Downstream Trade-offs of a Family of Text Watermarks

Anirudh Ajith*
Princeton University
anirudh.ajith@princeton.edu

Sameer Singh
University of California, Irvine
sameer@uci.edu

Danish Pruthi
Indian Institute of Science
danishp@iisc.ac.in

Abstract

Watermarking involves implanting an imperceptible signal into generated text that can later be detected via statistical tests. A prominent family of watermarking strategies for LLMs embeds this signal by upsampling a (pseudorandomly-chosen) subset of tokens at every generation step. However, such signals alter the model's output distribution and can have unintended effects on its downstream performance. In this work, we evaluate the performance of LLMs watermarked using three different strategies over a diverse suite of tasks including those cast as k-class classification (CLS), multiple choice question answering (MCQ), short-form generation (e.g., open-ended question answering) and long-form generation (e.g., translation) tasks. We find that watermarks (under realistic hyperparameters) can cause significant drops in LLMs' effective utility across all tasks. We observe drops of 10 to 20% in CLS tasks in the average case, which shoot up to 100% in the worst case. We notice degradations of about 7% in MCQ tasks, 10–15% in short-form generation, and 5–15% in long-form generation tasks. Our findings highlight the trade-offs that users should be cognizant of when using watermarked models.[1]

1 Introduction

Large Language Models (LLMs), and derived chatbots of the likes of ChatGPT, can generate human-like responses to a variety of requests like writing emails, translating or summarizing content (Brown et al., 2020; Chowdhery et al., 2022). As these systems gain popularity, there are looming concerns about their misuse for spreading targeted misinformation, influencing public opinion (Panditharatne and Giansiracusa, 2023) or conducting social engineering attacks (Grbic and Dujlovic, 2023).

Such concerns have spurred research towards distinguishing human-written and LLM-generated content. Naive approaches such as training post-hoc classifiers for this purpose have been shown to be ineffective, as they typically have large false-positive rates that can lead to false accusations of plagiarism (O'Neill, 2023; OpenAI, 2023). These classifiers can further degrade in accuracy when LLM developers like OpenAI continually finetune and update their public models. Additionally, the output distributions of future LLMs may grow even more similar to that of human-written text causing the efficacy of such approaches to wane.

A promising alternative is to intentionally embed a *watermark* signal (Atallah et al., 2001; Chiang et al., 2004; Topkara et al., 2006; Jalil and Mirza, 2009) into LLM-generated text that is imperceptible to unsuspecting readers but can be algorithmically detected using statistical tests. A popular watermarking scheme, often referred to as KGW, works by boosting the probabilities of a psuedo-randomly chosen subset of the model's vocabulary at every generation step (Kirchenbauer et al., 2023a). This scheme has been extensively studied and extended (Kirchenbauer et al., 2023b; Liu et al., 2024; Lu et al., 2024). The original approach and its derivatives, collectively called the KGW family in the literature, comprise the most popular watermarking strategies for LLMs today.

Previous works studying the trade-offs of watermarking LLMs mostly restrict their analysis to intrinsic evaluations of watermarked models generation quality such as perplexity or GPT4 judgements (Singh and Zou, 2023), eschewing evaluation on downstream task benchmarks. But since it is likely that all strong LLMs made available to the public (eg. through internet APIs) will be watermarked in the near future (as promised by several leading LLM developers (Press, 2023)), it is important to understand how watermarks impact LLMs' performance on downstream tasks.

*Work done during an internship at IISc, Bangalore.

[1]We make our code available at https://github.com/FLAIR-IISc/watermark_tradeoffs.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 14039–14053
November 12-16, 2024

In our work, we evaluate the downstream impact of 3 popular watermarks from the KGW family including the original KGW approach (Kirchenbauer et al., 2023a), EWD (Lu et al., 2024) and SIR (Liu et al., 2024) over a diverse selection of tasks. Since KGW-based watermarks perturb output probability distributions at the token level, we categorize the tasks as follows for our analysis:

1. CLS: Tasks framed as k-class classification problems with static labels.
2. MCQ: Tasks framed as multiple choice question-answering problems with choices that differ across test examples.
3. SGEN: Tasks requiring generation of a short output sequence via sampling.
4. LGEN: Tasks involving the generation of a long output sequence by repeatedly sampling from the LLM's probability distributions.

We categorize the examined tasks into these four buckets as we expect similar effects of watermarking for tasks in a given category. For instance, for CLS tasks, there is a possibility of systematic bias against some labels for every test example. In SGEN and LGEN tasks, the modifications to the output distributions due to watermarking can directly impact the correctness of generated content.[2]

We evaluate the performance of watermarked LLaMA (Touvron et al., 2023a), Mistral (Jiang et al., 2023) and OPT (Zhang et al., 2022) models and observe that, under realistic watermark settings, watermarking can cause significant drops in LLMs' effective utility across all tasks. We notice drops of 10–20% in CLS tasks in the average case which can rise up to 100% in the worst case. We see drops of about 7% in MCQ tasks, 10–15% in short-form generation, and 5–15% in long-form generation.

We believe that our findings will allow model developers and users to make informed choices about watermarked models and spur interest into developing novel watermarking schemes and decoding strategies that may exhibit better performance trade-offs. We make our code available at https://github.com/FLAIR-IISc/watermark_tradeoffs to facilitate research in this area, and holistically evaluate future watermarking approaches.

[2]We treat short-form and long-form generation tasks differently due to differences in how they are evaluated.

2 Background

The KGW watermark (Kirchenbauer et al., 2023a) is a deterministic algorithm parameterized by 3 hyperparameters γ, δ and k, and a keyed psuedo-random function $F. : \mathbb{N} \to \{g, r\}^m$.

Generation. The algorithm works by modifying the logits obtained from the language model at each generation step. Formally, given a model $\mathcal{M}$ with vocabulary V, and a prefix comprising tokens $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n$ the scheme involves first computing the logits $\mathcal{M}(\mathbf{w}_1 \ldots, \mathbf{w}_n) = (l_1, \ldots, l_{|V|})$ of the language model that would ordinarily be used to predict the subsequent token. The terminal prefix token $\mathbf{w}_n$ is then fed to F under the key k to obtain a partition of V into a green list G and a red list R such that $|G| = \lfloor \gamma |V| \rfloor$. That is,

$$F_k(\mathbf{w}_n) \in \{g, r\}^{|\mathcal{V}|}$$

such that $\sum_{x \in F_k(\mathbf{w_n})} \mathbb{1}[x = g] = \lfloor \gamma |V| \rfloor$. Finally, watermarked logits $(\lambda_1, \ldots, \lambda_{|V|})$ are computed as $\lambda_i = l_i + \delta \cdot \mathbb{1}[i \in G]$. These watermarked logits can then be used for sampling tokens (for generation) or even computing the likelihood or perplexity of a given sequence.

Detection. The detection scheme proposed by Kirchenbauer et al. (2023a) works by assessing the probability of a null hypothesis that the given text was written without knowledge of the watermarking scheme (specifically hash key k). Precisely, given a token sequence x of length T that was written without knowledge of the scheme, the number of green list tokens in x, denoted by $|x|_G$, can be assumed to be normally distributed with a mean of γT and a standard deviation of $\sqrt{T\gamma(1-\gamma)}$. The detection algorithm computes a z-score,

$$z = (|x|_G - \gamma T)/\sqrt{T\gamma(1-\gamma)}, \qquad (1)$$

and rejects the null hypothesis if this z-score exceeds a chosen threshold.

Variations. Entropy-based Watermarking Detection (EWD) is a variation of the KGW approach that uses the same generation algorithm but differs in its detection algorithm (Lu et al., 2024). EWD seeks to improve the trade-off between watermark detectability and language modeling ability by proposing a novel entropy-based detection strategy that involves reweighting individual tokens

using their entropies during detection. It computes an adjusted z score as

$$z' = (|x|_G - \gamma \sum_{i=m}^{|T|-1} W_i) / \sqrt{T\gamma(1-\gamma) \sum_{i=m}^{|T|-1} W_i^2}$$

where W_i is computed based on the entropy of the i^{th} token of the sequence x.

Another KGW-based derivative called Semantic-Invariant and Robust (SIR) watermarking aspires to improve the robustness against paraphrasing attacks (Liu et al., 2024). The SIR watermarking scheme computes G as a pseudorandom function of a semantic embedding of its prefix, departing from KGW's strategy of computing G using the prefix's lexical properties. For SIR,

$$F'_k(\text{embed}(\mathbf{w}_1 \ldots, \mathbf{w}_n)) = \{g, r\}^{|V|}$$

Specifically, SIR utilizes a sentence encoder to generate a semantic embedding for the prefix and then uses a learned 'watermark model' to transform this embedding into a partition over the model's vocabulary. A notable feature of the SIR watermark is that the γ hyperparameter cannot be set explicitly, but instead is implicitly determined by the watermark model at every generation step. This model's training objective incentivizes the effective γ to always take on values close to 0.5.

3 Evaluation Setup

Datasets. To assess watermarks' effects on tasks that are framed as classification tasks (CLS), we work with SST-2, BoolQ and CB from the GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a) benchmarks. These correspond to sentiment analysis, yes/no question answering and textual entailment tasks respectively. We select the commonsense NLI dataset called HellaSwag (Zellers et al., 2019) and the question-answering dataset PIQA (Bisk et al., 2020) as MCQ tasks, the reading comprehension datasets DROP (Dua et al., 2019) and SQuADv2 (Rajpurkar et al., 2016) as SGEN tasks, and the WMT14-En-Fr (Bojar et al., 2014) and WMT20-En-De (Barrault et al., 2020) translation tasks as LGEN tasks. We evaluate models' performance on these tasks using the metrics typically associated with them. For instance, CLS and MCQ tasks are evaluated using accuracy while SGEN tasks are evaluated using F1 scores and LGEN translation tasks are evaluated using BLEU scores (Papineni et al., 2002).

Models. We analyze the performance trade-offs for the above tasks for watermarked and unwatermarked versions of LLaMA 7B (Touvron et al., 2023a), Mistral 7B (Jiang et al., 2023) and OPT 6.8B (Zhang et al., 2022) models.

Methodology. In KGW-based watermarks, the γ and δ hyperparameters control the strength of the watermark signal and accordingly, the shift in watermarked models' output distribution. However due to differences among these algorithms, a specific (γ, δ) setting does not imply the same signal strength (as measured by its empirical detectability) or impact on an LLM's language modeling ability.

To ensure a fair comparison of these schemes' downstream implications, we find the settings of hyperparameters for each watermarked model such that the resulting signal is of the same strength. In the watermarking literature, signal strengths are typically evaluated by computing the True Positive Rates (TPR) of their detection algorithm at a fixed False Positive Rate (FPR). Generally, FPR is set to a low value such as 0.01, to avoid the risk of false accusing someone of plagiarism. In our evaluation, we consider signal strengths of 0.5, 0.75 and 0.95 TPR@FPR=0.01 at 50 generated tokens to be *light*, *moderate* and *heavy* intensity settings respectively.

For each watermark and model, we use 200 prefixes sampled from the C4 corpus (Raffel et al., 2020) as prompts to isolate the δ values corresponding to light, moderate and heavy watermarks for each $\gamma \in \{0.1, 0.25, 0.5, 0.75\}$. Next, we obtain the perplexity values for each (γ, δ) setting corresponding to a particular signal strength over a disjoint sample of 200 C4 snippets. We then choose the tuple which least impacts the model's perplexity scores as the canonical hyperparameter setting for that signal strength. Through this process, we select the pareto-optimal set of hyperparameters with respect to language modeling performance under a target watermark strength. Some contemporary work (Tu et al., 2024) that performs downstream evaluations fails to conduct this type of pareto-optimal hyperparameter search, and instead arbitrarily chooses a (γ, δ) setting that achieves the target signal strength. We believe that this limits the practical applicability of their findings.

Measuring effective utility drop. We work under the assumption that the metrics (e.g., accuracy, F1, BLEU, etc.) that are typically used to evaluate these tasks are representative of human perception

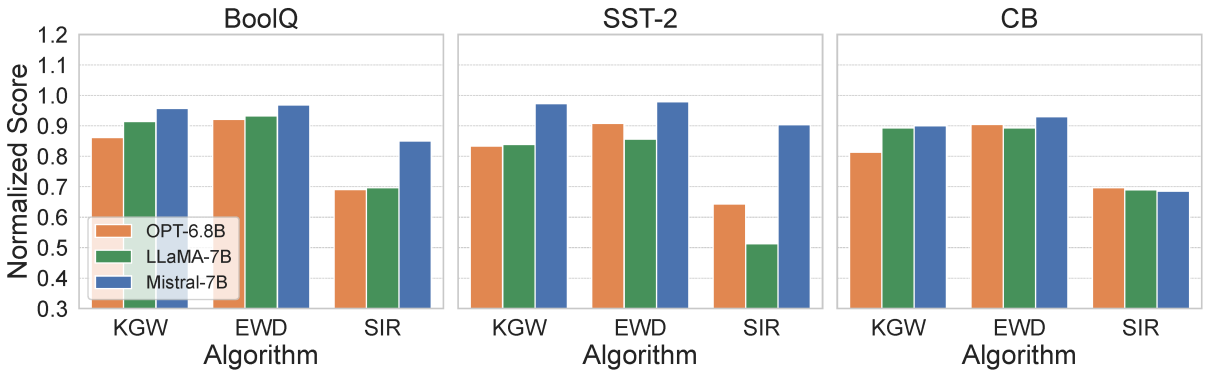


Figure 1: Expected normalized scores for classification datasets. KGW and EWD show 10 - 20% drops in normalized scores while while SIR can cause drops of 30 - 60%.

of performance on them. For CLS and MCQ tasks, even a random classifier achieves a non-zero accuracy in expectation while a random generator achieves negligible F1 or BLEU scores on the generation tasks we study. To measure effective utility, we compute how much the performance metric exceeds that of a random classifier/generator. Specifically, we report normalized scores for each task in section 4. A normalized score of 0 indicates performance equivalent to that of a random classifier/generator and a normalized score of 1 indicates performance equivalent to the corresponding unwatermarked model. Specifically if $\mathcal{M}$ is an unwatermarked model, $\mathcal{M}_{w,i}$ is the corresponding model watermarked using watermark w at signal strength i, if t is a downstream task and if $\mathcal{M}(t)$ denotes model's raw score on the task, then

$$\text{normalized score}(\mathcal{M}, w, i, t) = \frac{\mathcal{M}_{w,i}(t) - \mathcal{R}(t)}{\mathcal{M}(t) - \mathcal{R}(t)}$$

where $\mathcal{R}(t)$ indicates random performance on t. Note that this quantity can be negative if $\mathcal{M}_{w,i}(t) < \mathcal{R}(t) < \mathcal{M}(t)$. It can also exceed 1 if $\mathcal{M}_{w,i}(t) > \mathcal{M}(t)$.

4 Results

We present our main results over all models, tasks and watermarks under the *moderate* signal strength and provide our full set of results in Table 4 in Appendix B.1.

4.1 CLS tasks

When a fixed prompt template is used to prompt LLMs to solve classification tasks (where all test examples share a common label set), a systematic bias over the tokens comprising the label set can arise at the label generation position, and this bias can persist over all test examples. This could be true in the case of KGW and EWD if the terminal tokens in the prompt template (such as `Answer:`) remain fixed and in case of SIR if the semantic embedding of the prompt turns out to be similar for all test examples (e.g., on using fixed instructions and few-shot demonstrations). How specifically the label set tokens get segregated into green list G and red list R however, depends on the choice of the hash key k used with F.

Using the logits of the unwatermarked models and knowledge of the tokens comprising the labels for each CLS task, we compute the expected classification task score under a uniform choice of k for each watermarked model. In Figure 1, we show that expected normalized scores can drop by at least 10–20% in CLS tasks as is the case for all models under the KGW and EWD watermarks. However, these drops can be as high as 30–50% for some models under the SIR watermark. We provide some intuition for this discrepancy in Section 5.

If the label set consists of L tokens for a given task, then there could be $2^{|L|}$ possible partitions of this set into G and R. By enumerating these partitions and evaluating the watermarked model under each of these partitions independently, we isolate the partition that yields the worst test accuracy and plot the corresponding worst-case normalized scores in Figure 2. We see that even watermark signals of moderate strength can destroy effective utility with normalized scores dropping to near zero i.e. akin to random classifier performance in OPT and LLaMA. In tasks such as CB dataset where there is a class imbalance (the minority class, `Neither`, constitutes only 6% of test examples), placing the minority class token into G and the rest into R causes accuracy to fall far below random.

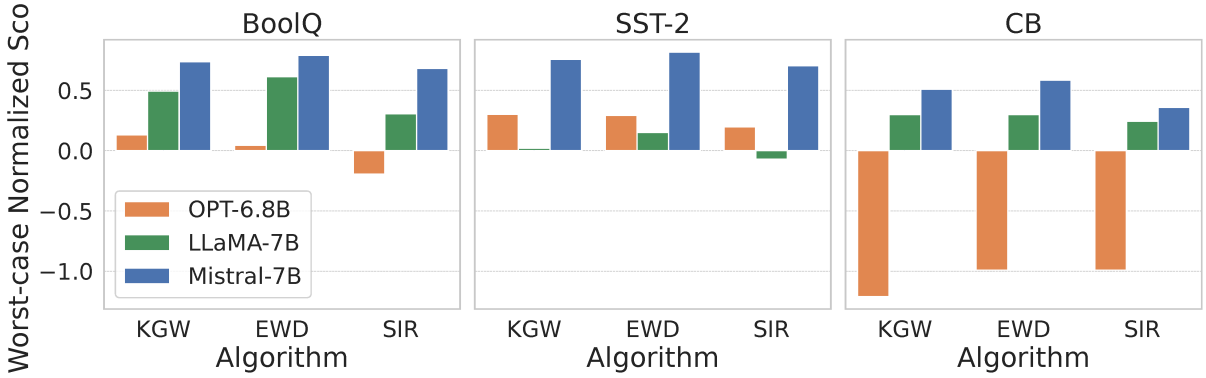
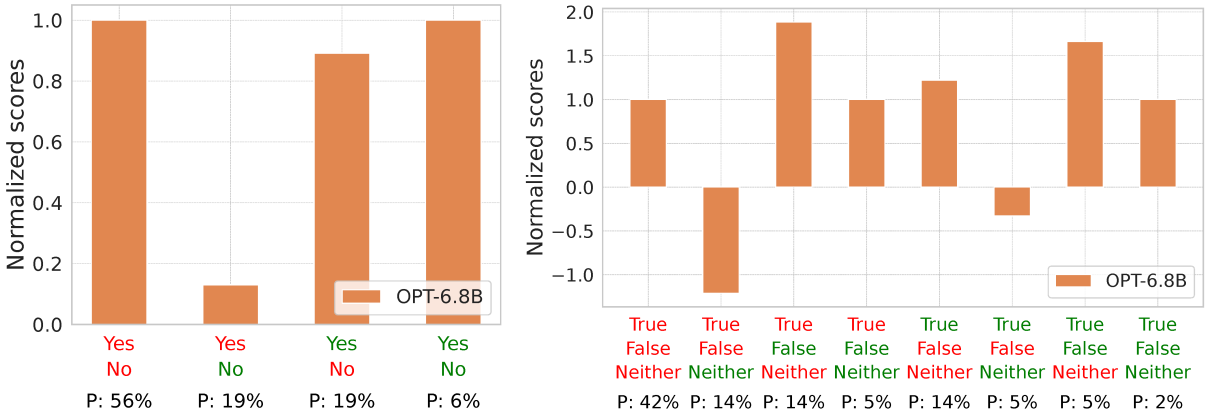


Figure 2: Normalized scores for classification datasets under the worst-case partition. For the 3 watermarking variations, we find that the effective utility of a model can be nearly or completely lost even by moderate watermarks.



(a) BoolQ

(b) CB

Figure 3: OPT 6.8-B's normalized scores on BoolQ and CB under various partitions of the corresponding class labels. Labels are partitioned differently with significant probability. Effective utility of the model can be completely destroyed even under moderate watermarks.

It is crucial to note that these types of worst-case partitions are not rare (under a uniformly random choice of k). In Figure 3, we show the impact of watermarking OPT-6.8B when evaluating over the BoolQ and CB datasets. We show all possible partitions of L into G and R, and the probabilities of such partitions occurring under a randomly chosen hash key. Since these labels are only single token long, $|L| \ll |V|$, we can approximate the probability of a given partition using the binomial distribution as

$$\binom{|L|}{n_G} \cdot \gamma^{n_G} \cdot (1-\gamma)^{|L|-n_G}$$

where $n_G = |L \cap G|$. We find that moderate watermarks can reduce BoolQ accuracy to that of a random classifier with an 19% probability and the normalized score for the CB task to near-zero with a similar probability. Also notice that boosting the logits of the minority class (i.e., `Neither`) without modifying the logits of the remaining two classes causes accuracy to drop far below the random guessing baseline. Imporantly, these observations about worst case performance are consistent over all 3 the KGW-based watermarks we study.

4.2 MCQ tasks

We observe that watermarks leave model performance on MCQ tasks relatively unaffected. We usually see only 5–10% drops in normalized scores (Figure 4) which is markedly lower than those observed in CLS tasks (Figure 1). In fact, Figure 5 shows that the model's preference ranking over all provided choices often remains unchanged by the watermark. We plot the proportion of examples from the HellaSwag task where the model's preference order over its top k most preferred choices remains unchanged upon watermarking. Although these proportions drop (as expected) on increasing

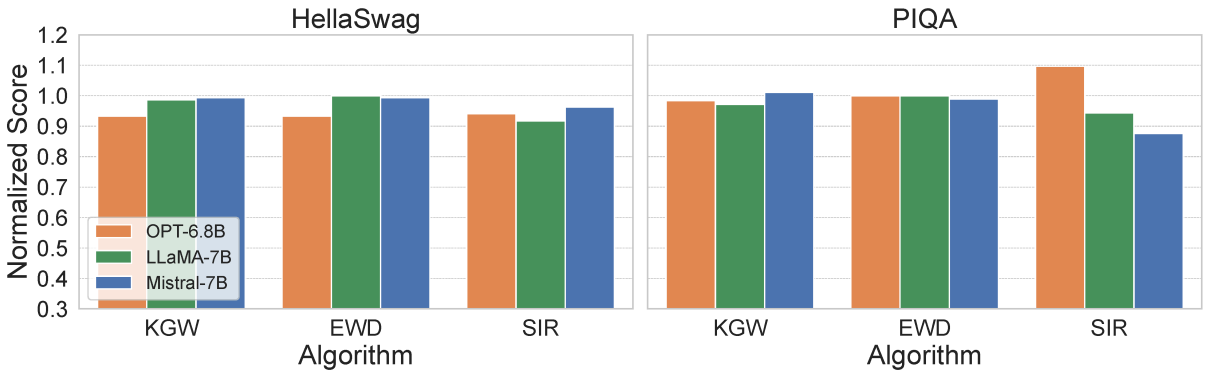


Figure 4: Expected normalized scores for multiple-choice question-answering datasets. These tasks are only mildly affected by watermarks with drops underate moderate watermarks usually restricted to $\leq 10\%$.

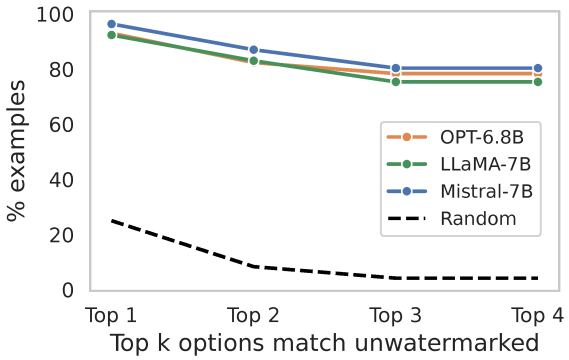


Figure 5: Proportion of HellaSwag examples where the watermark does not change the ranking of the model's top k preferred options. These proportions are consistently higher than expected from a random permutation.

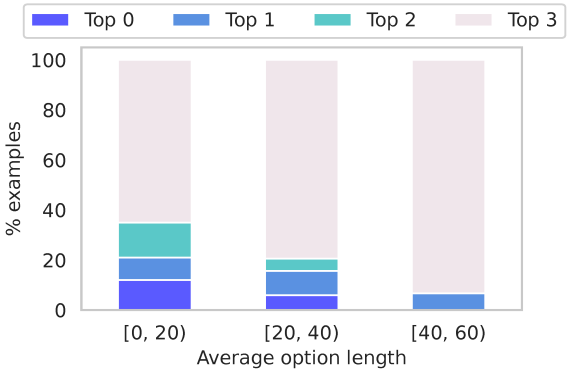


Figure 6: Proportion of HellaSwag examples where the model's top k most preferred choices remains unchanged on watermarking, stratified by average option length. Rankings of examples with longer options are more robust to watermarking.

k, they remain significantly higher than would be expected from a random permutation.

This discrepancy in the score drops we observe between CLS and MCQ tasks could be due to the fact that systematic biases present in CLS tasks with static label sets do not appear in MCQ tasks (where labels differ across examples). Such an effect, however, would diminish on averaging performance over multiple hash keys (as we do). This observation is instead due to the significantly longer choice lengths in HellaSwag and PIQA (averaging 34 and 17 words respectively). In a longer token sequence, the fraction of tokens that occur in G is more likely closely approximated by γ. Hence, each choice sees an almost uniform increase in perplexity on KGW-based watermark application, leaving the ordinal relationship among the the choices' perplexities unchanged. In Figure 6, we bucket HellaSwag examples by their average choice length (in words) and show the proportion of examples in each bucket whose top k most preferred choices (taken together) remains unchanged on watermarking. Notice that the rankings of examples with longer options are more robust.

4.3 *GEN tasks

SGEN Figure 7 shows that models' normalized F1 scores drop by up to 10% across all watermarks in SQuAD2 and by upto 15% on DROP. This impact is noticeable, but milder than might be expected for short generation lengths (considering our previous findings). However, it is unsurprising since in (unwatermarked) LLMs, the logits of the model's predictions at a generation step typically far exceed the logits over most other vocabulary

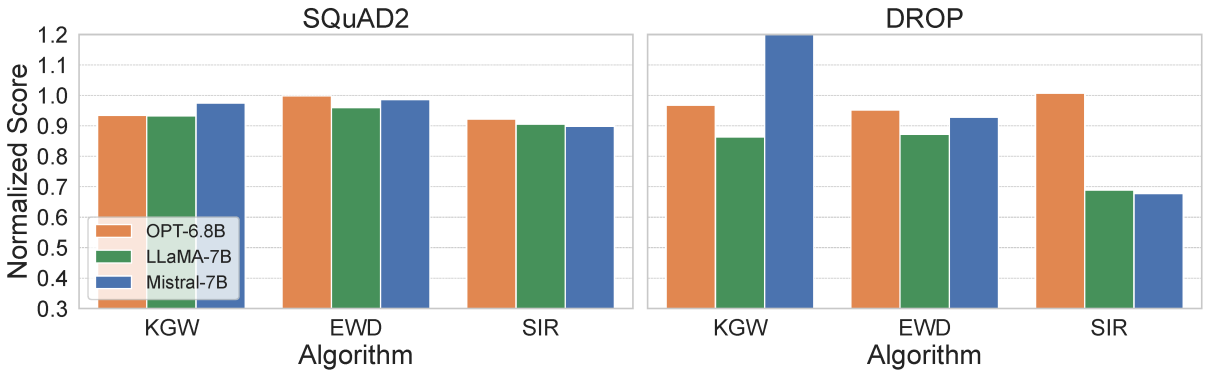


Figure 7: Expected normalized scores for short-form generation (SGEN) tasks.

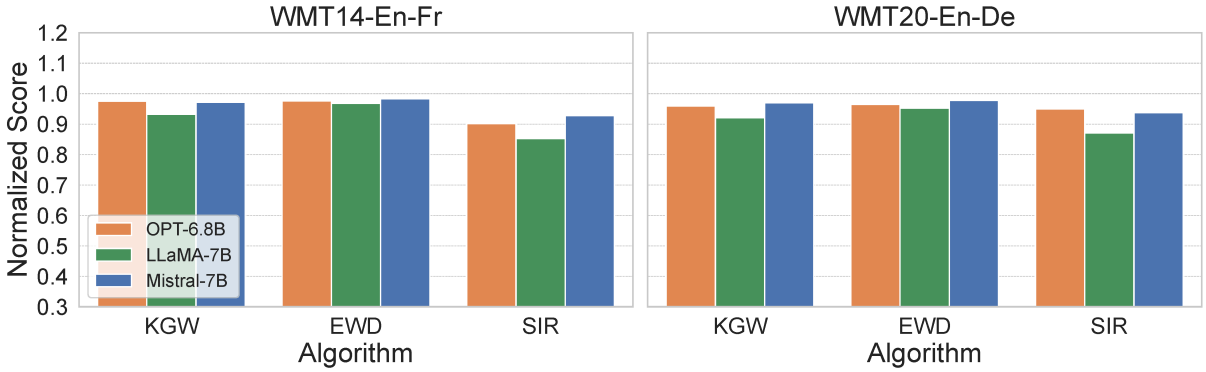


Figure 8: Expected normalized scores for long-form generation datasets. The normalized BLEU scores for both translation tasks remain almost unaffected by moderate KGW-based watermarks, with drops of about $5-10\%$.

tokens (example in Figure 9). Since we use greedy sampling to elicit a model's predictions on SGEN tasks, a watermark would only change these predictions when δ is large enough to boost the logits of arbitrary tokens in G to values larger than those of the original prediction. This appears to only rarely occur in most cases we evaluate. SIR again appears disproportionately impacted, especially in the DROP task.

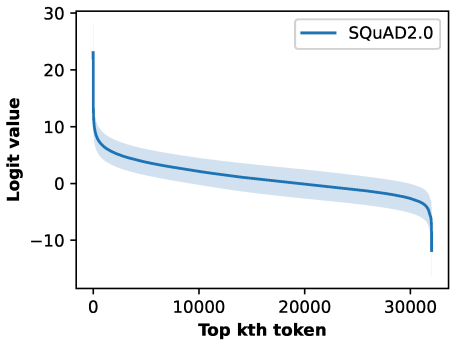


Figure 9: LLaMA 7B's logit-magnitudes for the top-kth token sorted in descending order, averaged over outputs at every generation step for SQuAD2 examples. We observe that at each generation step, a few tokens have significantly higher values than most others.

LGEN The models we study show normalized BLEU score drops of 5–10% under KGW and EWD watermarks, and up to 15% under SIR watermarks. We also observe qualitatively that moderate KGW-based watermarks can lead to occasional factual errors in model generations (examples in Tables 5 and 6 in Appendix B).

5 Analysis

Why does SIR perform worse? In Section 4, we evaluate the performance drops on various different tasks due to applying moderate strength KGW, EWD and SIR watermarks over 3 models, each of about 7 billion parameters. As explained in Section 3, we define the 'moderate' intensity as the watermark strength required to obtain an empirical TPR @ FPR $= 0.01$ of 0.75 using 50-token long freeform generations. Despite this uniformity in calibration, we find that the SIR watermark consistently leads to worse downstream tradeoffs than KGW and EWD over each CLS (Figure 1), MCQ

(Figure 4), SGEN (Figure 7) and LGEN (Figure 8) task we study.

One aspect that makes SIR different from KGW and EWD is that its γ hyperparameter cannot be explicitly set by the user, but instead varies dynamically around 0.5. As Table 3 in the appendix shows however, the γ settings for KGW and EWD we obtain through the calibration procedure described in Section 3 are all either 0.1 or 0.25. Noticing that the likelihood of an arbitrary (independently chosen) pair of tokens being segregated differently into G and R is given by $1-\gamma^2-(1-\gamma)^2$ and that this expression is maximized at $\gamma = 0.5$, we hypothesize that SIR's systematic underperformance may be due to the fact that $\gamma = 0.5$ "maximally" perturbs the model's output distribution. To verify this claim, we evaluate the performance drops due to KGW and EWD when (γ, δ) are set to the values that realize the 'moderate' intensity when $\gamma = 0.5$.

	moderate γ			$\gamma = 0.5$	
Dataset	**KGW**	**EWD**	**SIR**	**KGW**	**EWD**
BoolQ	0.91	0.93	0.70	0.88	0.93
SST2	0.84	0.86	0.51	0.82	0.81
CB	0.89	0.89	0.69	0.85	0.84
HellaSwag	0.99	1.00	0.92	0.97	0.98
PIQA	0.97	1.00	0.94	1.00	0.96
SQuAD2	0.93	0.96	0.91	0.89	0.86
DROP	0.86	0.87	0.69	0.87	0.85
WMT14-En-Fr	0.93	0.97	0.85	0.91	0.93
WMT20-En-De	0.92	0.95	0.87	0.91	0.94
Average	0.92	0.94	0.79	0.90	0.90

Table 1: Normalized scores for LLaMA 7B under the pareto-optimal 'moderate' calibration and under $\gamma = 0.5$. The effective γ for SIR is 0.5 by default.

Although Table 1 shows that the normalized scores under KGW and EWD become somewhat closer to those under SIR when enforcing $\gamma = 0.5$, they still remain significantly higher suggesting that the choice of γ cannot be the only reason for SIR's underperformance.

Effect of model strength. In Table 2, we show the effect of applying KGW-, EWD- and SIR-based watermarks to two stronger models in the LLaMA family: the larger LLaMA 13B and the similarly sized 7B model from the subsequent LLaMA2 generation (Touvron et al., 2023b). Both stronger models show slightly larger normalized scores than LLaMA 7B, suggesting that stronger models may see smaller utility drops upon watermarking.

6 Related Work

Watermarking text. Watermarking discrete-valued text data has classically been considered difficult (Petitcolas et al., 1999; Katzenbeisser and Petitcolas, 1999). Early attempts involved rule-based synonym substitutions and parse-tree modifications (Chiang et al., 2003; Topkara et al., 2006; Venugopal et al., 2011). Seeing that implanting strong watermarks without severely degrading text quality was challenging for these approaches, later work utilized LSTMs (Fang et al., 2017) and masked language-models (Ueoka et al., 2021) for generating watermarked text. The popularity of autoregressive LLMs has spurred fresh interest in text-watermarking techniques. Kirchenbauer et al. (2023a) introduced a method for implanting watermarks into LLM generations by upsampling a subset of tokens during the decoding phase. This has inspired much followup work to make LLM-watermarks robust to paraphrase attacks (Kirchenbauer et al., 2023b; Hou et al., 2024; Ren et al., 2024; Liu et al., 2024; Zhao et al., 2024), encode multibit information (Yoo et al., 2024; Qu et al., 2024; Wang et al., 2024), distill watermarks into standalone language models (Gu et al., 2024), and reduce the degree of watermark-induced text degradation (Wu et al., 2024; Takezawa et al., 2024; Lu et al., 2024; Chen et al., 2024). Although these works, collectively called the KGW family of watermarks are by far the most popular LLM watermarks used today, there also exist other cryptographically-inspired watermarking schemes (Christ et al., 2023; Aaronson and Kirchner, 2023; Kuditipudi et al., 2024).

Downstream effects of watermarking. Most prior work on watermarking has evaluated their resulting models using perplexity of the generated text. Kirchenbauer et al. (2023a) evaluates the performance of watermarked models on a single question -answering task. Some follow-up work (Fernandez et al., 2023) conducts small-scale evaluations but does not attempt to uncover the causes for observed performance drops. One contemporaneous work (Tu et al., 2024) performs a similar study to ours over a broad range of tasks but in contrast to our work, chooses watermark hyperparameters relatively arbitrarily (see Section 3) which we believe limits the practical applicability of their findings. To the best of our knowledge, our study is the first to conduct a principled analysis on the downstream

Dataset	LLaMA 7B			LLaMA 13B			LLaMA2 7B		
	KGW	EWD	SIR	KGW	EWD	SIR	KGW	EWD	SIR
BoolQ	0.91	0.93	0.70	0.96	0.97	0.87	0.96	0.98	0.90
SST2	0.84	0.86	0.51	0.94	0.97	0.84	0.93	0.93	0.79
CB	0.89	0.89	0.69	0.92	0.93	0.77	0.97	0.97	0.92
HellaSwag	0.99	1.00	0.92	1.02	0.99	0.95	1.00	1.00	0.97
PIQA	0.97	1.00	0.94	0.98	0.99	0.97	0.96	1.00	1.00
SQuAD2	0.93	0.96	0.91	0.91	0.95	0.89	0.87	0.97	0.83
DROP	0.86	0.87	0.69	0.81	0.72	0.58	0.79	0.92	0.69
WMT14-En-Fr	0.93	0.97	0.85	0.94	0.97	0.90	0.95	0.97	0.92
WMT20-En-De	0.92	0.95	0.87	0.95	0.98	0.93	0.91	0.99	0.94
Average	0.92	0.94	0.79	0.94	0.94	0.86	0.93	0.97	0.88

Table 2: Normalized scores of LLaMA 7B, LLaMA 13B and LLaMA2 7B models with KGW, EWD and SIR watermarks, along with the average scores (in the last row). Normalized scores appear slightly larger for LLaMA 13B and LLaMA2 7B (compared to LLaMA 7B), suggesting that stronger models do not see as much utility drop.

effects of watermarking schemes over a broader spectrum of tasks, shedding light on underlying reasons for the observed trade-offs.

7 Conclusion

We evaluate the extent to which watermarks from the KGW family hurt downstream performance by examining three watermark and three LLMs over a diverse suite of NLP tasks. We motivate a categorization of tasks into 4 buckets and analyze causes for the observed trade-offs in each category.

We find the performance trade-offs for each category vary in a manner that simple perplexity measurements cannot capture or predict (an assumption implicit in prior work). Watermarks, under realistic hyperparameters, can cause significant drops in LLMs' effective utility across all tasks. We observe drops of 10 to 20% in CLS tasks in the average case, which shoot up to 100% in the worst case. We notice degradations of about 7% in MCQ tasks, 10–15% in short-form generation, and 5–15% in long-form generation tasks. We also find some evidence that the downstream trade-offs posed by the KGW family of watermarks may diminish with increasing model strength.

We believe that our work will (i) allow developers and practitioners to make informed choices about watermarked LLMs and their adaptations, (ii) spur research into novel watermarking strategies that present better trade-offs, and (iii) inspire techniques for maintaining model performance under existing watermarking schemes.

Limitations

We restrict our analysis in this work to empirically evaluating the downstream performance of three representatives from the KGW family of watermarks. While we perform some analyses and give some theoretical intuitions, it may be possible to establish a concrete theoretical framework for (KGW-based) watermarked models' downstream trade-offs. We leave such analyses to future work.

Although our findings likely transfer to most KGW-based watermarks, unrelated schemes such as Aaronson and Kirchner (2023) and Christ et al. (2023), lie outside the scope of our work.

We only analyze the effect of watermarking under the typical decoding strategies used for each of the tasks. It is plausible that not all decoding strategies would be similarly affected by KGW-based watermarks. There may also exist watermark-aware decoding strategies designed to mitigate performance drops. This possibility presents an exciting avenue for future work.

Acknowledgments

We are grateful to reviewers for their constructive feedback. We also thank Navreet Kaur, Shashwat Singh, Saksham Rastogi, Nihar B. Shah, and Priyanka Agrawal for their insights and feedback for this work. DP acknowledges Adobe Inc., Google Research, Schmidt Sciences, National Payments Corporations of India (NPCI) and Pratiksha Trust for supporting his group's research.

References

Scott Aaronson and Hendrik Kirchner. 2023. Watermarking GPT Outputs.

Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina

Mohamed, and Sanket Naik. 2001. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding: 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings 4*, pages 185–200. Springer.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ale s Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Liang Chen, Yatao Bian, Yang Deng, Deng Cai, Shuaiyi Li, Peilin Zhao, and Kam-Fai Wong. 2024. WatME: Towards lossless watermarking through lexical redundancy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9166–9180, Bangkok, Thailand. Association for Computational Linguistics.

Yuei-Lin Chiang, Lu-Ping Chang, Wen-Tai Hsieh, and Wen-Chih Chen. 2003. Natural language watermarking using semantic substitution for chinese text. In *International Workshop on Digital Watermarking*.

Yuei-Lin Chiang, Lu-Ping Chang, Wen-Tai Hsieh, and Wen-Chih Chen. 2004. Natural language watermarking using semantic substitution for chinese text. In *Digital Watermarking: Second International Workshop, IWDW 2003, Seoul, Korea, October 20-22, 2003. Revised Papers 2*, pages 129–140. Springer.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Miranda Christ, Sam Gunn, and Or Zamir. 2023. Undetectable watermarks for language models.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proc. of NAACL*.

Tina Fang, Martin Jaggi, and Katerina Argyraki. 2017. Generating steganographic text with LSTMs. In *Proceedings of ACL 2017, Student Research Workshop*, pages 100–106, Vancouver, Canada. Association for Computational Linguistics.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. 2023. Three bricks to consolidate watermarks for large language models.

Dijana Vukovic Grbic and Igor Dujlovic. 2023. Social engineering with chatgpt. In *2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5. IEEE.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. 2024. On the learnability of watermarks for language models. In *The Twelfth International Conference on Learning Representations*.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2024. SemStamp: A semantic watermark with paraphrastic robustness for text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4067–4082, Mexico City, Mexico. Association for Computational Linguistics.

Zunera Jalil and Anwar M Mirza. 2009. A review of digital watermarking techniques for text documents. In *2009 International Conference on Information and Multimedia Technology*, pages 230–234. IEEE.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Stephan Katzenbeisser and Fabien Petitcolas. 1999. *Information Hiding Techniques for Steganography and Digital Watermaking*, volume 28.

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.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2024. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11724–11735, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI. 2023. New ai classifier for indicating ai-written text.

Natalie O'Neill. 2023. Texas professor flunked whole class after chatgpt wrongly claimed it wrote their papers.

Mekela Panditharatne and Noah Giansiracusa. 2023. How ai puts elections at risk — and the needed safeguards. *Ms. Magazine*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. 1999. Information hiding-a survey. *Proc. IEEE*, 87:1062–1078.

WH Press. 2023. Biden harris administration secures voluntary commitments from leading artificial intelligence companies to manage the risks posed by ai. *White House Press Statement*.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. 2024. Provably robust multi-bit watermarking for ai-generated text.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. 2024. A robust semantics-based watermark for large language model against paraphrasing. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 613–625, Mexico City, Mexico. Association for Computational Linguistics.

Karanpartap Singh and James Zou. 2023. New evaluation metrics capture quality degradation due to llm watermarking.

Yuki Takezawa, Ryoma Sato, Han Bao, Kenta Niwa, and Makoto Yamada. 2024. Necessary and sufficient watermark for large language models.

Mercan Topkara, Giuseppe Riccardi, Dilek Hakkani-Tür, and Mikhail J Atallah. 2006. Natural language watermarking: Challenges in building a practical system. In *Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 106–117. SPIE.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. 2024. WaterBench: Towards holistic evaluation of watermarks for large language models. In *Proceedings of the 62nd Annual Meeting*

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1517–1542, Bangkok, Thailand. Association for Computational Linguistics.

Honai Ueoka, Yugo Murawaki, and Sadao Kurohashi. 2021. Frustratingly easy edit-based linguistic steganography with a masked language model. *ArXiv*, abs/2104.09833.

Ashish Venugopal, Jakob Uszkoreit, David Talbot, Franz Josef Och, and Juri Ganitkevitch. 2011. Watermarking the outputs of structured prediction with an application in statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. 2024. Towards codable watermarking for injecting multi-bits information to LLMs. In *The Twelfth International Conference on Learning Representations*.

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. 2024. Dipmark: A stealthy, efficient and resilient watermark for large language models.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. 2024. Advancing beyond identification: Multi-bit watermark for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4031–4055, Mexico City, Mexico. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. 2024. Provable robust watermarking for AI-generated text. In *The Twelfth International Conference on Learning Representations*.

A Watermark Hyperparameters

We show the watermark hyperparameters we use for our main experiments (arrived at through our calibration procedure described in Section 3) in Table 3.

	KGW		EWD		SIR	
Model	γ	δ	γ	δ	γ	δ
OPT-6.8B	0.25	1.21	0.1	1.41	0.5	0.86
LLaMA-7B	0.1	2.13	0.1	1.72	0.5	1.31
Mistral-7B	0.1	1.8	0.1	1.5	0.5	1.02

Table 3: We show the (γ, δ) hyperparameters obtained through the calibration procedure described in Section 3 for each model and moderate watermark. Note that the effective γ values for SIR are determined dynamically at each generation step, but we empirically verify that they always take values very close to 0.5.

B Additional Results

B.1 Results for Light, Moderate and Heavy settings

We provide evaluation results for each task, model and watermark we study in Table 4.

B.2 LGEN Examples

We present representative examples for LGEN tasks. Examples from the WMT14-En-Fr task are tabulated in Table 5 and examples from WMT20-En-De are tabulated in Table 6.

C Task Evaluation Details

C.1 Decoding

1. **CLS** tasks are tasks framed as k-class classification problems with static (and often short) labels that are common across all test examples. These are evaluated by picking the class label that the model assigns the highest probability to. Formally, the input text $\mathbf{x}$ is formatted using a suitable prompt template T and the class $\mathbf{y}$ which maximizes $p(\mathbf{y}|T(\mathbf{x}))$ is chosen as the model's prediction.

$$\hat{y} = \arg\max_{\mathbf{y} \in L} p(\mathbf{y}|T(\mathbf{x}))$$

   These tasks are also typically evaluated using accuracy metrics.

2. The **MCQ** category includes several open-book question-answering, reading-comprehension and common-sense reasoning tasks that are posed as multiple-choice question-answering tasks to language models. In these tasks, every test input $\mathbf{x}$ is associated with a set of possible answer choices $L(\mathbf{x})$. When a test input $\mathbf{x}$ is formatted using a suitable template T, the answer choice that the language model assigns the highest average log likelihood to,

$$\arg\max_{\mathbf{y} \in L(\mathbf{x})} \text{avg-log-likelihood}(\mathbf{y}|T(\mathbf{x}))$$

   is chosen as the model's prediction. These tasks are also typically evaluated using accuracy metrics.

3. **SGEN** includes open-domain question-answering and reading-comprehension tasks are posed to language models as short-form conditional generation tasks and require models to output concise free-form responses. Given a test input $\mathbf{x}$ formatted using a prompt template T, the model produces a sequence $\mathbf{y}^*$ which maximizes the conditional likelihood $p(\mathbf{y}|T(\mathbf{x}))$.

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|T(\mathbf{x}))$$

   Typically, the generated sequence is bounded by a certain length or concludes when the model outputs an end-of-sequence token. The generated sequences are typically evaluated against gold sequences using F1 scores.

4. **LGEN** represents all long-form generation tasks including machine translation and summarization. For an input text $\mathbf{x}$, formatted using an appropriate prompt template T, the model is tasked with producing an extended sequence $\mathbf{y}^*$ that maximizes the conditional likelihood $p(\mathbf{y}|T(\mathbf{x}))$.

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|T(\mathbf{x}))$$

   The generated sequences are often evaluated against multiple ground truth references using metrics such as ROUGE and BLEU, allowing for some flexibility for paraphrases.

Model	Watermark	Intensity	BoolQ	SST2	CB	HellaSwag	PIQA	SQuADv2	DROP	WMT14-En-Fr	WMT20-En-De
OPT-6.8B	KGW	light	0.92	0.91	1.11	0.97	0.98	0.93	0.98	0.98	0.99
		moderate	0.86	0.83	0.81	0.93	0.98	0.93	0.97	0.98	0.96
		heavy	0.85	0.83	0.80	0.90	0.94	0.81	0.96	0.89	0.88
	EWD	light	1.15	0.91	0.96	1.12	1.06	1.00	0.97	1.01	0.97
		moderate	0.92	0.91	0.90	0.93	1.00	1.00	0.95	0.98	0.96
		heavy	1.04	0.83	0.83	1.08	0.94	0.90	0.87	0.95	0.91
	SIR	light	0.86	0.85	1.06	0.96	1.06	0.96	0.98	0.98	0.98
		moderate	0.69	0.64	0.70	0.94	1.10	0.92	1.01	0.90	0.95
		heavy	0.50	0.50	0.56	0.68	0.82	0.22	0.65	0.42	0.63
LLaMA-7B	KGW	light	0.93	0.86	0.89	1.00	0.97	0.95	0.88	0.96	0.93
		moderate	0.91	0.84	0.89	0.99	0.97	0.93	0.86	0.93	0.92
		heavy	0.86	0.82	0.89	0.99	0.97	0.84	0.78	0.83	0.76
	EWD	light	0.92	0.81	0.90	0.99	1.04	0.97	0.94	0.97	0.98
		moderate	0.93	0.86	0.89	1.00	1.00	0.96	0.87	0.97	0.95
		heavy	0.84	0.73	0.89	0.97	0.97	0.81	0.78	0.89	0.90
	SIR	light	0.91	0.75	0.73	0.97	1.06	0.91	0.86	0.96	0.99
		moderate	0.70	0.51	0.69	0.92	0.94	0.91	0.69	0.85	0.87
		heavy	0.50	0.50	0.48	0.26	0.42	0.37	0.37	0.40	0.46
Mistral-7B	KGW	light	0.97	0.98	0.94	1.01	1.01	1.01	1.26	0.99	0.98
		moderate	0.96	0.97	0.90	0.99	1.01	0.97	1.29	0.97	0.97
		heavy	0.92	0.95	0.83	0.98	0.97	0.96	1.22	0.93	0.92
	EWD	light	0.93	0.97	0.94	1.06	0.98	0.97	0.94	0.99	0.99
		moderate	0.97	0.98	0.93	0.99	0.99	0.99	0.93	0.98	0.98
		heavy	0.90	0.95	0.90	1.03	0.98	0.98	0.81	0.96	0.96
	SIR	light	0.97	0.98	0.92	1.02	0.97	0.96	0.89	0.99	1.00
		moderate	0.85	0.90	0.69	0.96	0.88	0.90	0.68	0.93	0.94
		heavy	0.50	0.50	0.45	0.30	0.37	0.44	0.24	0.30	0.37

Table 4: Normalized scores for all Model, Watermark, Intensity, Dataset combinations we evaluate.

Model		Content
	Input	Repainted Traffic Lights: MP Asks Brigitte Grouwels to Resign
	Ground Truth	Feux repeints: un député demande la démission de Brigitte Grouwels *[Repainted lights: a deputy calls for the resignation of Brigitte Grouwels]*
LLaMA 7B	Output	Les feux de circulation ont été repeints: le député a demandé à Brigitte Grouweels de démissionner. *[The traffic lights have been repainted: the MP asked Brigitte Grouweels to resign.]*
+ moderate KGW	Output	Les feux de circulation sont rénovés, mais c'est la ministre qui se trouve sur le gril, demande à être relevée de ses fonctions. *[The traffic lights are **renovated, but it is the minister who is on the grill, asking to be relieved of her duties.**]*
	Input	This really is a must for our nation.
	Ground Truth	Cela est vraiment indispensable pour notre nation. *[This is truly essential for our nation]*
Mistral 7B	Output	Cela est vraiment indispensable pour notre nation. *[This is truly essential for our nation]*
+ moderate SIR	Output	C'est une chose qui nous manque réellement. ***[This is something we really miss.]***

Table 5: Moderate watermarks can cause increase factual errors in model outputs in the WMT14-En-Fr task. We provide the input, the gold output, the unwatermarked model's output and the output generated by a model with a moderate watermark. We also provide the English back-translation (from Google Translate) for each of these.

Model		Content
	Input	He is now facing 10 counts of murder in Canada and 16 counts of attempted murder following the attack on April 23, 2018.
	Ground Truth	Nach dem Attentat vom 23. April 2018 wurden in Kanada nicht weniger als 10 Anklagepunkte wegen Mordes und 16 Anklagepunkte wegen versuchten Mordes gegen ihn erhoben. *[Following the assassination attempt on **April 23, 2018**, no fewer than 10 charges of murder and 16 charges of attempted murder were brought against him in Canada.]*
OPT 6.8B	Output	Der Mann ist jetzt auf 10 Mordverdachtsfälle in Kanada und 16 Mordverdachtungen gegenüber der Tat am 23. April 2018. *[The man is now facing 10 counts of murder in Canada and 16 counts of murder for the crime on April 23, 2018.]*
+ moderate EWD	Output	*[The man is now convicted of 10 counts of murder and 16 attempted murders in Canada.]*
	Input	However, despite fans being eager for the upcoming release, it seems we all need to sit tight as it won't be dropping on our screens until 2021.
	Ground Truth	Selbst wenn seine Fans die bevorstehende Ausstrahlung der Serie kaum erwarten können, bleibt Geduld angesagt, da diese erst für 2021 angekündigt wurde. *[Even if his fans can hardly wait for the upcoming broadcast of the series, patience remains as it has only been announced for 2021.]*
Mistral 7B	Output	Trotzdem, obwohl Fans eifrig auf die kommende Veröffentlichung warten, scheint es so, als würden wir alle auf die Sitze sitzen müssen, bis es 2011 auf unseren Bildschirmen erscheint. *[Nevertheless, although fans are eagerly awaiting the upcoming release, it seems we will all have to sit on the edge of our seats until it hits our screens in 2011.]*
+ moderate KGW	Output	Jedoch, obwohl Fans einschließlich mich ehrgeizig auf die nächste Veröffentlichung warten, müssen wir alle auf die nächstjährige Veröffentlichungsdatum warten. *[However, although fans including myself are eagerly waiting for the next release, we all have to wait for **next year's** release date.]*

Table 6: Moderate watermarks can cause increase factual errors in model outputs in the WMT20-En-De task. We provide the input, the gold output, the unwatermarked model's output and the output generated by a model with a moderate watermark. We also provide the English back-translation (from Google Translate) for each of these.