Factuality Beyond Coherence: Evaluating LLM Watermarking Methods for Medical Texts

Rochana Prih Hastuti, Rian Adam Rajagede,
Mansour Al Ghanim, Mengxin Zheng, Qian Lou
University of Central Florida
Orlando, FL
{rochana, rian, mansour.alghanim, mengxin.zheng, qian.lou}@ucf.edu

Abstract

As large language models (LLMs) are adapted to sensitive domains such as medicine, their fluency raises safety risks, particularly regarding provenance and accountability. Watermarking embeds detectable patterns to mitigate these risks, yet its reliability in medical contexts remains untested. Existing benchmarks focus on detection-quality tradeoffs and overlook factual risks. In medical text, watermarking often reweights low-entropy tokens, which are highly predictable and often carry critical medical terminology. Shifting these tokens can cause inaccuracy and hallucinations, risks that prior general-domain benchmarks fail to capture.

We propose a medical-focused evaluation workflow that jointly assesses factual accuracy and coherence. Using GPT-Judger and further human validation, we introduce the Factuality-Weighted Score (FWS), a composite metric prioritizing factual accuracy beyond coherence to guide watermarking deployment in medical domains. Our evaluation shows current watermarking methods substantially compromise medical factuality, with entropy shifts degrading medical entity representation. These findings underscore the need for domain-aware watermarking approaches that preserve the integrity of medical content.

1 Introduction

LLMs have advanced human-like text generation capability, raising concerns about potentially harmful or biased information in various use case, including in the medical domain (Xue et al., 2024; Zhang et al., 2025; Al Ghanim et al., 2023; Lou et al.; Hastuti et al., 2025). Watermarking techniques serve as a safeguard by embedding subtle statistical patterns into generated content that enable detection while preserving quality (Kirchenbauer et al., 2023; Gu et al., 2023; Kuditipudi et al., 2024).

While watermarking methods show promise for securing information in LLM outputs, their effectiveness in the medical domain remains underexplored (Kong et al., 2024). A key limitation is that watermarking often treats all tokens uniformly, ignoring low-entropy tokens (Lee et al., 2024). These are highly predictable words that often carry critical factual content in medical text, such as disease names. Reweighting them to embed a watermark can alter their meaning, compromising factual accuracy and increasing the risk of hallucinations. Current watermarking evaluation primarily assesses detection-quality tradeoffs from a domain-agnostic perspective, where quality assessment using human or LLM evaluators focuses on measuring text preference (Tu et al., 2024; Molenda et al., 2024) or semantic coherence (Singh and Zou, 2024), without considering the domain-specific factual risks associated with low-entropy tokens

To address the limitations of general evaluation workflows, our watermarking evaluation for medical text is illustrated in Figure 1. (1) **Evaluation Workflow**: a unified framework to accommodate both existing automatic metrics and GPT-Judger assessment while examining the critical dimensions of factuality and coherence. This workflow is deliberately flexible, allowing traditional metrics to be integrated alongside our fine-grained GPT-Judger assessments. The key innovation is in creating a structured evaluation that works across different measurement approaches and tasks. (2) **Factuality-Weighted Score (FWS)**: To reflect the critical nature of factual integrity in medical applications, we introduce a composite quality metric that emphasizes factual correctness beyond coherence. We validate this approach through human evaluation, confirming alignment between expert judgments, our GPT-Judger assessments, and the proposed FWS metric, creating a reliable workflow for evaluating watermarked medical text.

Our analysis reveals critical trade-offs in wa-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 15129–15147
November 4-9, 2025

termarking methods for medical text. We find that generation-time watermarks, where watermarks are embedded during the LLM's text generation process, achieve high detection rates but alter token entropy distributions, particularly affecting medical entities. Our evaluation workflow demonstrates that factuality degradation is often more severe than coherence loss, a distinction missed by standard quality metrics. Human evaluations show stronger agreement with our FWS. These results highlight the need for tailored watermarking approaches in medical domains where factual integrity is crucial.

We recommend future approaches: (1) develop domain-aware techniques that preserve low-entropy tokens of medical terminology and avoid hallucination while maintaining detectability, and (2) perform factuality-focused evaluation beyond general quality assessments. These insights aim to develop reliable watermarking methods balancing detectability with crucial medical factuality.

To summarize, our contributions are four-fold:

- We introduce a unified evaluation framework for medical texts that integrates coherence and factuality dimensions, compatible with both traditional metrics and our proposed GPT-Judger approach.
- We develop the Factuality-Weighted Score (FWS) that emphasizes factuality in medical contexts, validated through correlation analysis with human evaluations involving medical practitioners.
- We provide baseline results on medical tasks, showing that despite high detection capabilities, current watermarking methods face notable factuality degradation.
- We identify entropy distribution shifts in medical entities as a contributing factor behind factuality degradation, providing critical insights for developing a domain-aware watermarking approach.

The code and experimental data are available at: https://github.com/rochanaph/fact-eval-wllm

2 Related Work

Watermarking methods for LLMs. Methods to detect LLM-generated text has been studied in previous works, like leveraging the statistical properties without changing the text itself (Mitchell et al., 2023; Gehrmann et al., 2019), adding lexical feature as a pattern to distinguish (He et al., 2022), as well as training both traditional and neural-based classifier to classify human-generated and model-generated texts (OpenAI, 2023; Guo et al., 2023; Singh and Zou, 2024). Recently, a series of work embeds watermarks within LLM-generated texts by either modifying logits from the LLM (Kirchenbauer et al., 2023, 2024), adding a selective technique tailored for domain requirements (Lee et al., 2024), manipulating the sampling procedure (Christ et al., 2024; Kuditipudi et al., 2024), or investigate the learnability of watermarks in the distillation process (Gu et al., 2023).

Benchmarking watermarking methods. Existing benchmarking for watermarking methods has defined evaluation criteria, particularly in general-domain scenarios. Most benchmarks emphasize detectability to ensure watermark robustness (Tu et al., 2024) and complement this with quality assessments often based on LLM-as-evaluator preferences (Singh and Zou, 2024). However, these evaluations typically lack domain-specific considerations and rely heavily on automatic metrics (Ajith et al., 2024) or simplified pairwise comparisons, with no further human validation or detailed justification of results (Molenda et al., 2024).

Securing medical LLM's. The development of medical language models supports the applicability of various tasks in broader clinical and healthcare settings. Despite this progress, efforts to incorporate safeguards for medical language models remain limited, such as in the jailbreaking case or model watermarking via backdoor (Kong et al., 2024; Xue et al., 2023; Al Ghanim et al., 2024; Hastuti et al., 2025; Zheng et al., 2024). Given the relatively recent development of LLM watermarking techniques, their applicability and effectiveness in critical domains like medical remain unexplored and need further investigation.

3 Evaluation Workflow

General workflow Our framework is illustrated in Figure 1. First, we feed the Medical LLMs with a prompt such as a medical question, and then we evaluate the quality of its watermarked and unwatermarked output. We evaluate watermarking methods across three key quality aspects. **1) Coherence** captures the fluency of the generated text and is commonly used in watermarking evaluation. **2) Relevance/Completeness** measures whether the

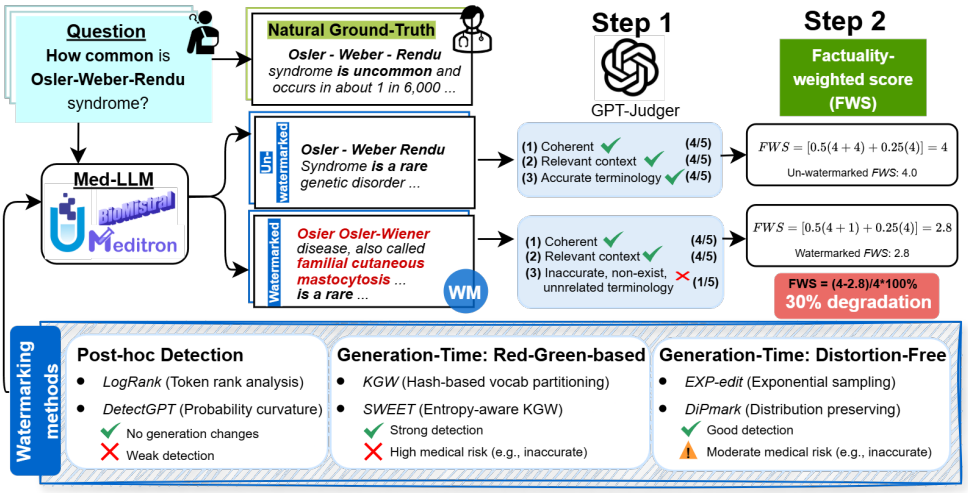


Figure 1: Factuality degradation in watermarked medical text illustrated through the proposed evaluation framework. **(1) Evaluation workflow** covers *coherence, relevance,* and *factual accuracy*, applicable to GPT-Judger and traditional metrics. **(2) Factuality-weighted Score (FWS)** emphasizes critical factual accuracy beyond coherence and serving as a unified metric to guide watermarking deployment in medical applications.

context surrounding critical medical terminology is preserved, ensuring watermarking mechanisms do not distort model confidence.[1] **3) Factual accuracy** assesses whether newly introduced medical terminology preserves correct semantics to avoid hallucination [2]. The latter two aspects emphasize potential factual corruption, with detailed analysis provided in Section 5.4.

In **Step 1**, we implement and evaluate these three quality aspects through two approaches:

- **GPT-judger evaluation.** We employ GPT-4o-2024-08-06 as an LLM evaluator to provide fine-grained judgments along the dimensions defined in Table 1.

- **Traditional automatic metrics.** Our framework also integrates established metrics to enable unified quality quantification. Coherence is measured using non-reference metrics such as Perplexity (Kirchenbauer et al., 2023) or reference-based similarity measures such as SimCSE (Huo et al., 2024). Accuracy is evaluated with task-oriented, reference-based metrics including `ROUGE-2`, `ROUGE-L`, `F1` (He et al., 2024), and factuality metric `AlignScore` (Zha et al., 2023), using natural text as the ground truth.

In **Step 2**, we evaluate the overall quality by proposing a quality metric **Factuality-weighted Score** to see how the watermarking method affects factuality.

Factuality-weighted Score (FWS) The proposed aspects allow us to quantify the overall quality metric as shown in Equation 1. To prioritize relevant context and factual accuracy over coherence, we set $\alpha = 0.4$ and $\beta = 0.2$ to indicate that Relevance or Completeness ($Rel.$) and Accuracy ($FactAcc.$) are equally important, and they are twice as important as Coherence ($Coh.$). This weighting was based on empirical consideration through sensitivity analysis provided in Table 5.

$$FWS = \alpha(\text{Rel} + \text{FactAcc}) + \beta(\text{Coh}) \quad (1)$$

FWS serves as a unified metric that enables actionable deployment decisions for watermarking methods in the medical domain and is also applicable to other high-stakes domains where factual accuracy is critical.

Human evaluation To evaluate both automatic metrics and GPT-Judger ratings on equal footing, we conducted a human evaluation with 6 respondents, including graduate students and medical practitioners, ensuring familiarity with LLMs and medical expertise. The evaluation focused on the KGW watermarking method applied to the Question Answering (QA) task, chosen as KGW underlies many LLM watermarking approaches and QA

[1] Variations in symptom descriptions may correspond to distinct diagnoses, a model should not fluctuate in confidence when generating crucial terms like *"cancer"*. See Figure 4.

[2] Entity errors source to factuality hallucination (Li et al., 2024). See Table 7 for rate and Table 13 for example.

Text Completion	Question Answering	Summarization
Coherence: Whether the language is coherent, clear, and understandable.		
Relevance: Whether the text includes **relevant information** for the prompt.	**Relevance:** Whether the answer addresses the question **without going off-topic and covering all essential parts**	**Completeness:** Whether the generated summary **misses any important information** from the original text.
Factual Accuracy: Whether the generated text **introduces inaccurate or unrelated medical terms not present** in the original reference.		

Table 1: Proposed quality aspects for evaluating watermarked medical text include *coherence* (**Coh.**), *relevance* (**Rel.**), and *factual accuracy* (**FactAcc.**) across text completion, question answering, and summarization. **Rel.** assesses contextual information retention, and **FactAcc.** ensures the semantics of medical terminology are preserved, both emphasized in the FWS metric as critical for medical applications.

represents the most realistic practical use case for medical language models. Each item was rated by 3 respondents to obtain an average score using the quality aspects in Table 1. The full evaluation template is provided in Appendix B.2.

Pearson correlation was computed to measure agreement between human judgments and both traditional automatic metrics and GPT-Judger ratings. Prior to this, a Nemenyi test was applied to detect significant differences among the proposed aspects, with lower p-values indicating greater distinction (Fu et al., 2024).

4 Experiments

4.1 Research Questions

Our experiments aim to answer the following research questions:

- **RQ1** How do watermarking methods perform in *detection* and *coherence quality*?
- **RQ2** How do watermarking methods perform on *task-oriented* evaluations?
- **RQ3** How does the proposed *factuality-weighted score* perform, and is it aligned with human evaluation?
- **RQ4** What are the risks and contributing factors of medical *factuality corruption*?

4.2 Watermarking Methods

We evaluate six watermarking methods across two categories (Liu et al., 2024). **Post-hoc watermarking** applies after text generation. We evaluate two schemes from this category: LogRank (Gehrmann et al., 2019), which classifies text by thresholding the average log-rank of tokens, as machine-generated text typically has smaller average log-ranks, and DetectGPT (Mitchell et al., 2023), which exploits the tendency of model-generated text to occupy negative curvature regions in the log probability function by comparing original passages with semantically similar perturbations.

The second category is **generation-time watermarking**, where watermarks are embedded during the LLM's text generation process. We evaluate four approaches: *KGW* (Kirchenbauer et al., 2023), a foundational logit-based method that partitions the vocabulary into red and green lists based on a previous token hash; *SWEET* (Lee et al., 2024), an improvement of the KGW method on addressing low entropy tokens for code generation domain However, another work (Huo et al., 2024) showed that SWEET works well in the general domain, thus we believe potentially useful in medical text where low entropy tokens can be found; *DiPmark* (Wu et al., 2024) a distribution-preserving watermark that uses complementary reweighting strategies with random token permutations to maintain original text quality; and *EXP-edit* (Kuditipudi et al., 2024), a token sampling-based method modifying probabilities based on token rank.

This selection allows comparison across three strategies ***post-hoc* (PH), *logit-based* (LB)**, distribution preserving or ***distortion-free* (DF)**, while also includes methods addressing potential domain-specific challenges like low entropy. All generation-time watermarks used the MarkLLM toolkit (Pan et al., 2024) with default parameters. Brief methods are available in Appendix B.4, and parameter details are available in Appendix A.2, Table 10.

4.3 Tasks and Datasets

To assess watermark performance in realistic medical scenarios, we employ three distinct tasks using established medical datasets. First, **Text Completion** using the HealthQA dataset (Zhu et al., 2019) requires models to complete medical passages (approx. 200 tokens), evaluating the impact on fundamental coherence and fluency. Second, **Question Answering (QA)**, also on HealthQA, challenges models to answer medical questions (approx. 200 tokens), specifically testing factual accuracy preservation under watermarking. Third, **Summarization** using the MeQSum dataset (Abacha and Demner-Fushman, 2019) involves generating con-

Schm.	Waterm. Methods	Meditron 7B				BioMistral 7B				MedLlama 8B			
		Detection		Quality		Detection		Quality		Detection		Quality	
		TPR↑	AUROC↑	PPL↓	SimCSE↑	TPR↑	AUROC↑	PPL↓	SimCSE↑	TPR↑	AUROC↑	PPL↓	SimCSE↑
	Text Completion												
	Un-watermarked	–	–	9.7	1.000	–	–	9.2	1.000	–	–	4.5	1.000
PH	LogRank	0.063	0.764	9.7	1.000	0.146	0.750	9.2	1.000	0.042	0.700	4.5	1.000
	DetectGPT	0.010	0.704	9.7	1.000	0.021	0.592	9.2	1.000	0.010	0.619	4.5	1.000
LB	KGW	0.999	0.999	12.6	**0.645**	**1**	**1**	11.6	0.688	**0.890**	**0.995**	4.9	**0.782**
	SWEET	**1**	**1**	12.7	0.642	**1**	**1**	11.4	0.685	0.860	0.994	4.6	0.765
DF	DiPmark	0.990	0.999	**10.9**	0.643	**1**	**1**	**9.9**	**0.709**	0.290	0.944	**4.5**	0.776
	EXP-edit	0.935	0.991	26.6	0.625	0.980	0.996	12.8	0.654	0.040	0.689	9.8	0.758
	Question Answering												
	Un-watermarked	–	–	8.7	1.000	–	–	8.5	1.000	–	–	4.2	1.000
PH	LogRank	0.438	0.967	8.7	1.000	0.875	0.985	8.5	1.000	0.573	0.977	4.2	1.000
	DetectGPT	0.160	0.741	8.7	1.000	0.000	0.459	8.5	1.000	0.000	0.543	4.2	1.000
LB	KGW	0.995	0.999	11.9	0.628	**1**	**1**	11.0	0.682	0.620	0.990	4.8	0.792
	SWEET	**1**	**1**	11.2	**0.633**	**1**	**1**	10.2	0.685	**0.890**	**0.995**	4.6	0.791
DF	DiPmark	0.975	0.999	**9.9**	0.628	0.980	0.999	**9.7**	**0.693**	0.550	0.964	**4.4**	**0.797**
	EXP-edit	0.965	0.988	24.3	0.620	0.985	**1**	12.2	0.666	0.020	0.824	6.1	0.774
	Summarization												
	Un-watermarked	–	–	41.4	1.000	–	–	59.4	1.000	–	–	32.4	1.000
PH	LogRank	0.050	0.303	41.4	1.000	0.000	0.195	59.4	1.000	0.281	0.767	32.4	1.000
	DetectGPT	0.087	0.548	41.4	1.000	0.087	0.494	59.4	1.000	0.080	0.551	32.4	1.000
LB	KGW	0.565	0.962	47.7	0.376	0.327	0.914	81.3	0.494	**0.091**	**0.786**	64.2	0.722
	SWEET	**0.820**	**0.985**	45.2	**0.410**	0.375	0.922	286.5	0.449	0.015	0.631	48.6	**0.728**
DF	DiPmark	0.090	0.859	**41.7**	0.393	0.083	0.741	115.9	**0.504**	0.005	0.678	41.9	0.718
	EXP-edit	0.015	0.949	65.8	0.393	**0.960**	**0.994**	**20.1**	0.402	0.005	0.528	**33.2**	0.724

Table 2: (RQ1) Detection and quality performance of six watermarking methods under three schemes: PH (*post-hoc*), LB (*logit-based*), and DF (*distortion-free*) evaluated on three medical generation tasks with three different models. TPR was set at FPR = 0%.

cise summaries (< 100 tokens) of medical questions, evaluating performance on challenging short-text generation (Kirchenbauer et al., 2023) and the crucial preservation of essential factual information. These tasks represent diverse medical tasks and output lengths, offering a comprehensive view of watermark impact on text quality and factual integrity in the medical domain. Details on how we built each task are available in Appendix B.3.

4.4 Model and Hardware

We use Meditron-7B (Chen et al., 2023) as the main model for most experiments. This model included training on clinical guidelines, more suitable for consumer QA, unlike most models, which limited their training on abstracts and articles. Meditron is a medical adaptation model from Llama-2-7B (Touvron et al., 2023) through continued pretraining on the medical GAP-Replay corpus. We also evaluate other medical models like MedLlama-3-8B[3] and BioMistral 7B (Labrak et al., 2024). Though this variation is not our main focus, we only include RQ1 results using these models and provide additional analysis in Appendix A.1. All experiments are conducted on a workstation equipped with an AMD Ryzen Threadripper PRO 3955WX (16-Cores) processor and two NVIDIA GeForce RTX 3090 GPUs (24GB VRAM each).

5 Results

We present our analysis of watermarking methods in answering a series of research questions defined in § 4.1. We begin by examining traditional automatic evaluation metrics in § 5.1 (RQ1), comparing detectability measures against standard quality metrics to establish baseline performance. We then extend our investigation to task-specific metrics in § 5.2 (RQ2), providing deeper insights regarding its resemblance of factuality aspects.

Building on the observed findings, we provide our Factuality-weighted Score (FWS) metrics in § 5.3 (RQ3), which emphasize the factuality quality preservation. We validate the proposed metric through correlation analysis, comparing existing traditional automatic metrics and GPT-Judger assessments, and verify its alignment with human perception. Finally, we conduct an in-depth observation in § 5.4, to check the possible contributing factors to factuality corruption in watermarked texts (RQ4), offering insights for future watermarking method improvements.

5.1 Detectability and Quality (RQ1)

Table 2 presents comparative performance of watermarking methods across generation tasks, measuring both detection capability and text quality preservation. `TPR` measures detection rate, `AUROC` overall robustness, `PPL` text fluency, and `SimCSE` semantic similarity. The results show a clear trade-off between detectability and output quality among different watermarking approaches.

Post-hoc methods (LogRank and DetectGPT) maintain the best possible text quality across all tasks, as they introduce no alterations to the generated text. However, this advantage comes at the cost of significantly reduced detectability. Lo-

[3]https://hf.co/johnsnowlabs/JSL-MedLlama-3-8B-v2.0

gRank consistently outperforms DetectGPT across all tasks, reaching its highest detection performance in Question Answering (AUROC 0.967), though its TPR remains relatively low at 0.438. These low detectabilities are expected since no additional pattern is added to distinguish the generated text.

In contrast, generation-time methods demonstrate superior detection capabilities, with SWEET achieving perfect TPR and AUROC scores of 1 across Text Completion and Question Answering, and still best in Summarization. Quality-wise, DiPmark's distribution-preserving approach shows the best PPL score across all tasks with a significant gap compared to other generation-time methods, though SWEET and KGW performed slightly better in SimCSE score.

Observation 1: *Post-hoc methods preserve text quality but suffer from weak detection, especially in* TPR. *In contrast,* ***generation-time methods offer near-perfect detection with minimal quality loss****, making them a more balanced and promising approach.*

To better understand how generation-time watermarking quality can be optimized from a downstream task perspective, the next subsection presents a focused evaluation of task-oriented performance.

5.2 Task-oriented Performance (RQ2)

Table 3 presents reference-based quality metrics evaluating generated text against natural text ground-truth, deliberately to give insights of factual preservation across different tasks. Unlike the general quality metrics in RQ1, task-specific metrics provide finer granularity ROUGE-2 measures bigram overlap between generated and reference text, ROUGE-L captures the longest common subsequence to reflect order matching, F1 balances precision and recall to quantify overall matching accuracy and AlignScore evaluates factual consistency. Higher values across all metrics indicate better performance.

Waterm. Methods	Text Completion			Question Answering			Summarization		
	RG-2	RG-L	AS	RG-2	F1	AS	RG-2	RG-L	AS
Un-waterm.	.038	.142	.241	.027	.130	.264	.046	.156	.114
KGW	.030	.135	**.255**	.021	**.127**	**.273**	.025	.125	.096
SWEET	.029	.132	.227	.022	.125	.235	.032	.128	**.121**
DiPmark	**.033**	**.137**	.216	.021	.125	.238	.046	**.153**	.110
EXP-edit	.029	.129	.223	**.023**	.123	.254	**.049**	.142	.104

Table 3: (RQ2) Watermarking methods Task-oriented performance across various tasks. RG* stands for ROUGE* and AS stands for AlignScore.

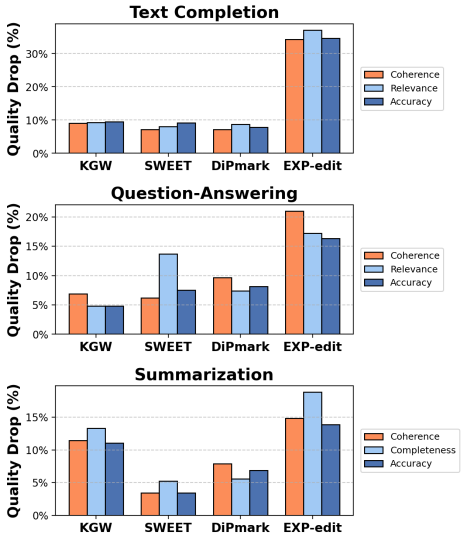


Figure 2: (RQ3) GPT-Judger quality aspects cover the dimensions of *coherence*, *relevance*, and *factual accuracy*, showing degradation across watermarking methods. On average, relevance and factual accuracy suffer greater degradation than coherence.

In Text Completion, DiPmark achieved the best performance with ROUGE-2 (0.033) and ROUGE-L (0.137), slightly above other watermarking methods, while getting the lowest AlignScore (0.216). For Question Answering, KGW gets the best performance with the highest F1 score (0.127) and AlignScore (0.273), though all methods show minimal degradation from un-watermarked text. In Summarization, EXP-edit achieves ROUGE-2 (0.049) that exceeds even the un-watermarked text (0.046), likely because its alternative token sampling approach modifies probability distributions rather than directly applying specific token selections.

Observation 2: *DiPmark demonstrates strong performance in Text Completion, while EXP-edit notably exceeds un-watermarked quality in Summarization. However, overall differences between methods are marginal (<1%), indicating* ***task-specific metrics are insufficient for drawing clear factuality quality conclusions****.*

Addressing traditional automatic metrics' limitation, the next subsection (RQ3) introduces GPT-Judger and our Factuality-Weighted Score (FWS) for a more tailored, factuality-focused evaluation in the medical domain.

5.3 Factuality-weighted Score FWS (RQ3)

GPT-Judger quality aspects Figure 2 shows results of GPT-Judger Quality aspects defined in Table 1, evaluating coherence (orange) and factuality dimensions (blue). The results highlight quality

degradations when watermarking techniques are applied to language models, ranging between 3%-37% across tasks. This pattern is most pronounced with EXPEdit in Text Completion, showing significant degradations (34-37% drops) across all dimensions. Another interesting sight is in the Question Answering task, where SWEET's relevance quality degrades 2x the coherence's degradation.

Additionally, by averaging percentage drops across all tasks and methods, we observe that relevance suffers the most substantial degradation at approximately 12.42%, compared to coherence's average drop of 11.58% and accuracy's 11.09%. Though the difference seems not significant, Figure 2 shows contradictory relations (SWEET on QA, DiPmark on Summarization), where eliminating one over the other aspect will not be insightful. It is also worth considering that the drop in accuracy aspect itself indicates factuality corruption, as the Judger agrees that it introduces inaccurate medical terms, potentially signaling hallucinations. These findings indicate that coherence alone is inadequate and can be misleading. While a model's output may maintain reasonable coherence, its factual reliability may experience more compromises. Based on these insights, we incorporate these detailed assessments into a unified Factuality-weighted Score (FWS).

FWS of Automatic metrics and GPT-Judger The results of unified quality metrics using FWS can be seen in Table 4 through both traditional automatic metrics (`Auto`) and GPT-Judger (`GPTJ`) quality assessment. FWS of GPT-Judger metrics provides more interpretable evaluation of watermarking methods, with wider score differentials (ranging from 0.369 to 0.556) compared to FWS using traditional automatic metrics, which are tighter (0.135 to 0.195). This distinction provides a clearer assessment of the performance.

Watermarking Methods	Text Completion		Question Answering		Summarization	
	Auto	GPTJ	Auto	GPTJ	Auto	GPTJ
KGW	0.195	0.540	0.185	0.411	0.135	0.419
SWEET	0.193	0.556	0.185	0.393	0.146	0.451
DiPmark	0.197	0.552	0.184	0.408	0.158	0.432
EXP-edit	0.188	0.369	0.183	0.369	0.155	0.405

Table 4: (RQ3) Factuality-weighted score (FWS) across watermarking methods, generation tasks and evaluation schemes. FWS using GPT-Judger aspects (`GPTJ`) presents clearer distinctions than traditional automatic metrics (`Auto`).

For the task-specific performance across watermarking approaches: SWEET demonstrates superior performance in Text Completion (0.556), while KGW achieves the highest scores in both Question Answering (0.411) and Summarization (0.419). EXP-edit consistently shows the weakest performance across all tasks. This is likely due to its alternative token sampling strategy, which causes the model to behave almost like a different language model compared to the un-watermarked version. In this case, it introduces more drawbacks than the intended benefit of distortion-free watermark, as claimed (Kuditipudi et al., 2024).

Another interesting result in the Question Answering task is that SWEET and KGW scores are the same under automatic metrics, but SWEET falls behind in the GPT-Judger result. This is due to greater factuality degradation, as supported in Figure 2. Since factuality is weighted more in FWS, SWEET's score behind KGW's by about 2%. Overall, indicating the reliability of our proposed FWS. Next, we calculate the correlation of both FWS (`Auto` and `GPTJ`) and human ratings to assess alignment with human judgment.

Configuration	FWS Param. (α, β)	Auto-Human	GPTJ-Human
Coherence-Heavy (1:2)	(0.25, 0.5)	0.089	0.814
Equal Weighting (1:1)	(0.33, 0.33)	0.181	0.833
Current (Ours) (2:1)	(0.4, 0.2)	**0.248**	**0.840**
Factuality-Heavy (4:1)	(0.44, 0.11)	0.276	0.841
Factuality-Heavy (6:1)	(0.46, 0.08)	0.283	0.841

Table 5: Sensitivity analysis comparing correlations of automatic and GPT-Judger with human evaluation under different weighting configurations, with stronger correlations observed when factuality is emphasized.

Human Evaluation Human ratings were evaluated using the Nemenyi test and p-value significance on quality aspects defined in Table 1. The p-values for (`Coherence, Relevance`), (`Coherence, FactualAccuracy`), and (`Relevance, FactualAccuracy`) are 0.635, 0.001, and 0.004, indicating clear distinctions among the aspects. **These quality aspects exhibit distinct characteristics** consistent with intuition, highlighting the importance of detailed factuality aspects in the medical domain. Sensitivity analysis of Equation 1 shows **a consistent trend: greater emphasis on factuality improves correlation with human judgment**. While the 2:1 weighting did not yield the highest correlation, it outperformed coherence-heavy settings and was chosen as an intuitive benchmark for a factuality-focused configuration (Table 5).

Metrics	Coh.	Rel.	FactAcc.	FWS
Auto	0.070	0.355	0.497	0.256
GPTJ	**0.701**	**0.881**	**0.613**	**0.839**

Table 6: (RQ3) Pearson correlation of Human evaluation in QA task. GPT-Judger (GPTJ) correlated strongly for each aspect and aligned better to human evaluation than traditional automatic metrics (Auto).

Consequently, Table 6 shows the Pearson correlation of human evaluation in the Question Answering task. The correlation for each aspect ranging from 0.6 to 0.8, indicate that **human ratings aligned closer to GPT-Judger (GPTJ)** than traditional automatic metrics (Auto), provided higher absolute values indicate stronger correlations.

Taken together, these analyses offer some key observations regarding the impact of watermarking on factuality, FWS reliability, and human alignment.

Observation 3:

- *Watermarking techniques introduce **significant factuality degradation**, with EXPEdit showing the most severe drops (up to 37%).*
- ***Factuality-weighted Scores (FWS)** based on GPT-Judger **offer clearer and more interpretable distinctions** than traditional automatic metrics.*
- ***Human evaluations confirm the distinct role of factuality dimensions beyond coherence** and show strong alignment with GPT-Judger, **validating the proposed evaluation framework**.*

5.4 Risk of Factuality Corruption (RQ4)

The previous subsection highlighted the degradation of factuality quality dimensions caused by watermarking techniques. In this subsection, we further investigate the potential contributing factors behind this quality drop. To ensure a fair analysis, we use natural text from the datasets as the reference, assuming it represents factuality from medical domain expertise.

Token entropy distribution In Figure 3, we observe a distinct pattern: the reference text's entropy is primarily concentrated in low-entropy regions. While the watermarked texts follow a similar overall shape, their **density shifts by approximately 10%, spreading across both low and mid-entropy** areas. Similar pattern of low-entropy distribution shifts for Question Answering and Summarization described in Appendix A.3. Given that this shift may have a non-trivial impact, specifically in the second entropy range (1.5–4), which hypothetically corresponds to medical terminology, we conduct a deeper analysis of low-entropy tokens and their types to assess the potential risk of reduced model confidence in predicting critical terms.

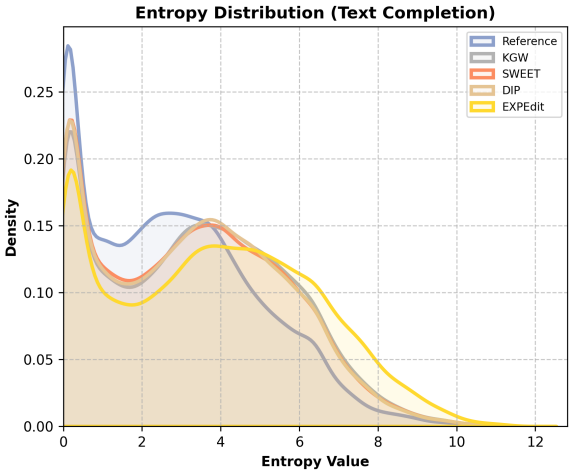


Figure 3: (RQ4) All watermarking methods cause token entropy distribution shifts, especially on the low-entropy side where two hills appear. EXPEdit shows the biggest change overall.

Entropy distribution by disease entity As shown in Figure 4, the reference token entropy for disease entities (Neumann et al., 2019) generally aligns with the second entropy range (2–4) in Figure 3. Within this range, we observe noticeable shifts in median token entropy across disease entities shown in the plot, such as *pain*, *infection*, and *cancer*. These shifts suggest a **degradation in the model's contextual confidence when handling critical domain-specific terms**. It also indicates that the more uncertain a model is on a disease entity, the more likely it is to hallucinate about that entity. This intuition will further be observed in a more detailed entity-level comparison.

Entity hallucinations Inspired by LLMs factuality hallucination described by Li et al. (2024), which categorizes erroneous entities as entity-error hallucination, we analyze factuality corruption in watermarked medical texts. Using a disease entity recognizer (Neumann et al., 2019), we measure *Introduced entities* (new disease entities not found in reference text) and *Hallucination rate* (percentage of semantically divergent entities). We compute pairwise contextual BERT embeddings with a co-

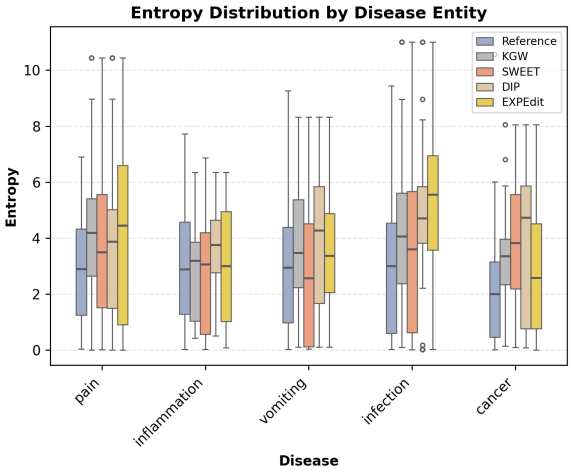


Figure 4: (RQ4) Entropy distribution by most frequent disease entities. Median shifts toward higher entropy reflect reduced contextual confidence in generating medical terminology.

sine similarity (Zhang et al., 2020) threshold of 0.6, which allows us to distinguish acceptable synonyms from true semantic shifts that corrupt the text's medical context.

Watermarking Methods	Avg. Introduced Entities	Increased Hallucination rate (%)
KGW	4.13	3.1
SWEET	4.39	3.2
DiPmark	4.10	**0.6**
EXP-edit	**3.63**	1.8

Table 7: (RQ4) Semantic corruption in watermarked texts measured by new entity introduction and hallucination rates. New disease entities occupy up to 11-13% of content and produce varying degrees of semantic distortion (0.6-3.2%), revealing how watermarking risks factual integrity in the medical domain.

Table 7 presents entity hallucination statistics across watermarking methods. On average, each watermarking technique introduces 3.6-4.4 new disease entities per 200 tokens of text, with each entity typically spanning about 6 tokens, meaning up to 11-13% of watermarked content consists of introduced medical terms. The increased hallucination rate, on the other hand, shows that **all methods led to higher hallucination rates compared to un-watermarked texts, ranging from 0.6% to 3.2%**. Samples of the hallucinated text can be found in Appendix A.4. These findings suggest watermarking methods risk their semantic preservation capabilities by introducing entity hallucinations, which further support the quality degradation result in the previous result Figure 2.

Observation 4:

- ***Entropy shift***: *token entropy distribution shifts by 10% in low and mid-entropy regions, disrupting natural language patterns.*
- ***Entity confidence***: *reduced model confidence when handling disease entities, potentially correlated with hallucination.*
- ***Entity hallucination***: *new disease entities introduced covering 11-13% of content and increased hallucination rates by up to 3.2%, creating semantic corruption of medical factual information.*

6 Conclusion

Our evaluation of LLM watermarking methods for medical texts reveals a critical trade-off: while current approaches achieve high detection capabilities, they compromise factual integrity. The proposed Factuality-Weighted Score addresses this concern by prioritizing factual accuracy over coherence, validated through human evaluations. Our analysis shows watermarking shifts token distributions, affecting medical terminology and introducing entity hallucinations.

These findings highlight the need for domain-aware watermarking techniques that preserve medical content integrity while maintaining detectability. We recommend developing approaches that specifically account for medical terminology alongside more robust factuality-focused evaluation protocols. As LLMs continue to be deployed in healthcare settings, ensuring content authenticity and factual reliability remains important.

Limitations

This study includes a human evaluation to validate the alignment between GPT-Judger assessments and human judgments, restricted to the Question Answering (QA) task. Our objective was not to establish comprehensive human evaluation across all tasks, but to assess the reliability of our proposed workflow, which we demonstrate through a strong correlation with human judgments in the QA setting. We also identified contributing factors to factuality degradation, such as entropy distribution shifts and entity hallucinations, while other potential factors likely exist beyond our analysis. The connections between these mechanisms and various watermarking approaches represent a promising area for further research toward developing more domain-sensitive watermarking techniques.

Ethical Statement

This work investigates the implementation of existing LLM watermarking methods in medical texts. We note that applying these methods without consideration of domain-specific factuality may contribute to misinformation. While this study provides high level insights of factuality concerns, further refinement to different use case might be necessary to ensure safe and responsible deployment in real-world medical applications.

References

Scott Aaronson and Hendrik Kirchner. 2022. Watermarking gpt outputs. *Scott Aaronson, 2022b. URL https://www. scottaaronson. com/talks/watermark. ppt*.

Asma Ben Abacha and Dina Demner-Fushman. 2019. On the summarization of consumer health questions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2228–2234.

Anirudh Ajith, Sameer Singh, and Danish Pruthi. 2024. Downstream trade-offs of a family of text watermarks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14039–14053.

Mansour Al Ghanim, Saleh Almohaimeed, Mengxin Zheng, Yan Solihin, and Qian Lou. 2024. Jailbreaking LLMs with Arabic transliteration and Arabizi. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18584–18600, Miami, Florida, USA. Association for Computational Linguistics.

Mansour Al Ghanim, Muhammad Santriaji, Qian Lou, and Yan Solihin. 2023. Trojbits: A hardware aware inference-time attack on transformer-based language models. In *ECAI 2023*, pages 60–68. IOS Press.

Zeming Chen, Alejandro Hernández-Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. Meditron-70b: Scaling medical pretraining for large language models. *Preprint*, arXiv:2311.16079.

Miranda Christ, Sam Gunn, and Or Zamir. 2024. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1125–1139. PMLR.

Weiping Fu, Bifan Wei, Jianxiang Hu, Zhongmin Cai, and Jun Liu. 2024. Qgeval: Benchmarking multi-dimensional evaluation for question generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11783–11803.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. 2023. On the learnability of watermarks for language models. *arXiv preprint arXiv:2312.04469*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Rochana Prih Hastuti, Rian Adam Rajagede, Mengxin Zheng, and Qian Lou. 2025. Clinic-prompt: Few-shot discrete clinical prompt optimization. In *Workshop on Large Language Models and Generative AI for Health at AAAI 2025*.

Xuanli He, Qiongkai Xu, Lingjuan Lyu, Fangzhao Wu, and Chenguang Wang. 2022. Protecting intellectual property of language generation apis with lexical watermark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10758–10766.

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

Mingjia Huo, Sai Ashish Somayajula, Youwei Liang, Ruisi Zhang, Farinaz Koushanfar, and Pengtao Xie. 2024. Token-specific watermarking with enhanced detectability and semantic coherence for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 20746–20767.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning*, 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. 2024. On the reliability of watermarks for large language models. In *ICLR*.

Cong Kong, Rui Xu, Weixi Chen, Jiawei Chen, and Zhaoxia Yin. 2024. Protecting copyright of medical

pre-trained language models: Training-free backdoor watermarking. *arXiv preprint arXiv:2409.10570*.

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

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickaël Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5848–5864.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2024. Who wrote this code? watermarking for code generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4890–4911, Bangkok, Thailand. Association for Computational Linguistics.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10879–10899, Bangkok, Thailand. Association for Computational Linguistics.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip Yu. 2024. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 57(2):1–36.

Qian Lou, Yepeng Liu, and Bo Feng. Trojtext: Test-time invisible textual trojan insertion. In *The Eleventh International Conference on Learning Representations*.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

Piotr Molenda, Adian Liusie, and Mark Gales. 2024. WaterJudge: Quality-detection trade-off when watermarking large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3515–3525, Mexico City, Mexico. Association for Computational Linguistics.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

OpenAI. 2023. New ai classifier for indicating ai-written text. Accessed: 2025-05-12.

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.

Karanpartap Singh and James Zou. 2024. New evaluation metrics capture quality degradation due to LLM watermarking. *Transactions on Machine Learning Research*.

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

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.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. 2024. A resilient and accessible distribution-preserving watermark for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 53443–53470.

Jiaqi Xue, Qian Lou, and Mengxin Zheng. 2024. BadFair: Backdoored fairness attacks with group-conditioned triggers. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8257–8270, Miami, Florida, USA. Association for Computational Linguistics.

Jiaqi Xue, Mengxin Zheng, Ting Hua, Yilin Shen, Yepeng Liu, Ladislau Bölöni, and Qian Lou. 2023. Trojllm: A black-box trojan prompt attack on large language models. *Advances in Neural Information Processing Systems*, 36:65665–65677.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348.

Hang Zhang, Qian Lou, and Yanshan Wang. 2025. Towards safe ai clinicians: A comprehensive study on large language model jailbreaking in healthcare. *arXiv preprint arXiv:2501.18632*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Mengxin Zheng, Jiaqi Xue, Xun Chen, Yanshan Wang, Qian Lou, and Lei Jiang. 2024. Trojfsp: Trojan insertion in few-shot prompt tuning. 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 1141–1151.

Ming Zhu, Aman Ahuja, Wei Wei, and Chandan K Reddy. 2019. A hierarchical attention retrieval model for healthcare question answering. In *The World Wide Web Conference*, pages 2472–2482.

A Additional Analysis

A.1 Model Analysis

Table 8 presents a comparison of watermarking methods across medical language models for text QA tasks: Meditron 7B[4], MedLlama-3 8B[5] and BioMistral 7B[6]. High detection metrics (TPR=1, AUROC=1) can be seen on smaller models (Meditron 7B and BioMistral 7B), with a decrease observed in the larger MedLlama-3 8B model. As expected, quality metrics improve with model size, with MedLlama-3 8B showing substantially higher SimCSE scores compared to smaller models, and better task performance in all metrics.

Watermarking Methods	Task		
	RG-2↑	F1↑	AS↑
	Meditron v1.0 7B		
KGW	.021	**.127**	**.273**
SWEET	.022	.125	.235
DiPmark	.021	.125	.254
EXP-edit	**.023**	.123	.238
	BioMistral 7B		
KGW	.026	.135	**.311**
SWEET	.028	.137	.286
DiPmark	**.030**	**.141**	.297
EXP-edit	.029	.135	.297
	MedLlama-3 8B		
KGW	.042	**.154**	.364
SWEET	.045	**.154**	**.387**
DiPmark	.045	**.154**	.373
EXP-edit	**.048**	.153	.375

Table 8: Performance comparison of watermarking methods across medical language models of different sizes for QA tasks.

When evaluating the quality using GPT-Judger, we still can see that the more recent models, BioMistral 7B and MedLlama-3 8B, still experience a quality drop as shown in Table 9.

A.2 Hyperparameter Effect on Detection

The parameters used for each watermarking method are shown in Table 10. These are the default parameters used in MarkLLM[7] based on each original paper. For details on the usage of each parameter, please refer to the original papers.

Additionally, we conducted a systematic investigation into how hyperparameter values affect detection performance. First, we explored the impact of varying $\delta \in \{0.5, 1, 2\}$ and $\gamma \in \{0.1, 0.25, 0.5\}$ in logit-based watermarking, KGW, and SWEET,

[4]https://hf.co/epfl-llm/meditron-7b
[5]https://hf.co/johnsnowlabs/JSL-MedLlama-3-8B-v2.0
[6]https://hf.co/BioMistral/BioMistral-7B
[7]https://github.com/THU-BPM/MarkLLM

Watermarking Methods	GPT-Judger Quality Drop (%)		
	Coherence↓	Relevance↓	Factual Accuracy↓
	Meditron v1.0 7B		
KGW	6.9	**4.8**	**4.8**
SWEET	**6.2**	13.7	7.5
DiPmark	9.6	7.4	8.1
EXP-edit	20.9	17.2	16.3
	BioMistral 7B		
KGW	12.4	12.8	12.1
SWEET	8.5	9.2	6.6
DiPmark	**3.3**	**5.9**	**6.0**
EXP-edit	17.9	24.1	15.9
	MedLlama-3 8B		
KGW	7.5	9.7	**3.7**
SWEET	9.5	15.0	11.9
DiPmark	**6.9**	**5.3**	5.3
EXP-edit	7.5	5.8	6.3

Table 9: Performance comparison of watermarking methods across medical language models of different sizes for QA tasks.

Watermarking Methods	Parameters
KGW	$\gamma = 0.5, \delta = 2.0$
SWEET	$\gamma = 0.5, \delta = 2.0$, entropy_threshold $= 0.9$
DiPmark	$\gamma = 0.5, \alpha = 0.45$
EXP-edit	pseudo_length $= 256$

Table 10: Hyperparameters used in main experiments for each watermarking method.

with results presented in Figure 5. For these experiments, we utilized BioGPT[8] (Luo et al., 2022) model, which is computationally less demanding than the larger models used in our main analysis, enabling more efficient hyperparameter exploration. Our findings demonstrate that for both KGW and SWEET schemes, larger values of δ and γ correspond to improved detection rates. However, as discussed in § 5.1, there exists a fundamental trade-off between detection efficacy and the quality of generated content.

EXP-edit parameters	TPR	AUROC
pseudo_length $= 100$	**0.890**	**0.974**
pseudo_length $= 200$	0.875	0.938
pseudo_length $= 300$	0.830	0.949

Table 11: TPR and AUROC for EXP-edit with different pseudo length (n) in Text Completion task

Hyperparameter exploration for EXP-edit and DiPmark is shown in Table 11 and Table 12, respectively. We can see that EXP-edit achieves optimal performance with shorter pseudo lengths ($n = 100$), yielding TPR of 0.890 and AUROC of 0.974. While DiPmark demonstrates consistently higher detection rates with its best configuration at $\alpha = 0.45$, achieving near-perfect results

[8]https://hf.co/microsoft/biogpt

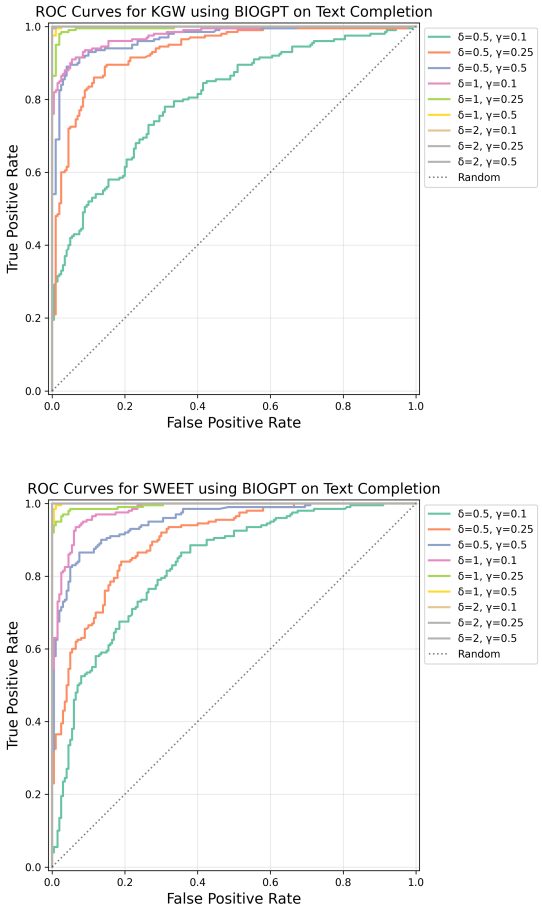


Figure 5: ROC Curve for KGW (top) and SWEET (bottom) with different δ and γ in Text Completion task.

DiPmark parameters	TPR	AUROC
$\alpha = 0.4$	0.985	0.990
$\alpha = 0.45$	**0.995**	**0.999**
$\alpha = 0.5$	0.980	**0.999**

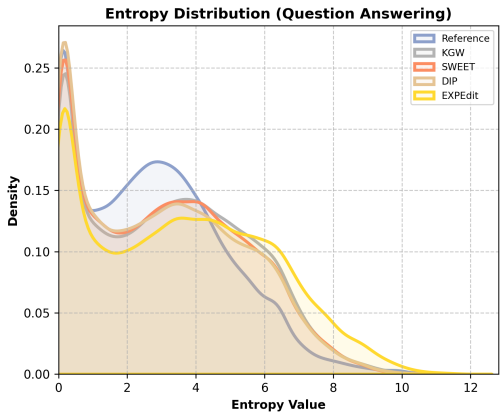
Table 12: TPR and AUROC for DiPmark with different α threshold in Text Completion task

(TPR=0.995, AUROC=0.999).

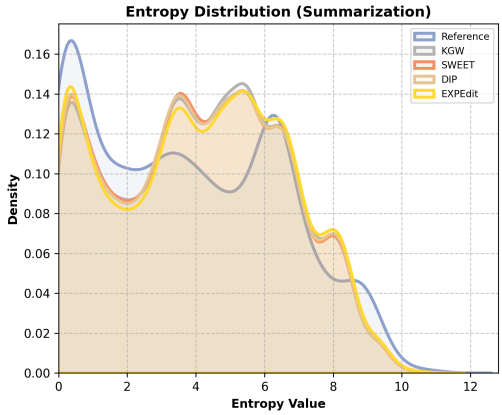
A.3 Risk of Factuality Corruption

Token entropy shifts in Question Answering task are more profound in the mid-region entropy as seen in Figure 6a, where medical terms supposedly exist. This pattern differs in Summarization task where the increase shifts present in the low-entropy region, and significant decrease shifts exist in the mid-entropy region as seen in Figure 6b.

This difference is mainly related to each task having a different text length, QA with 200 tokens and Summarization with fewer than 100 tokens. The QA task, naturally giving the easier prompt for a language model to understand, will generate



(a) Question Answering



(b) Summarization

Figure 6: All watermarking methods cause the entropy distribution to shift in low-to-mid entropy regions for Question Answering and Summarization.

straightforward answers related to medical topics, thus the general distribution is more preserved than in Summarization task. The summarization, on the other hand, is a question summarization task, which is a more difficult and less straightforward task. The language model has to understand the medical topic and then needs to summarize it according to the given short context limitation, which also potentially becomes the lead of the distorted generated text.

A.4 Hallucination Samples

In Table 13, we showed two samples where the watermarked text introduces hallucination, according to low BERT similarity score. We use a text completion task and pick samples from watermarked text generated using KGW and SWEET. We can see from the table that the extracted entities from the un-watermarked text are still relevant to the original prompt. In contrast, the watermarked texts introduce unrelated terms to the original prompt.

Sample #1
Prompt: short time . Decongestant tablets or syrup . For example, a medication called pseudoephedrine. This can be obtained from a pharmacy without a prescription. Take the
Un-watermarked Extracted Entities: pseudoephedrine
KGW-Watermarked Extracted Entities: pseudoephedrine, amphetamine, ®, amitriptyline, chlorpheniramine, amylopectin chlorophyllin chlorphenoxyethanone, chlorphenamine, low back pain, sciatica, lower leg numbness, maleate, amiloride, allopurinol/chlorphenaminesine
Hallucination: One sample from the extracted entities, Amphetamine, is considered below threshold with score -0.204. Amphetamine is not related to decongestant or pseudoephedrine
Sample #2
Prompt: to develop as a result of long - standing (chronic) stress and irritation of a plantar digital nerve . This may be due to the nerve being squashed
Un-watermarked Extracted Entities: trauma, haemorrhage, loss of skeletal muscle mass, muscle wasting diseases, obesity, muscle mass, muscle loss
SWEET-Watermarked Extracted Entities: shoe - counter, fection, infection, chancre, tbc, herpes zoster, lupus, lupus erythematosus, psoriasis, arthritis, leprosy, verruca, ings
Hallucination: One sample from the extracted entities, TBC, is considered below threshold with score 0.299. From the prompt, the discussion is about nerve in the foot, which is unrelated to TBC

Table 13: Hallucination example in Text Generation task.

B Implementation Details and Methods Preliminary

B.1 GPT-Judger

The prompt for GPT-Judger formerly referred to the template of Singh and Zou (2024), where the aspects are tailored for the factuality dimension designed in Table 1. We add detailed questions following each aspect to further trigger deeper reasoning. These questions per aspect are also used as in the Human Evaluation questionnaire detailed in Appendix B.2. The example of prompt and response of GPT-Judger for Summarization task is available in Table 14.

We use OpenAI GPT-4o-2024-08-06 as an LLM evaluator, and in total, we spent around 60 USD on the API usage during experiments.

B.2 Human Evaluation

The human evaluation is conducted for the QA task as the nature of the task is typically easier for humans to interpret and rate based on the ground-truth answer, compared to other tasks like Text

Table 14: GPT-Judger prompt and example response set for Summarization task. We utilize GPT4o-2024-08-06.

PROMPT	**JUDGER RESPONSE**
Please act as an impartial judge and evaluate the quality of the Summarized Medical Question provided by two large language models to the prompt displayed below. Assess each response according to the criteria outlined, using a 1-5 Likert scale where 1 indicates strong disagreement or the lowest quality, and 5 indicates strong agreement or the highest quality. Criteria: 1. Coherence. Is the language coherence, clear, understandable for a general audience? 2. Completeness. Does the generated summary miss any important information from the original question? 3. Factual Accuracy. Does the generated summary introduce any inaccurate or unrelated medical terms not found in the original question? After scoring each criterion, provide a short summary for each response, including specific examples that influenced your scoring. Additionally, don't let the length of the responses influence your evaluation. Be as objective as possible and ensure that the order in which the responses are presented does not affect your decision. Start with a brief statement about which response you think is superior. Then, for each response and criterion, provide a score, followed by a brief justification for that score. At the very end of your response, declare your verdict by choosing one of the choices below, strictly following the given format: [[A]]: [list of scores for LLM A output, in order of Coherence, Completeness, Factual Accuracy] [[B]]: [list of scores for LLM B output, in order of Coherence, Completeness, Factual Accuracy]	Overall Superior Response: LLM A provides a more detailed and coherent summarized question while maintaining clarity and relevance to the original query LLM A Evaluation: Coherence: 4 - The language used in LLM A's response is generally clear and understandable. However, the addition of the word "or" between "depression" and "migraines" introduces slight ambiguity, slightly reducing coherence. Completeness: 5 - LLM A captures all the critical elements of the original question, such as the focus on vagal nerve stimulation and its potential to alleviate both depression and migraines. Factual Accuracy: 5 - The summary does not introduce any inaccurate medical terms or unrelated concepts. It stays true to the original topic, discussing the relevance of vagal nerve stimulation to depression and migraines. LLM B Evaluation: ⋮
[Prompt] [LLM A's Answer] [LLM B's Answer]	Verdict [[A]]: [4, 5, 5] [[B]]: [3, 3, 4]

Completion. Sample of questionnaire shown in Figure 7. We collected results from 6 respondents (4 graduate students and 2 medical practitioners) to evaluate sample of 10% of the total generated texts, while making sure each item is rated by 3 different respondents. This study has been approved by IRB under exempt human subject research with IRB ID STUDY00007853. Respondents were recruited through social media groups and domain background were checked later. All participation was voluntary and was clearly stated in the consent page of the questionnaire.

B.3 Tasks Details

B.3.1 Text Completion

For this task, we utilize the HealthQA dataset (Zhu et al., 2019), which consists of medical question-answer pairs. We focus solely on the answer portions, selecting 200 answers that exceed 230 words in length. From each answer, we extract the last 230 words, using the first 30 words as the input prompt and tasking the model with generating the subsequent 200 words. This experimental design aligns with prior watermarking evaluation frameworks (Kirchenbauer et al., 2023), allowing for consistent comparison across domains.

B.3.2 Question Answering

We employ the same HealthQA dataset (Zhu et al., 2019) for our QA task, but utilize both question and answer components. To maintain consistency with the Text Completion task, we select 200 data points containing questions of exactly 10 words and answers of fewer than 250 words. The questions (ending with ?) serve as input prompts without additional instructions, while the corresponding answers constitute the ground truth for evaluation.

B.3.3 Summarization

For the Summarization task, we employ the MeQSum dataset (Abacha and Demner-Fushman, 2019), which provides pairs of detailed clinical questions and their concise summaries. We construct model prompts by prefixing the clinical question text with `Write a short question that summarizes this question:` and appending `Summarized Question:` after the question text. To introduce diversity, we limit our selection to inputs with a maximum of 60 words and summaries containing at least 10 words, creating more variation in input-output length ratios compared to the other tasks.

B.4 Watermarking Methods

B.4.1 KGW (Kirchenbauer et al., 2023)

We provide brief preliminaries for KGW watermarking approach. For a given language model f_{LM} with vocabulary $\mathcal{V}$, the likelihood probability of a token y_t is calculated as follows:

$$l_t = f_{\text{LM}}(x, y_{[:t]}) \tag{2}$$

$$p_{t,i} = \frac{e^{l_{t,i}}}{\sum_{j=1}^{|\mathcal{V}|} e^{l_{t,j}}} \tag{3}$$

where $x = x_0, \ldots, x_{M-1}$ and $y_{[:t]} = y_1, \ldots, y_{t-1}$ are an M-length tokenized prompt and the generated token sequence respectively, and $l_t \in \mathbb{R}^{|\mathcal{V}|}$ is the logit vector.

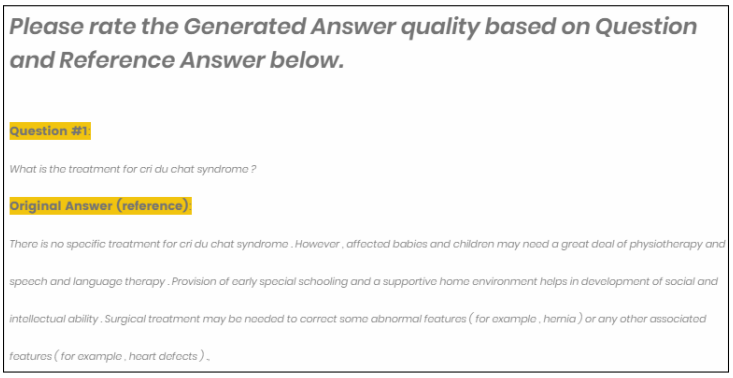
Watermarking In the watermarking procedure, the entire tokens in $\mathcal{V}$ at each time-step are randomly binned into green ($\mathcal{G}_t$) and red ($\mathcal{R}_t$) groups in proportions of γ and $1-\gamma$ ($\gamma \in (0,1)$), respectively. The method increases the logits of green group tokens by adding a fixed bias δ, promoting these tokens to be sampled at each position. Consequently, watermarked LM-generated text is more likely than γ to contain tokens from the green group. In contrast, since humans have no knowledge of the hidden green-red partition rule, the proportion of green group tokens in human-written text is expected to be close to γ.

Detection The presence of watermarking in text is detected through a one-sided z-test by testing the null hypothesis that the text is not watermarked. The z-score is calculated using the number of recognized green tokens in the test text. Subsequently, the text is determined to be watermarked if the z-score exceeds a predetermined threshold $z_{\text{threshold}}$.

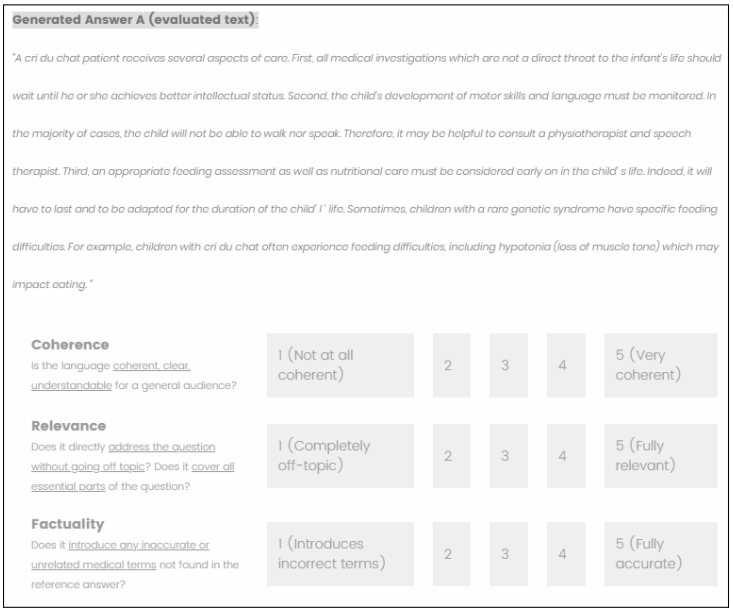
B.4.2 SWEET (Lee et al., 2024)

SWEET improve KGW (Kirchenbauer et al., 2023) by distinguishing watermark applicable tokens, meaning it embed and detect watermarks only within tokens with high entropy.

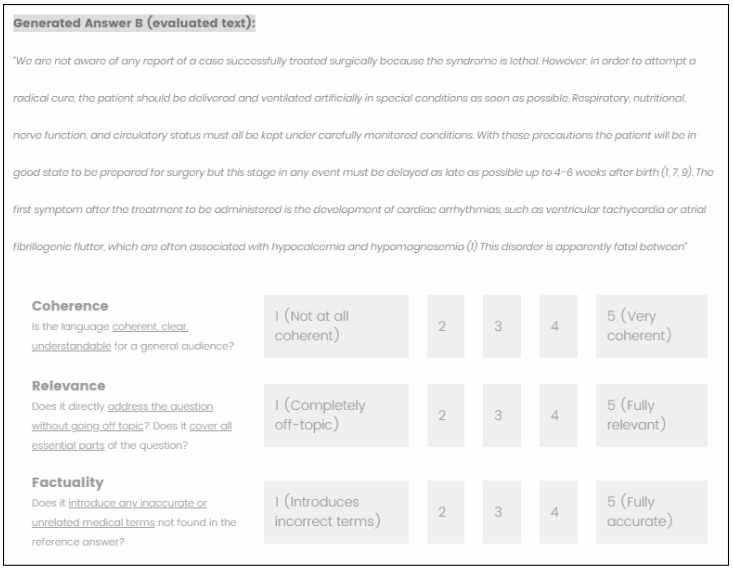
Watermarking Given a tokenized prompt $x = x_0, \ldots, x_{M-1}$ and already generated tokens $y_{[:t]} = y_0, \ldots, y_{t-1}$, a model calculates an entropy value (H_t) of the probability distribution for y_t.The watermarking only applied when H_t exceeds the threshold, τ. Then, randomly split the vocabulary into green and red groups with a fixed green token ratio γ. If a token is selected to be watermarked, a constant δ is added to the logits of green tokens, thereby promoting the sampling of these tokens. By



(a) Reference answer help non-expert respondent evaluate the watermarked text.



(b) Generated Answer A: watermarked text or un-watermarked text.



(c) Generated Answer B: watermarked text or un-watermarked text.

Figure 7: Questionnaire for human evaluation on Question Answering task. Respondents were presented Ground-truth text from the dataset, Watermarked text, and Un-watermarked text at the same page to fairly evaluate each generated text. Positioning for the generated text was randomized to prevent bias.

limiting the promotion of green tokens only to positions with high entropy, it prevent modifications to the model's logit distribution for tokens where the model has high confidence (and, therefore, low entropy).

Detection Given a token sequence $y = y_0, \ldots, y_{N-1}$, the task is to detect watermarks within y, thereby determining whether it was generated by the specific language model. As in the generation phase, SWEET computes the entropy values H_t for each y_t. Let N_h denote the number of tokens that have an entropy value H_t higher than the threshold τ, and let N_h^G denote the number of green tokens among those in N_h. Finally, with the green list ratio γ used in the generation step, SWEET computes a z-score under the null hypothesis that the text is not watermarked:

$$z = \frac{N_h^G - \gamma N_h}{\sqrt{N_h \gamma (1-\gamma)}} \tag{4}$$

The presence of a watermark can be assessed with increasing confidence as the z-score increases. SWEET sets $z_{\text{threshold}}$ as a cut-off score. If $z > z_{\text{threshold}}$ holds, it determines that the watermark is embedded in y and thus the text was generated by the LLM.

B.4.3 EXP-edit (Kuditipudi et al., 2024)

This method apply the watermarking via Exponential Minimum Sampling (Aaronson and Kirchner, 2022) with slight modification of detection using Levenshtein Alignment.

Watermarking Given a tokenized prompt $x = x_0, \ldots, x_{M-1}$ and already generated tokens $y_{[:t]} = y_0, \ldots, y_{t-1}$, a watermark key sequence $\xi \in [0,1]^N$ is used to deterministically map to samples from the language model. The decoder function Γ for each token is defined as:

$$\Gamma(\xi, \mu) := \arg\min_{i \in [N]} -\log(\xi_i)/\mu(i) \tag{5}$$

where $\mu \in \Delta([N])$ is the probability distribution over the vocabulary from the language model. This decoder is distortion-free, as marginalizing over the watermark key sequence, the distribution of generated tokens remains equivalent to sampling directly from the language model.

Detection Given a token sequence $y = y_0, \ldots, y_{N-1}$ that might be watermarked, a Levenshtein alignment cost function d_γ allows for robust detection even when the text has been modified through insertions or deletions:

$$d_\gamma(y, \xi) := \min \begin{cases} d_\gamma(y_{2:}, \xi_{2:}) + \\ d_0(y_1, \xi_1)\ d_\gamma(y, \xi_{2:}) + \\ \gamma\ d_\gamma(y_{2:}, \xi) + \\ \gamma \end{cases} \tag{6}$$

where $d_0(y, \xi) = \sum_{i=1}^{\text{len}(y)} \log(1 - \xi_{i,y_i})$ is the base alignment cost for EXP, and γ is a hyperparameter controlling the penalty for insertions and deletions. For EXP-edit, the optimal value is $\gamma = 0.0$. The detection algorithm computes a test statistic by finding the minimum cost alignment between blocks of the input text and the watermark key sequence. A permutation test is then used to generate a p-value by comparing this test statistic against randomly generated watermark keys, determining whether the text is likely watermarked.

B.4.4 DiPmark (Wu et al., 2024)

DiPmark operates on a vocabulary set V with size $N = |V|$ and employs a distribution-preserving reweight strategy that modifies token probabilities without changing the original distribution $P_M(x_{n+1}|x_{1:n})$.

Watermarking During generation, the system derives a texture key s_i from previously generated tokens and combines it with a secret key k to produce a cipher $\theta_i = h(k, s_i)$ using a hash function h. This cipher represents a permutation of the vocabulary tokens. The core of DiPmark is its DiP-reweight strategy, defined as:

$$\begin{aligned} &P_W(t_i|x, \theta) := \\ &(1-\alpha) P_W^{\alpha}(t_i|x, \theta) + \alpha P_W^{1-\alpha}(t_i|x, \theta) \end{aligned} \tag{7}$$

where $\alpha \in [0,1]$ is the reweight parameter and P_W^{α} adjusts token probabilities within interval $[0, \alpha]$ to 0 and scales others by $\frac{1}{1-\alpha}$. The next token is then sampled according to this modified distribution. This approach mathematically guarantees that $E_\theta[P_W(t|x,\theta)] = P_M(t|x)$, ensuring the original distribution is preserved across multiple generations, which distinguishes DiPmark from other watermarking approaches.

Detection The detection process uses a statistical test based on the "green token ratio" to identify watermarked text. For a given text sequence $x_{1:n}$, the detector computes the cipher θ_i for each position using the same hash function and secret key. Each permutation θ_i is divided using a green list

separator $\gamma \in [0, 1]$, with the last $(1 - \gamma)N$ tokens in the permutation designated as "green tokens."

The detector counts the number of green tokens $L_G(\gamma)$ in the sequence and calculates the green token ratio:

$$\Phi(\gamma, x_{1:n}) := \frac{L_G(\gamma)}{n} - (1 - \gamma) \tag{8}$$

This ratio is compared against a threshold z; if $\Phi > z$, the text is classified as watermarked.