermark Smoothing Attacks against Language Models

Hongyan Chang[1] **Hamed Hassani**[2] **Reza Shokri**[3]

[1]Mohamed bin Zayed University of Artificial Intelligence [2]University of Pennsylvania
[3]National University of Singapore

Abstract

Watermarking is a key technique for detecting AI-generated text. In this work, we study its vulnerabilities and introduce the *Smoothing Attack*, a novel watermark removal method. By leveraging the relationship between the model's confidence and watermark detectability, our attack selectively smoothes the watermarked content, erasing watermark traces while preserving text quality. We validate our attack on open-source models ranging from 1.3B to 30B parameters on ten different watermarks, demonstrating its effectiveness. Our findings expose critical weaknesses in existing watermarking schemes and highlight the need for stronger defenses.

1 Introduction

Detecting whether a text is generated by language models is critical in domains such as fraud detection, fake news identification, and plagiarism prevention. A common approach is watermarking, where subtle patterns are embedded in the generated text for later detection (Aaronson, 2023; Christ et al., 2023; Huang et al., 2023; Li et al., 2024). Watermarking has gained traction in both academia and industry (Dathathri et al., 2024) as a key safeguard for language model applications. While various watermarking techniques exist, they share a core principle: biasing the token distribution to favor certain tokens over others (see Section 2).

In this work, we identify key scenarios where token-based watermarks fail and introduce a general watermark removal attack that exploits this weakness, exposing fundamental limitations in current watermarking schemes.

Effectiveness of watermarks. We say a watermark as effective if (i) the watermarked text maintains high quality, comparable to that of the corresponding un-watermarked model, and (ii) the detector reliably identifies watermark traces without incurring high error. We show both analytically and empirically that these goals are inherently in tension: higher text quality typically reduces detectability, and vice versa. This tradeoff is closely linked to the model's confidence in generating the next token. When confidence is high, watermarking has negligible impact, leaving only faint traces. When confidence is low, watermarking biases the model toward otherwise unlikely tokens, producing stronger traces at the cost of text quality.

Smoothing Attack. Based on this observation, we propose the *Smoothing Attack*, a general approach for removing token-based watermarks. For each prefix, the attack estimates the watermarked model's confidence in the output token. If confidence is low (where watermarking is strongest), we replace the token with a freshly sampled one, erasing traces while preserving fluency. If confidence is high, we retain the original output. This systematic procedure eliminates watermark signals without sacrificing quality, going beyond ad-hoc heuristics.

We evaluate our attack across ten watermarking schemes and three open-source model families: OPT (Zhang et al., 2022) (from 1.3B to 30B parameters), Llama3-8B (Dubey et al., 2024), and Qwen2-1.5B (Chu et al., 2024). In certain cases, our attack completely removes the watermark (reducing watermark detection rates to zero) while preserving the text quality. It also outperforms the state-of-the-art *Paraphrasing Attack*, which relies on GPT-3.5-turbo. Compared with paraphrasing, our method is far more cost-efficient, requiring only weak reference models (e.g., OPT-125M (Zhang et al., 2022) when attacking OPT models up to 30B). These findings reveal fundamental weaknesses in token-based watermarks and highlight the need for more robust defenses.

We evaluate our attack across ten diverse watermarking schemes and three different families of open-sourced models, OPT (Zhang et al., 2022) (from 1.3B to 30B parameters), Llama3-8B (Dubey et al., 2024) and Qwen2-1.5B (Chu et al., 2024). In certain cases, our attack completely removes the watermark (reducing watermark detection rates to zero) while preserving the text quality. Our attack can also outperform the state-of-the-art *Paraphrasing Attack*, which uses the strong GPT-3.5-turbo to paraphrase the watermarked text. Compared with *Paraphrasing Attack*, our attack is more cost-efficient, as it uses only much weaker reference models, e.g., OPT-125M (Zhang et al., 2022) when attacking OPT models from 1.3B to 30B parameters. These findings underscore critical weaknesses in existing watermarks and highlight the need for more robust defenses.

2 Preliminaries and Related Work

Given an auto-regressive language model (LM) M with vocabulary $\mathcal{V}$, the model outputs a probability distribution over tokens at each position t in a prompt by computing logits $l_t(v)$ and applying a softmax:

$$P_t(v) = \frac{\exp\big(l_t(v)\big)}{\sum_{v' \in \mathcal{V}} \exp\big(l_t(v')\big)}. \tag{1}$$

To sample the next token, common strategies include top-k sampling (Fan et al., 2018; Holtzman et al., 2018), selecting from the top k tokens by probability, and top-p (nucleus) sampling (Holtzman et al., 2019), selecting from the smallest set whose cumulative probability exceeds p.

Watermarking schemes subtly modify this sampling to embed patterns in the generated text $(v_1, \ldots, v_T)$. These patterns are later detected via a scoring function $d(v_1, \ldots, v_T)$; if the score exceeds a threshold τ, the text is deemed watermarked.

Below we briefly review representative watermarking approaches.

Green–red watermark (Kirchenbauer et al., 2023a). For each position t, a secret key and the current prefix deterministically partition the vocabulary $\mathcal{V}$ into a green list $\mathcal{G}_t$ (size $\gamma|\mathcal{V}|$) and a red list. The logits of green tokens are then boosted by a constant δ, giving the modified distribution

$$\widetilde{P}_t(v) = \frac{\exp\big(l_t(v) + \delta \cdot \mathbf{1}\{v \in \mathcal{G}_t\}\big)}{\sum_{v' \in \mathcal{V}} \exp\big(l_t(v') + \delta \cdot \mathbf{1}\{v' \in \mathcal{G}_t\}\big)}. \tag{2}$$

The detector checks whether green tokens appear more frequently than expected, computing the score

$$d(v_1, \ldots, v_T) = \frac{\sum_{t=1}^{T} \big(\mathbf{1}\{v_t \in \mathcal{G}_t\} - \gamma\big)}{\sqrt{T\gamma(1-\gamma)}}. \tag{3}$$

The score is high when green tokens are overrepresented, indicating a watermark.

Gumbel and Tournament watermarks (Aaronson, 2023; Dathathri et al., 2024). These methods introduce randomness via a secret key and selection process without altering the distribution P_t, preserving average text quality. In Gumbel sampling, noise values $u_t(v) \in [0, 1]$ are generated using a seed derived from recent tokens and a secret key; the token v_t^* is selected as $v_t^* = \arg\max_v -\frac{\log u_t(v)}{P_t(v)}$. The detector uses the score $d(v_1, \ldots, v_T) = -\sum_t \log(1 - u_t(v_t))$.

Tournament sampling similarly uses m secret functions $g^{(1)}, \ldots, g^{(m)}$ to score each token based on a seed r_t. The token is chosen through m rounds of pairwise comparisons among 2^m sampled candidates. Detection relies on the average tournament score:

$$d(v_1, \ldots, v_T) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{m}\sum_{l=1}^{m} g^{(l)}(v_t, r_t). \tag{4}$$

If the score significantly exceeds 0.5, the text is predicted as watermarked.

Other related work. The Green-red list watermark (Kirchenbauer et al., 2023a) introduces token-level bias by amplifying the probabilities of green-listed tokens and is thus considered *distortionary*. Variants differ in list construction and detection methods (Kirchenbauer et al., 2023b; Lee et al., 2023; Liu et al., 2023; Wu et al.). In contrast, Gumbel and Tournament sampling (Kuditipudi et al., 2023; Aaronson, 2023; Dathathri et al., 2024) preserve the token distribution in expectation and are *distortion-free*. Other distortion-free schemes include those by Hu et al.; Christ et al. (2023); see Zhao et al. (2024a) for a broader review. We evaluate 10 representative watermarks from both categories to demonstrate the generality of our attack.

Watermark removal techniques typically disrupt token patterns using homoglyphs, emojis, or control characters (Pajola and Conti, 2021; Boucher et al., 2022; Goodside, 2023), but often degrade fluency. A more effective approach is *paraphrasing*, where a separate model rewrites the text (Kirchenbauer et al., 2023b; Krishna et al., 2023; Piet et al., 2023). These attacks depend on strong paraphrasers—e.g., using GPT-3.5-turbo to rewrite outputs from smaller models like LLaMA-7B (OpenAI, 2023; Touvron et al., 2023).

Recent work explores adaptive paraphrasers trained to evade specific watermark detectors (Diaa et al., 2024), often requiring detailed knowledge of the watermarking algorithm. Other methods combine paraphrasing with auxiliary models to select high-quality rewrites from large candidate pools (Jovanović et al., 2024; Zhang et al., 2024). In contrast, our smoothing attack is lightweight: it leverages only top-K token probabilities from the watermarked model and a small reference model—without needing a strong paraphraser, large-scale sampling, or exact knowledge of the watermarking scheme.

3 On the Effectiveness of Watermarks

We investigate two key aspects of watermark effectiveness: *detectability* and *text quality*. Our key finding is that these aspects are inter-connected via the model's confidence in prediction and are inherent tension—improving detectability typically decreases text quality, and vice versa. This trade-off arises directly from how watermarking algorithms exploit token-level decisions, revealing a fundamental vulnerability leveraged by our proposed attack. Full derivations and further analysis are provided in Appendix C.

3.1 Token-level detectability

Detectability. Watermark detectors aggregate token-wise signals. For the Green–red scheme, the contribution of position t is

$$S_t = \underbrace{\mathbb{E}_{v\sim\widetilde{P}_t}[1\{v\in\mathcal{G}_t\}]}_{\text{watermarked}} - \underbrace{\mathbb{E}_{v\sim P_t}[1\{v\in\mathcal{G}_t\}]}_{\text{original}},$$

where P_t and $\widetilde{P}_t$ are the original and logit-shifted distributions. The detector score in Eq. equation 3 is just the normalised sum of these S_t.

Link to model confidence. With logit shift δ, S_t can be expressed solely through $\mathbb{E}_{v\sim P_t}[1\{v\in\mathcal{G}_t\}]$ (full derivation in Appendix):

$$S_t = \frac{(e^\delta-1)\left(1-\mathbb{E}_{v\sim P_t}[1\{v\in\mathcal{G}_t\}]\right)}{1+(e^\delta-1)\,\mathbb{E}_{v\sim P_t}[1\{v\in\mathcal{G}_t\}]}. \quad (5)$$

The variance of $1\{v\in\mathcal{G}_t\}$ under P_t equals $\gamma(1-\gamma)\,\|P_t\|^2$; hence departures of the expectation from its mean γ grow with the confidence measure $\|P_t\|^2=\sum_v P_t(v)^2$. Figure 1 plots S_t versus $\|P_t\|^2$ for 400 prefixes (OPT-1.3B, $\gamma=0.5$, $\delta=1.0$). High-confidence locations ($\|P_t\|^2\approx 1$) yield small S_t, whereas low-confidence ones ($\|P_t\|^2\approx|\mathcal{V}|^{-1}$) maximise S_t. The same inverse relation holds for Gumbel and Tournament watermarks (see Fig. 7 in Appendix C.5). The take-away is that when the model is confident about a token, that output token leaves little trace for watermark detection (i.e., difficult to detect); uncertainty amplifies the watermark signal (i.e., easy to detect).

3.2 Model confidence governs text quality

Ultimately, watermarking should minimize its negative impact on downstream *text quality*. While we empirically assess quality via perplexity and diversity in Section 5, we first quantify how watermarking affects token-level distributional *fidelity*, as changes at this level directly influence downstream quality. To measure fidelity loss, we use the total-variation distance:

$$D_{TV}(P_t,\widetilde{P}_t)=\tfrac{1}{2}\sum_{v\in\mathcal{V}}|P_t(v)-\widetilde{P}_t(v)|.$$

This metric is prompt- and task-agnostic, precisely capturing how watermarking alters token probabilities. Importantly, D_{TV} provides an upper bound on any smooth token-level objective—including perplexity and diversity—thus directly linking distributional fidelity to measurable text quality.

We evaluate fidelity loss under a *fixed secret key*, which deterministically partitions tokens (e.g., into green/red lists) based on the prefix. This matches the practical setting, where detectors must know the exact watermark key used during text generation.

Link to model confidence. Figure 2 (left) shows that distortion shrinks as confidence $\|P_t\|^2$ rises: sharply peaked distributions are barely perturbed, while flat ones incur substantial fidelity loss. Gumbel and Tournament watermarks exhibit the same pattern (Appendix Fig. 8). In addition, if we focus on the part where $\|P_t\|^2$ is small, we note there is very little difference between the D_{TV} measured between the original model and the watermarked model and the the D_{TV} measured between the original model and the watermarked reference model. Effectively, that means *replacing low-confidence tokens with samples from a small reference model* harms fidelity no more (and often less, according to our experiments) than the watermark itself. Conversely, at high-confidence positions the reference model can be worse than the watermarked model,

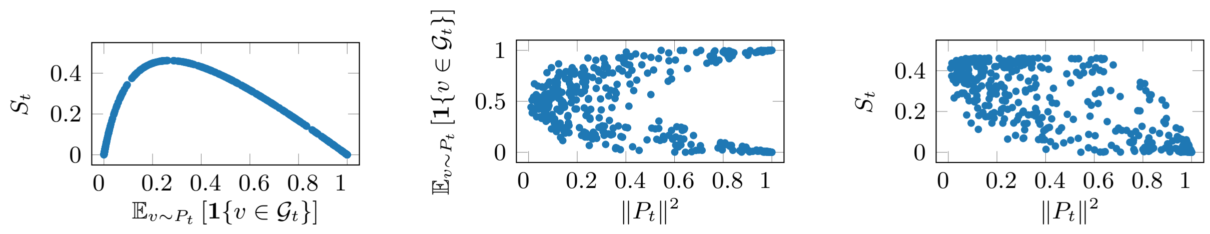

Figure 1: Correlation between the token-level watermark signal S_t, the original model's expected green-token rate $\mathbb{E}_{v\sim P_t}[1v \in \mathcal{G}_t]$, and model confidence $\|P_t\|^2$, measured on OPT-1.3B with the Green-red watermark (γ=0.5, ; δ=1.0). Prefixes are drawn from the Harry Potter Wikipedia article. Corresponding results for Gumbel and Tournament watermarks are provided in Fig. 7 (Appendix).

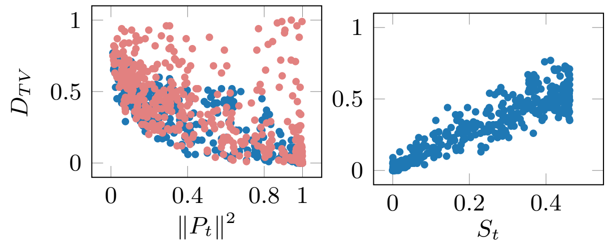

Figure 2: **Left:** Token-level distributional shift $D_{TV}(P_t, \widetilde{P}_t)$ vs. model confidence $\|P_t\|^2$ evaluated on OPT-1.3B. The blue plot measures the distributional shift due to the watermark distortion for Green–red watermark (with $\gamma = 0.5$, $\delta = 1.0$). The red plot measures the distributional shift between the reference model (OPT-125M) and the original model. **Right:** $D_{TV}(P_t, \widetilde{P}_t)$ vs. token-level detectability S_t for the Green-red watermark. Lower confidence leads to larger fidelity loss and stronger watermark detectability. Gumbel and Tournament show identical patterns (see Fig. 9).

which, in turn, underscores the watermark's muted effect per se.

3.3 Detectability–quality trade-off

Combining the findings of the previous two subsections, we now answer the question: are the tokens that are easiest to detect are also those that most distort the distribution?

The answer is yes. In Figure 2 (right), we see that tokens that boost detectability scores (S_t) are exactly those with the largest fidelity loss (D_{TV}). Hence token-level watermarks cannot achieve high detectability without sacrificing distributional fidelity—and, by extension, downstreaming text quality. This inherent trade-off motivates our smoothing attack to be introduced in Section 4: **by targeting at low-confidence positions, we can successfully remove the watermark signal while preserving the overall text quality**.

4 Smoothing Attack

Our attack aims to remove watermarks while preserving text quality. At each token position, we first estimate the model's confidence and then selectively smooth low-confidence tokens to weaken the watermark detection signal. Detailed justification and derivations are provided in Appendix C; here we present the core algorithm clearly.

Adversary's model access. We assume a practical scenario: the adversary queries the target watermarked model via an API, obtaining only the top-K token probabilities for a given prefix (with $K \ll |\mathcal{V}|$). The original, un-watermarked model is not available to the adversary.

Estimating model confidence. We estimate model confidence at position t by approximating the squared ℓ_2-norm of the token probability vector P_t:

$$\widehat{c}_t = \sum_{v\in\mathcal{V}_{\text{top}}} P_t(v)^2 + \frac{\left(1 - \sum_{v\in\mathcal{V}_{\text{top}}} P_t(v)\right)^2}{|\mathcal{V}| - K},$$

where $\mathcal{V}_{\text{top}}$ denotes the set of top-K tokens returned by the model. Here we assume uniform probabilities among unobserved tokens. - For distortion-free watermarks (Gumbel, Tournament), the observed probabilities $P_t(v)$ directly reflect the original distribution. - For distortionary watermarks (Green-red), the probabilities from the watermarked model slightly deviate from the original. However, as shown in Appendix C, this approximation still correctly ranks tokens by their confidence; hence, it remains effective without additional correction.

Normalizing confidence scores. To convert the confidence estimate $\widehat{c}_t$ into a normalized confidence score $c_t \in [0, 1]$, we first establish empirical bounds L (lower) and U (upper). Specifically, we query the watermarked model using N random prefixes (e.g., $N = 200$) and record their $\widehat{c}_t$ values.

Given these bounds, we set

$$c_t = \frac{\widehat{c}_t - L}{U - L}.$$

Smoothing procedure. Using a smoothing parameter $\alpha > 0$, our attack proceeds as follows at each token position. **1)** For *distortion-free watermarks*, with probability c_t^α, we retain the token originally produced by the watermarked model; otherwise, we resample from the observed top-K probabilities. **2)** For *distortionary watermarks*, we query a small reference model (e.g., smaller than the watermarked model) using the same prefix to obtain its top-K token probabilities to construct a mixture distribution:

$$c_t^\alpha \cdot P_{\text{wm}} + (1 - c_t^\alpha) \cdot P_{\text{ref}},$$

where P_{wm} and P_{ref} represent the watermarked and reference model distributions, respectively. We then sample from this mixture distribution. Intuitively, large α favors keeping tokens from the watermarked model, while small α smooths the watermark more aggressively by replacing tokens more frequently.

Finally, if the adversary is uncertain about whether the watermark is distortion-free or distortionary, they simply follow the procedure designed for distortionary watermarks (using the reference-model mixture) and the attack's success is not affected, as demonstrated in Section 5.

Efficiency and assumptions. Our attack is computationally efficient: it requires a modest initial overhead (a few hundred queries) to establish confidence-score normalization bounds (L, U), followed by one query per token position during generation. Importantly, our attack requires no knowledge of the watermark's algorithm, secret key, or detector internals, ensuring practical applicability.

5 Experiments

Setup. We evaluate our attack on three open-source model families: Llama3 (8B parameters) (Dubey et al., 2024), OPT (1.3B to 30B parameters) (Zhang et al., 2022), and Qwen2 (1.5B parameters) (Chu et al., 2024). When attacking distortionary watermarks, we use smaller models as references: Llama3-1B, OPT-125M, and Qwen2-0.5B, respectively.

Following prior work (Kirchenbauer et al., 2023a; Pan et al., 2024), we conduct evaluations on the C4 dataset (Raffel et al., 2020), where watermark performance has been shown effective. We avoid datasets with inherently low entropy (e.g., code generation), since previous studies (Kirchenbauer et al., 2023b; Lee et al., 2023) have demonstrated that watermark effectiveness significantly reduces in such settings. For each text in the dataset, the first 30 tokens form the prompt, and models generate the subsequent 200 tokens. All reported results are averaged over 100 prompts. Experiments were conducted using RTX-Titan GPUs.

We evaluate our attack against 10 representative watermarking algorithms covering both distortionary and distortion-free methods: KGW (Kirchenbauer et al., 2023a), Unigram (Zhao et al., 2023), UPV (Liu et al., 2023), X-SIR (He et al., 2024), DIP (Wu et al.), SWEET (Lee et al., 2023), EWD (Lu et al., 2024), Unbiased (Hu et al.), SynthID (Tournament sampling) (Dathathri et al., 2024), and Gumbel sampling (Aaronson, 2023). Our implementations build on the MarkLLM toolkit (Pan et al., 2024). Note that Gumbel and X-SIR evaluations are restricted to OPT models, since Gumbel sampling exceeds 100 GB GPU memory requirements for Llama/Qwen due to their large vocabulary sizes, and X-SIR's official implementation currently supports only OPT models.

We compare our Smoothing Attack with the state-of-the-art *Paraphrasing Attack* (Piet et al., 2023), which employs GPT-3.5-turbo to rewrite texts, as well as its enhanced variants: paraphrasing multiple times and using GPT-4o as a stronger paraphraser.

Performance metric. We measure attack effectiveness along two dimensions: *watermark removal* and *text quality preservation*. For watermark removal, we report the true positive rate (TPR) of watermark detection, under a fixed false positive rate (FPR) of less than 1%. A lower TPR indicates better watermark removal (TPR is 1% for un-watermarked texts and 100% for fully watermarked texts without any attack). To evaluate text quality, we follow established protocols (Kirchenbauer et al., 2023a; Pan et al., 2024; Kirchenbauer et al., 2023b) by reporting perplexity (lower is better) and diversity (higher is better).

Unless otherwise noted, we set the smoothing parameter $\alpha = 1.0$ and use the top-10 tokens from both the watermarked and reference models. Additional experimental details and parameter variations are provided in the appendix.

Table 1: Performance of watermark removal attacks on OPT-1.3B, Llama3-8B, and Qwen-1.5B models. We report the watermark true positive rate (TPR, lower is better), perplexity (PPL, lower is better), and diversity (Div, higher is better). All TPR values are measured at a fixed false positive rate below 1%. Additional results for models from 1.3B to 30B parameters are presented in Appendix B.6, demonstrating consistent trends.

Watermark	Attack	OPT-1.3B			Llama3-8B			Qwen2-1.5B		
		TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div
Un-watermarked	-	1	11.39	8.22	1	3.47	6.82	1	12.26	8.10
Reference	-	1	19.57	7.69	1	4.40	6.52	1	16.02	8.06
KGW (Kirchenbauer et al., 2023a)	None	100	14.61	8.07	99	4.60	6.92	100	16.46	8.11
	Paraphrasing	3	14.82	9.56	2	5.35	8.0	2	10.45	9.42
	Smoothing	0	9.57	6.72	2	3.20	5.63	0	8.02	6.91
Unigram (Zhao et al., 2023)	None	100	14.99	7.29	99	4.61	6.56	100	15.41	7.37
	Paraphrasing	53	14.51	8.75	54	5.60	8.02	5	10.40	8.56
	Smoothing	5	9.44	6.73	24	3.10	5.44	1	7.77	6.71
SynthID (Dathathri et al., 2024)	None	100	7.12	7.41	99	4.83	7.31	100	6.94	7.05
	Paraphrasing	1	10.57	9.11	1	5.62	8.18	1	6.90	8.43
	Smoothing	0	10.40	8.64	0	3.40	6.86	0	10.21	8.04
DIP (Wu et al.)	None	100	13.73	8.44	84	4.03	7.35	100	14.34	8.27
	Paraphrasing	0	13.95	9.25	0	5.25	8.34	2	10.10	8.85
	Smoothing	6	9.34	6.84	6	3.17	5.67	11	7.62	6.92
Unbiased (Hu et al.)	None	100	13.61	8.29	84	4.02	7.29	100	14.64	8.21
	Paraphrasing	3	14.45	10.39	2	5.36	8.57	1	9.97	8.82
	Smoothing	27	9.19	6.84	5	3.17	5.75	5	7.68	6.94
UPV (Liu et al., 2023)	None	99	11.65	8.22	83	4.38	6.80	86	11.93	7.49
	Paraphrasing	34	13.73	9.92	2	5.43	8.00	2	9.03	8.58
	Smoothing	20	10.01	6.89	1	3.12	5.49	0	8.16	6.91
EWD (Lu et al., 2024)	None	100	15.23	7.92	100	4.56	6.71	100	16.31	7.85
	Paraphrasing	0	14.95	9.95	7	5.73	7.83	1	10.18	9.28
	Smoothing	0	9.93	6.78	3	3.13	5.38	0	7.82	6.85
SWEET (Lee et al., 2023)	None	100	14.36	8.02	99	4.53	6.69	100	15.89	7.65
	Paraphrasing	0	14.57	9.45	14	5.64	8.05	4	10.18	9.30
	Smoothing	0	9.59	6.72	4	3.09	5.40	0	7.85	6.92

Performance in watermark removal. Our main results are summarized in Tables 1 and 2. The key finding is that the *Smoothing Attack* effectively removes watermarks across diverse models and watermarking algorithms, consistently outperforming the strong paraphrasing attacks. Specifically, our attack achieves very low watermark detection rates (TPR around 5%, occasionally reaching 0%), whereas paraphrasing attacks perform notably worse. For instance, on OPT-1.3B with the Unigram watermark (Table 1), paraphrasing leaves 53% of texts detectable, whereas our smoothing reduces detection to just 5%, despite using only a much smaller OPT-125M reference model.

Moreover, our attack is computationally inexpensive and practical: watermarking text with KGW on OPT-1.3B requires about 4.2 seconds, while our smoothing attack using OPT-125M takes just 6.5 seconds (on two TITAN RTX GPUs). This brings us an important message: ***existing watermarking schemes are vulnerable to even resource-limited adversaries, underscoring the significant real-world applicability of the smoothing attack and fundamental limitations of existing watermark defenses.***

Table 2: Performance of watermark removal attacks on OPT-1.3B with Gumbel (Aaronson, 2023) and X-SIR (He et al., 2024) watermarks (with FPR $< 1\%$).

Watermark	Attack	TPR (%)	PPL	Div
Gumbel	None	98	2.96	4.35
	Paraphrasing	13	14.21	11.13
	Smoothing	9	19.25	8.30
X-SIR	None	94	13.99	7.96
	Paraphrasing	34	14.13	8.80
	Smoothing	9	9.47	6.75

Performance in text quality. Our smoothing attack effectively preserves text quality, maintaining ***low perplexity (PPL) and competitive diversity (Div)*** while significantly improving watermark removal. For instance, on OPT-1.3B with the Unigram watermark (Table 1), our attack achieves notably better perplexity (9.44 vs. 14.51) and comparable diversity (6.73 vs. 8.75) relative to the paraphrasing attack, while dramatically reducing watermark detection (TPR of 5% vs. 50%). Further details on these quality metrics are provided in Appendix B.3 (Figures 4 and 5).

Table 3: Effectiveness of watermark removal attacks on OPT-1.3B under Unigram and UPV schemes. Translation can reduce TPR but significantly degrades quality. Repeated paraphrasing using GPT-3.5 lowers TPR further, with higher cost. GPT-4o improves fluency (lower PPL) but still leaves detectable watermark traces. Smoothing achieves the best overall trade-off by reducing TPR while preserving fluency and diversity.

Watermark	Attack	TPR(%)	PPL	Div
Unigram	None	100	15.0	7.3
	Translate (ZH)	100	22.2	7.6
	Translate (FR)	0	23.0	7.3
	GPT-3.5 Paraphrase (1x)	53	14.5	8.8
	GPT-3.5 Paraphrase (2x)	0	15.7	9.9
	GPT-3.5 Paraphrase (3x)	0	15.6	9.9
	GPT-4o Paraphrase (1x)	0	11.2	8.4
	Smoothing	5	9.4	6.7
UPV	None	99	11.7	8.2
	Translate (ZH)	58	17.0	8.0
	Translate (FR)	83	17.2	7.7
	GPT-3.5 Paraphrase (1x)	34	13.7	9.9
	GPT-3.5 Paraphrase (2x)	23	15.1	9.9
	GPT-3.5 Paraphrase (3x)	24	14.7	10.1
	GPT-4o Paraphrase (1x)	51	9.8	8.7
	Smoothing	20	10.0	6.9

The effectiveness of our smoothing method is also evident for Gumbel sampling (Table 2). Although Gumbel sampling itself yields artificially low perplexity by generating repetitive content, it substantially reduces text diversity and overall quality. Our smoothing attack, by comparison, slightly increases perplexity but significantly reduces undesirable repetition, thereby improving actual text readability and coherence (examples in Appendix B.5, Table 14).

While our attack may sometimes show slightly lower diversity compared to paraphrasing, this primarily arises because our method samples only from the top-K most likely tokens instead of the entire vocabulary. Increasing the value of K (see Table 5) effectively restores diversity, though it requires additional probability information from the model. Interestingly, by restricting token selection to the top-K tokens, our smoothing attack consistently achieves *even lower perplexity than unwatermarked texts*. Thus, beyond effectively removing watermarks, our method also enhances overall text quality by preventing the selection of extremely unlikely tokens commonly encountered in standard sampling.

Comparison with translation and repeated paraphrasing attacks. We further contextualize our smoothing attack by evaluating two additional attack types—translation-based and repeated paraphrasing attacks—particularly targeting watermark schemes (Unigram and UPV) resilient to single-pass GPT-3.5 paraphrasing.

For translation-based attacks, texts are translated to an intermediate language (Chinese or French) and then back to English using Google Translate (Google, 2024). For repeated paraphrasing, we apply GPT-3.5 up to three times iteratively, simulating more aggressive rewriting. Additionally, we explore GPT-4o, a stronger paraphraser with enhanced fluency and coherence.

Table 3 demonstrates that translation-based attacks often significantly degrade text quality (higher perplexity) while only inconsistently removing watermarks. Repeated paraphrasing improves watermark removal but substantially increases computational cost. Paraphrasing with GPT-4o—a stronger model than GPT-3.5 turbo—enhances text quality (lower perplexity) but does not guarantee better watermark removal; detection rates remain inconsistent and can even worsen, as observed with the UPV watermark.

Overall, these baselines emphasize the core advantage of our smoothing attack: it achieves robust watermark removal while preserving or even enhancing text quality, *without relying on costly LMs*.

Comparison with adaptive paraphrasers. We compare our smoothing attack with the adaptive paraphraser proposed by Diaa et al. (2024), which relies on white-box knowledge of watermark algorithms, including the secret key generation and embedding mechanisms. In contrast, our smoothing attack operates purely under black-box assumptions, requiring no detailed watermark knowledge. Despite this weaker assumption, Table 4 demonstrates our smoothing attack achieves comparable or superior watermark removal performance, while significantly preserving text quality (lower PPL). Thus, our method offers practical effectiveness without demanding unrealistic adversarial knowledge.

Ablation studies on K, α, and model size. We extensively study the sensitivity of our smoothing attack to critical parameters (K and α) and the target model's size.

Increasing K typically improves watermark removal (lower TPR) and diversity, at a slight cost of increased perplexity (Table 5). Notably, even with a minimal setting ($K = 1$), our attack remains effective, achieving a TPR of only 18%, significantly

Table 4: Comparison with the adaptive paraphraser from Diaa et al. (2024) on Llama3-8B with the Unigram watermark. The adaptive paraphraser is fine-tuned on Llama3-3B. Our attack use the same base, non-fine-tuned model as our reference model.

Attack	TPR (%)	PPL	Div
None	67	4.6	6.6
Adaptive Paraphraser	0	6.5	7.4
Smoothing	0	3.1	6.6

Table 5: Impact of K and α on *Smoothing Attack* performance on OPT-1.3B with Unigram watermark.

K	α	TPR (%)	PPL	Div
Fixed to 10	0.5	42	9.9	6.86
	1.0	5	9.44	6.73
	2.0	0	9.38	6.58
	3.0	1	9.25	6.43
1	Fixed to 1	18	3.21	4.62
5		10	7.46	6.11
10		5	9.44	6.73
15		5	11.73	7.11

lower than GPT-3.5 paraphrasing (53%). Increasing the smoothing parameter α makes our attack more aggressive in replacing uncertain tokens, thus enhancing watermark removal and generally improving perplexity. However, higher α values can slightly reduce diversity. By adjusting α, adversaries can effectively balance between watermark removal and text diversity (see Table 5). Additional results across diverse watermarks and models are presented in Appendix B.4.

Finally, we evaluate the effect of varying the target model size within the OPT family (from 1.3B to 30B), using the much smaller OPT-125M as the reference model. Our results (Table 5) indicate minimal sensitivity to the target model's size, confirming the scalability of our attack. Comprehensive results are detailed in Appendix B.6.

Effect of reference model size. To evaluate the influence of reference model size, we apply our smoothing attack using OPT models ranging from 125M to 1.3B parameters as the reference model. Table 7 clearly indicates that using larger reference models improves text quality (lower PPL, higher diversity), and can further reduce watermark detectability. These results confirm that the expressiveness of the reference model positively impacts overall attack performance.

Impact of watermark type knowledge. Our smoothing attack requires knowledge of the watermark type—specifically, whether it is distortion-free or distortionary—only for deciding whether and how to re-sample tokens. To assess the importance of this assumption, we evaluate our attack on the distortion-free SynthID watermark both with and without access to this information (Table 8). The results show that our attack achieves identical performance (TPR of 0%) in both settings, demonstrating its robustness and practical applicability even without watermark-type knowledge.

Table 6: *Smoothing Attack* against Unigram watermarking on models of *different sizes*, with OPT-125M as the reference model.

Target model size	Setting	TPR (%)	PPL	Div
1.3B	Unwatermarked	0	12.95	8.67
	Watermarked	99	16.53	7.29
	Smoothing	6	10.37	6.83
2.7B	Unwatermarked	0	11.75	8.36
	Watermarked	100	14.31	7.41
	Smoothing	4	10.35	6.66
6.7B	Unwatermarked	0	10.20	8.45
	Watermarked	100	12.94	7.48
	Smoothing	6	10.54	6.68
13B	Unwatermarked	0	10.14	8.39
	Watermarked	100	12.44	7.39
	Smoothing	5	10.32	6.70
30B	Unwatermarked	0	8.46	8.44
	Watermarked	100	10.45	7.56
	Smoothing	7	10.15	6.75

Table 7: Impact of size of reference model size on the performance of the smoothing attack. Larger models reduce perplexity; lower TPR; and maintain diversity.

Watermark	Attack / Model Size	TPR (%)	PPL	Div
KGW	None (Watermarked)	100	14.6	8.1
	Smoothing (OPT-125M)	0	9.6	6.7
	Smoothing (OPT-350M)	0	8.5	7.0
	Smoothing (OPT-1.3B)	0	7.0	7.3
DIP	None (Watermarked)	100	13.7	8.4
	Smoothing (OPT-125M)	6	9.3	6.8
	Smoothing (OPT-350M)	9	8.2	7.0
	Smoothing (OPT-1.3B)	6	6.9	7.2
UPV	None (Watermarked)	99	11.7	8.2
	Smoothing (OPT-125M)	20	10.0	6.9
	Smoothing (OPT-350M)	5	8.9	7.2
	Smoothing (OPT-1.3B)	4	7.4	7.3

Table 8: Impact of watermark-type knowledge on smoothing attack against the distortion-free SynthID watermark. Smoothing attack reduces true positive rate to zero whether or not it knows the watermark type.

Setting	TPR (%)	PPL	Div
Watermarked (no attack)	100	7.1	7.4
Smoothing (type known)	0	10.4	8.6
Smoothing (type unknown)	0	9.6	6.7

6 Conclusion

We revealed limitations in existing watermarks for language models and examined their robustness against watermark removal attacks. We introduced *Smoothing Attack*, a novel method that leverages model confidence to selectively remove watermark traces while preserving text quality. Comprehensive evaluations demonstrated that *Smoothing Attack* can completely remove watermarks, outperforming the state-of-the-art attack and highlighting a critical gap in current watermarks, and calling for more robust solutions.

7 Acknowledgments

We would like to thank the anonymous reviewers.

8 Limitations and Ethical Considerations

In conducting this study, we have carefully considered several factors that could influence the generalizability and applicability of our findings. Here we outline these considerations explicitly, along with the steps taken to address potential limitations.

Access to confidence-related information. Our smoothing attack relies on estimating model confidence from top-K token probabilities, commonly provided by public APIs (e.g., OpenAI's API). Although limiting access to probability information could theoretically mitigate our attack, in practice, such restrictions are challenging to enforce and may conflict with broader ethical goals around AI transparency and interpretability (OECD, 2019; National Institute of Standards and Technology (NIST), 2023). Recognizing this tension, we highlight the need for watermarking methods robust to scenarios where confidence estimates remain partially accessible.

Dependence on reference models. For distortionary watermark removal, our attack uses a reference model to generate alternative token candidates. We explicitly evaluated different reference model sizes (Table 7), confirming strong performance even when using significantly smaller reference models. However, selecting an extremely mismatched or lower-quality reference model could impact both watermark removal and text quality. We recommend using reference models carefully matched to the domain or distribution of the target model.

Dataset selection and evaluation scope. We evaluated our methods extensively on the C4 dataset due to its well-documented suitability for watermark evaluation (Kirchenbauer et al., 2023a; Pan et al., 2024). Prior research indicated that datasets with low-entropy generation (e.g., code) already significantly reduce watermark effectiveness (Kirchenbauer et al., 2023b; Lee et al., 2023). Thus, while our findings clearly establish the effectiveness of our attack under typical high-entropy generation conditions, results may differ in specialized, low-entropy contexts.

Future watermarking approaches. Our attack exploits inherent structural vulnerabilities shared by current token-level watermark schemes. We explicitly acknowledge that future watermarking algorithms could be designed specifically to counter such confidence-based attacks. Recognizing this potential evolution, we strongly encourage further research into watermark robustness and the development of methods resilient to confidence-based adversaries.

9 Ethical Considerations

Our work demonstrates that an adversary, under realistic assumptions, can successfully remove watermarks from texts without compromising text quality. Although robustness concerns regarding watermarking have been highlighted by prior studies, our research underscores that these risks may be even greater than previously assessed.

We conducted experiments on LLama (Dubey et al., 2024), OPT (Zhang et al., 2022), and Qwen (Chu et al., 2024), each of which has been released under their respective licenses, as detailed in their documentation. Our implementation is based on MarkLLM (under the Apache License), and all modifications we have introduced are clearly documented in the README file included with our submitted code. Furthermore, any external packages used in our evaluations have been explicitly presented in the code we submitted.

Artifact use in this research has been consistent with intended purposes. Our dataset derives from the publicly available C4 dataset, which, to the best of our knowledge, does not contain personally identifiable information or offensive content. Additionally, relevant statistics regarding the data utilized in our experiments have been comprehensively reported C4 dataset (Raffel et al., 2020).

We utilized ChatGPT to revise portions of the manuscript; however, all revisions were performed under direct human supervision, ensuring that the final text accurately reflects our intent and ethical standards.

We approach this research with a firm commitment to ethical standards and responsible disclosure. By openly illustrating vulnerabilities, recommending effective mitigation, and transparently sharing our methods and outcomes, our objective is to inform and assist the broader research community. Our goal is to facilitate advancements in watermarking techniques that effectively balance transparency, innovation, and security, aligning with emerging regulatory standards such as the EU AI Act and the U.S. AI safety policies (European Commission, 2021). Moreover, we explicitly discuss potential defensive strategies and evaluate their efficacy (see Appendix D and Table10), providing actionable guidance for enhancing the resilience of watermarking techniques. Overall, by clearly outlining these ethical considerations and limitations, we believe our research contributes robust and actionable insights, responsibly addressing the ethical implications and boundaries inherent in our study.

References

Scott Aaronson. 2023. Simons institute talk on watermarking of large language models. https://simons.berkeley.edu/talks/scott-aaronson-ut-austin-openai-2023-08-17.

Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. 2022. Bad characters: Imperceptible nlp attacks. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1987–2004. IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Deepak Subbiah, Jack Kaplan, Prafulla Dhariwal, A. Neelakantan, Long Ouyang, and Dario Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*. Seminal paper introducing GPT-3, discussing the importance of probabilities in understanding model behavior.

Miranda Christ and Sam Gunn. 2024. Pseudorandom error-correcting codes. In *Annual International Cryptology Conference*, pages 325–347. Springer.

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

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, and 1 others. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Aloni Cohen, Alexander Hoover, and Gabe Schoenbach. 2024. Watermarking language models for many adaptive users. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 84–84. IEEE Computer Society.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, and 1 others. 2024. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823.

Abdulrahman Diaa, Toluwani Aremu, and Nils Lukas. 2024. Optimizing adaptive attacks against content watermarks for language models. *arXiv preprint arXiv:2410.02440*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

European Commission. 2021. The EU Artificial Intelligence Act. Available at: https://artificialintelligenceact.eu/. Proposed regulation focusing on transparency and accountability in high-risk AI systems.

Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. 2023. Publicly detectable watermarking for language models. *Cryptology ePrint Archive*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Jiayi Fu, Xuandong Zhao, Ruihan Yang, Yuansen Zhang, Jiangjie Chen, and Yanghua Xiao. 2024. Gumbelsoft: Diversified language model watermarking via the gumbelmax-trick. *arXiv preprint arXiv:2402.12948*.

Surendra Ghentiyala and Venkatesan Guruswami. 2024. New constructions of pseudorandom codes. *Cryptology ePrint Archive*.

Noah Golowich and Ankur Moitra. Edit distance robust watermarks via indexing pseudorandom codes. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Riley Goodside. 2023. There are adversarial attacks for that proposal as well — in particular, generating with emojis after words and then removing them before submitting defeats it. Twitter. URL: https://twitter.com/goodside/status/1610682909647671306.

Google. 2024. Google Translate. https://translate.google.com. Accessed: 2024-05-01.

Zhiwei He, Binglin Zhou, Hongkun Hao, Aiwei Liu, Xing Wang, Zhaopeng Tu, Zhuosheng Zhang, and Rui Wang. 2024. Can watermarks survive translation? on the cross-lingual consistency of text watermark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4115–4129, Bangkok, Thailand. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649.

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

Baihe Huang, Banghua Zhu, Hanlin Zhu, Jason D Lee, Jiantao Jiao, and Michael I Jordan. 2023. Towards optimal statistical watermarking. *arXiv preprint arXiv:2312.07930*.

Mingjia Huo, Sai Ashish Somayajula, Youwei Liang, Ruisi Zhang, Farinaz Koushanfar, and Pengtao Xie. Token-specific watermarking with enhanced detectability and semantic coherence for large language models. In *Forty-first International Conference on Machine Learning*.

Nikola Jovanović, Robin Staab, and Martin Vechev. 2024. Watermark stealing in large language models. *arXiv preprint arXiv:2402.19361*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

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

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *arXiv preprint arXiv:2303.13408*.

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

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2023. Who wrote this code? watermarking for code generation. *arXiv preprint arXiv:2305.15060*.

Xiang Li, Feng Ruan, Huiyuan Wang, Qi Long, and Weijie J Su. 2024. A statistical framework of watermarks for large language models: Pivot, detection efficiency and optimal rules. *arXiv preprint arXiv:2404.01245*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

Aiwei Liu, Leyi Pan, Xuming Hu, Shuang Li, Lijie Wen, Irwin King, and S Yu Philip. 2023. An unforgeable publicly verifiable watermark for large language models. In *The Twelfth International Conference on Learning Representations*.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2024. A semantic invariant robust watermark for large language models. In *The Twelfth International Conference on Learning Representations*.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. 2024. An entropy-based text watermarking detection method. *arXiv preprint arXiv:2403.13485*.

National Institute of Standards and Technology (NIST). 2023. Ai risk management framework. Technical report, U.S. Department of Commerce. Framework to improve AI system trustworthiness and manage risks, emphasizing transparency.

OECD. 2019. Oecd ai principles. Available at: `https://oecd.ai/en/dashboards/ai-principles/`. Guidelines for ethical, trustworthy AI. Transparency and accountability are key principles.

OpenAI. 2023. Openai api documentation. Available at: `https://platform.openai.com/docs/`. Developer documentation highlighting the use of top-K sampling, beam search, and probability outputs.

R OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5).

Luca Pajola and Mauro Conti. 2021. Fall of giants: How popular text-based mlaas fall against a simple evasion attack. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 198–211. IEEE.

Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, and 1 others. 2024. Markllm: An open-source toolkit for llm watermarking. *arXiv preprint arXiv:2405.10051*.

Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. 2023. Mark my words: Analyzing and evaluating language model watermarks. *arXiv preprint arXiv:2312.00273*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2022. Paraphrastic representations at scale. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 379–388.

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

Hanlin Zhang, Benjamin L Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. 2024. Watermarks in the sand: Impossibility of strong watermarking for language models. In *Forty-first International Conference on Machine Learning*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

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

Xuandong Zhao, Sam Gunn, Miranda Christ, Jaiden Fairoze, Andres Fabrega, Nicholas Carlini, Sanjam Garg, Sanghyun Hong, Milad Nasr, Florian Tramer, and 1 others. 2024a. Sok: Watermarking for ai-generated content. *arXiv preprint arXiv:2411.18479*.

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. 2024b. Permute-and-flip: An optimally robust and watermarkable decoder for llms. *arXiv preprint arXiv:2402.05864*.

Tong Zhou, Xuandong Zhao, Xiaolin Xu, and Shaolei Ren. 2024. Bileve: Securing text provenance in large language models against spoofing with bi-level signature. *arXiv preprint arXiv:2406.01946*.

A More on Related Work

Variations of Green-red list watermark. Different variations of Green-red list watermark, e.g., see (Kirchenbauer et al., 2023b; Lee et al., 2023; Liu et al., 2023; Wu et al.; Huo et al.; Zhou et al., 2024; Lu et al., 2024; Liu et al., 2024; He et al., 2024; Zhao et al., 2023; Kirchenbauer et al., 2023a), mainly differ in the assignment of the green lists and the detection process. In particular, the assignment of $\mathcal{G}_t$ could depend on the prefix, e.g., the preceding h tokens in the generated text. When $h = 0$, we say the assignment is context-independent and is referred to as the *Unigram* watermark (Zhao et al., 2023); when $h = 1$, the assignment depends on the previous token and is referred to as the *KGW* watermark (Kirchenbauer et al., 2023a)

Scalable Tournament sampling. As shown in their paper, the original tournament process in (Dathathri et al., 2024) can be costly to implement, as there are $O(2^m)$ times of sampling and pair-wise comparison of tokens. Instead, they obtain a modified distribution for tokens. With $\widetilde{P}_t^{(0)} = P_t$, they iteratively compute $\widetilde{P}_t^{(l)}(v) = \Big(1 + g^{(l)}(v, r_t) - \sum_{v' \in \mathcal{V}} \big(g^{(l)}(v', r_t) \cdot P_t^{(l-1)}(v')\big)\Big) \cdot \widetilde{P}_t^{(l-1)}(v)$, for $l = 1, \ldots, m$, and then sample the token from $\widetilde{P}^{(m)}$.

Distortion-free watermark. There are also other distortion-free watermarks, which aim to preserve the original model's token distribution and avoid detectable shifts in probabilities of output tokens, e.g., see Hu et al.; Zhao et al. (2024b); Fu et al. (2024); Christ et al. (2023); Fairoze et al. (2023); Christ and Gunn (2024); Cohen et al. (2024); Ghentiyala and Guruswami (2024); Golowich and Moitra; Dathathri et al. (2024); Wu et al..

Comparison with paraphrasing attacks. When attacking OPT models (from 1.3B to 30B parameters), our attack only leverages the OPT-125M as the reference model when attacking distortionary watermarks such as the Unigram watermark. When attacking distortion-free watermarks, our attack sometimes resamples from the target watermarked model. In either case, the resource used in our attack is significantly smaller than the state-of-the-art paraphrasing attack, which uses the much larger

Table 9: Table of notation definitions and their locations.

Notation	Meaning	Definition Location
M	Auto-regressive language model (LM), which generates text sequentially based on a given prompt.	Section 2
$\widetilde{M}$	Watermarked model, a variant of M that embeds watermarks into generated text.	Section 3.1
$\mathcal{V}$	Vocabulary of the LM, the set of all possible tokens that can be generated.	Section 2
t	Token position in the generated sequence, indicating the index of a specific token.	Section 2
$l_t(v)$	Logit assigned by the model to token v at position t before applying softmax.	Eq. equation 1
$P_t(v)$	Probability of token v at position t after applying the softmax function.	Eq. equation 1
$\widetilde{P}_t(v)$	Modified probability distribution in the watermarked model after logit manipulation.	Eq. equation 2
$(v_1, \ldots, v_T)$	Sequence of tokens forming the output text from the language model.	Section 2
$d(v_1, \ldots, v_T)$	Detection score function used to determine whether a text is watermarked.	Section 2
τ	Threshold value for watermark detection; if $d(v_1, \ldots, v_T) > \tau$, the text is classified as watermarked.	Section 2
$\mathcal{G}_t$	Green list, a subset of vocabulary containing tokens whose logits are increased in green-red list watermarking.	Section 2
γ	Fraction of the vocabulary included in the green list $\mathcal{G}_t$, determining the probability of token selection.	Section 2
δ	Logit increase applied to tokens in the green list $\mathcal{G}_t$, influencing token selection probabilities.	Eq. equation 2
T	Length of the generated sequence, i.e., the total number of tokens in the output text.	Section 2
$u_t(v)$	Randomly sampled value from $[0, 1]$ for token v in Gumbel sampling watermarking.	Section 2
v_t^*	Token selected using Gumbel sampling watermarking by maximizing a transformed probability.	Section 2
S_t	Contribution of the token at position t to the overall watermark detection score.	Eq. equation 5
$\mathbb{E}_{v \sim P_t}[\mathbf{1}\{v \in \mathcal{G}_t\}]$	Expected probability mass assigned to green tokens at position t from probability distribution P_t.	Eq. equation 5
$\|P_t\|^2$	$\mathcal{L}_2$ norm of the probability vector, measuring model confidence at position t. A higher value means greater confidence.	Section 3.1
$D_{TV}(P_t, \widetilde{P}_t)$	Total variation distance between original and watermarked probability distributions, measuring distortion.	Section 3.1
$D_{TV}(P_t, P_t^{\text{ref}})$	Total variation distance between the original model and a reference model's probability distributions.	Section 3.1
K	Number of most probable tokens that the adversary has access to from the watermarked model.	Section 4
$\mathcal{V}_{\text{Top-K}}$	Set of top-K most probable tokens observed by the adversary.	Section 4
β	Scaling factor used to estimate $\|P_t\|^2$ from watermarked probabilities in Green-red list watermarking.	Section 4
c	Normalized confidence score in $[0, 1]$ based on estimated $\mathcal{L}_2$ norm.	Section 4
U, L	Upper and lower bounds for normalizing $\mathcal{L}_2$ norms into the confidence score c.	Section 4
α	Exponential factor controlling the aggressiveness of the smoothing attack. A larger α favors keeping watermarked tokens, while a smaller α favors replacement.	Section 4
P_t^{ref}	Token probability distribution from a much smaller, un-watermarked reference model.	Section 4

GPT-3.5-turbo. Despite using fewer resources, our approach achieves higher watermark removal rates and comparable text quality. This highlights that even resource-limited adversaries can thwart watermarks, underscoring the need for stronger watermark defenses.

B More on Experiments

B.1 Implementation

We evaluate the smoothing attack on eight different watermarking algorithms, including KGW (Kirchenbauer et al., 2023a), Unigram (Zhao et al., 2023), SWEET (Lee et al., 2023), UPV (Liu et al., 2023), EWD (Lu et al., 2024), X-SIR (He et al., 2024), DIP (Wu et al.), Unbiased (Hu et al.), SynthID (Dathathri et al., 2024) and Gumbel (Aaronson, 2023). We use the implementations and default configurations provided by MarkLLM (Pan et al., 2024). For completeness, we provide details of the algorithms below.

- KGW (Kirchenbauer et al., 2023a): The green set $\mathcal{G}_t$ at each position t is selected based on the previous h tokens and a secret key known to the service provider. The hyperparameters are set as follows: $\gamma = 0.5$, $\delta = 2.0$, and $h = 1$.
- Unigram (Kirchenbauer et al., 2023a): The green set $\mathcal{G}_t$ is fixed for each token t and each prefix, depending solely on the secret key known to the service provider. No dynamic updates are performed based on previous tokens. The parameters are: $\gamma = 0.5$, $\delta = 2.0$.
- SWEET (Lee et al., 2023): A shift is applied only when the entropy of the probability distribution at position t is high, improving text quality, particularly for code generation tasks. The parameters are set as: $\gamma = 0.5$, $\delta = 2.0$, the entropy threshold is 0.9, and $h = 1.0$.
- UPV (Liu et al., 2023): The green token selection process is similar to the previous approaches. However, this method requires training two additional models: a generator network to separate red and green tokens and a detector network for classification based on the input text. The watermarks are introduced using $\gamma = 0.5$, $\delta = 2.0$, and $h = 1.0$. The detector produces a binary prediction rather than a continuous score like a z-score.
- EWD (Lu et al., 2024): Watermark introduction follows a similar process as the previous methods. The hyperparameters are $\gamma = 0.5$, $\delta = 2.0$, and $h = 1.0$. During detection, tokens are assigned different weights based on their entropy, with higher entropy tokens receiving greater weight to improve detectability in low-entropy scenarios.
- X-SIR (He et al., 2024): Instead of operating at the token level, the red-green partition is applied at the level of semantic clusters, grouping similar words together and adding bias at the group level. This improves robustness against Cross-lingual Watermark Removal Attacks (CWRA).
- DIP (Wu et al.): Similar to Kirchenbauer et al. (2023), this method selects green tokens but uses a distribution-preserving reweight function to adjust token probabilities. This increases the probability of green tokens while maintaining the overall distribution. The reweighting is controlled by the parameter α. The hyperparameters are set as $\gamma = 0.5$, $h = 5$.

Implementation of the paraphrasing attack. We include the strongest baseline that paraphrases the given text based on the GPT-3.5-turbo (Piet et al., 2023), denoted as P-GPT3.5 using the prompt: "Please rewrite the following text:". As shown in (Kirchenbauer et al., 2023b), GPT-3.5-turbo is more powerful in removing the watermarks compared to Dipper model (Krishna et al., 2023).

Text quality metric. We use Llama3-8B, Qwen2-7B, and OPT-2.7B to evaluate the perplexity of the text generated from Llama3, Qwen2, and OPT models. We also report the log diversity of the text (Welleck et al.; Kirchenbauer et al., 2023b; Li et al., 2022), following the definition in (Kirchenbauer et al., 2023b) considering the 2-gram, 3-gram, and 4-gram repetition in the generated text. A higher diversity score represents a more diverse text.

B.2 Performance of the smoothing attack

Figure 3 shows three scatterplots of TPR vs. PPL for text generated under different watermarking and attack settings. Each point is colored by the watermarking method and corresponds to one of three models (OPT-1.3B, Llama3-8B, Qwen2-1.5B). Overall, the smoothing attack yields substantially lower TPR relative to the watermarked setting, demonstrating its performance at watermark removal. Notably, smoothing's TPR is on par with that of the paraphrasing attack, which uses a more powerful model (GPT-3.5-turbo). In terms of perplexity (PPL), smoothing also generates text that is competitive with (and sometimes lower than) both the watermarked text and the paraphrased text, indicating that it preserves text quality while removing the watermark.

B.3 Text Quality Evaluation

Figure 4 and Figure 5 present boxplots of the perplexity (PPL) and diversity of text generated from different sources using the OPT-1.3B model. We observe that the smoothing attack generally yields text with lower PPL than the watermarked model, except in cases involving the Gumbel watermark. This suggests that, according to the PPL metric, the smoothing attack can generate high-quality text. In terms of diversity, the constrained selection process—where sampling is restricted to the top-K candidates from both the reference and target models—results in lower diversity for the smoothing attack. These findings are consistent with the average PPL results reported in Table 1 in the main paper.

In addition, we compute the P-SP score (Wieting et al., 2022), which quantifies the similarity between pairs of texts in the embedding space, with higher scores indicating greater similarity. Specifically, we calculate P-SP scores for text generated from different sources and visualize the results in the heatmap shown in Figure 6. We observe that, aside from the paraphrasing case, texts from different sources generally exhibit low similarity. For instance, text generated by the watermarked model has a P-SP score of 53.6 on Unigram, whereas the similarity between the watermarked text and its paraphrased version reaches 82.3. Our smoothing attack produces a P-SP score (measuring similarity between text from the smoothing attack and unwatermarked text) comparable to that of the watermarked text (measuring similarity between watermarked text and unwatermarked text). The generally low P-SP scores across different sources reflect the natural variability in generated responses, as multiple reasonable outputs can exist for the same prompt. Therefore, P-SP metrics may not be a reliable measure for assessing text quality degradation due to watermarking or smoothing.

B.4 Effect of K and α

Table 10 and Table 12 show the performance of smoothing attacks against different watermarking algorithms under varying values of K. In a smoothing attack, the adversary has access only to the top-K tokens and their probabilities from both the reference and target models. Even with $K = 1$, the attack can drastically reduce the true positive rate (TPR) from 99% (without any attack) to an extremely low value, sometimes reaching 0%. This indicates that even with minimal access to both models, the smoothing attack can effectively remove watermarks. Furthermore, we observe that increasing K leads to more diverse text generation, as discussed in the main paper. This is because a higher K provides the attack with a larger selection of candidate tokens, allowing for greater variation in the generated text. This observation remains consistent across both the OPT-1.3B and Llama3-8B models.

Table 11 and Table 13 analyze the performance of smoothing attacks against different watermarking algorithms under varying values of α. In this attack, the weight assigned to the top-K tokens from the watermarked model is given by c^{α}, while the weight for the top-K tokens from the reference model is $1 - c^{\alpha}$, where c is a confidence score between 0 and 1. A larger α shifts the token selection preference toward the reference model, making the generated text more aligned with it. Conversely, a smaller α biases the attack toward the watermarked model, producing text that more closely resembles the watermarked output. As α increases, the true positive rate (TPR) decreases, leading to a higher watermark removal rate—an effect consistently observed across all watermarking methods for both the OPT-1.3B and Llama3-8B models.

In terms of text quality, when α is lower, the generated text is more influenced by the watermarked model, which generally exhibits higher quality than the reference model. Consequently, decreasing α can improve text quality. This provides an adversary with a way to adjust α to balance watermark removal and text quality preservation.

B.5 Text example

Table 14 presents text generated by the Gumbel sampling algorithm and the smoothing attack. We observe that, although the perplexity of the watermarked text is significantly lower than that of the text from the smoothing attack, this is primarily due to repetition in the generated text. This behavior may stem from the deterministic nature of Gumbel sampling, which can lead to less diverse outputs.

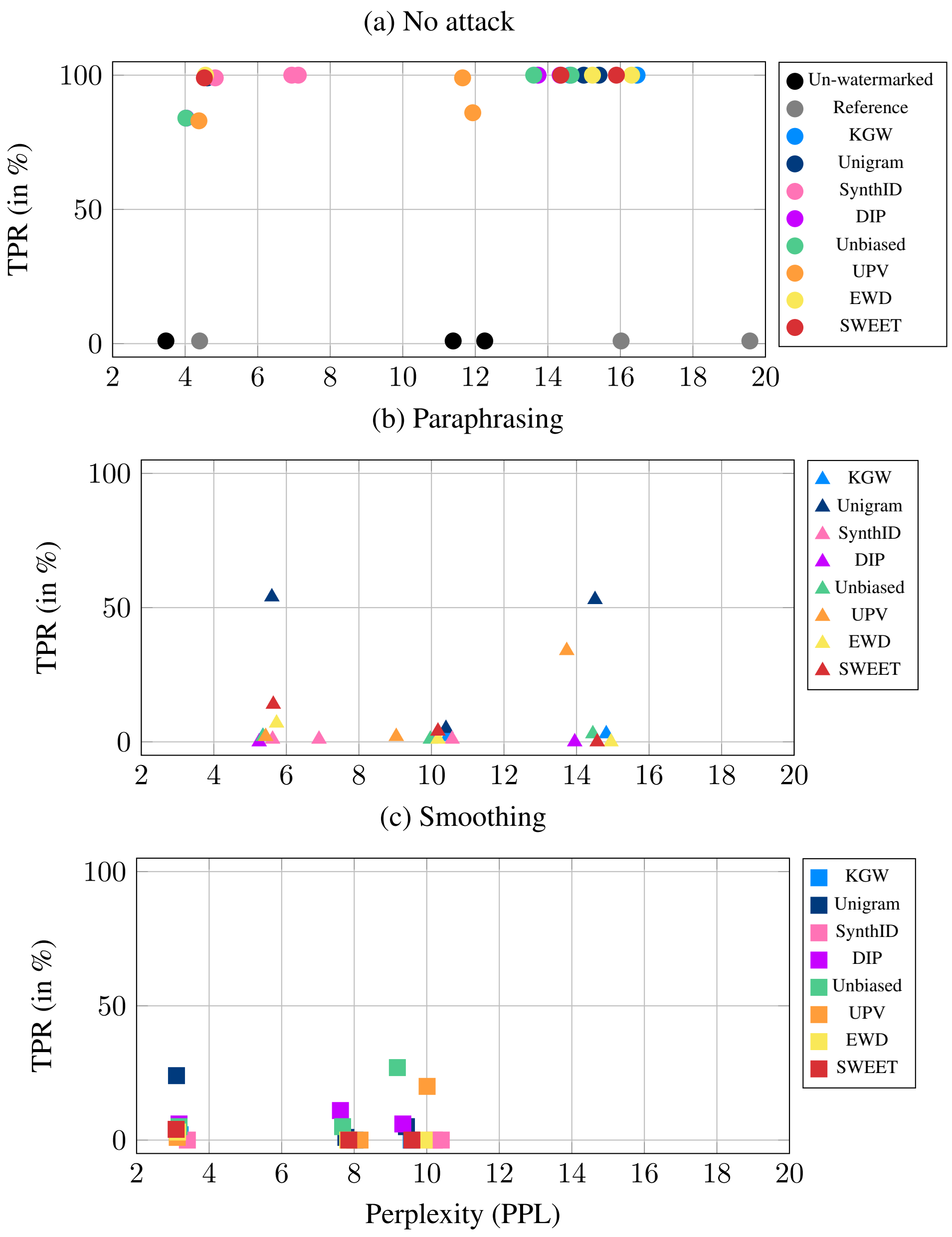

Figure 3: Each subfigure shows how the true positive rate (TPR) varies with perplexity (PPL) for a specific attack. No attack (a) corresponds to watermarked text without modifications, paraphrasing (b) uses GPT-3.5-turbo to rewrite the text, and smoothing (c) randomly replaces some tokens to remove the watermark. Colors indicate the particular watermarking method and each point corresponds to one of three models (OPT-1.3B, Llama3-8B, Qwen2-1.5B).

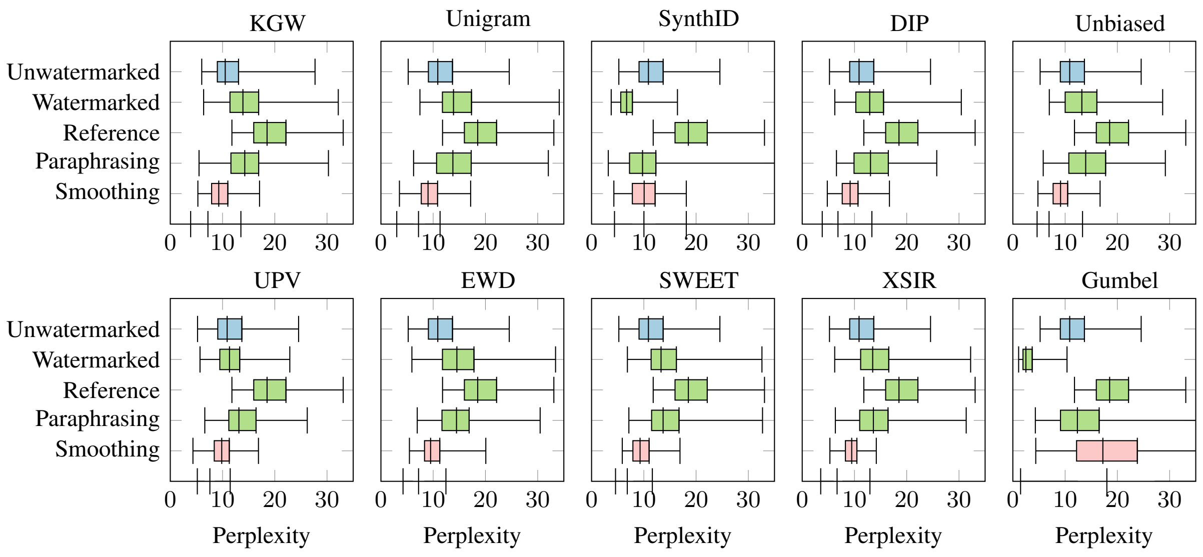

Figure 4: Text Quality Comparison – Perplexity (OPT-1.3B). Box plots of perplexity for text generated from different sources, with perplexity computed using the OPT-2.7B model. Our smoothing attack produces text with quality comparable to, and in some cases better than, that of the watermarked model.

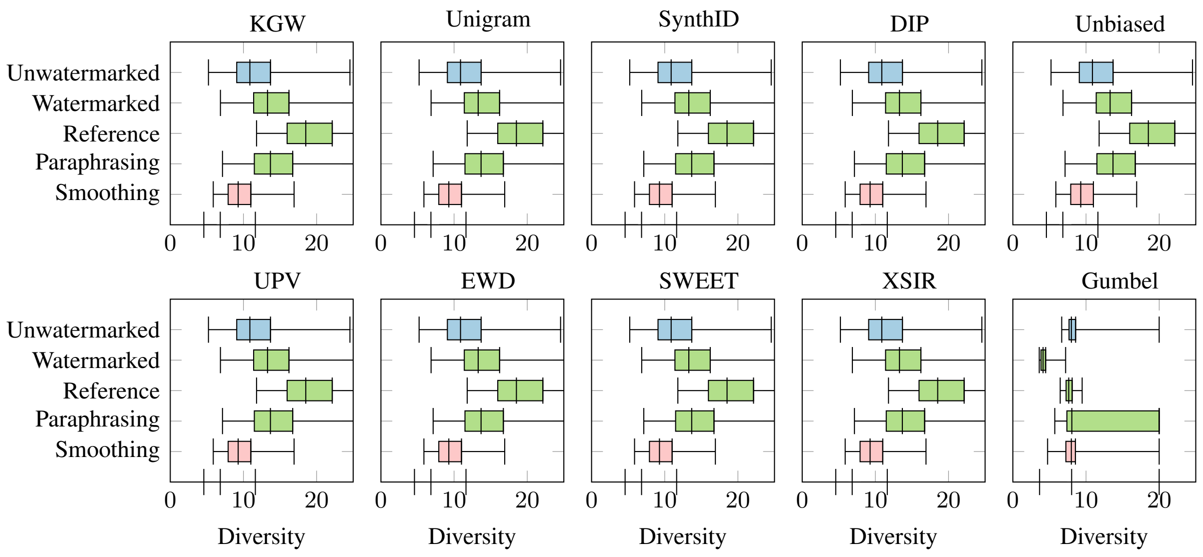

Figure 5: Text Quality Comparison – Diversity (OPT-1.3B). Box plots of text diversity for outputs generated from different sources. Our smoothing attack produces text with diversity comparable to, and in some cases lower than, that of the watermarked model due to its constrained selection process.

B.6 Impact of model size

Table 15 presents the performance of the smoothing attack across different watermarking algorithms and varying sizes of OPT models. Perplexity (PPL) is computed with respect to the OPT-30B model, while the reference model remains consistent across all settings—the OPT-125M.

For unwatermarked models, the True Positive Rate (TPR) is consistently 0%. In contrast, watermarked models achieve near-perfect TPR. However, the smoothing attack significantly reduces TPR across all model sizes, with its impact increasing as the model size grows—for instance, TPR drops to 0% for the KGW watermark in the 30B model.

Watermarked models exhibit a notable increase in perplexity, indicating that watermarking impacts text fluency. The smoothing attack reduces perplexity, bringing it closer to unwatermarked levels, suggesting a partial recovery of fluency. Regarding diversity, the unwatermarked text demonstrates the highest variation, while watermarking constrains generation patterns, resulting in a noticeable drop in diversity. The smoothing attack

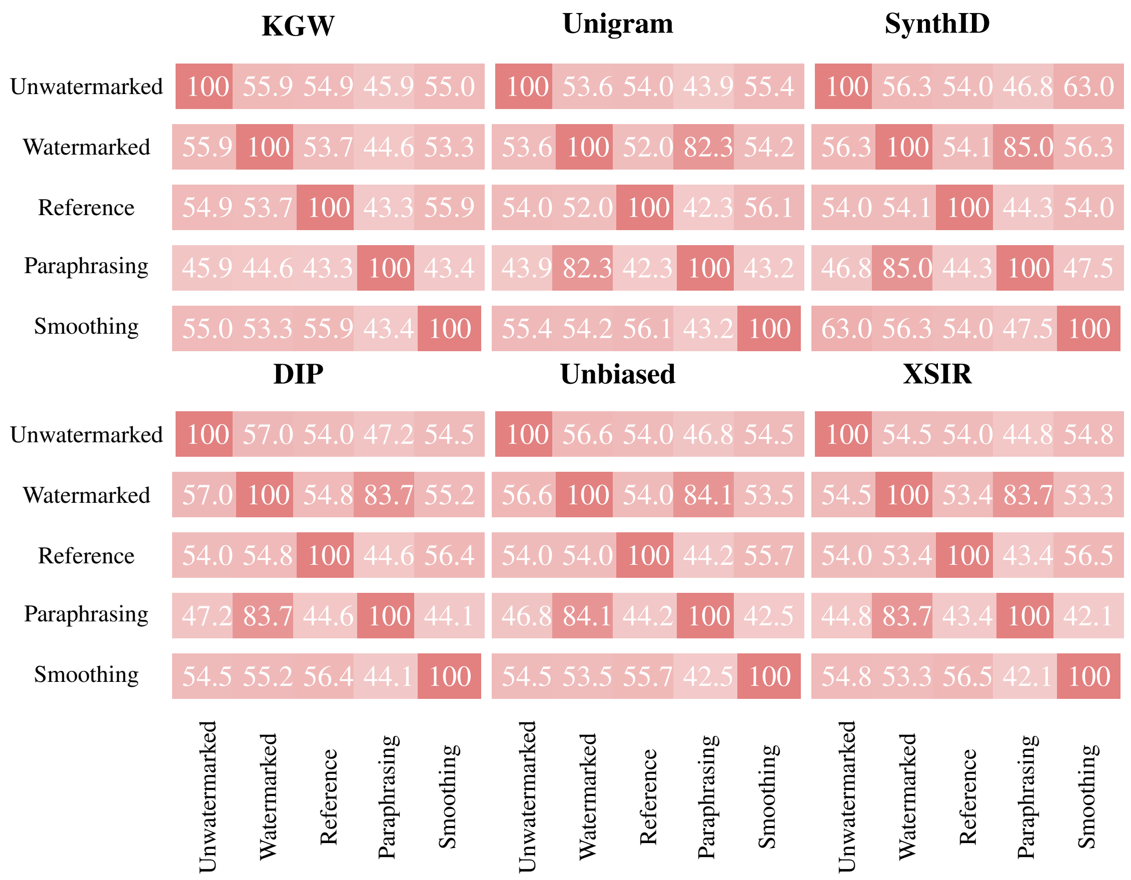

Figure 6: Text Quality Comparison – P-SP (OPT-1.3B). Heatmap comparing the similarity of text generated by different models in the sentence embedding space. Text from the watermarked model has a low similarity score compared to unwatermarked text, reflecting the inherent variability in generated responses. However, the paraphrased text (Paraphrasing vs. watermarked) exhibits a high similarity score, suggesting that the P-SP metric is more suitable for evaluating paraphrasing rather than assessing text quality degradation due to watermarking or smoothing.

Table 10: Effect of K on Smoothing Attack Performance (OPT-1.3B). Evaluation of the smoothing attack's effectiveness against different watermarking algorithms on the OPT-1.3B model, varying the number of top-K tokens accessible to the attacker.

K	KGW			Unigram			SynthID			DIP			Unbiased		
	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div
1	9%	3.22	4.54	18%	3.21	4.62	0%	10.45	8.47	1%	3.36	4.57	7%	3.36	4.56
3	0.0%	5.76	5.71	8.0%	5.9	5.68	0.0%	10.5	8.31	4.0%	5.58	5.66	14.0%	5.59	5.68
5	2.0%	7.27	6.17	10.0%	7.46	6.11	0.0%	10.35	8.71	3.0%	6.97	6.23	19.0%	7.11	6.29
7	1.0%	8.14	6.46	5.0%	8.48	6.55	0.0%	10.42	8.63	7.0%	7.97	6.46	29.0%	8.06	6.47
10	0.0%	9.57	6.72	5.0%	9.44	6.73	0.0%	10.4	8.64	6.0%	9.34	6.84	27.0%	9.19	6.84

K	XSIR			UPV			Gumbel			EWD			SWEET		
	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div
1	14%	3.31	4.5	22%	3.62	4.63	0%	20.8	8.2	1%	3.31	4.49	2%	3.41	4.57
3	17.0%	5.69	5.52	14.0%	6.22	5.87	2.0%	21.72	8.47	0.0%	5.78	5.71	0.0%	5.64	5.75
5	8.0%	6.8	6.04	16.0%	7.68	6.3	8.0%	20.3	8.23	0.0%	7.32	6.18	0.0%	7.15	6.23
7	10.0%	8.26	6.48	7.0%	8.75	6.65	9.0%	21.15	8.15	0.0%	8.65	6.52	0.0%	8.47	6.45
10	9.0%	9.47	6.75	20.0%	10.01	6.89	9.0%	19.25	8.3	0.0%	9.93	6.78	0.0%	9.59	6.72

further reduces diversity, primarily because tokens are sampled only from the top-K tokens of both the watermarked and reference models, limiting the range of possible candidates.

Table 11: Effect of α on Smoothing Attack Performance (OPT-1.3B). Evaluation of the smoothing attack's effectiveness against different watermarking algorithms on the OPT-1.3B model, varying the parameter α. A larger α indicates that the attack relies more on the reference model's output, while a smaller α means the attack is more influenced by the watermarked text.

α	KGW			Unigram			SynthID			DIP			Unbiased			X-SIR			UPV			Gumbel			EWD			SWEET		
	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div
0.5	11.0%	10.03	7.02	42.0%	9.9	6.86	2.0%	9.33	7.9	29.0%	9.27	7.11	63.0%	8.92	7.09	28.0%	9.47	6.94	42.0%	10.01	7.14	80.0%	13.73	7.54	0.0%	9.76	7.01	6.0%	9.66	7.13
1.0	0.0%	9.57	6.72	5.0%	9.44	6.73	0.0%	10.4	8.64	6.0%	9.34	6.84	27.0%	9.19	6.84	9.0%	9.47	6.75	20.0%	10.01	6.89	9.0%	19.25	8.3	0.0%	9.93	6.78	0.0%	9.59	6.72
2.0	0.0%	9.35	6.65	0.0%	9.38	6.58	0.0%	11.16	8.26	1.0%	9.03	6.71	9.0%	8.89	6.59	6.0%	9.45	6.46	4.0%	9.28	6.59	0.0%	25.39	9.04	0.0%	9.63	6.58	0.0%	9.29	6.45
3.0	0.0%	9.45	6.46	1.0%	9.25	6.43	0.0%	11.33	8.61	0.0%	9.32	6.82	1.0%	9.05	6.65	0.0%	9.12	6.41	1.0%	9.85	6.57	0.0%	25.77	9.5	0.0%	9.43	6.68	0.0%	9.33	6.53

Table 12: Effect of K on Smoothing Attack Performance (Llama3-8B). Evaluation of the smoothing attack's effectiveness against different watermarking algorithms on the Llama3-8B model, varying the number of top-K tokens accessible to the attacker.

K	KGW			Unigram			SynthID			DIP			Unbiased			UPV			EWD			SWEET		
	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div
1	6%	2.37	4.67	19%	2.41	4.67	0%	3.6	6.86	2%	2.53	4.84	1%	2.5	4.8	1%	2.48	4.76	3%	2.43	4.68	3%	2.41	4.72
3	1%	2.81	5.17	27%	2.8	5.2	0%	3.42	6.87	4%	2.91	5.47	4%	2.9	5.44	1%	2.97	5.37	4%	2.94	5.33	3%	2.91	5.27
5	3%	2.99	5.36	24%	2.92	5.31	0%	3.41	6.89	1%	2.97	5.55	2%	2.95	5.53	0%	3.02	5.55	3%	3.06	5.48	4%	3.01	5.43
7	2%	3.14	5.55	23%	3.03	5.43	0%	3.41	6.86	4%	3.1	5.78	7%	3.14	5.72	1%	3.1	5.54	6%	3.09	5.43	5%	3.01	5.37
10	2%	3.2	5.63	24%	3.1	5.44	0%	3.4	6.86	6%	3.17	5.67	5%	3.17	5.75	1%	3.12	5.49	3%	3.13	5.38	4%	3.09	5.4

Table 13: Effect of α on Smoothing Attack Performance (Llama3-8B). Evaluation of the smoothing attack's effectiveness against different watermarking algorithms on the Llama3-8B model, varying the parameter α. A larger α indicates greater reliance on the reference model's output, while a smaller α means the attack text is more influenced by the watermarked model.

α	KGW			Unigram			SynthID			DIP			Unbiased			UPV			EWD			SWEET		
	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div	TPR	PPL	Div
0.5	13%	3.45	5.92	62%	3.4	5.77	0%	3.78	6.88	35%	3.34	6.19	26%	3.37	6.09	10%	3.47	6.08	28%	3.44	5.84	44%	3.38	5.9
1.0	2%	3.2	5.63	24%	3.1	5.44	0%	3.4	6.86	6%	3.17	5.67	5%	3.17	5.75	1%	3.12	5.49	3%	3.13	5.38	4%	3.09	5.4
2.0	0%	3.05	5.28	12%	2.93	5.21	0%	3.49	6.87	3%	2.99	5.23	3%	2.98	5.28	0%	2.96	5.2	0%	3.0	5.36	1%	3.06	5.38
3.0	0%	2.93	5.17	12%	2.99	5.26	0%	3.52	6.83	1%	2.96	5.16	3%	2.96	5.21	0%	2.99	5.2	0%	2.89	5.18	0%	2.93	5.22

Table 14: Text generated by watermarked model and smoothing attack (OPT-1.3B against Gumbel sampling).

Watermarked	**Smoothing Attack**
Cluster comprises IBM's Opteron-based eServer 325 server and systems management software and storage devices that can run Linux and Windows operating systems. The data center will be built on top of existing IBM Power servers. The company will offer a variety of services, including cloud-based services, as well as a "plug and play" environment that will allow users to transition to other hardware. The data center will be built on top of existing IBM Power servers. The company will offer a variety of services, including cloud-based services, as well as a "plug and play" environment that will allow users to transition to other hardware. (Image: IBM) The data center will be built on top of existing IBM Power servers. The company will offer a variety of services, including cloud-based services, as well as a "plug and play" environment that will allow users to transition to other hardware. The data center will be built on top of existing IBM Power servers. The company will offer a variety of services, including cloud-based services, as well as a "plug and play" environment that will allow users to transition to other hardware.	Cluster comprises IBM's Opteron-based eServer 325 server and systems management software and storage devices that can run Linux and Windows operating systems. IBM will start selling customers a prototype of the cluster by July, according to Jim Bessen, the executive vice president of middleware architecture and services at IBM. "The cluster is just the tip of the iceberg," he said. The cluster firm computing is encased in a step-by-step process starting with activities such as how to set up clinks that back up disks, store data, create back-up schemes for data, and at what resolutions to save the data. "We are not done yet," he said. As new kinds of programming software and other technologies come out, users need to act on it. While IBM expects internal clusters with the capability to run multiple operating systems in fuse over the next year, this capability will be available only to Enterprise Software Group (ESG) customers. ESG will not sell its cluster technology to anyone else, Bessen said.

C Analysis

C.1 Contribution Depends on the Confidence Score of the Unwatermarked Model

We first demonstrate that the contribution of each token to the detection score is influenced by the confidence score of the unwatermarked model, as measured by its probability distribution.

C.1.1 Case Study: Green-Red List Watermark

Suppose that l_t is the logit vector for predicting the t-th token from the unwatermarked model, and $\mathcal{G}_t$ is the green list used by the watermarked model at position t, with size $\gamma|\mathcal{V}|$. Given the watermark shift δ, the probabilities assigned by the unwatermarked and watermarked models are expressed as:

$$P_t(v) = \frac{\exp\big(l_t(v)\big)}{\sum_{v' \in \mathcal{V}} \exp\big(l_t(v')\big)}. \tag{6}$$

$$\widetilde{P}_t(v) = \frac{\exp\big(l_t(v) + \delta \cdot \mathbf{1}_{\{v\in\mathcal{G}_t\}}\big)}{\sum_{v' \in \mathcal{V}} \exp\Big(l_t(v') + \delta \cdot \mathbf{1}_{\{v'\in\mathcal{G}_t\}}\Big)}. \tag{7}$$

Rewriting $\widetilde{P}_t(v)$, we observe:

$$\widetilde{P}_t(v) = P_t(v) \times \frac{\exp\big(\delta\, \mathbf{1}_{\{v\in\mathcal{G}_t\}}\big)}{\sum_{v' \in \mathcal{V}} P_t(v')\, \exp\big(\delta\, \mathbf{1}_{\{v'\in\mathcal{G}_t\}}\big)}.$$

Define the normalization factor:

$$Z_\delta = \frac{\sum_{v'\in\mathcal{V}} \exp\big(l_t(v') + \delta\, \mathbf{1}_{\{v'\in\mathcal{G}_t\}}\big)}{\sum_{v'\in\mathcal{V}} \exp\big(l_t(v')\big)} \tag{8}$$

$$= \sum_{v' \in \mathcal{V}} P_t(v')\, \exp\big(\delta\, \mathbf{1}_{\{v'\in\mathcal{G}_t\}}\big). \tag{9}$$

Then:

$$\widetilde{P}_t(v) = \begin{cases} \dfrac{e^\delta}{Z_\delta}\, P_t(v), & v \in \mathcal{G}_t, \\ \dfrac{1}{Z_\delta}\, P_t(v), & v \notin \mathcal{G}_t. \end{cases}$$

The expected fraction of tokens belonging to the green list under the unwatermarked model is given by:

$$\mathbb{E}_{v\sim P_t}[\mathbf{1}(v \in \mathcal{G}_t)] = \sum_{v\in\mathcal{G}_t} P_t(v) = P_{\mathcal{G}_t},$$

where $P_{\mathcal{G}_t}$ represents the probability mass assigned to green tokens in the unwatermarked model.

Similarly, the expected fraction of green tokens in the watermarked model is:

$$\mathbb{E}_{v\sim\widetilde{P}_t}[\mathbf{1}(v \in \mathcal{G}_t)] = \sum_{v\in\mathcal{G}_t} \widetilde{P}_t(v) = \frac{e^\delta}{Z_\delta}\, P_{\mathcal{G}_t}. \tag{10}$$

Table 15: Impact of Model Size on the Smoothing Attack (OPT). Performance of the smoothing attack across different watermarking algorithms and various sizes of OPT models. The perplexity (PPL) is computed with respect to the OPT-30B model, while the reference model is consistently the OPT-125M. The table reports True Positive Rate (TPR), Perplexity (PPL), and Diversity (Div.) for unwatermarked, watermarked, and smoothed settings.

Size	Setting	KGW			Unigram			SynthID			DIP			Unbiased		
		TPR	PPL	Div.	TPR	PPL	Div.	TPR	PPL	Div.	TPR	PPL	Div.	TPR	PPL	Div.
1.3B	Unwatermarked	0.0%	12.95	8.67	0.0%	12.95	8.67	0.0%	12.95	8.67	0.0%	12.95	8.67	0.0%	12.95	8.67
	Watermarked	100.0%	15.94	8.09	99.0%	16.53	7.29	100.0%	7.7	7.41	100.0%	15.16	8.44	99.0%	15.14	8.29
	Smoothing	4.0%	10.48	6.72	6.0%	10.37	6.83	1.0%	11.37	8.67	6.0%	10.03	7.03	4.0%	9.94	6.79
2.7B	Unwatermarked	0.0%	11.75	8.36	0.0%	11.75	8.36	0.0%	11.75	8.36	0.0%	11.75	8.36	0.0%	11.75	8.36
	Watermarked	100.0%	13.94	7.88	100.0%	14.31	7.41	99.0%	6.86	7.55	97.0%	13.86	8.61	97.0%	13.6	8.69
	Smoothing	4.0%	10.35	6.77	4.0%	10.35	6.66	6.0%	9.84	8.0	13.0%	9.87	6.84	6.0%	9.85	6.88
6.7B	Unwatermarked	0.0%	10.2	8.45	0.0%	10.2	8.45	0.0%	10.2	8.45	0.0%	10.2	8.45	0.0%	10.2	8.45
	Watermarked	100.0%	13.16	8.06	100.0%	12.94	7.48	98.0%	6.21	7.48	98.0%	11.8	8.48	97.0%	11.79	8.59
	Smoothing	4.0%	10.07	6.92	6.0%	10.54	6.68	3.0%	8.98	8.31	8.0%	9.78	6.86	8.0%	9.68	6.74
13B	Unwatermarked	0.0%	10.14	8.39	0.0%	10.14	8.39	0.0%	10.14	8.39	0.0%	10.14	8.39	0.0%	10.14	8.39
	Watermarked	100.0%	12.88	8.56	100.0%	12.44	7.39	100.0%	5.88	7.8	96.0%	11.67	9.34	93.0%	11.42	8.77
	Smoothing	2.0%	10.24	6.82	5.0%	10.32	6.7	8.0%	8.07	7.8	8.0%	9.6	6.88	7.0%	9.37	6.77
30B	Unwatermarked	0.0%	8.46	8.44	0.0%	8.46	8.44	0.0%	8.46	8.44	0.0%	8.46	8.44	0.0%	8.46	8.44
	Watermarked	100.0%	10.23	8.34	100.0%	10.45	7.56	100.0%	5.27	7.72	94.0%	9.43	8.78	97.0%	9.89	9.08
	Smoothing	0.0%	9.5	6.8	7.0%	10.15	6.75	5.0%	6.96	8.04	4.0%	9.34	6.89	4.0%	9.36	6.88

Size	Setting	X-SIR			UPV			Gumbel			EWD			SWEET		
		TPR	PPL	Div.	TPR	PPL	Div.	TPR	PPL	Div.	TPR	PPL	Div.	TPR	PPL	Div.
1.3B	Unwatermarked	1.0%	12.95	8.67	0.0%	12.95	8.67	0.0%	12.95	8.67	0.0%	12.95	8.67	0.0%	12.95	8.67
	Watermarked	94.0%	15.42	7.96	99.0%	12.79	8.22	98.0%	3.15	4.35	100.0%	16.88	7.92	100.0%	15.99	8.02
	Smoothing	13.0%	10.3	6.72	20.0%	10.78	6.89	9.0%	20.94	8.30	1.0%	10.71	6.75	1.0%	10.54	6.81
2.7B	Unwatermarked	3.0%	11.75	8.36	0.0%	11.75	8.36	0.0%	11.75	8.36	0.0%	11.75	8.36	0.0%	11.75	8.36
	Watermarked	91.0%	14.07	8.25	99.0%	12.30	8.01	99.0%	2.96	4.38	100.0%	14.88	7.98	100.0%	14.07	8.32
	Smoothing	10.0%	10.34	6.77	18.0%	10.56	6.90	10.0%	19.46	8.41	1.0%	10.43	6.86	3.0%	10.49	6.86
6.7B	Unwatermarked	0.0%	10.2	8.45	0.0%	10.20	8.45	0.0%	10.20	8.45	0.0%	10.20	8.45	0.0%	10.20	8.45
	Watermarked	91.0%	13.04	8.19	97.0%	10.92	7.75	100.0%	2.97	4.49	100.0%	13.42	8.69	100.0%	13.05	8.41
	Smoothing	9.0%	10.01	6.7	8.0%	10.60	7.05	9.0%	14.85	8.62	0.0%	10.60	6.79	1.0%	10.07	6.89
13B	Unwatermarked	0.0%	10.14	8.39	0.0%	10.14	8.39	0.0%	10.14	8.39	0.0%	10.14	8.39	0.0%	10.14	8.39
	Watermarked	88.0%	12.29	8.05	99.0%	10.59	7.91	98.0%	2.96	4.63	100.0%	13.09	8.74	100.0%	12.32	8.35
	Smoothing	11.0%	9.84	6.79	12.0%	10.84	6.88	12.0%	15.06	8.27	0.0%	10.16	6.73	2.0%	10.15	6.74
30B	Unwatermarked	0.0%	8.46	8.44	0.0%	8.46	8.44	0.0%	8.46	8.44	0.0%	8.46	8.44	0.0%	8.46	8.44
	Watermarked	91.0%	10.43	8.43	97.0%	8.59	8.13	97.0%	2.89	4.79	100.0%	10.75	8.54	100.0%	9.98	8.25
	Smoothing	16.0%	9.65	6.74	17.0%	10.06	7.11	9.0%	11.92	8.39	2.0%	10.02	6.99	2.0%	9.55	6.84

Since $Z_\delta = (e^\delta - 1)P_{\mathcal{G}_t} + 1$, the difference in green token probabilities (i.e., the detection contribution at token position t) is:

$$S_t = \mathbb{E}_{v \sim \widetilde{P}_t}[\mathbf{1}(v \in \mathcal{G}_t)] - \mathbb{E}_{v \sim P_t}[\mathbf{1}(v \in \mathcal{G}_t)] \tag{11}$$

$$= \frac{-(e^\delta - 1)P_{\mathcal{G}_t} + (e^\delta - 1)}{(e^\delta - 1) + \frac{1}{P_{\mathcal{G}_t}}}. \tag{12}$$

In other words, the token-level detection contribution S_t is a function of the probability mass $P_{\mathcal{G}_t}$ assigned to green tokens by the unwatermarked model.

C.1.2 Case Study: Tournament Sampling Watermark

In the Tournament Sampling watermark, when generating the t-th token, the algorithm assigns scores to each token using m independent watermarking functions $g^{(1)}, ..., g^{(m)}$. These scores depend on a random seed generated based on the recent context and a secret watermarking key. The token selection follows a multi-round elimination process, where 2^m tokens are first sampled from $P_t(\cdot)$, then compete in m rounds to determine the final output.

Despite the complex sampling mechanism, the probability of each token in the modified distribution $\widetilde{P}_t$ is adjusted by a factor dependent on its assigned g value. Specifically, for any token v:

$$\widetilde{P}_t(v) = \begin{cases} P_t(v) \cdot (1 - P_{\mathcal{G}_t}) & \text{if } g(v) = 0, \\ P_t(v) \cdot (2 - P_{\mathcal{G}_t}) & \text{if } g(v) = 1. \end{cases} \tag{13}$$

During watermark detection, the detector computes the average g value across all tournament layers, i.e., $\frac{1}{m}\sum_{l=1}^{m} g^{(l)}(v)$, as the watermark score for the token.

Single Tournament Layer ($m = 1$). Consider the simplest case where $m = 1$, meaning only one tournament round is used. Let $\mathcal{G}_t$ denote the set of tokens where $g^{(1)}(v) = 1$. The probability modification simplifies to:

$$\widetilde{P}_t(v) = \begin{cases} P_t(v) \cdot (1 - P_{\mathcal{G}_t}) & \text{if } v \notin \mathcal{G}_t, \\ P_t(v) \cdot (2 - P_{\mathcal{G}_t}) & \text{if } v \in \mathcal{G}_t. \end{cases} \tag{14}$$

The expected g value for tokens sampled from $\widetilde{P}_t$ is $(2 - P_{\mathcal{G}_t}) \cdot P_{\mathcal{G}_t}$, while the expectation under P_t is simply $P_{\mathcal{G}_t}$. Thus, the detection contribution S_t is:

$$S_t = (1 - P_{\mathcal{G}_t}) \cdot P_{\mathcal{G}_t}. \tag{15}$$

This mirrors the Green-Red List watermark, showing that the detection contribution per token is fundamentally tied to $P_{\mathcal{G}_t}$.

C.2 Low Model Confidence Leads to Large Variance in the Watermark Score for Unwatermarked Text

Thus far, we have established that the contribution of each token to the detection score is correlated with the expected watermark score under the unwatermarked model. We now analyze what affects the watermark score of the unwatermarked model.

Let $P_t = (p_1, p_2, \ldots, p_d)$ be the probability vector from the unwatermarked model at token position t, where $p_i \in [0, 1]$ and $\sum_{i=1}^{d} p_i = 1$. Typically, $d = |\mathcal{V}|$ is large. We randomly select a subset $\mathcal{G}_t \subset \{1, \ldots, d\}$ of indices of size $\gamma|\mathcal{V}|$. Define the random variable:

$$P_{\mathcal{G}_t} \;=\; \sum_{i \in \mathcal{G}_t} p_i.$$

We analyze how $P_{\mathcal{G}_t}$ is distributed over all possible assignments of $\mathcal{G}_t$. Define the indicator variable X_i as follows:

$$X_i \;=\; \begin{cases} 1, & \text{if } i \in \mathcal{G}_t, \\ 0, & \text{otherwise.} \end{cases}$$

Since each token is independently assigned to $\mathcal{G}_t$ with probability γ, we have:

$$\mathbb{E}[X_i] = \gamma, \quad \text{and} \quad \mathrm{Var}(X_i) = \gamma(1 - \gamma).$$

For different token indices $i \neq j$, the covariance between their assignments is:

$$\mathrm{Cov}(X_i, X_j) = \mathbb{E}[X_i X_j] - \mathbb{E}[X_i]\mathbb{E}[X_j].$$

For Poisson sampling (i.e., assigning each token to $\mathcal{G}_t$ independently with probability γ), the covariance is zero. However, under a fixed-size sampling setup (i.e., selecting exactly $\gamma|\mathcal{V}|$ tokens), we have:

$$\begin{aligned} \mathrm{Cov}(X_i, X_j) &= \frac{\gamma|\mathcal{V}|}{d} \cdot \frac{\gamma|\mathcal{V}| - 1}{d - 1} - \gamma^2 \\ &= -\frac{\gamma(1 - \gamma)}{|\mathcal{V}| - 1}. \end{aligned}$$

Expressing $P_{\mathcal{G}_t}$ in terms of X_i, we obtain:

$$P_{\mathcal{G}_t} \;=\; \sum_{i=1}^{d} X_i\, p_i.$$

Expectation and Variance of $P_{\mathcal{G}_t}$. The expectation is:

$$\mathbb{E}[P_{\mathcal{G}_t}] \;=\; \sum_{i=1}^{d} \mathbb{E}[X_i]\, p_i \;=\; \gamma \sum_{i=1}^{d} p_i \;=\; \gamma.$$

The variance $\mathrm{Var}(P_{\mathcal{G}_t})$ is:

$$\sum_{i=1}^{d} p_i^2 \mathrm{Var}(X_i) + \sum_{i \neq j} p_i p_j \mathrm{Cov}(X_i, X_j).$$

Substituting $\mathrm{Var}(X_i) = \gamma(1 - \gamma)$ and $\mathrm{Cov}(X_i, X_j) = -\frac{\gamma(1-\gamma)}{|\mathcal{V}|-1}$:

$$\mathrm{Var}(P_{\mathcal{G}_t}) = \gamma(1-\gamma) \sum_{i=1}^{d} p_i^2 - \frac{\gamma(1-\gamma)}{|\mathcal{V}| - 1} \sum_{i \neq j} p_i p_j.$$

For the first term,

$$\gamma(1-\gamma) \sum_{i=1}^{d} p_i^2 = \gamma(1-\gamma)\sigma^2,$$

where $\sigma^2 = \sum_{i=1}^{d} p_i^2$ represents the squared ℓ_2 norm of the probability vector.

For the second term, using the identity:

$$\sum_{i \neq j} p_i p_j = \left(\sum_{i=1}^{d} p_i\right)^2 - \sum_{i=1}^{d} p_i^2 = 1 - \sigma^2,$$

and we obtain:

$$\frac{\gamma(1-\gamma)}{|\mathcal{V}|-1}\sum_{i\neq j} p_i p_j = \frac{\gamma(1-\gamma)}{|\mathcal{V}|-1}(1-\sigma^2).$$

For large $|\mathcal{V}|$, the correction term $\frac{\gamma(1-\gamma)}{|\mathcal{V}|-1}(1-\sigma^2)$ becomes negligible, and we approximate:

$$\mathrm{Var}(P_{\mathcal{G}_t}) \approx \gamma(1-\gamma)\sigma^2.$$

Interpretation. This analysis shows that $P_{\mathcal{G}_t}$ depends on the probability mass distribution.

High-Uncertainty Case (Uniform Distribution): If $p_i = \frac{1}{|\mathcal{V}|}$ for all i, then

$$\sigma^2 = \sum_{i=1}^{|\mathcal{V}|} \frac{1}{|\mathcal{V}|^2} = \frac{1}{|\mathcal{V}|}.$$

For large $|\mathcal{V}|$, σ^2 is small, meaning that the distribution of $P_{\mathcal{G}_t}$ concentrates tightly around γ with small variance. This corresponds to a scenario where the model has high uncertainty, spreading probability mass nearly uniformly over all tokens.

Low-Uncertainty Case (Dominant Tokens): In practice, language models often assign high probability mass to a small number of dominant tokens. Suppose $p_j \geq 0.8$ for some token j, then:

$$\sigma^2 \geq p_j^2 = 0.64.$$

In this case, σ^2 is much larger than $1/|\mathcal{V}|$ (which is on the order of 10^{-5} for large models). Consequently, $P_{\mathcal{G}_t}$ exhibits a bimodal distribution: it is either close to 0 or close to 1, depending on whether the dominant tokens are in $\mathcal{G}_t$. The probability of $P_{\mathcal{G}_t} \approx \gamma$ is nearly zero.

Thus, when the model is confident in its predictions (low uncertainty), the variance of $P_{\mathcal{G}_t}$ is large, leading to a higher variance in the watermark score. Conversely, when the model is uncertain, the watermark score is more stable and centered around γ.

Connection to Watermark Detection. Since the contribution to the detection score S_t depends on $P_{\mathcal{G}_t}$ (Eq. equation 5), its variance is governed by $\mathrm{Var}(P_{\mathcal{G}_t})$. This means that tokens generated with high confidence contribute more variability to the detection score, whereas tokens generated under uncertainty contribute less variability.

C.3 Estimating the Confidence Score of the Unwatermarked Model Using the Watermarked Model

Our goal is to estimate the squared ℓ_2 norm of the probability distribution $\|P_t\|^2$, which serves as a confidence measure for the unwatermarked model, using only access to the watermarked model $\widetilde{P}_t$. This estimation is critical for adaptive attacks and for understanding how watermarking affects text quality.

Setup. We consider the Green-Red List watermarking scheme, where the probability distribution $\widetilde{P}_t$ is obtained by modifying P_t as:

$$\widetilde{P}_t(v) \ = \ \frac{e^{\delta \mathbf{1}_{\{v\in\mathcal{G}_t\}}}}{Z_\delta} P_t(v),$$

where the normalization factor Z_δ is defined as:

$$Z_\delta = (1-P_{\mathcal{G}_t}) + e^\delta P_{\mathcal{G}_t}.$$

We aim to construct an estimator $\widehat{U}$ for the confidence measure:

$$\|P_t\|^2 = \sum_{v\in\mathcal{V}} P_t(v)^2.$$

Expected Squared Norm of the Watermarked Model. Since each probability mass in P_t is scaled by either e^δ/Z_δ (if in $\mathcal{G}_t$) or $1/Z_\delta$ (if not in $\mathcal{G}_t$), we have:

$$\mathbb{E}[\widetilde{P}_t(v)^2] \ = \ (1-\gamma)\frac{1}{Z_\delta^2}P_t(v)^2 + \gamma\frac{e^{2\delta}}{Z_\delta^2}P_t(v)^2.$$

Summing over all tokens in $\mathcal{V}$, we obtain:

$$\mathbb{E}[\|\widetilde{P}_t\|^2] \ = \ \frac{(1-\gamma)+\gamma e^{2\delta}}{Z_\delta^2}\|P_t\|^2.$$

Unbiased Estimator. Rearranging the above expression, we define an unbiased estimator:

$$\widehat{U} = \frac{Z_\delta^2}{(1-\gamma)+\gamma e^{2\delta}}\|\widetilde{P}_t\|^2.$$

Taking expectation, we confirm:

$$\mathbb{E}[\widehat{U}] = \|P_t\|^2.$$

Practical Approximation. Since Z_δ depends on $P_{\mathcal{G}_t}$, which is unknown to an adversary, we approximate it using γ:

$$Z_\delta \approx (1-\gamma) + \gamma e^\delta.$$

Thus, the practical estimator becomes:

$$\widetilde{U} = \frac{[(1-\gamma)+\gamma e^\delta]^2}{(1-\gamma)+\gamma e^{2\delta}} \|\widetilde{P}_t\|^2.$$

This provides a computationally efficient way to estimate $\|P_t\|^2$ using only $\widetilde{P}_t$, making it useful for designing attacks.

C.4 Estimating the ℓ_2 Norm Using Top-K Probabilities

While we have established the connection between the squared ℓ_2 norm $\|P_t\|^2$ of the probability distribution and its contribution to the watermark detection score, direct access to this quantity is often unavailable, even for the watermarked model. In this section, we show how to estimate $\|P_t\|^2$ using only limited access to the model's top-K probabilities.

Suppose we only have access to the top-K probabilities:

$$p_1 \geq p_2 \geq \cdots \geq p_K,$$

where the remaining probabilities $p_{K+1}, \ldots, p_{|\mathcal{V}|}$ are unknown. Define the remaining probability mass of the tail as:

$$R = 1 - \sum_{i=1}^{K} p_i.$$

Our goal is to estimate the squared ℓ_2 norm:

$$\|P_t\|^2 = \sum_{i=1}^{|\mathcal{V}|} p_i^2,$$

given only $p_1, \ldots, p_K$ and R.

We bound $\|P_t\|^2$ by considering two extreme ways in which the unknown tail probabilities could be distributed:

1. Uniform Tail: The remaining probability mass R is evenly distributed across the unknown tokens, minimizing the sum of squares.

2. Concentrated Tail: The entire probability mass R is assigned to a single token, maximizing the sum of squares.

Uniform Tail (Lower Bound) If the tail probability mass R is *uniformly* spread among the remaining $|\mathcal{V}| - K$ tokens, then each unknown probability is $\frac{R}{|\mathcal{V}|-K}$. The squared sum of the tail probabilities is then:

$$\sum_{i=K+1}^{|\mathcal{V}|} p_i^2 = (|\mathcal{V}| - K)\left(\frac{R}{|\mathcal{V}| - K}\right)^2 = \frac{R^2}{|\mathcal{V}| - K}.$$

Since distributing the mass uniformly minimizes the squared sum (due to convexity), this scenario provides a lower bound for $\|P_t\|^2$:

$$\|P_t\|^2 \;\geq\; \sum_{i=1}^{K} p_i^2 + \frac{R^2}{|\mathcal{V}| - K}.$$

Concentrated Tail (Upper Bound) At the other extreme, if the entire remaining probability mass R is assigned to a single token, then the squared sum of the tail probabilities is simply:

$$\sum_{i=K+1}^{|\mathcal{V}|} p_i^2 = R^2.$$

Since concentrating all probability mass in one entry maximizes the sum of squares, this provides an upper bound for $\|P_t\|^2$:

$$\|P_t\|^2 \;\leq\; \sum_{i=1}^{K} p_i^2 + R^2.$$

Combining both bounds, we obtain:

$$\sum_{i=1}^{K} p_i^2 + \frac{R^2}{|\mathcal{V}| - K} \;\leq\; \|P_t\|^2 \;\leq\; \sum_{i=1}^{K} p_i^2 + R^2,$$

where $R = 1 - \sum_{i=1}^{K} p_i$.

Practical Approximation. A commonly used practical heuristic is to assume that the remaining probability mass R follows a uniform distribution across the unknown probabilities. Under this assumption, we approximate:

$$\|P_t\|^2 \;\approx\; \sum_{i=1}^{K} p_i^2 + \frac{R^2}{|\mathcal{V}| - K}.$$

This estimate tends to be slightly lower than the true value, since in reality, the tail probabilities are rarely perfectly uniform—some tokens may have slightly higher probabilities than others. However, in the case of language modeling, probability distributions often exhibit a “long tail” where the remaining probability mass is spread across many small values. In such cases, the uniform assumption serves as a reasonable first-order approximation.

C.5 Additional Numerical Analysis

Generalization to other watermarking solutions. For Gumbel sampling, we define the token-level contribution to watermark detection as $S_t = -\log(1 - U_{v^*}) - \mathbb{E}_{v \sim P_t}[-\log(1 - U_v)]$, where v^* is the token selected by the watermarked model. Note that the choice of v^* is deterministic after the secret key held by the LM provider and the prefix content are fixed. For Tournament sampling, we define the token-level contribution as $S_t = \mathbb{E}_{v \sim \widetilde{P}_t}\left[\frac{1}{m}\sum_{l=1}^{m} g^{(l)}(v, r)\right] - \mathbb{E}_{v \sim P_t}\left[\frac{1}{m}\sum_{l=1}^{m} g^{(l)}(v, r)\right]$, where $\widetilde{P}_t$ is the modified probability distribution.

For these two watermarks, we still observe the same correlation between S_t and $\|P_t\|^2$ as we have for Green-list watermarks, as shown in Figure 7. Namely, the token-level contribution S_t to the watermark detectability is negatively correlated to the model’s confidence at position t.

Impact of watermarking on text quality We also plot $D_{TV}(P_t, P_t^{\text{ref}})$, which measures the negative impact on text quality if we alternatively sample from the reference model OPT-125M (in color red). We note that when the model is not confident in its output, i.e., when $\|P_t\|^2$ is small, sampling from the reference model’s token distribution, i.e., P_t^{ref}, does not hurt the text quality. In particular, under the Green-red list watermarking scheme, $D_{TV}(P_t, P_t^{\text{ref}})$ is comparable to $D_{TV}(P_t, \widetilde{P}_t)$ when $\|P_t\|^2$ is small (observe that the red points generally overlap with the blue ones). For Gumbel and Tournament sampling, $D_{TV}(P_t, P_t^{\text{ref}})$ is even smaller than $D_{TV}(P_t, \widetilde{P}_t)$ when $\|P_t\|^2$ is small (observe that the red points are generally below the blue ones). Conversely, when the model is confident in its output, i.e., when $\|P_t\|^2$ is large, replacing the watermarked model with a reference model may hurt the text quality (observe that the red points are above the blue ones).

Trade-off between detectability and text quality In Figure 9, we plot the correlation between $D_{TV}(P_t, \widetilde{P}_t)$ and S_t, empirically measured on OPT-1.3B model using the same setup as the above simulations. When the watermark has little impact on text quality (i.e., smaller total variation distance), the watermark is also less detectable (i.e., smaller S_t). Conversely, tokens that contribute more to watermark detection also lead to more notable text quality degradation. This finding, in turn, reveals the crucial limitation of existing watermarking schemes: high watermark detectability and high text quality cannot be achieved at the same time, since the very same set of tokens causes quality degradation while contributing to watermark detectability simultaneously.

D Possible Defenses to Smoothing Attack

Our attack exploits the correlation between a token’s contribution to the watermark detection score and the confidence level of the unwatermarked model in predicting that token. One possible defense against this attack is to restrict access to confidence-related information, such as returning only the most probable token without revealing its probability. Note that, if the probability of the most likely token is available, our attack remains effective. However, such a defense is challenging to enforce in practice. Many existing LLM services provide *top-K probabilities* (e.g., OpenAI’s API returns probabilities for the top 20 tokens), which is already sufficient to approximate model confidence and execute our attack. Moreover, service providers often release these probabilities to enhance transparency and build trust by providing insights into the model’s reasoning, addressing concerns about the opacity of AI systems (European Commission, 2021; OECD, 2019). Access to probability distributions is also essential for debugging and evaluating model performance, as it allows developers to identify biases, diagnose overconfidence, and improve reliability (National Institute of Standards and Technology (NIST), 2023). Probabilities support explainable AI (XAI) by revealing model uncertainty, enabling users to interpret predictions and explore alternative suggestions (Brown et al., 2020). From an ethical standpoint, making probability distributions available facilitates bias auditing and aligns with broader efforts to promote fairness and accountability in AI (OECD, 2019). Additionally, probability

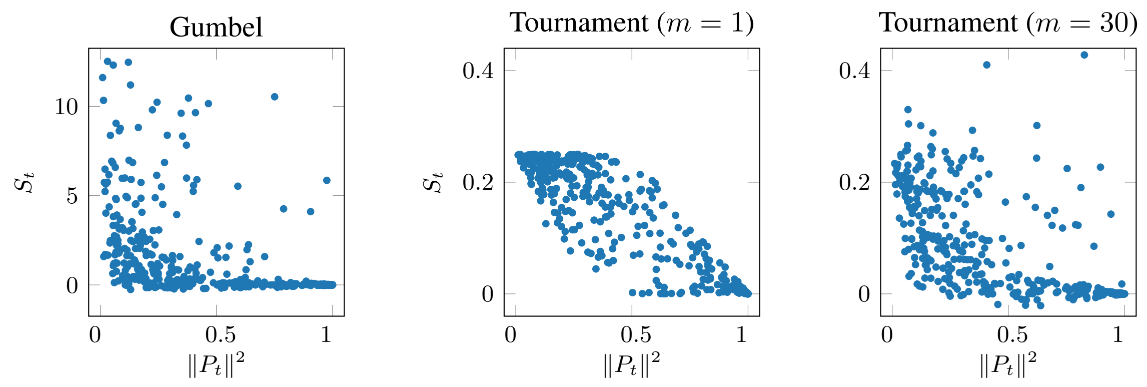

Figure 7: The correlation between S_t (watermark contribution score) and $\|P_t\|^2$ (model confidence) evaluated on model OPT-1.3B with the Gumbel and Tournament sampling (with m tournaments) watermarks, using the same setup as in Figure 1. Each sample corresponds to a specific prefix and secret key. $\|P_t\|^2$ is computed from the original un-watermarked model. The overall observation is similar to what we have for the *Green-red list watermarking*: S_t decreases as $\|P_t\|^2$ increases.

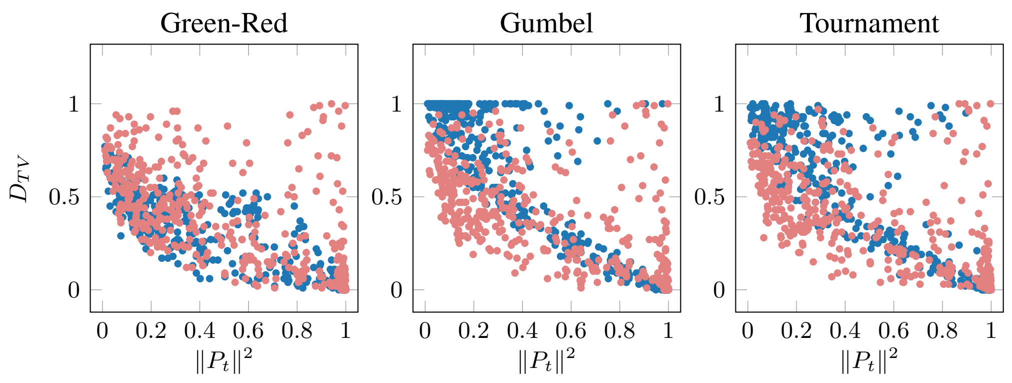

Figure 8: The correlation between $D_{TV}(P_t, \widetilde{P}_t)$, i.e., the negative impact on text quality due to watermarks (in color blue), and $\|P_t\|^2$, measured on OPT-1.3B with the Green-red list and Gumbel and Tournament sampling watermarks. We also plot $D_{TV}(P_t, P_t^{\text{ref}})$, which measures the negative impact on text quality if we use tokens sampled from the reference model OPT-125M (in color red).

information empowers developers and end users by enabling advanced decision-making strategies, such as re-ranking, rejection sampling, and beam search (OpenAI, 2023). Furthermore, it helps mitigate risks associated with model overconfidence and hallucinations, which is particularly crucial in high-stakes domains such as healthcare and law (National Institute of Standards and Technology (NIST), 2023). Given the practical difficulties in restricting access to confidence-related information, our findings suggest that existing watermarking techniques may be vulnerable when model confidence can be estimated. This highlights the need for developing watermarking schemes that remain effective even in scenarios where adversaries have partial access to confidence estimates. Future research should explore watermarking methods that explicitly account for the model's confidence and ensure robustness against adversarial attacks that exploit confidence information.

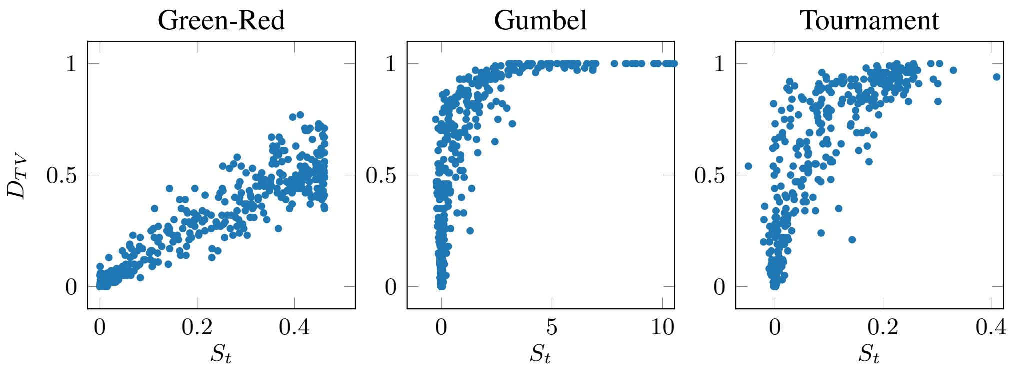

Figure 9: The correlation between $D_{TV}(P_t, \widetilde{P}_t)$, i.e., the negative impact of watermarking on the text quality, and S_t, i.e., the token-level contribution to watermark detectability. We measure this on OPT-1.3B. For all three watermarking schemes, $D_{TV}(P_t, \widetilde{P}_t)$ increases as S_t increases.