Survey: Evaluating LLM Watermarking Robustness Across Model Scales Using MarkLLM

Zahran Yahia Khan **Sofia Cobo Navas**
University of South Florida

Abstract

Watermarking large language models (LLMs) embeds detectable statistical signals into generated text, enabling attribution and misuse detection. While several algorithms have been proposed, their behavior across different model scales remains poorly understood. We use the MarkLLM framework (Pan et al., 2024) to benchmark three algorithms KGW (Kirchenbauer et al., 2023), Unigram (Zhao et al., 2024), and EXP (Aaronson and Kirchner, 2022) across three OPT model sizes: 125M, 1.3B, 2.7B, two datasets: C4 and WMT16, and three adversarial attacks: word deletion, synonym substitution, and GPT-3.5 paraphrasing. Our central finding is that the low entropy scale floor problem is *algorithm-specific*: KGW fails on OPT-125M (TPR @ 10% FPR = 0.09), while Unigram and EXP achieve almost perfect detection even at that scale. Above the floor, detectability remains stable between 1.3B and 2.7B. GPT 3.5 paraphrasing is by far the most damaging attack since the average TPR drop of 57.6 pp vs. 6.2 pp for word deletion, but Unigram is more paraphrase resistant and imposes the lowest perplexity cost, suggesting a practical advantage for deployment under adversarial conditions.

1 Introduction

The widespread adoption of large language models has made it easier to generate fluent, convincing text at scale. While this unlocks valuable applications, it also creates serious social risks like AI generated misinformation, academic plagiarism, and a drop in trust in written content which are growing concerns that post-hoc classifiers are poorly equipped to address (Gehrmann et al., 2019; Mitchell et al., 2023). Watermarking offers a structural alternative. Rather than detecting AI generated text after the fact, watermarking techniques embed a detectable statistical signal directly into the generation process of a model. These signals can later be identified using detection algorithms, enabling distinguishing model generated content.

Recent work has proposed several watermarking approaches with different design philosophies, like token biasing methods (Kirchenbauer et al., 2023; Zhao et al., 2024) and sampling based strategies (Aaronson and Kirchner, 2022). However, most evaluations focus on a single model architecture or size. This leaves an important practical question unanswered: *does watermarking effectiveness hold up as model scale changes, and does the answer differ by algorithm?* A watermark that works on a large model may fail on a smaller one, and deployment decisions depend on understanding this relationship.

In this survey study, we use MarkLLM (Pan et al., 2024), an open source watermarking toolkit, to conduct a systematic benchmark addressing three research questions:

- **RQ1:** Does detectability depend on model scale, and is the dependency algorithm specific?
- **RQ2:** Does robustness to adversarial attacks change with scale or algorithm choice?
- **RQ3:** What is the text quality cost of watermarking across scales and algorithms?

Our key finding is that the smallest model we tested, OPT-125M, is too small for KGW to embed a detectable watermark but this failure is specific to KGW's context dependent logit biasing mechanism and does not affect Unigram or EXP, which both achieve near perfect detection even at that scale. For models above this minimum viable size, increasing capacity further provides no benefit: OPT-1.3B and OPT-2.7B produce virtually identical detection rates across all three algorithms. Unigram emerges as the strongest practical choice, most paraphrase resistant and lowest perplexity cost, despite being the simplest algorithm.

To our knowledge, no prior work has systematically compared watermarking effectiveness across model scales for multiple algorithms; our study provides the first such cross scale, cross algorithm benchmark.

2 Background and Related Work

MarkLLM. MarkLLM (Pan et al., 2024) is an open source Python toolkit providing a unified framework for implementing, visualizing, and evaluating LLM watermarking algorithms. It implements nine algorithms from two families: KGW-family and Christ-family, provides evaluation scripts for detectability, robustness, and text quality, and offers mechanism visualization tools. We use MarkLLM as both the generation and evaluation infrastructure for all experiments in this study.

KGW Watermark. The KGW method (Kirchenbauer et al., 2023) partitions the vocabulary into a green list and a red list at each generation step, using the previous token as a hash key. A logit bias δ is added to all green list tokens during generation, increasing their selection probability. Detection uses a one sided z-test on the proportion of green tokens in the output text. The original paper notes that watermark embedding may fail at low entropy positions where the model's distribution is sharply peaked, but does not systematically evaluate this failure across model scales.

Unigram Watermark. Unigram (Zhao et al., 2024) simplifies KGW by using a single fixed global green list rather than regenerating it at each step. This eliminates prefix dependence, making the watermark statistic depend only on the marginal token distribution. It is computationally cheaper and, as our results show, structurally more robust to attacks that disrupt token sequence structure.

EXP Watermark. The EXP method (Aaronson and Kirchner, 2022) belongs to a family of sampling based watermarks. Rather than biasing logits, it uses a pseudorandom key derived from the recent token context to reseed the sampling distribution at each step. Detection correlates observed token choices against the expected distribution under the pseudorandom key. EXP uses an inverted score polarity (lower p-value = watermarked), which we handle explicitly in our detection function.

Related Work. Prior watermarking evaluations have primarily benchmarked algorithms on single model sizes. Kirchenbauer et al. (2023) evaluate KGW on OPT-6.7B; Zhao et al. (2024) evaluate Unigram on LLaMA-7B; Pan et al. (2024) provide cross-algorithm comparisons on OPT-1.3B. Several studies have examined robustness to paraphrasing (Krishna et al., 2023; Sadasivan et al., 2023), but these likewise hold model size constant. The question of how watermarking behavior changes across model scales and whether that change is algorithm dependent has not been addressed. Our study fills this gap by evaluating three algorithms across three OPT sizes, providing the first multi-scale, multi-algorithm watermarking benchmark.

3 Experimental Setup

3.1 Algorithms, Datasets, and Models

We evaluate KGW, Unigram, and EXP as implemented in MarkLLM. These were selected to represent different design philosophies context-dependent logit-biasing, fixed logit-biasing, and sampling-based, with no additional model training or fine-tuning required.

We evaluate on two datasets included in the MarkLLM repository. **C4** (Raffel et al., 2020) is a large scale corpus derived from Common Crawl web text, providing diverse general domain content and serving as the standard benchmark in prior watermarking evaluations. **WMT16 (en-de)** (Bojar et al., 2016) provides formally edited, structured English source sentences from English German MT benchmarks, offering a different vocabulary distribution and lower lexical diversity than C4. For each dataset, the first 30 tokens of each document serve as the generation prompt, with up to 200 new tokens generated.

To study scale effects we evaluate three OPT model sizes (Zhang et al., 2022): **OPT-125M** (small baseline to explore the lower-end scale regime), **OPT-1.3B** (primary experimental model, widely used in prior watermarking research), and **OPT-2.7B** (an additional larger scale point above 1.3B). All models are loaded in FP16 on a T4 GPU; no quantization was required. Generation parameters are `max_new_tokens=200`, `temperature=0.7`, `top_k=50`, `top_p=0.9`.

Iterative design. The original experimental plan included only OPT-125M and OPT-1.3B 12 combinations. During the clean detectability phase, KGW on OPT-125M produced almost zero detection rates across both datasets, indicating a genuine embedding failure rather than a pipeline error. To

obtain a three point scale comparison and verify whether the failure is a sharp cutoff or a gradual transition, we added OPT 2.7B as a midpoint, expanding the experiment to 18 combos and 3,600 total generated texts. OPT-125M was retained in detectability and quality experiments because its failure is itself evidence for RQ1. The OPT-2.7B results were generated under an updated environment configuration (see Section 6) and validated as compatible with earlier outputs via fixed seed equivalence testing that confirmed byte identical generation and detection scores identical to 15 decimal places.

3.2 Evaluation Protocol

For each of the 18 combinations (3 algorithms $\times$ 3 models $\times$ 2 datasets), we generate 100 watermarked and 100 non-watermarked texts 3,600 in total. Detection metrics are computed using a custom implementation that explicitly handles EXP's inverted score polarity (lower p-value = watermarked, opposite to KGW and Unigram's z-score direction). The implementation was cross validated against MarkLLM's `DynamicThresholdSuccessRateCalculator` to numerical precision on the KGW+OPT-1.3B+C4 control combo (TPR = 0.99, F1 = 0.9519, max absolute difference = 0.0). We report TPR at 10% FPR (primary operating-point metric) and F1 at the threshold maximizing F1.

3.3 Robustness Attacks

Robustness experiments are conducted on OPT-1.3B and OPT-2.7B only (12 combos). OPT-125M was excluded to maintain a consistent cross-algorithm comparison matrix: since KGW's baseline detection on OPT-125M is at chance level (TPR $\leq$ 0.16), degradation under attack cannot be meaningfully measured for that algorithm. Although Unigram and EXP embed reliably on OPT-125M, including them without KGW would make cross-algorithm robustness comparisons incomplete. Evaluating Unigram and EXP robustness on OPT-125M remains an interesting direction for future work.

Three attacks are applied: **Word Deletion** (ratio = 0.3) randomly removes 30% of words via MarkLLM's `WordDeletion`; **Synonym Substitution** (ratio = 0.5) replaces 50% of tokens using context-aware BERT-large (Devlin et al., 2019) masked LM via `ContextAwareSynonymSubstitution`; and **GPT-3.5 Paraphrase** (OpenAI, 2022) rewrites texts via the OpenAI Batch API with unconstrained output length, matching MarkLLM's published configuration. Robustness is measured as the TPR drop at 10% FPR relative to the clean baseline.

Two methodological caveats apply to the attack configurations. Context-aware synonym substitution may alter factual content when BERT proposes contextually fluent but semantically different replacements (e.g., "current fiscal year" $\rightarrow$ "2013 fiscal year"). We follow MarkLLM's published attack configuration without modification, matching prior evaluations. GPT-3.5 paraphrase output length was unconstrained; some outputs were severe compressions of the original rather than length-preserving rewrites. Zero API errors and zero empty responses were observed across 1,200 requests. This matches MarkLLM's published configuration and represents a realistic adversarial scenario where the attacker accepts whatever the paraphrasing model produces.

3.4 Text Quality Metrics

Text quality is evaluated on all 18 combos, including OPT-125M—the question is whether watermarking *attempts* hurt quality, not whether they succeed, and a failed-to-embed watermark may still distort the output distribution. **Perplexity (PPL)** is scored using GPT-2 (Radford et al., 2019) as a neutral external evaluator, following MarkLLM's protocol. PPL is reported for C4 only; WMT16 PPL values are omitted because GPT-2 (an English-trained model) assigns inflated perplexity to WMT16 continuations due to cross-lingual token distribution mismatch rather than watermark-induced quality degradation, conflating language-model behavior with watermark cost. **Log Diversity** is computed using MarkLLM's `LogDiversityAnalyzer`, measuring the entropy of the n-gram distribution (n = 2,3,4 pooled across all 100 texts per combo); lower values indicate more repetitive vocabulary. One caveat: OPT-125M log diversity values are visibly inflated relative to larger models, particularly on WMT16. Diagnostic inspection showed OPT-125M produces partially degenerate text on cross-lingual prompts (verbatim prompt repetition followed by topic-drifting sentences), and n-gram diversity metrics reward this surface novelty without measuring quality. Cross-scale diversity comparisons should therefore focus on the 1.3B–2.7B contrast.

Table 1: Detectability. TPR and F1 at 10% FPR and best-threshold. Red = failed embedding; green = near-perfect; orange = partial.

Algo	Model	Data	@ 10% FPR		@ Best	
			TPR	F1	TPR	F1
KGW	125M	C4	0.09	0.15	0.99	0.66
KGW	125M	WMT16	0.16	0.25	0.98	0.67
KGW	1.3B	C4	0.99	0.95	0.99	0.99
KGW	1.3B	WMT16	0.96	0.93	0.96	0.96
KGW	2.7B	C4	0.99	0.95	0.96	0.97
KGW	2.7B	WMT16	1.00	0.95	1.00	1.00
Unig.	125M	C4	1.00	0.95	0.99	0.99
Unig.	125M	WMT16	0.95	0.93	0.95	0.96
Unig.	1.3B	C4	1.00	0.95	1.00	1.00
Unig.	1.3B	WMT16	1.00	0.95	0.99	0.99
Unig.	2.7B	C4	1.00	0.95	0.99	0.99
Unig.	2.7B	WMT16	1.00	0.95	0.99	0.99
EXP	125M	C4	1.00	0.95	1.00	1.00
EXP	125M	WMT16	0.96	0.93	0.96	0.98
EXP	1.3B	C4	0.98	0.94	0.97	0.98
EXP	1.3B	WMT16	0.99	0.95	0.99	0.99
EXP	2.7B	C4	0.99	0.95	0.99	0.99
EXP	2.7B	WMT16	1.00	0.95	1.00	1.00

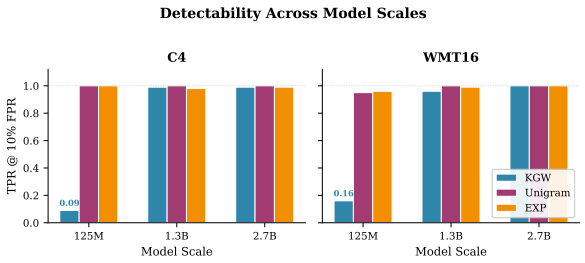

Figure 1: Detectability (TPR @ 10% FPR) across model scales. KGW fails on OPT-125M (0.09 on C4, 0.16 on WMT16) while Unigram and EXP maintain near-perfect detection at all scales. Above 125M, all algorithms plateau.

4 Results

4.1 Detectability (RQ1)

Table 1 and Figure 1 present clean detection results for all 18 combinations.

The results reveal an algorithm-specific boundary. KGW on OPT-125M fails to embed reliably (TPR @ 10% FPR = 0.09 on C4, 0.16 on WMT16), consistent with the low-entropy embedding failure described in Kirchenbauer et al. (2023). Critically, this failure is *exclusive to KGW*: Unigram and EXP achieve near-perfect detection even on OPT-125M (TPR $\geq$ 0.95 across both datasets), as shown in Figure 1. This is a key empirical refinement of the general claim that small models cannot be watermarked reliably. Above OPT-125M, all three algorithms plateau: OPT-1.3B and OPT-2.7B show virtually identical detectability across all algorithm–dataset combinations, indicating that additional capacity above a minimum threshold does not improve watermark signal strength.

A note on interpreting the KGW OPT-125M "@ Best" columns in Table 1: the best-F1 threshold selects a value below nearly every observed score, causing the classifier to predict almost everything as watermarked. The resulting F1 = 0.66 is mathematically correct but operationally meaningless—at any deployment-realistic operating point, these watermarks are not separable. Both columns should always be read side by side.

Table 2: Robustness: TPR @ 10% FPR under each attack. WD = word deletion; Syn = synonym substitution; Para = GPT-3.5 paraphrase. Bold = highest surviving TPR per model–dataset group in the paraphrase column.

Algo	Mdl	Data	Clean	WD	Syn	Para
KGW	1.3B	C4	0.99	0.96	0.95	0.28
KGW	1.3B	WMT16	0.96	0.93	0.96	0.25
KGW	2.7B	C4	0.99	0.82	0.89	0.15
KGW	2.7B	WMT16	1.00	0.96	0.99	0.48
Unig.	1.3B	C4	1.00	1.00	0.99	**0.83**
Unig.	1.3B	WMT16	1.00	0.98	0.99	**0.42**
Unig.	2.7B	C4	1.00	0.99	0.98	**0.92**
Unig.	2.7B	WMT16	1.00	0.94	0.96	**0.39**
EXP	1.3B	C4	0.98	0.85	0.96	0.31
EXP	1.3B	WMT16	0.99	0.97	0.98	0.37
EXP	2.7B	C4	0.99	0.79	0.93	0.23
EXP	2.7B	WMT16	1.00	0.97	0.99	0.36
Avg TPR drop (12 combos)			–	*6.2 pp*	*2.8 pp*	***57.6 pp***

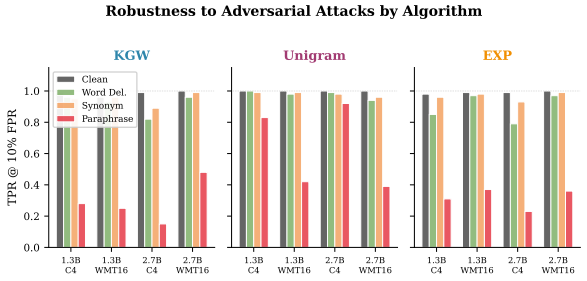

Figure 2: Post-attack TPR @ 10% FPR by algorithm. Paraphrasing (red) is catastrophically more damaging than lexical attacks across all algorithm–model–dataset combinations. Unigram retains substantially higher detection rates post-paraphrase, particularly on C4.

4.2 Robustness to Adversarial Attacks (RQ2)

Table 2 and Figures 2–3 present TPR @ 10% FPR under each attack for OPT-1.3B and OPT-2.7B.

Three patterns emerge. First, GPT-3.5 paraphrasing is catastrophically more damaging than lexical attacks, causing an average TPR drop of

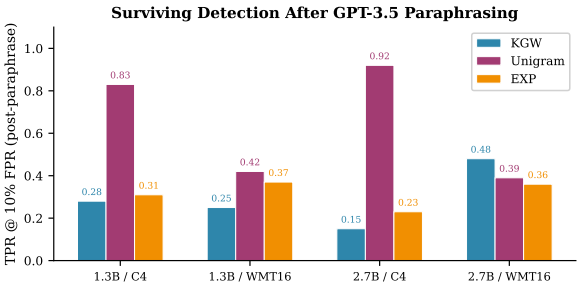

Figure 3: Surviving detection after GPT-3.5 paraphrasing. Unigram dominates across all model–dataset combinations, retaining TPR = 0.92 on OPT-2.7B + C4. KGW and EXP drop to the 0.15–0.48 range.

57.6 pp compared to 6.2 pp for word deletion and 2.8 pp for synonym substitution. This roughly 9× gap highlights a fundamental difference between surface-level text modifications and full semantic rewriting as attack strategies. Second, Unigram is markedly more paraphrase-resistant: its average paraphrase-TPR drop is 36.0 pp versus 69.5 pp for KGW and 67.3 pp for EXP. The effect is particularly striking on C4, where Unigram+OPT-2.7B retains TPR = 0.92 even after paraphrasing (Figure 3). Third, synonym substitution (ratio = 0.50) is consistently *less* damaging than word deletion (ratio = 0.30)—a counter-intuitive result discussed in Section 5.

Our original hypothesis—that EXP's sampling-based mechanism would make it specifically more vulnerable to paraphrase than KGW—is not supported; EXP's paraphrase drop (67.3 pp) is nearly identical to KGW's (69.5 pp), indicating that prefix-context dependence rather than generation mechanism is the primary vulnerability driver.

4.3 Text Quality (RQ3)

Table 3 shows GPT-2 perplexity on C4 and Table 4 shows corpus-level log diversity for all 18 combos.

KGW consistently increases perplexity across all model scales (mean ΔPPL = +1.79), indicating a measurable fluency cost at every scale (Figure 4). Unigram systematically *reduces* GPT-2-scored perplexity (mean ΔPPL = −0.86)—an artifact discussed in Section 5. EXP shows scale-dependent behavior: it reduces perplexity on OPT-125M (−1.49, where embedding largely fails so text is near-unwatermarked) but imposes a cost at larger scales (+1.60 at 1.3B; +0.72 at 2.7B). All three algorithms increase log diversity across nearly all combos—watermarked texts use more varied vocabulary than unwatermarked texts—with

Table 3: Perplexity on C4 (GPT-2 scorer). WMT16 omitted due to cross-lingual scorer mismatch; see Section 3.4. ΔPPL > 0 means watermarking hurts fluency.

Algo	Model	WM PPL	UWM PPL	ΔPPL
KGW	125M	10.78	7.82	+2.96
KGW	1.3B	9.88	9.13	+0.75
KGW	2.7B	10.83	9.19	+1.65
Unig.	125M	6.29	7.25	−0.97
Unig.	1.3B	8.32	8.99	−0.67
Unig.	2.7B	8.90	9.85	−0.95
EXP	125M	6.96	8.46	−1.49
EXP	1.3B	9.57	7.98	+1.60
EXP	2.7B	9.95	9.23	+0.72

Table 4: Log diversity (corpus-level, n-gram orders 2–4 pooled across 100 texts). ΔDiv > 0 means watermarking increases vocabulary diversity. Values for 125M are inflated due to degenerate text; see Section 3.4.

Algo	Model	Data	WM	UWM	ΔDiv
KGW	125M	C4	1.292	1.234	+0.059
		WMT	1.275	1.216	+0.059
KGW	1.3B	C4	0.832	0.664	+0.168
		WMT	0.504	0.541	−0.037
KGW	2.7B	C4	0.836	0.612	+0.224
		WMT	0.512	0.377	+0.135
Unig.	125M	C4	1.941	1.200	+0.742
		WMT	2.310	1.208	+1.102
Unig.	1.3B	C4	1.223	0.643	+0.581
		WMT	0.939	0.453	+0.485
Unig.	2.7B	C4	1.132	0.576	+0.556
		WMT	0.687	0.375	+0.313
EXP	125M	C4	1.673	1.290	+0.383
		WMT	1.620	1.201	+0.419
EXP	1.3B	C4	1.061	0.833	+0.227
		WMT	0.798	0.459	+0.339
EXP	2.7B	C4	1.016	0.702	+0.314
		WMT	0.753	0.444	+0.309

Unigram producing the largest diversity increase at every scale.

5 Discussion

RQ1: An algorithm-specific scale floor. The most important empirical contribution of our study is demonstrating that the low-entropy scale-floor problem is specific to KGW and does not generalize to Unigram or EXP. KGW hashes the previous token to generate a new green list at each step, then applies a fixed logit bias $\delta = 2.0$ to shift token probabilities toward green tokens. On OPT-125M, the model's distribution is sharply peaked at low-entropy positions—numbers, named entities, formulaic phrasing—and the logit nudge is insufficient to flip choices at these positions without breaking coherence. The watermark fails to embed,

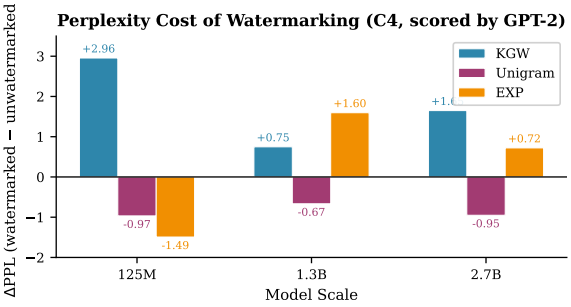

Figure 4: Perplexity cost of watermarking on C4 (ΔPPL, scored by GPT-2). KGW consistently increases perplexity (positive = fluency degradation). Unigram systematically reduces it—an artifact of scorer-prior overlap discussed below. EXP shows scale-dependent behavior.

keeping the green-token ratio near the 50% baseline. Unigram's fixed global list does not depend on the previous token's hash, and EXP's sampling-based mechanism does not bias logits at all; both embed reliably regardless of the model's per-token entropy profile. This has immediate practical implications: deployers using small models should prefer Unigram or EXP over KGW, and should not assume that scale constraints are shared across algorithms.

Above OPT-125M, the 1.3B → 2.7B comparison shows detectability is essentially flat for all three algorithms (e.g., KGW @ C4: TPR = 0.99 at both sizes). This plateau suggests that watermark signal strength is bounded by the embedding mechanism rather than by model capacity, once the model is large enough to have a sufficiently diffuse token distribution. The practical implication is that, for OPT-family models, deployers gain no detectability benefit from scaling beyond 1.3B—the investment in larger models must be justified on other grounds.

RQ2: Prefix dependence drives paraphrase vulnerability. Paraphrasing is by far the most destructive attack (Figure 2). Both KGW and EXP depend on prefix context: KGW regenerates the green list at each step keyed on the previous token, and EXP keys its pseudorandom sequence on a 4-token context window. GPT-3.5 paraphrasing completely rewrites surface form, breaking these prefix structures at every position and thereby disrupting the watermark signal throughout the text. Unigram's fixed global green list is immune to this mechanism—paraphrasing cannot change which tokens are green—which explains its substantially lower vulnerability (36.0 pp average drop vs. 67–70 pp for KGW and EXP). The effect is most pronounced for Unigram on C4 with OPT-2.7B, where TPR remains 0.92 even after GPT-3.5 rewriting (Figure 3), suggesting near-attack-proof behavior against this class of adversary in general-domain text. This provides strong empirical support for Zhao et al. (2024)'s design rationale for fixed-list watermarking.

The finding that synonym substitution (50%) is less damaging than word deletion (30%) is counter-intuitive but mechanistically explainable for KGW. Word deletion shifts the token sequence, changing the prefix hash for every subsequent position and propagating green-list disruption throughout the text. Synonym substitution preserves positional prefix structure: even when content tokens change, the preceding token (and hence the green-list hash) at unchanged positions remains the same. Disruption therefore does not cascade in the same way. This finding suggests that attack severity cannot be predicted from the proportion of tokens affected alone; the *type* of modification matters, and sequence-disrupting attacks are more effective against prefix-dependent watermarks than content-replacing attacks at comparable or even lower ratios.

We also note that our original hypothesis—that EXP, as a sampling-based method, would be specifically more vulnerable to paraphrase than KGW—is not supported by the data. EXP's average paraphrase drop (67.3 pp) is nearly equal to KGW's (69.5 pp). Both share a dependence on prefix context that paraphrasing exploits; the difference in generation mechanism (logit-biasing vs. sampling-based) confers no robustness advantage. The corrected finding is that the common vulnerability of KGW and EXP to paraphrase is driven by their shared reliance on prefix structure, not by their different embedding strategies.

RQ3: Quality costs and the Unigram GPT-2 artifact. KGW imposes a consistent and scale-stable perplexity cost (ΔPPL ranging from +0.75 to +2.96), confirming that its logit biasing meaningfully shifts the generation distribution across all model sizes. The cost is largest on OPT-125M (+2.96), where the model has the least distributional headroom to absorb the logit perturbation, and smallest on OPT-1.3B (+0.75). EXP's behavior is scale-dependent: near-zero cost where the watermark fails to embed (125M) and positive cost where it succeeds (+1.60 at 1.3B). Unigram's negative ΔPPL warrants careful interpreta-

tion. Matched-prompt qualitative inspection shows watermarked Unigram text is not visibly degraded; the most likely explanation is an overlap between Unigram's fixed green-list tokens and GPT-2's high-probability tokens, causing the scorer to favor Unigram-watermarked outputs due to a scorer-prior interaction rather than genuine fluency improvement. This interpretation is further supported by the log diversity data: Unigram-watermarked text is substantially *more* diverse than unwatermarked text (mean $\Delta\text{Div} = +0.57$ across 1.3B and 2.7B on C4), which is inconsistent with a quality improvement in the usual sense but consistent with the model being steered toward a broader and systematically different set of tokens. This artifact highlights a broader methodological concern: automated perplexity scoring may not reliably capture watermark-induced quality degradation when the scorer's token preferences overlap with the watermarking algorithm's token selection.

All algorithms increase log diversity, with Unigram showing the largest increase at every scale and dataset. This is stable from 1.3B to 2.7B, confirming it is an algorithmic property of the fixed green-list mechanism rather than a scale effect.

Domain effects. WMT16 consistently yields lower post-paraphrase TPR for Unigram than C4 (0.42 and 0.39 vs. 0.83 and 0.92). A plausible explanation is that WMT16 prompts elicit shorter, more formulaic completions, leaving fewer green-list tokens per text; paraphrasing a small green-token budget removes proportionally more of the signal than paraphrasing a longer text. This domain sensitivity suggests that watermarking benchmarks should use domain-matched text, and that C4-based evaluations may overestimate practical robustness for deployment in structured-text domains such as news.

Practical recommendations. Our findings support three deployment-relevant conclusions. First, KGW should not be used on small models where the token distribution is concentrated; Unigram or EXP are reliable alternatives that do not share this constraint. Second, once above the minimum viable scale ($\geq$ 1.3B for OPT-family models), scaling up provides no detectability benefit. Third, Unigram is the most resilient algorithm under realistic adversarial conditions—substantially more paraphrase-resistant and with the lowest perplexity cost—making it the preferred choice for practical deployment despite being the simplest of the three methods studied. It is worth noting that Unigram's advantage is not just marginal: on C4 with OPT-2.7B, Unigram retains TPR = 0.92 after paraphrasing while KGW drops to 0.15 and EXP to 0.23—a gap of 69–77 percentage points that represents the difference between a usable and unusable watermark.

6 Implementation Details

Detection metrics. Detection metrics (TPR and F1 at 10% FPR and at the F1-maximizing threshold) were computed using a custom `compute_metrics()` implementation rather than MarkLLM's `DynamicThresholdSuccessRateCalculator`. Two reasons motivated this choice: first, explicit handling of EXP's inverted score polarity (lower p-value = watermarked, opposite of KGW/Unigram); second, the custom function returns auxiliary diagnostic fields (selected thresholds, sample counts) useful for debugging. Cross-validation against MarkLLM's library on the KGW+OPT-1.3B+C4 control combo showed the two implementations agree to numerical precision: TPR = 0.99 (both), F1 @ 10% FPR = 0.9519 (16-decimal match), F1 @ best = 0.99 (both).

Memory management. BERT-large ($\sim$1.3 GB) and OPT-2.7B ($\sim$5.3 GB) cannot coexist on the T4's 16 GB without OOM errors. We adopted a strict sequential loading strategy: all synonym-substitution attacks across the 12 combos were completed with BERT loaded, then BERT was explicitly freed (`del` + `torch.cuda.empty_cache()`), and only then was the OPT model loaded for detection scoring. Consecutive OPT-1.3B and OPT-2.7B detection sweeps were also split into separately executed cells to avoid residual memory contention. This pattern proved essential for reliable execution and is documented in the repository.

Environment stability. Initial setup pinned `torchvision==0.18.1`, which was correct during Week 2 but conflicted with Colab's updated base image (torch 2.10, CUDA 12.8) by Week 3, forcing a torch downgrade that broke modern numpy/scipy. Resolution: drop torch/torchvision pins entirely, retain only `scipy`$\geq$`1.14,<1.16` and `transformers==4.41.2`, and let Colab supply current numpy and torch. A fixed-seed equivalence test confirmed byte-identical gener-

ation and 15-decimal-place identical detection scores across the old and new environments, validating that OPT-125M and OPT-1.3B files generated under the earlier configuration could be combined with new OPT-2.7B files without any methodological caveat. The lesson is general: pin only the constraints a framework actually needs, not defensive version locks on rapidly-evolving base packages.

7 Limitations

Several limitations should be considered when interpreting our results. First, all experiments use OPT-family models; the scale-floor threshold and plateau behavior may differ for other architectures (e.g., LLaMA, Pythia), particularly those with different tokenizer vocabularies or pre-training objectives. Second, our sample size of 100 texts per combination, while standard in prior work, limits statistical power for fine-grained comparisons—confidence intervals on individual TPR values are on the order of ± 5 pp. Third, we evaluate only three attack types; stronger adaptive attacks (Sadasivan et al., 2023) or combinations of attacks may reveal different robustness patterns. Fourth, text quality is assessed only by GPT-2 perplexity and n-gram diversity, without human evaluation; the Unigram negative-ΔPPL artifact illustrates the limitations of automated quality metrics and suggests that future work should incorporate human fluency judgments. Fifth, robustness on OPT-125M was not evaluated for Unigram and EXP, which embed reliably at that scale; characterizing their robustness behavior at that scale remains an open and practically relevant question. Finally, our study uses only the OPT-family OPT model architecture; extending the benchmark to decoder-only models from other families would strengthen the generalizability of our conclusions.

8 Resources

All experiments were conducted using the MarkLLM framework (Pan et al., 2024) on Google Colab Pro with a Tesla T4 GPU (16 GB VRAM). Models were loaded in FP16; no quantization was applied. The GPT-3.5 paraphrase attack used the OpenAI Batch API (1,200 requests, zero errors, estimated cost under $2). Model weights and outputs were persisted on Google Drive across Colab sessions. No model training was performed; all experiments involve inference and evaluation only.

9 Code

All experiment notebooks, evaluation scripts, and a `README` with reproduction instructions are available at: https://github.com/zahran001/MarkLLM-Survey. The repository contains a single self-contained Colab notebook covering all experimental phases (generation, detectability, robustness attacks, and text quality), a README with reproduction instructions, and the full set of saved generation, attack, and detection outputs. The shipped artifacts let reviewers replay detection, aggregation, and table generation in roughly 30 minutes on a Colab T4 without rerunning the 3-hour generation phase.

10 Conclusion

We presented a systematic benchmark of three LLM watermarking algorithms across three OPT model scales, two text domains, and three adversarial attacks using the MarkLLM framework. Our primary contribution is an empirical refinement of the scale-floor narrative: the failure of watermarking on small models is not universal—it is specific to KGW's context-dependent logit-biasing mechanism. Unigram and EXP detect reliably even on OPT-125M. Above the minimum viable scale, detectability plateaus between 1.3B and 2.7B for all algorithms. GPT-3.5 paraphrasing remains the dominant attack, destroying 57.6 pp of TPR on average, while Unigram retains substantially higher detection rates post-paraphrase (up to 0.92 on C4) and imposes the lowest perplexity cost. These findings position Unigram as the strongest practical watermarking choice under adversarial conditions. Future work should extend this analysis to larger model families (e.g., LLaMA, Pythia), stronger adaptive attacks, the interaction between watermarking and quantization, and robustness evaluation of Unigram and EXP on sub-billion-parameter models where KGW fails.

References

Scott Aaronson and Hendrik Kirchner. 2022. Watermarking GPT outputs. Talk at the Simons Institute. https://www.scottaaronson.com/talks/watermark.ppt.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, and 1 others. 2016. Findings of the 2016 conference on machine translation.

In *Proceedings of the First Conference on Machine Translation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*.

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. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*.

OpenAI. 2022. ChatGPT: Optimizing language models for dialogue. https://openai.com/blog/chatgpt.

Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. 2024. MarkLLM: An open-source toolkit for LLM watermarking. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 61–71, Miami, Florida, USA. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

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.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can AI-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

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. 2024. Provable robust watermarking for AI-generated text. In *The Twelfth International Conference on Learning Representations*.