# Topic-Based Watermarks for Large Language Models

**Alexander Nemecek**[1]*, **Yuzhou Jiang**[1], **Erman Ayday**[1]
[1]Case Western Reserve University

ACL Anthology ID 2026.findings-acl.1220 / revision 2 / 04 Sep 2026

## Abstract

The indistinguishability of large language model (LLM) output from human-authored content poses significant challenges, raising concerns about potential misuse of AI-generated text and its influence on future model training. Watermarking algorithms offer a viable solution by embedding detectable signatures into generated text. However, existing watermarking methods often involve trade-offs among attack robustness, generation quality, and additional overhead such as specialized frameworks or complex integrations. We propose a lightweight, topic-guided watermarking scheme for LLMs that partitions the vocabulary into topic-aligned token subsets. Given an input prompt, the scheme selects a relevant topic-specific token list, effectively "green-listing" semantically aligned tokens to embed robust marks while preserving fluency and coherence. Experimental results across multiple LLMs and state-of-the-art benchmarks demonstrate that our method achieves text quality comparable to industry-leading systems and simultaneously improves watermark robustness against paraphrasing and lexical perturbation attacks, with minimal performance overhead. Our approach avoids reliance on additional mechanisms beyond standard text generation pipelines, enabling straightforward adoption and suggesting a practical path toward globally consistent watermarking of AI-generated content. Our code is available at https://github.com/ANCP2021/Topic-Based-Watermarks.

## 1 Introduction

The rapid expansion of Large Language Model (LLM) capabilities has led to unprecedented accuracy and fluency in tasks such as text generation, summarization, and dialogue. Models like OpenAI's ChatGPT (OpenAI, 2022) and Google's Gemini (Google, 2024) can produce text nearly indistinguishable from human-authored content. While these advancements offer significant benefits across multiple domains, they also pose security and ethical challenges. One central issue is the misuse of LLM-generated text for malicious purposes, such as misinformation, copyright infringement, or plagiarism (Chen and Shu, 2024; Mueller et al., 2024; Lee et al., 2023). Additionally, many large-scale language models are trained on massive corpora scraped from the web, and the prevalence of LLM-generated data raises concerns about "model collapse," wherein repeatedly ingesting AI-generated text as training data leads to a gradual erosion in quality (Shumailov et al., 2024).

In response to these growing concerns, the research community has focused on methods for reliably attributing text to its source, specifically, to distinguish LLM-generated from human-authored text (Li et al., 2021; OpenAI, 2023; Mitchell et al., 2023). Early approaches to text attribution primarily involved training classification-based detection methods on labeled corpora of human- and machine-generated text. While such classifiers can achieve respectable accuracy under controlled conditions, they are often susceptible to adversarial paraphrasing or stylistic alterations (Liang et al., 2023). Furthermore, these methods rely on maintaining large, curated training sets that reflect the rapidly evolving landscape of LLMs which in turn pose substantial scalability challenges.

Researchers have explored *watermarking* as a complementary or alternative solution. Rather than detecting AI-authored text post-hoc, watermarking algorithms embed a detectable signature into text during the generation process (Kirchenbauer et al., 2023; Scott Aaronson, 2023). By introducing controlled token-level modifications, watermarks can remain identifiable even after moderate transformations. However, existing approaches face trade-offs where methods prioritizing computational efficiency lack robustness to paraphrasing (Dathathri

*Correspondence: ajn98@case.edu.

et al., 2024), while more robust techniques require costly architectural modifications or multiple inference passes that degrade practical deployability (Huo et al., 2024; Kuditipudi et al., 2024).

**Our Contributions.** We propose a *lightweight, topic-based watermarking (TBW) scheme* that addresses these limitations by integrating semantic information into the watermarking process while preserving generation pipeline simplicity. Rather than randomly partitioning the vocabulary, we map tokens to predefined topic embeddings. During generation, we **(i)** identify relevant topics from the input prompt, **(ii)** select the corresponding semantically-aligned "green" token list, and **(iii)** bias generation toward these thematically appropriate tokens. This semantic alignment preserves fluency while embedding robust watermark signals that resist paraphrasing attacks. We evaluate TBW on robustness, text quality, and efficiency, demonstrating performance comparable to leading techniques with perplexity on par with current production-grade systems, all without additional inference steps beyond standard generation pipelines.

## 2 Related Work

Traditional AI-generated text detection relied on post-hoc classifiers trained to differentiate human and machine writing, but these often fail under adversarial transformations like paraphrasing. To address this, watermarking techniques have gained prominence as a more robust alternative, categorized into post-processing and generation methods. **Post-processing watermarking** modifies generated text to embed hidden patterns but remains vulnerable to edits that erase markers (Sato et al., 2023). **Generation watermarking** embeds signals during token generation, ensuring greater resilience against text modifications.

The KGW algorithm (Kirchenbauer et al., 2023) introduced model vocabulary partitioning into "green" and "red" lists, biasing sampling toward green tokens. While computationally efficient, it remains vulnerable to paraphrasing and token perturbation. The Unigram watermark (Zhao et al., 2024) improves detection by assigning tokens based on unigram statistics rather than random partitioning. However, both methods suffer from partition recovery attacks where adversaries can approximate token subsets through repeated querying (Jovanović et al., 2024).

Google's SynthID-Text (Dathathri et al., 2024) employs Tournament Sampling, ranking candidate tokens using random watermarking functions before selecting the highest-scoring option, designed for minimal generation overhead and enhanced text quality. While scalable, SynthID-Text has shown limited resistance to paraphrasing and token perturbations. Similarly, DiPmark (Wu et al., 2024), another lightweight biasing-based watermark, struggles under text rewriting.

To enhance robustness while preserving text quality, the SIR watermark (Liu et al., 2024) incorporates user-provided context to guide token selection, achieving greater durability at the cost of requiring decoder modifications and prompt access, hindering adoption in large-scale commercial LLMs. While semantic-based watermarks (Liu et al., 2024; Hou et al., 2024a; Lee et al., 2024) can offer greater robustness, they still remain vulnerable to paraphrasing attacks.

In pursuit of stronger detection resilience, EXP, EXP-Edit, and ITS-Edit (Kuditipudi et al., 2024) introduce iterative decoding or re-ranking to strengthen watermark robustness. While these methods improve detection under adversarial conditions, they significantly increase computational overhead, potentially degrading fluency or inflating perplexity. Beyond these iterative approaches, some watermarking algorithms employ architectural changes that may survive more aggressive attack models (e.g., heavy paraphrasing, near-synonym substitutions), but at the cost of increased generation time and heightened implementation complexity (Zhang et al., 2024; Liu and Bu, 2024), making them less attractive for practical deployments in latency-sensitive applications.

**Overall gaps** persist: in-generation methods balance robustness, computational cost, and text quality. KGW-like and SynthID-Text approaches minimize perplexity shifts but struggle against adversarial edits, whereas more durable methods require costly repeated passes or model modifications. Our work occupies this middle ground by integrating semantic information without adding significant complexity, enabling efficient watermarking with improved resilience.

## 3 Preliminaries

We introduce the notation and key concepts utilized in our topic-guided watermarking approach.

### 3.1 Notation and Setup

Let $V$ denote the vocabulary of an LLM with parameters $\theta$. Each token $v \in V$ is associated with an embedding $e_v \in \mathbb{R}^d$, typically obtained from the model's embedding layer or another off-the-shelf embedding source. We assume a predefined set of topics $\{t_1, t_2, \ldots, t_K\}$ with corresponding embedding $e_{t^i} \in \mathbb{R}^d$.

As a preparatory step, we assign each token $v$ to exactly one topic-aligned list based on its semantic similarity. Specifically, for each token $v$, we compute

$$\text{sim}(v, t_i) = \frac{e_v \cdot e_{t_i}}{\|e_v\| \, \|e_{t_i}\|}, \tag{1}$$

and compare this value to a threshold $\tau$. If $\text{sim}(v, t_i) \geq \tau$ for some topic $t_i$, then $v$ is appended to $t_i$'s list. Tokens not meeting or exceeding $\tau$ for any topic are evenly distributed among all topic lists in a round-robin fashion to ensure balanced coverage. Thus, each topic list $G_{t_i} \subseteq V$ serves as the "green list" for $t_i$, analogous to the subsets in prior watermarking schemes such as KGW. The number of topics $K$ determines each active green list's vocabulary fraction, analogous to KGW's $\gamma$ parameter. For example, $K = 4$ yields an effective fraction of $0.25$, comparable to $\gamma = 0.2$ used in KGW's reported experiments.

During text generation, given an input prompt $x^{\text{prompt}}$, an LLM predicts the next token from a probability distribution $p_\theta(v \mid x^{\text{prompt}})$ over $V$. To embed a watermark, we can reweight this distribution toward tokens belonging to a chosen topic list. Specifically, we extract the most relevant topics from $x^{\text{prompt}}$ using a lightweight topic extraction model, then use $k$-means clustering to map them to the closest predefined topic if no direct match is found. This yields a single "green list" $G_{t^*}$, which is then biased during the generation process.

### 3.2 Threat Model

We assume a standard black-box adversary with access only to LLM outputs, seeking to remove watermarks while preserving semantic fidelity. Following (Kirchenbauer et al., 2023; Zhang et al., 2024), the adversary cannot access model parameters, watermarking keys, or topic-partition logic. While the detection algorithm is public, specific parameters (thresholds, embeddings) remain private. We evaluate robustness against two primary attack surfaces: **(1) text degradation** via token-level perturbations (e.g., substitutions) that disrupt the watermark signal at the cost of fluency; and **(2) semantic rephrasing** using external models for paraphrasing or summarization to rewrite content while maintaining meaning.

We additionally consider **(3) partition recovery attacks**, where an adversary attempts to reconstruct the topic-aligned green lists by exploiting the semantic structure of TBW's vocabulary partitioning. Prior work has demonstrated such attacks against KGW-style schemes through repeated positional querying (Jovanović et al., 2024). While TBW's topic-aligned partitions introduce semantic co-grouping that could appear more predictable than random assignment, the full partition remains unrecoverable without the private random seed used during round-robin distribution of residual tokens (§4). Detailed adversary assumptions, excluded attack vectors (e.g., statistical comparison), detection-based attacks, and the full partition recovery analysis are provided in B.

## 4 Proposed Method

In this section, we detail our lightweight topic-based watermarking (TBW) scheme. §4.1 explains how tokens are mapped to topic-aligned subsets, which serve as "green lists" for watermarking. In §4.2, we describe the generation procedure for embedding watermarks. §4.3, presents multiple watermark detection algorithms that identify watermarked text.

### 4.1 Token-to-Topic Mappings

We begin by clustering tokens in the LLM vocabulary $V$ into semantically aligned lists, each associated with one of a small set of high-level "generalized topics" $\{t_1, \ldots, t_K\}$. For illustration, one might choose topics such as {`animals`, `technology`, `sports`, `medicine`} to capture common themes. Using a sentence embedding model (e.g., all-MiniLM-L6-v2 (Reimers and Gurevych, 2020)), we encode each token $v \in V$ into an embedding $\mathbf{e}_v$ and compute its similarity to each topic embedding $\mathbf{e}_{t_i}$: $\text{sim}(v, t_i)$. If the maximum similarity across all topics exceeds a threshold $\tau$, the token is assigned to the corresponding topic's "green list" $G_{t_i}$. Tokens that do not exceed $\tau$ for any topic are collected into a residual set, which is subsequently distributed among $\{G_{t_1}, \ldots, G_{t_K}\}$ in a round-robin fashion. This ensures comprehensive coverage of the entire vocabulary, preventing any token from being discarded.

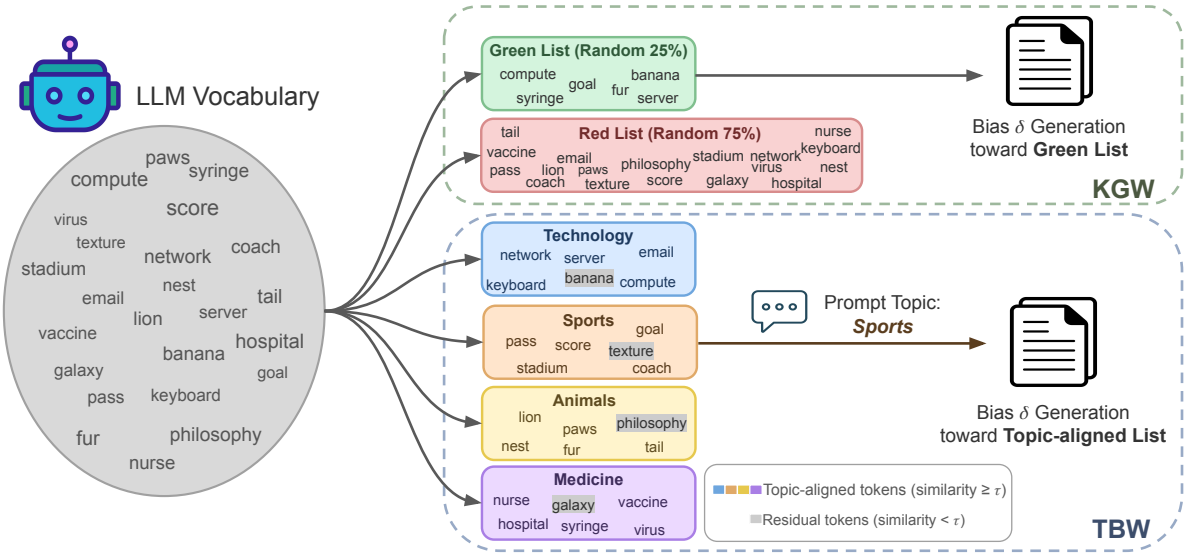



Figure 1: **Comparison of KGW and TBW vocabulary partitioning.** KGW (top) randomly partitions vocabulary $V$ into green/red lists using parameter $\gamma$. TBW (bottom) creates semantically meaningful partitions by assigning tokens to predefined topic lists. Prompts are mapped to corresponding topics, making that topic list the active green list. Both methods bias generation toward green lists using parameter $\delta$ to adjust logit values.

The hyperparameter $\tau$ controls the granularity of semantic alignment and comprehensive topic coverage where a higher $\tau$ enforces stronger coherence but increases the proportion of tokens allocated via the round-robin mechanism. Although we consider only four broad topics in our main evaluations, the same procedure naturally extends to a larger number of topics for more fine-grained coverage as the vocabulary $V$ increases. We use a general-purpose sentence model (all-MiniLM-L6-v2) for this implementation as a lightweight approach; however, any semantic embedding framework can be substituted, allowing practitioners to tailor the mapping for domain-specific or resource-constrained environments. Figure 1 illustrates this token-to-topic assignment process, with the detailed implementation procedure provided in Algorithm 1 (A).

### 4.2 Topic-Based Watermarks

Building on the token-to-topic mappings, we now detail how to embed watermarks during text generation by selectively biasing tokens in a single topic list. This procedure is analogous to the KGW scheme, where a targeted subset of the vocabulary (the "green list") receives a higher sampling probability. However, in our approach, the specific green list is chosen via a semantic matching process that depends on the user's prompt.

Given an input prompt $x^{\text{prompt}}$, we first identify relevant keywords or topics using a lightweight extractor (KeyBERT (Grootendorst, 2020)). If one or more of these extracted topics $\mathcal{T}_{\text{detected}}$ exactly matches an entry in the predefined set $\{t_1, \ldots, t_K\}$, we select the corresponding list $G_{t^*}$. Otherwise, we cluster the detected topic embeddings into a few centroids and compute their cosine similarity to each $t_i$'s embedding. We designate the topic list whose embedding is most similar to the centroid as $G_{t^*}$. This ensures that even if an exact match is unavailable, the system still picks the most semantically aligned topic.

At each generation step, the model produces logits $p_\theta(v \mid x^{\text{prompt}}, \mathbf{z})$ over the vocabulary $V$. We add a small bias $\delta$ to all tokens $v \in G_{t^*}$ before normalizing with a softmax function. Intuitively, this raises the selection probability of tokens in $G_{t^*}$, embedding a watermark without introducing multiple decoding passes or inflating perplexity. A larger $\delta$ yields a more robust watermark signal at the cost of potentially more noticeable shifts in text style or quality. After adjusting logits, the model samples the next token via standard methods (e.g., top-$k$ sampling, beam search). The entire generation loop is detailed in Algorithm 2 (A), highlighting that topic extraction and logit biasing constitute minimal overhead compared to typical LLM pipelines.

### 4.3 Topic-Based Detection

While our watermarking approach embeds topic-aligned signals during generation, practical detection faces challenges including topic ambiguity,

topic drift within longer documents, and potential generation-detection misalignment. To address these challenges systematically, we propose three detection schemes with increasing robustness and decreasing reliance on perfect topic knowledge.

All detection schemes, compute a $z$-score as:

$$z = \frac{g - \gamma \cdot n}{\sqrt{n \cdot \gamma \cdot (1 - \gamma)}} \tag{2}$$

where $g$ is the observed green token count, $n$ is the total tokens, and $\gamma$ is the expected green token fraction. Text is classified as watermarked if $z$ exceeds a threshold.

#### 4.3.1 Strict Topic Matching

Our baseline detection scheme assumes consistent topic alignment between generation and detection phases, establishing an upper bound on performance while highlighting the challenges that arise when this assumption fails in practice.

Given text $z_{test}$, we extract topics using KeyBERT. The process follows a hierarchical matching strategy: if extracted topics $T_{detected}$ directly match predefined topics $\{t_1, \ldots, t_K\}$, we select the corresponding green list $G_{t^*}$. Otherwise, we use semantic mapping to ensure consistency with the generation-time topic selection using one of two approaches. **Embedding averaging** computes the mean embedding of all detected topics and identifies the predefined topic with highest cosine similarity, where $\bar{e}_{detected} = \frac{1}{m}\sum_{i=1}^{m} e_{d_i}$ and selects $t^* = \text{argmax}_{t_j} \text{sim}(\bar{e}_{detected}, e_{t_j})$. **K-means clustering** captures topic diversity within the detected set by applying k-means to detected topic embeddings and evaluating centroids against predefined topics: $t^* = \text{argmax}_{t_j} \max_{c_k} \text{sim}(c_k, e_{t_j})$. Detection proceeds by computing the $z$-score using $G_{t^*}$.

#### 4.3.2 Sliding Window Detection

While strict matching works under ideal conditions, real text exhibits topic drift. This approach addresses local topic inconsistencies while maintaining semantic awareness through temporal aggregation.

We partition the input text into windows of size $w$. For each window, we extract topics using the hierarchical matching above, then assign the final topic via majority voting: $t^* = \text{argmax}_{t_j} |\{w_i : \text{topic}(w_i) = t_j\}|$ where $\text{topic}(w_i)$ represents the assigned topic for window $w_i$. An adaptive fallback mechanism selects embedding averaging for windows with fewer than 3 detected keywords. Detection proceeds using the same $z$-score computation as strict matching.

#### 4.3.3 Maximum $z$-Score Detection

Both previous methods rely on successful topic extraction, which can fail with ambiguous or multi-topic content. Our most robust detection scheme eliminates the topic matching assumption entirely by evaluating text against each predefined green list $G_{t_i}$, computing the corresponding $z$-score $z_i$ using the same statistical framework as previous methods. The final classification uses the maximum $z$-score across all topics: $t^* = \text{argmax}_{t_i} z_i$. This approach effectively allows the watermark signal itself to determine the most likely topic alignment.

This parameter-free approach provides several key advantages: (i) it requires no topic extraction or mapping steps, eliminating potential failure modes; (ii) it leverages the embedded watermark signal to guide topic selection, aligning detection with the generation process; and (iii) it provides robustness against topic ambiguity, drift, and misalignment, making it suitable for practical deployment where topic alignment cannot be guaranteed.

## 5 Evaluation

We present a comprehensive evaluation of TBW against watermarking schemes across four key dimensions: text quality, robustness against both full-text paraphrasing and lexical perturbations, evaluation of watermark detection performance under our proposed schemes, and computational efficiency via average generation times. Our results demonstrate that TBW achieves an optimal balance across these metrics, offering strong robustness and high text quality while maintaining computational efficiency comparable to production-ready systems.

### 5.1 Experimental Setup

**Data and Models.** All experiments use sampled subsets of the C4 dataset (Raffel et al., 2020), where we truncate the first 100 tokens as the input prompt and let each model generate 200 additional tokens (allowing a tolerance of $\pm 5$ to accommodate decoding variation). We primarily evaluate two LLMs: OPT-6.7B (Zhang et al., 2022) and GEMMA-7B (Team et al., 2024). All evaluations use NVIDIA V100 or A100 GPUs.

**Watermarking Comparisons.** We compare TBW against several baselines (see §2

for details): No Watermark (standard decoding), KGW (Kirchenbauer et al., 2023), DiPMark (DiP) (Wu et al., 2024), Unigram (Zhao et al., 2024), SynthID-Text (SynthID) (Dathathri et al., 2024), SIR (Liu et al., 2024), and EXP, EXP-Edit, ITS-Edit (Kuditipudi et al., 2024). We use the MARKLLM library (Pan et al., 2024) implementations with parameter settings from the original papers to ensure fair comparison (details in D). We additionally compare TBW against semantic watermarking schemes that operate via post-processing rather than in-generation biasing, specifically PostMark (Chang et al., 2024) and k-SemStamp (Hou et al., 2024b); these results are reported separately in E.6 due to their different generation pipelines.

**Implementation Details:** Our TBW uses KeyBERT (Grootendorst, 2020) for topic extraction alongside a partition of topic-aligned tokens (§4.1). We employ a predefined set of four generalized topics, {`animals`, `technology`, `sports`, `medicine`}, which we found to be sufficiently generic to cover large vocabulary partitions while avoiding overspecialization. In our setup, four topics correspond to an effective green-list fraction of 0.25, which is close to the $\gamma = 0.2$ used in KGW's experiments, while still maintaining broad topic coverage. Topic assignment is performed using a sentence embedding model (all-MiniLM-L6-v2) with a similarity threshold of $\tau = 0.7$, though any semantic embedding framework can be substituted. We also study scalability beyond four topics by varying $K \in \{4, 8, 16, 32\}$; as $K$ increases the watermark signal attenuates as expected yet remains strong, and text quality stays flat (see 5.6 and F.4).

## 5.2 Text Quality

We evaluate text quality preservation through perplexity metrics (§5.2.1) with additional human evaluation in E.2 and LLM-as-a-Judge analysis in E.3.

### 5.2.1 Perplexity

We evaluate text fluency via perplexity (Niess and Kern, 2025; Mao et al., 2025; Feng et al., 2025; Qu et al., 2025) using a larger "oracle" model, LLAMA-3.1-8B (Grattafiori et al., 2024), which is from a different family than both OPT and GEMMA to avoid bias. We fix the watermark strength at $\delta = 2.0$ for direct comparability with prior work (Kirchenbauer et al., 2023; Zhao et al., 2024) and optimal quality-detection trade-off (see F.1 for sensitivity analysis). Perplexity is measured on the generated $200 \pm 5$ tokens only of 100 samples.

Figure 2 shows distributions for OPT-6.7B and GEMMA-7B under all watermarking schemes, with values above 100 clipped for clarity. We observe

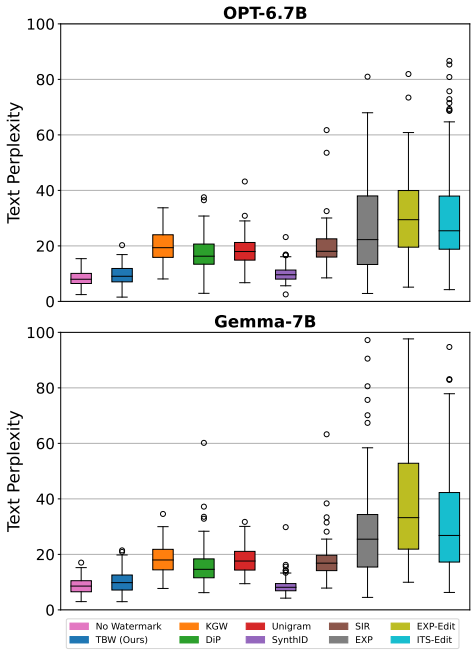



Figure 2: Text perplexity comparison using baseline watermarking schemes. Boxplots show distribution of perplexity values for each watermarking method. Lower text perplexity indicates higher generated text quality.

that TBW achieves significantly lower perplexity (higher text quality) compared to other watermarking schemes, specifically SynthID, closely matching non-watermarked outputs. TBW improves perplexity by approximately 42% over Unigram on OPT-6.7B and by 48% on GEMMA-7B. These results indicate that TBW embeds a robust watermark without degrading text fluency, making it unlikely that watermarked text would be perceptibly less natural to human readers. Additional results for a higher bias strength setting ($\delta = 3.0$) on OPT-6.7B are provided in E.1, illustrating the trade-off between increased robustness and text quality.

## 5.3 Robustness

Having established TBW's high text quality and efficiency, we next evaluate its robustness to adversarial transformations. Since EXP-based methods (e.g., ITS-Edit, EXP-Edit) exhibit substantially degraded text quality at comparable watermark strengths, we focus our main comparisons on TBW, KGW, DiP, Unigram, SynthID, and SIR. Results for ITS-Edit, the strongest EXP-based scheme, are deferred to E.5.

All robustness experiments use oracle conditions, where the generation topic is assumed to be known at detection time, as well as a higher watermark strength ($\delta = 3.0$) to establish an upper bound on detection performance independent of topic extraction errors. Such conditions approximate the scenario of near-perfect topic detection achievable with our best-performing algorithm (see §5.4) and provide a clean benchmark for evaluating the intrinsic resilience of each watermarking method. Detailed attack configurations are provided in G.

### 5.3.1 Text Degradation

We evaluate how each scheme's detection score deteriorates under lexical perturbations using OPT-6.7B as the representative model. This choice provides a conservative robustness estimate, as its smaller vocabulary makes maintaining watermark signal more challenging (§5.2). Two perturbation types are considered: **random perturbations** (a mix of word insertions, deletions, and substitutions applied uniformly across the text affecting both high- and low-importance words) and **targeted perturbations** (the same operations applied selectively to semantically important words such as nouns and verbs, while avoiding low-information stop words like "a," "the," and "to," thereby maximizing disruption to watermark signal).

For each scheme, we generate 100 watermarked texts, then apply perturbations in 5% increments up to 50%. At each level, we average the binary detection outcome ("watermarked" vs. "not watermarked") across 20 independent trials. Figure 3 presents the resulting score trajectories.

All watermarking schemes show a gradual decline in classification scores as perturbation levels increase, except for DiP, which does not rely on a $z$-statistic for detection. Unigram, despite its robustness to paraphrasing, deteriorates under even modest perturbations, crossing its classification threshold earlier than TBW. This highlights a key vulnerability that simple, low-effort edits can bypass detection despite strong paraphrasing resistance. In contrast, TBW maintains higher detection rates across all perturbation levels for both random and targeted attacks, demonstrating resilience to lexical perturbations.

### 5.3.2 Semantic Paraphrasing

We evaluate detection robustness against strong paraphrasing by generating 500 watermarked and 500 non-watermarked samples from both OPT-6.7B and GEMMA-7B. Each 200-token completion is transformed using two paraphrasers widely adopted in watermarking research, PEGASUS (Zhang et al., 2020a) and DIPPER (Krishna et al., 2023), consistent with prior evaluation (Liu and Bu, 2024; Hou et al., 2024a,b).

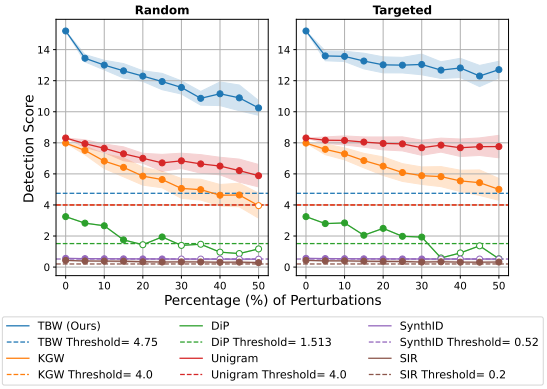



Figure 3: Detection scores under random (left) and targeted (right) word perturbations. Solid ticks indicate scores above threshold; white ticks indicate scores below. Higher scores indicate higher robustness to attacks.

We first report ROC-AUC and Best F1 scores in Table 1. Without attack, all methods reliably separate watermarked from non-watermarked text. Under paraphrasing, however, DIPPER and PEGASUS substantially degrade the performance of SynthID and DiP. TBW and Unigram retain the highest ROC-AUC and Best F1 scores across both attack scenarios, with TBW often matching Unigram and outperforming all other baselines. Full ROC-AUC curves are included in E.7.

In practical applications, maintaining a high True Positive Rate (TPR) at consistently low False Positive Rates (FPR) ensure non-watermarked texts are not misclassified as watermarked. We therefore report TPR@1%FPR and TPR@10%FPR for OPT-6.7B and GEMMA-7B in Table 2, with additional human-text FPR analysis in E.8.

Across both paraphrasers, TBW achieves the highest TPR@1%FPR of all watermarking schemes, indicating superior detection in conservative operating setting. At TPR@10%FPR, TBW remains competitive with Unigram while outperforming other baselines, demonstrating that it maintains strong paraphrasing robustness without compromising the practical constraints of real-world watermark detection.

| Model | Attacks | ROC-AUC | | | | | | Best F1 Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **TBW** | KGW | DiP | Unigram | SynthID | SIR | **TBW** | KGW | DiP | Unigram | SynthID | SIR |
| OPT-6.7B | No Attack | **1.000** | **1.000** | 0.999 | **1.000** | 0.999 | 0.995 | 0.995 | **0.998** | 0.994 | 0.994 | 0.995 | 0.978 |
| | Pegasus | **0.990** | 0.975 | 0.824 | 0.987 | 0.910 | 0.971 | 0.960 | 0.933 | 0.756 | **0.970** | 0.837 | 0.920 |
| | DIPPER | 0.945 | 0.826 | 0.576 | **0.955** | 0.650 | 0.891 | 0.888 | 0.770 | 0.667 | **0.893** | 0.675 | 0.829 |
| Gemma-7B | No Attack | 0.998 | 0.995 | **1.000** | 0.998 | **1.000** | 0.990 | **0.999** | 0.997 | **0.999** | 0.996 | 0.997 | 0.973 |
| | Pegasus | 0.981 | 0.983 | 0.836 | **0.985** | 0.912 | 0.952 | 0.951 | **0.962** | 0.759 | 0.959 | 0.842 | 0.903 |
| | DIPPER | 0.871 | 0.825 | 0.546 | **0.911** | 0.656 | 0.822 | 0.811 | 0.773 | 0.668 | **0.851** | 0.676 | 0.775 |

Table 1: ROC-AUC and Best F1 scores for watermarking approaches under no attack, PEGASUS paraphrasing, and DIPPER paraphrasing. Best results are in **bold**; second-best results are underlined.

| Model | Attacks | TPR@1% FPR | | | | | | TPR@10% FPR | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **TBW** | KGW | DiP | Unigram | SynthID | SIR | **TBW** | KGW | DiP | Unigram | SynthID | SIR |
| OPT-6.7B | No Attack | 0.994 | **0.996** | 0.992 | **0.996** | 0.992 | 0.964 | **1.000** | **1.000** | 0.996 | 0.996 | 0.996 | 0.986 |
| | Pegasus | **0.910** | 0.578 | 0.228 | 0.900 | 0.446 | 0.726 | 0.980 | 0.948 | 0.552 | **0.986** | 0.768 | 0.930 |
| | DIPPER | **0.536** | 0.124 | 0.028 | 0.516 | 0.058 | 0.248 | 0.866 | 0.534 | 0.170 | **0.872** | 0.258 | 0.702 |
| Gemma-7B | No Attack | **1.000** | **1.000** | **1.000** | 0.998 | **1.000** | 0.890 | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | 0.998 |
| | Pegasus | **0.842** | 0.246 | 0.282 | 0.598 | 0.484 | 0.476 | 0.960 | 0.974 | 0.614 | **0.980** | 0.750 | 0.896 |
| | DIPPER | **0.196** | 0.052 | 0.022 | 0.034 | 0.024 | 0.190 | 0.612 | 0.568 | 0.164 | **0.766** | 0.288 | 0.524 |

Table 2: True Positive Rate (TPR) at fixed False Positive Rate (FPR) for watermarking approaches under no attack, PEGASUS paraphrasing, and DIPPER paraphrasing. Best results are in **bold**; second-best results are underlined.

### 5.4 Detection Analysis

While our robustness experiments in §5.3 established watermark resilience under oracle conditions, practical deployment requires detection schemes that operate without prior topic knowledge. We evaluate the detection approaches introduced in §4.3, moving from theoretical validation to real-world applicability, using 500 watermarked samples under our standard experimental configuration. We further provide discussion in C on practical deployment contexts of detection schemes.

Table 3 compares all detection schemes for both OPT-6.7B and GEMMA-7B. Strict topic matching slightly outperforms sliding window detection despite the latter's intent to mitigate topic drift. Embedding averaging yields higher detection rates than k-means in both contexts. We attribute sliding window's weaker performance to the 50-word window size: while offering local topic resolution, it limits available context compared to processing the full text. Exploring alternative window sizes is considered out of scope but presents a possible avenue for future work.

The maximum $z$-score detection method delivers near-perfect results, achieving 99.6% and 100% detection rates for OPT-6.7B and GEMMA-7B, respectively, with the highest mean $z$-scores and minimal variance. This performance stems from its independence from topic extraction by directly evaluating all topic partitions and selecting the highest $z$-score, it avoids the alignment errors that constrain topic-dependent approaches. We confirm this with 500 watermarked and 500 non-watermarked samples with detailed performance metrics provided in E.10. These results connect our oracle-condition evaluations in §5.3 with practical, topic-agnostic detection, showing that maximum $z$-score preserves robustness while providing tunable separation adaptable to different operational constraints.

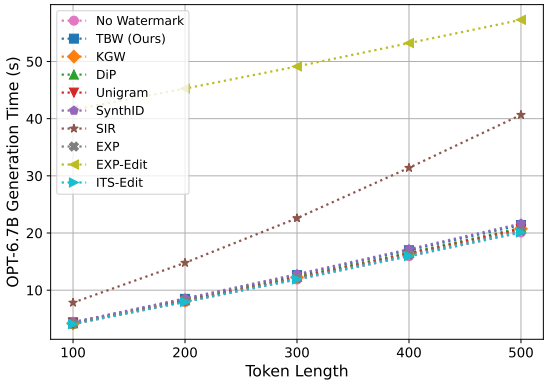



Figure 4: Comparison of average generation time (seconds) over various output token lengths from multiple watermarking schemes on OPT-6.7B.

### 5.5 Efficiency

We measure the computational overhead imposed by each watermarking method on OPT-6.7B, generating sequences of lengths {100, 200, 300, 400, 500}. For each token-length, we record the average generation time over 10 samples from C4.

| Detection Scheme | OPT-6.7B | | | GEMMA-7B | | |
|---|---|---|---|---|---|---|
| | Detection Rate | $z$-Score (Mean±Sd) | Topic Accuracy | Detection Rate | $z$-Score (Mean±Sd) | Topic Accuracy |
| Strict K-means | 0.540 | $6.32 \pm 10.80$ | 0.542 | 0.570 | $6.71 \pm 10.58$ | 0.570 |
| Strict Embedding | 0.574 | $7.05 \pm 10.68$ | 0.624 | 0.584 | $6.99 \pm 10.53$ | 0.584 |
| Sliding K-means | 0.516 | $5.80 \pm 10.77$ | 0.520 | 0.532 | $5.95 \pm 10.69$ | 0.532 |
| Sliding Embedding | 0.566 | $6.91 \pm 10.67$ | 0.602 | 0.516 | $5.63 \pm 10.70$ | 0.524 |
| Sliding K-means+Emb | 0.526 | $6.02 \pm 10.79$ | 0.530 | 0.528 | $5.85 \pm 10.68$ | 0.528 |
| **Max. $z$-Score** | **0.996** | $\mathbf{15.88 \pm 3.03}$ | **1.000** | **1.000** | $\mathbf{15.92 \pm 1.39}$ | **1.000** |

Table 3: Comparison of watermark detection schemes on watermarked text under realistic (non-oracle) detection conditions. **Bold** indicates the highest performing detection scheme in each metric.

Figure 4 shows that TBW introduces negligible overhead compared to non-watermarked generation, matching lightweight methods such as KGW and SynthID. In contrast, EXP-Edit requires multiple re-ranking passes and SIR incurs additional complexity, resulting in noticeably higher generation times. We observe consistent trends on OPT-2.7B (see E.4), confirming that TBW's efficiency advantage holds across model scales.

### 5.6 Topic Scalability

While our main experiments use $K = 4$ topics, practitioners may require finer-grained coverage. We evaluate TBW with $K \in \{4, 8, 16, 32\}$ on GEMMA-7B, whose larger vocabulary is more representative of deployed LLMs. We apply $\delta = 2.0$ and $\tau = 0.5$, relaxed relative to the main experiments to compensate for smaller per-topic coverage at larger $K$. Detection uses the maximum $z$-score scheme. Full experimental details, topic inventories, and additional visualizations are provided in F.4.

Figure 5 reports detection strength as a function of $K$. The $z$-score decreases gracefully from $\sim$11 at $K = 4$ to $\sim$7 at $K = 32$, yet remains comparable to KGW and Unigram baselines at equivalent watermark strength. Text quality (BERTScore F1) remains flat across all $K$ values (see F.4), confirming that increased topic granularity does not degrade fluency or semantics. Table 4 reports detection runtime, which scales approximately linearly with $K$ but remains tractable; because detection is performed offline by the model owner, this overhead does not impact user-facing latency.

## 6 Conclusion

Watermarking has emerged as a critical tool for mitigating risks in large language models, yet existing schemes face a persistent trade-off where lightweight methods such as SynthID-Text or KGW offer efficiency and text quality but collapse under paraphrasing, while heavyweight multi-pass

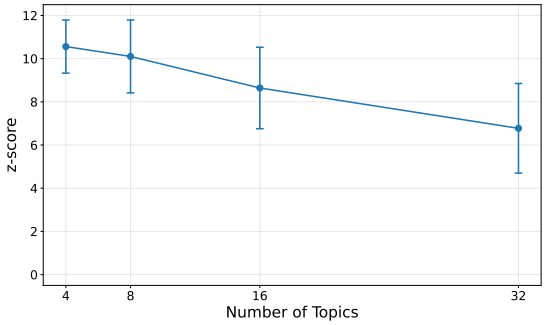



Figure 5: Detection strength vs. number of topics. Mean detector $z$-score (max-$z$) as we scale $K \in \{4, 8, 16, 32\}$ on GEMMA-7B with $\delta = 2.0$, $\tau = 0.5$ over 100 prompts. Error bars denote ±s.d.

| Number of Topics ($K$) | Detection Time (s) |
|---|---|
| 4 | $2.002 \pm 0.041$ |
| 8 | $2.910 \pm 0.047$ |
| 16 | $3.833 \pm 0.046$ |
| 32 | $6.565 \pm 0.066$ |

Table 4: Detector runtime vs. number of topics. Mean ± s.d. per-sample time in seconds.

methods deliver robustness at the cost of fluency and practicality. We presented Topic-Based Watermarking (TBW), which addresses this divide by embedding semantically aligned signals through topic-driven token partitioning. TBW preserves the speed and fluency of lightweight approaches while achieving robustness closer to heavyweight schemes. Our evaluation confirms that TBW provides strong detection accuracy, resilience to lexical and semantic attacks, and negligible runtime overhead. By overcoming this trade-off, TBW offers a practical path toward deployable watermarking.

## Limitations

Our results indicate that models with smaller vocabularies may exhibit slightly reduced performance compared to larger vocabulary models. For instance, OPT-6.7B (50,272 tokens) shows marginally lower detection scores and higher false positive rates than GEMMA-7B (256,000 tokens) (E.10). This disparity arises from how vocabulary size interacts with topic partitioning. When the vocabulary is divided among $K$ predefined topics, larger vocabularies yield denser topic-aligned green lists, providing more semantically coherent tokens to bias during generation. Consequently, if practitioners require finer-grained topic coverage (e.g., $K = 32$ topics), smaller vocabulary models may experience more pronounced performance degradation. We empirically examine this scaling behavior in F.4, demonstrating that while detection strength ($z$-score) decreases as $K$ increases, text quality remains stable and the watermark signal persists even at $K = 32$ topics.

The interaction between vocabulary size and topic granularity suggests an avenue for domain-specific adaptation. Rather than increasing $K$ to achieve broader coverage, practitioners deploying TBW on smaller models may instead tailor the predefined topic lists to their specific use case, maintaining a modest number of topics while ensuring semantic alignment with the target domain. Prior work has demonstrated the feasibility of such domain-specific topic-based watermarking in specialized contexts such as AI-assisted peer review (Nemecek et al., 2025), suggesting that thoughtful topic selection can preserve watermark effectiveness without requiring large vocabularies or extensive topic sets.

If TBW were to be deployed with a large number of topics, we would expect larger vocabulary models to yield better results. Our maximum $z$-score detection scheme, which achieves near-perfect separation between watermarked and non-watermarked text, requires evaluating all $K$ topic partitions to identify the strongest watermark signal. As the number of topics increases, detection time scales approximately linearly with $K$. We report these timing results in F.4, observing an increase from approximately 2 seconds at $K = 4$ to 6.5 seconds at $K = 32$. While this overhead remains tractable, it represents a consideration for deployments requiring rapid detection at scale. Importantly, detection is performed offline by model owners rather than during user-facing generation, mitigating latency concerns in practice.

Beyond TBW's specific implementation, a limitation it shares with other list-based watermarking schemes, including KGW (Kirchenbauer et al., 2023) and SynthID-Text (Dathathri et al., 2024), is reduced detection reliability on shorter text sequences. Our primary evaluations use 200-token generations, which we consider relatively short for watermark detection. The statistical nature of $z$-score-based detection requires sufficient tokens to distinguish the watermark signal from random variation; as text length decreases, the gap between watermarked and non-watermarked $z$-score distributions narrows, increasing the likelihood of misclassification. We hypothesize that detection performance would degrade on shorter sequences, a constraint inherent to the underlying framework.

## Ethical Considerations

Watermarking techniques inherently carry dual-use risks. While watermarking enables provenance tracking to combat misuse, it could potentially be exploited for punitive enforcement based on probabilistic signals. Adversarial actors may further weaponize watermarking to falsely attribute content to specific organizations or models. We additionally acknowledge residual concerns regarding false positives, which could lead to wrongful accusations. We recommend deployment with conservative thresholds, transparent documentation of error rates, and appeals processes for affected users, calibrated to the specific deployment context.

Our evaluation used a public dataset (C4) and locally-run models without probing live services or violating terms of service. The human evaluation study (E.2) involved only non-sensitive, model-generated text; participants provided informed consent, and no personally identifiable information was collected.

## Acknowledgments

This work was supported in part by the National Science Foundation under Grant No. OAC-2112606. ChatGPT and Claude were used in the preparation of this work for editing and grammar checking. No passages were copied without full author review, and all substantive ideas, analysis, and conclusions are the product and responsibility of the authors. The AI tools were also utilized for code development and early manuscript reviews.

## References


Association for the Advancement of Artificial Intelligence (AAAI). 2025. AAAI Launches AI-Powered Peer Review Assessment System. https://aaai.org/aaai-launches-ai-powered-peer-review-assessment-system/. Accessed: 2025-08-25.

Jainit Sushil Bafna, Hardik Mittal, Suyash Sethia, Manish Shrivastava, and Radhika Mamidi. 2024. Mast kalandar at semeval-2024 task 8: On the trail of textual origins: Roberta-bilstm approach to detect ai-generated text. *Preprint*, arXiv:2407.02978.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online. Association for Computational Linguistics.

Yapei Chang, Kalpesh Krishna, Amir Houmansadr, John Frederick Wieting, and Mohit Iyyer. 2024. Postmark: A robust blackbox watermark for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8969–8987.

Canyu Chen and Kai Shu. 2024. Combating misinformation in the age of llms: Opportunities and challenges. *AI Magazine*, 45(3):354–368.

Cleveland Clinic. 2025. Cleveland Clinic Announces Rollout of Ambience Healthcare's AI Platform. https://newsroom.clevelandclinic.org/2025/02/19/cleveland-clinic-announces-the-rollout-of-ambience-healthcares-ai-platform. Accessed: 2025-08-25.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

Xiaoyan Feng, He Zhang, Yanjun Zhang, Leo Yu Zhang, and Shirui Pan. 2025. Bimark: Unbiased multilayer watermarking for large language models. In *Forty-second International Conference on Machine Learning*.

Google. 2024. Introducing gemini 2.0: Our new ai model for the agentic era. https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/. Accessed: 2025-07-29.

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

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert. https://maartengr.github.io/KeyBERT/. Accessed: 2025-07-29.

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

Abe Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. 2024b. k-SemStamp: A clustering-based semantic watermark for detection of machine-generated text. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1706–1715, Bangkok, Thailand. Association for Computational Linguistics.

Hugging Face. 2025. Hugging face - the ai community building the future. https://huggingface.co. Accessed: 2025-08-26.

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*, ICML'24. JMLR.org.

Niful Islam, Debopom Sutradhar, Humaira Noor, Jarin Tasnim Raya, Monowara Tabassum Maisha, and Dewan Md Farid. 2023. Distinguishing human generated text from chatgpt generated text using machine learning. *Preprint*, arXiv:2306.01761.

Nikola Jovanović, Robin Staab, and Martin Vechev. 2024. Watermark stealing in large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

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.

Kalpesh Krishna. 2023. ai-detection-paraphrases. https://github.com/martiansideofthemoon/ai-detection-paraphrases. Accessed: 2025-08-01.

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 *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2024. Robust distortion-free watermarks for language models. *Preprint*, arXiv:2307.15593.

Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. 2023. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*, pages 3637–3647.

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.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2021. A survey on text classification: From shallow to deep learning. *Preprint*, arXiv:2008.00364.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7).

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2024. A semantic invariant robust watermark for large language models. *Preprint*, arXiv:2310.06356.

Yepeng Liu and Yuheng Bu. 2024. Adaptive text watermark for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Minjia Mao, Dongjun Wei, Zeyu Chen, Xiao Fang, and Michael Chau. 2025. Watermarking low-entropy generation for large language models: An unbiased and low-risk method. *Preprint*, arXiv:2405.14604.

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.

Felix B Mueller, Rebekka Görge, Anna K Bernzen, Janna C Pirk, and Maximilian Poretschkin. 2024. Llms and memorization: On quality and specificity of copyright compliance. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pages 984–996.

Alexander Nemecek, Yuzhou Jiang, and Erman Ayday. 2025. The feasibility of topic-based watermarking on academic peer reviews. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 616–634, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Newsweek. 2025. World's Best Hospitals 2025 - United States of America. `https://rankings.newsweek.com/worlds-best-hospitals-2025/united-states-america`. Accessed: 2025-08-25.

Georg Niess and Roman Kern. 2025. Ensemble watermarks for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2903–2916, Vienna, Austria. Association for Computational Linguistics.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/index/chatgpt/`. Accessed: 2025-07-29.

OpenAI. 2023. New ai classifier for indicating ai-written text. `https://openai.com/index/new-ai-classifier-for-indicating-ai-written-text`. Accessed: 2025-07-29.

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.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. 2025. Provably robust multi-bit watermarking for ai-generated text. *Preprint*, arXiv:2401.16820.

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. *J. Mach. Learn. Res.*, 21(1).

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

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

Scott Aaronson. 2023. Watermarking of large language models. `https://simons.berkeley.edu/talks/scott-aaronson-ut-austin-openai-2023-08-17`. Accessed: 2025-07-29.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

SynthID-Team. 2024. Watermarking ai-generated text and video with synthid. Accessed: 2025-08-26.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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*, ICML'24. JMLR.org.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. Pegasus: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Ruisi Zhang, Shehzeen Samarah Hussain, Paarth Neekhara, and Farinaz Koushanfar. 2024. Remark-llm: a robust and efficient watermarking framework for generative large language models. In *Proceedings of the 33rd USENIX Conference on Security Symposium*, SEC '24, USA. USENIX Association.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

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

## Appendix Contents


A Algorithm Implementations . . . . . 14

B Extended Threat Model . . . . . 14

C Practical Use of Detection Schemes . . . . . 16

C.1 Strict and Sliding Window Detection 16

C.2 Maximum $z$-Score Detection . . . . . 17

D Watermarking Evaluation Parameters . . . . 17

E Additional Evaluation Results . . . . . 17

E.1 Perplexity at Higher Bias Strength . . 17

E.2 Human Evaluation . . . . . 18

E.3 LLM-as-a-Judge . . . . . 18

E.4 Efficiency Results . . . . . 20

E.5 Comparison to ITS-Edit . . . . . 20

E.6 Comparison to Semantic Watermarks 20

E.7 TBW Robustness ROC Curves . . . . . 20

E.8 False Positive Rate (FPR) on Human-Written Text . . . . . 22

E.9 Detection Attacks . . . . . 22

E.10 Maximum $z$-Score Detection: Detailed Analysis . . . . . 25

F Ablation Studies . . . . . 26

F.1 TBW Strength Sensitivity . . . . . 26

F.2 Parameter Space Analysis . . . . . 27

F.3 Task-Specific Evaluation on Summarization . . . . . 28

F.4 Scaling the Number of Topics . . . . . 29

G Attack Configuration Details . . . . . 30

G.1 Paraphrasing Attacks . . . . . 30

G.2 Lexical Perturbation Attacks . . . . . 30

H Watermarked Text Examples . . . . . 30


## A Algorithm Implementations

We provide implementation details for our topic-based watermarking approach. Algorithm 1 presents the token-to-topic mapping procedure, while Algorithm 2 details the watermark generation process.

## B Extended Threat Model

This section provides additional formal constraints and justifications for the adversary assumptions introduced in §3.2.

**Adversary Assumptions and Constraints.** We assume an adversary consistent with standard threat models adopted across the watermarking literature (Kirchenbauer et al., 2023; Zhao et al., 2024; Zhang et al., 2024). In particular, the adversary cannot fine-tune or retrain the model, nor inspect its parameters or internal mechanisms. Interaction is strictly via black-box queries to the generator; the adversary has no oracle access to the watermark detection algorithm and receives neither per-text decisions nor detection scores from the system. While the detection algorithmic form may be public, the operating keys, topic partitions, and thresholds remain private, and detection is performed offline by a trusted party. This scope reflects realistic end-user capabilities and explicitly excludes insider or white-box attacks where model internals are visible.

**Excluded and Impractical Attacks.** We further assume the adversary cannot mount impractical strategies that rely on excessive querying or access to paired datasets (watermarked vs. non-watermarked versions of the same prompt). Statistical comparison or supervised classification attacks are excluded from our primary model as they require assumptions that are rarely feasible for typical end-users. We provide an empirical evaluation of these "detection-based" attacks in E.9 for completeness, though they fall outside our primary security focus.

**Partition Recovery Attacks.** Prior work has demonstrated partition recovery attacks against KGW-style watermarks, where adversaries approximate green/red list assignments through repeated positional querying (Jovanović et al., 2024). Because TBW uses a single fixed green list per generation rather than per-token hashing, this specific attack vector does not directly transfer. However, TBW's semantic partitioning introduces a different concern: an adversary who knows the algorithm, topic list, and embedding approach might attempt to reconstruct the topic-aligned green lists through structural inference.

To evaluate this, we consider a strong attacker who knows TBW's partitioning algorithm, the predefined topic set, and the general embedding approach, but not the private parameters: the specific

**Algorithm 1** Token-to-Topic Mapping

**Input:** Vocabulary $V$, predefined topic set $\{t_1, \ldots, t_K\}$, embedding function $\text{Enc}(\cdot)$, similarity threshold $\tau$.
Compute topic embeddings: $E_T = \{\mathbf{e}_{t_i} \mid t_i \in \{t_1, \ldots, t_K\}\}$
Compute token embeddings: $E_V = \{\mathbf{e}_v \mid v \in V\}$
Initialize topic-aligned lists: $G_{t_i} = \emptyset, \forall i \in \{1, \ldots, K\}$
Initialize residual set: $\mathcal{B} = \emptyset$
**for** each token $v \in V$ **do**
  Compute similarity scores: $\textit{sim}(v, t_i) = \frac{\mathbf{e}_v \cdot \mathbf{e}_{t_i}}{\|\mathbf{e}_v\| \|\mathbf{e}_{t_i}\|}, \forall i$
  $m, i^* \leftarrow \max(\textit{sim}(v, t_i)), \arg\max(\textit{sim}(v, t_i))$
  **if** $m \geq \tau$ **then**
    Assign $v$ to topic $t_{i^*}$: $G_{t_{i^*}} \leftarrow G_{t_{i^*}} \cup \{v\}$
  **else**
    Add $v$ to residual set: $\mathcal{B} \leftarrow \mathcal{B} \cup \{v\}$
  **end if**
**end for**
Distribute remaining tokens:
Initialize counter: $i \leftarrow 1$
**for** each token $v \in \mathcal{B}$ **do**
  Assign $v$ to $t_{\text{target}} = t_{(i \bmod K)+1}$
  $G_{t_{\text{target}}} \leftarrow G_{t_{\text{target}}} \cup \{v\}$
  $i \leftarrow i + 1$
**end for**
**Return** $\{G_{t_1}, \ldots, G_{t_K}\}$ {Final topic-aligned token lists}

**Algorithm 2** Topic-Based Watermark Generation

**Input:** Prompt $\mathbf{x}^{\text{prompt}}$, topic set $\{t_1, \ldots, t_K\}$, topic-aligned lists $\{G_{t_1}, \ldots, G_{t_K}\}$, logit bias $\delta$.
Extract topics: $\mathcal{T}_{\text{detected}} \leftarrow \text{KeyBERT}(\mathbf{x}^{\text{prompt}})$
**if** $\exists t_i \in \{t_1, \ldots, t_K\}$ such that $t_i \in \mathcal{T}_{\text{detected}}$ **then**
  Select direct match: $t^* \leftarrow t_i$
**else**
  Assign via clustering:
  $t^* \leftarrow \text{KMeans}(\mathcal{T}_{\text{detected}}, \{t_1, \ldots, t_K\})$
**end if**
Retrieve topic-aligned list: $G_{t^*} \leftarrow G_{t^*}$
Initialize output sequence: $\mathbf{z} \leftarrow \emptyset$
**while** not end-of-sequence **do**
  Compute logits: $\mathbf{logits} \leftarrow p_\theta(\cdot \mid \mathbf{x}^{\text{prompt}}, \mathbf{z})$
  **for** each token $v \in V$ **do**
    **if** $v \in G_{t^*}$ **then**
      Adjust logit: $\mathbf{logits}[v] \leftarrow \mathbf{logits}[v] + \delta$
    **end if**
  **end for**
  Compute probabilities: $\mathbf{p} \leftarrow \text{Softmax}(\mathbf{logits})$
  Sample next token: $v_{\text{next}} \leftarrow \text{SampleToken}(\mathbf{p})$
  Append token to sequence: $\mathbf{z} \leftarrow \mathbf{z} \cup \{v_{\text{next}}\}$
**end while**
**Return** $\mathbf{z}$ {Watermarked output text}

embedding model weights, similarity threshold $\tau$, or the random seed used for round-robin distribution of residual tokens. We measure partition recovery across 12 attacker configurations, varying the embedding model (3 models, including an oracle configuration using the identical architecture as the watermark deployer) and similarity threshold ($\tau \in \{0.3, 0.5, 0.7, 0.9\}$). For each configuration, we compute Jaccard similarity, precision, recall, and F1 between the attacker's reconstructed partition and the true partition, averaged across all four topic lists. Tables 5 and 6 report results for OPT-6.7B and GEMMA-7B, respectively.

| Model | $\tau$ | Jaccard | Precision | Recall | F1 |
|---|---|---|---|---|---|
| all-MiniLM | 0.3 | 0.1433 | 0.2512 | 0.2514 | 0.2505 |
| all-MiniLM | 0.5 | 0.1448 | 0.2529 | 0.2529 | 0.2529 |
| all-MiniLM | 0.7 | 0.1419 | 0.2486 | 0.2486 | 0.2486 |
| all-MiniLM | 0.9 | 0.1415 | 0.2478 | 0.2478 | 0.2478 |
| all-mpnet | 0.3 | 0.1423 | 0.2491 | 0.2492 | 0.2491 |
| all-mpnet | 0.5 | 0.1414 | 0.2478 | 0.2478 | 0.2478 |
| all-mpnet | 0.7 | 0.1413 | 0.2476 | 0.2476 | 0.2476 |
| all-mpnet | 0.9 | 0.1415 | 0.2478 | 0.2478 | 0.2478 |
| bge-small | 0.3 | 0.1299 | 0.2546 | 0.2532 | 0.2256 |
| bge-small | 0.5 | 0.1301 | 0.2542 | 0.2532 | 0.2259 |
| bge-small | 0.7 | 0.1423 | 0.2491 | 0.2491 | 0.2491 |
| bge-small | 0.9 | 0.1438 | 0.2515 | 0.2515 | 0.2515 |

Table 5: Partition recovery attack results on OPT-6.7B across 12 attacker configurations. The all-MiniLM rows represent an oracle attacker using the identical embedding model as the watermark deployer.

| Model | $\tau$ | Jaccard | Precision | Recall | F1 |
|---|---|---|---|---|---|
| all-MiniLM | 0.3 | 0.1427 | 0.2498 | 0.2498 | 0.2497 |
| all-MiniLM | 0.5 | 0.1422 | 0.2490 | 0.2490 | 0.2490 |
| all-MiniLM | 0.7 | 0.1425 | 0.2494 | 0.2494 | 0.2494 |
| all-MiniLM | 0.9 | 0.1434 | 0.2509 | 0.2509 | 0.2509 |
| all-mpnet | 0.3 | 0.1181 | 0.2489 | 0.2491 | 0.2064 |
| all-mpnet | 0.5 | 0.1428 | 0.2499 | 0.2499 | 0.2499 |
| all-mpnet | 0.7 | 0.1445 | 0.2525 | 0.2525 | 0.2525 |
| all-mpnet | 0.9 | 0.1434 | 0.2509 | 0.2509 | 0.2509 |
| bge-small | 0.3 | 0.0940 | 0.2502 | 0.2498 | 0.1617 |
| bge-small | 0.5 | 0.0940 | 0.2497 | 0.2497 | 0.1618 |
| bge-small | 0.7 | 0.1432 | 0.2506 | 0.2506 | 0.2506 |
| bge-small | 0.9 | 0.1431 | 0.2503 | 0.2503 | 0.2503 |

Table 6: Partition recovery attack results on GEMMA-7B across 12 attacker configurations. Results are consistent with OPT-6.7B, confirming chance-level recovery across both models.

Even the oracle attacker using the identical embedding model (all-MiniLM) and matching $\tau$ achieves only chance-level recovery across both models. With $K = 4$ topics, random assignment would yield Jaccard $= 0.25$ and F1 $= 0.25$ in expectation; the observed values fall at or below this baseline across all configurations. Some non-oracle configurations (e.g., bge-small at $\tau = 0.3$) perform below chance, reflecting topic-specific misalignment that further underscores the difficulty of partition inference. This result stems from TBW's dual assignment mechanism: tokens exceeding the similarity threshold $\tau$ are assigned semantically, while residual tokens are distributed via a randomly shuffled round-robin process (§4). Since the random seed governing the round-robin assignment is private, the full partition is unrecoverable regardless of the attacker's knowledge of the embedding model or threshold. Semantic co-grouping alone is insufficient without the private round-robin seed. As complementary evidence, our word cloud analysis (Figure 9) confirms that watermarked outputs show no systematic elevation of topic-specific tokens, preventing observational inference of the active partition from generated text.

# C Practical Use of Detection Schemes

## C.1 Strict and Sliding Window Detection

As discussed in §5.4, both the strict and sliding window detection schemes exhibit degraded performance due to imperfect topic matching between input prompts and output text. This limitation arises because the detection pipeline assumes alignment between topics extracted from the input and the generated output, an assumption that often fails in general-purpose, black-box deployment. However, there are domains where such access to the input text is natural and the trade-off becomes favorable.

For example, §F.4 shows that increasing the number of topics from $K = 4$ to $K = 32$ increases detection runtime under maximum $z$-score detection (from approximately 2 seconds to 6.5 seconds). In latency-sensitive applications, strict or sliding detection may therefore be more practical when input prompts are accessible, since they avoid evaluating all topic partitions. Healthcare settings provide a clear illustration: Cleveland Clinic, ranked among the world's top hospitals (Newsweek, 2025), recently announced deployment of an ambient AI platform for clinical documentation (Cleveland Clinic, 2025). In such environments, LLM-generated text is deeply integrated into provider workflows, and provenance guarantees are paramount to avoid blind reliance on automatically produced notes. With multiple campuses and a high volume of patients, runtime efficiency is critical, and strict/sliding detection can provide timely watermark checks while still leveraging topic-aware alignment. The natural pairing

of spoken prompts (provider voice memos) with textual outputs further supports this alignment assumption.

Similarly, the Association for the Advancement of Artificial Intelligence (AAAI) initiative on AI-assisted peer review for 2026 (Association for the Advancement of Artificial Intelligence (AAAI), 2025) offers another case where the input, the submitted paper, is always available alongside generated reviewing text. Here, strict and sliding window detection align directly with the review pipeline, enabling topic-aware watermark verification without the cost of iterating across all topic partitions. Looking forward, these use cases also suggest an avenue for future work allowing for hybrid "sliding-chunk" watermarking that adapts to multiple topical segments within long-form input documents, such as scientific papers, while retaining prompt-to-output alignment.

### C.2 Maximum $z$-Score Detection

In contrast, the maximum $z$-score detection scheme (§5.4) is best suited for general deployment where access to input prompts cannot be assumed. In such settings, strict or sliding detection would fail due to topic misalignment, whereas maximum $z$-score avoids this assumption by evaluating all topic partitions and selecting the strongest watermark signal. While detection incurs additional cost as the number of topics increases (e.g., 6.5 seconds at $K = 32$), our evaluation shows that this overhead remains tractable, particularly relative to the robustness gains. Maximum z-score detection delivers near-perfect separation of watermarked and non-watermarked text even under strong adversarial attacks, making it the preferred scheme for open-world scenarios where robustness is the overriding priority.

## D Watermarking Evaluation Parameters

To conduct our evaluations, we utilize MarkLLM (Pan et al., 2024), an open-source framework designed to facilitate the implementation and evaluation of LLM watermarking methods. MarkLLM provides an approach to watermarking by integrating different watermarking schemes within a unified framework. Its modular structure supports both the KGW-based family, which modifies token selection probabilities through logit adjustments, and the EXP-based family, which introduces pseudo-random guided sampling to embed watermarks.

We apply MarkLLM to evaluate our diverse set of watermarkings we compared to our proposed topic-based watermark (TBW). We use this framework exclusively for watermark generation and detection in alignment with the respective watermarking approach. Other utilities within the framework, such as robustness evaluation or text quality analysis, are not utilized in our study. The framework ensures that the configurations used in our study remain consistent with the original parameter choices presented in the respective papers, enabling a rigorous and reproducible assessment of each method.

## E Additional Evaluation Results

### E.1 Perplexity at Higher Bias Strength

We report perplexity results for OPT-6.7B when applying a stronger watermark bias parameter of $\delta = 3.0$ *for TBW only*. This matches the strength used in our robustness evaluation (§5.3) and allows a direct quality-robustness trade-off comparison. The experimental setup follows §5.2: we measure perplexity on the generated $200 \pm 5$ tokens using the LLAMA-3.1-8B oracle, clipping values above 100 for visualization. All baseline watermarking methods are reported with their default hyperparameters from the open-source MARKLLM library; only TBW uses the increased $\delta$.

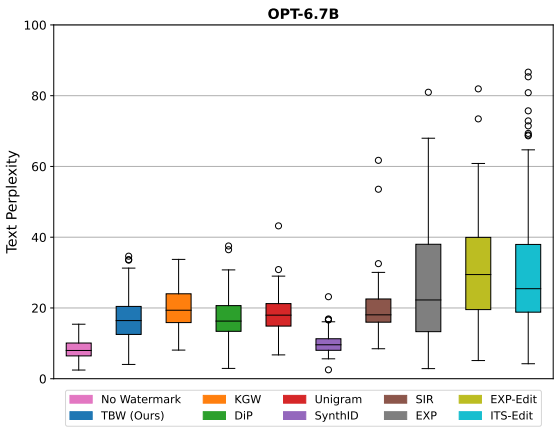



Figure 6: Perplexity comparison for OPT-6.7B with TBW at higher watermark strength ($\delta = 3.0$) and all other schemes at their standard settings. Compared to TBW's $\delta = 2.0$ results (see §5.2), the increased bias leads to moderate quality degradation while retaining the lowest perplexity among watermarking methods. Lower values indicate higher text quality.

Results indicate that increasing TBW's $\delta$ from 2.0 to 3.0 boosts robustness (§5.3) at the cost of a

| Model | Method | Fluency | Coherence | Grammar | Lexical Variety | Overall Quality |
|---|---|---|---|---|---|---|
| OPT-6.7B | TBW | $3.23 \pm 1.16$ | $3.33 \pm 1.16$ | $3.63 \pm 0.80$ | $3.20 \pm 1.02$ | $3.17 \pm 1.14$ |
| | No Watermark | $\mathbf{3.47 \pm 0.95}$ | $\mathbf{3.45 \pm 1.11}$ | $\mathbf{3.90 \pm 0.90}$ | $\mathbf{3.37 \pm 0.97}$ | $\mathbf{3.45 \pm 1.00}$ |
| GEMMA-7B | TBW | $3.53 \pm 1.05$ | $\mathbf{3.48 \pm 1.13}$ | $\mathbf{3.77 \pm 0.96}$ | $\mathbf{3.63 \pm 0.86}$ | $\mathbf{3.50 \pm 1.02}$ |
| | No Watermark | $\mathbf{3.55 \pm 1.05}$ | $3.42 \pm 1.15$ | $3.75 \pm 1.05$ | $3.50 \pm 0.93$ | $3.40 \pm 1.14$ |

Table 7: Human evaluation results for single-text quality assessment. Values represent average ratings ($\pm$standard deviation) on a 5-point Likert scale from 12 evaluators.

moderate rise in perplexity. Despite this stronger bias, TBW still achieves lower perplexity than other baselines, suggesting that semantic alignment mitigates some of the quality loss typically associated with high bias values.

### E.2 Human Evaluation

To complement our automated text quality metrics, we conduct a human evaluation to assess the perceptual quality of TBW text. Human evaluation provides crucial insights into whether watermarking artifacts are detectable to end users, which is essential for practical deployment.

We conduct a single-text evaluation where human evaluators assess individual text samples across multiple quality dimensions. Evaluators are presented with 20 generated text samples, comprising 10 samples from OPT-6.7B and 10 samples from GEMMA-7B. Within each model's samples, 5 are watermarked with TBW and 5 are non-watermarked. Again, we fix the watermark strength at $\delta = 2.0$ to mimic our perplexity as a metric study.

Evaluators rate each text sample on five quality dimensions using a 5-point Likert scale (1 = Very Poor, 5 = Excellent): **fluency** (how smooth and natural the writing flows), **coherence** (logical consistency and ease of following), **grammar** (correctness of grammar, spelling, and punctuation), **lexical variety** (appropriateness and diversity of word choices), and **overall quality** (holistic judgment). We collect responses from 12 evaluators, ensuring sufficient inter-rater reliability for statistical analysis. Table 7 summarizes the results across all quality dimensions.

The results demonstrate that TBW maintains competitive text quality across all evaluated dimensions, with performance variations that align with our automated perplexity assessments. OPT-6.7B exhibits modest quality decrements when watermarked, with performance gaps ranging from 0.12 to 0.28 points across the five evaluation criteria. However, these differences remain within acceptable bounds, as no metric, whether watermarked or non-watermarked, exceeds a score of 4.0, indicating that all generated text falls within a respectable quality range. Conversely, TBW demonstrates particularly stronger performance on GEMMA-7B, where watermarked text achieves comparable or better quality relative to non-watermarked baselines. Notably, TBW underperforms non-watermarked text only in fluency, with a negligible difference of 0.02 points, while achieving higher scores in lexical variety and overall quality metrics.

We attribute this difference to the disparity in vocabulary sizes between the two models. OPT-6.7B has a vocabulary of 50,272 tokens, whereas GEMMA-7B uses 256,000 tokens. A smaller vocabulary limits the number of semantically coherent tokens that exceed our similarity threshold $\tau$ during topic partitioning (Algorithm 1), slightly reducing lexical flexibility in watermarked outputs. In contrast, GEMMA-7B's larger vocabulary yields denser topic-aligned partitions, enabling TBW to preserve or even improve perceived quality. This pattern is consistent with modern production-grade LLMs, which typically adopt vocabularies exceeding 50k tokens, suggesting that TBW's benefits are most fully realized in contemporary large-scale deployments. Automated LLM-as-a-Judge evaluation (E.3) confirms these findings, showing TBW outperforms Unigram and SynthID-Text across all quality dimensions while matching non-watermarked text.

### E.3 LLM-as-a-Judge

While human evaluation provides valuable qualitative insight, its scalability is inherently limited. To extend our comparative analysis across a wider range of watermarking schemes, we employ an LLM-as-a-Judge setup using the chat functionality of GPT-4o. At the time of our study, GPT-4o was the strongest publicly available general-purpose model, offering high accuracy, consistent reasoning, and strong natural language understanding. This approach enables systematic, large-scale eval-

| Method | Fluency | Coherence | Grammar | Lexical Variety |
|---|---|---|---|---|
| No Watermark | $\mathbf{3.20 \pm 0.62}$ | $\mathbf{3.15 \pm 0.59}$ | $3.85 \pm 0.59$ | $\mathbf{3.00 \pm 0.32}$ |
| TBW | $3.10 \pm 0.72$ | $3.05 \pm 0.69$ | $\mathbf{3.90 \pm 0.72}$ | $\mathbf{3.00 \pm 0.46}$ |
| Unigram | $2.65 \pm 0.67$ | $2.70 \pm 0.73$ | $3.35 \pm 0.88$ | $2.90 \pm 0.55$ |
| SynthID | $3.00 \pm 0.73$ | $3.05 \pm 0.76$ | $3.70 \pm 0.73$ | $2.90 \pm 0.55$ |

Table 8: Automated GPT-4o-based text quality evaluation across watermarking methods. Values represent average scores (±standard deviation) on a 5-point scale over 20 prompts. **Bold** shows best values while underline indicates second best results.

uation of TBW's performance not only against non-watermarked baselines, but also relative to other state-of-the-art watermarking methods, including SynthID-Text. Similar LLM-judge protocols have been successfully adopted in prior watermarking research (Pan et al., 2024; Tu et al., 2024), demonstrating the validity and growing acceptance of this methodology in the field.

We conduct this evaluation using OPT-6.7B to provide a conservative assessment of TBW's quality advantages. Given that our human evaluation demonstrated TBW's stronger performance on models with larger vocabularies, using OPT-6.7B (with its smaller 50,272-token vocabulary) represents a more challenging test case for our method.

We evaluate text quality across four of the five dimensions used in our human evaluation using 20 C4 prompts and the same watermark strength ($\delta = 2.0$). For each prompt, we generate watermarked text using TBW, Unigram, and SynthID-Text, alongside non-watermarked baselines. Each evaluation uses a fresh GPT-4o conversation to avoid memory effects that could bias assessments. We use the following evaluation prompt to ensure consistent and impartial assessment:

> You are an impartial evaluator judging the fluency and naturalness of a text sample. Please rate the sample without knowing how it was produced. Based on the readability and quality, answer the following criteria using a 1–5 scale:
>
> - Fluency (1 = disfluent/unnatural, 5 = perfectly smooth and human-like)
> - Grammatical correctness (1 = frequent grammar issues, 5 = grammatically perfect)
> - Coherence (1 = ideas are disjointed or confusing, 5 = ideas flow logically)
> - Lexical variety (1 = repetitive/basic word use, 5 = rich, diverse vocabulary)
>
> Sample:
> *{Sample Text}*

Table 8 presents the automated evaluation results. The findings demonstrate that TBW maintains text quality nearly identical to non-watermarked output across all evaluated dimensions. Most notably, TBW achieves comparable fluency (3.10 vs. 3.20), equivalent lexical variety (3.00 vs 3.00), and even slightly superior grammatical correctness (3.90 vs 3.85) compared to non-watermarked text.

In comparison to competing watermarking methods, TBW demonstrates clear advantages. Against Unigram, which showed similar robustness in our adversarial evaluations (§5.3), TBW achieves quality improvements: +0.45 in fluency, +0.55 in grammatical correctness, and +0.35 in coherence. Similarly, TBW outperforms the industry-standard SynthID-Text across all dimensions.

Overall, TBW consistently outperforms other watermarking schemes, matching or exceeding the

text quality of non-watermarked outputs across most dimensions. These findings are consistent with our human evaluation, where TBW maintained comparable quality scores to non-watermarked text, confirming that its robustness advantages do not come at the expense of fluency, coherence, or grammatical correctness.

### E.4 Efficiency Results

To verify that TBW's efficiency advantage holds across model scales, we repeat the generation time evaluation from §5.5 on OPT-2.7B. Figure 7 confirms that the same trends observed on OPT-6.7B hold at smaller scale since TBW introduces negligible overhead relative to non-watermarked generation, matching lightweight methods across all sequence lengths.

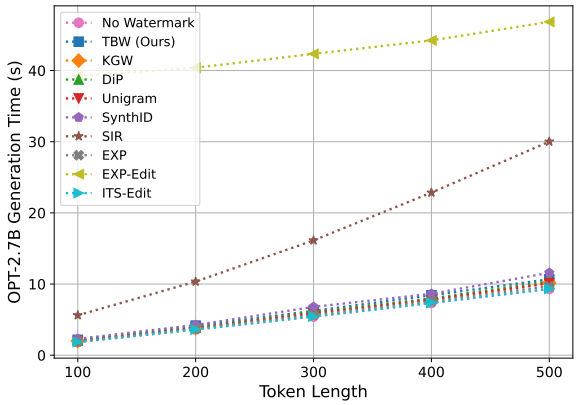



Figure 7: Comparison of average generation time (seconds) over various output token lengths from multiple watermarking schemes on OPT-2.7B.

### E.5 Comparison to ITS-Edit

In the main evaluation, we exclude EXP-Edit and ITS-Edit due to their poor perplexity values, which make them impractical for real-world use. However, for completeness, we assess the robustness of ITS-Edit in terms of ROC-AUC score, Best F1 score, TPR@1%FPR, and TPR@10%FPR, comparing it to TBW.

We follow the same evaluation procedure used in the main paper under a smaller sample size, generating 50 watermarked and 50 non-watermarked (baseline) samples using OPT-6.7B. We then apply two paraphrasers, PEGASUS and DIPPER, to the 200-token completions. ITS-Edit performs 500 detection runs per sample during the detection phase to better estimate the presence of the watermark.

As shown in Table 9, TBW consistently outperforms ITS-Edit across all evaluated robustness metrics, demonstrating superior detection performance while maintaining practical efficiency.

### E.6 Comparison to Semantic Watermarks

To situate TBW relative to semantic watermarking approaches that operate via post-processing, we compare against PostMark (Chang et al., 2024) and k-SemStamp (Hou et al., 2024b). A key methodological distinction is that both PostMark and k-SemStamp embed watermarks by rewriting already-generated text through additional LLM inference passes, whereas TBW embeds the watermark during a single standard decoding pass. This difference has direct implications for generation cost and deployment complexity.

We evaluate all methods over 100 C4 prompts on OPT-6.7B, measuring perplexity, generation time, ROC-AUC, and TPR@1%FPR under both no-attack and DIPPER paraphrasing conditions. TBW is evaluated at two watermark strengths ($\delta = 2.0$ and $\delta = 3.0$) to illustrate the quality-robustness trade-off. Results are reported in Table 10.

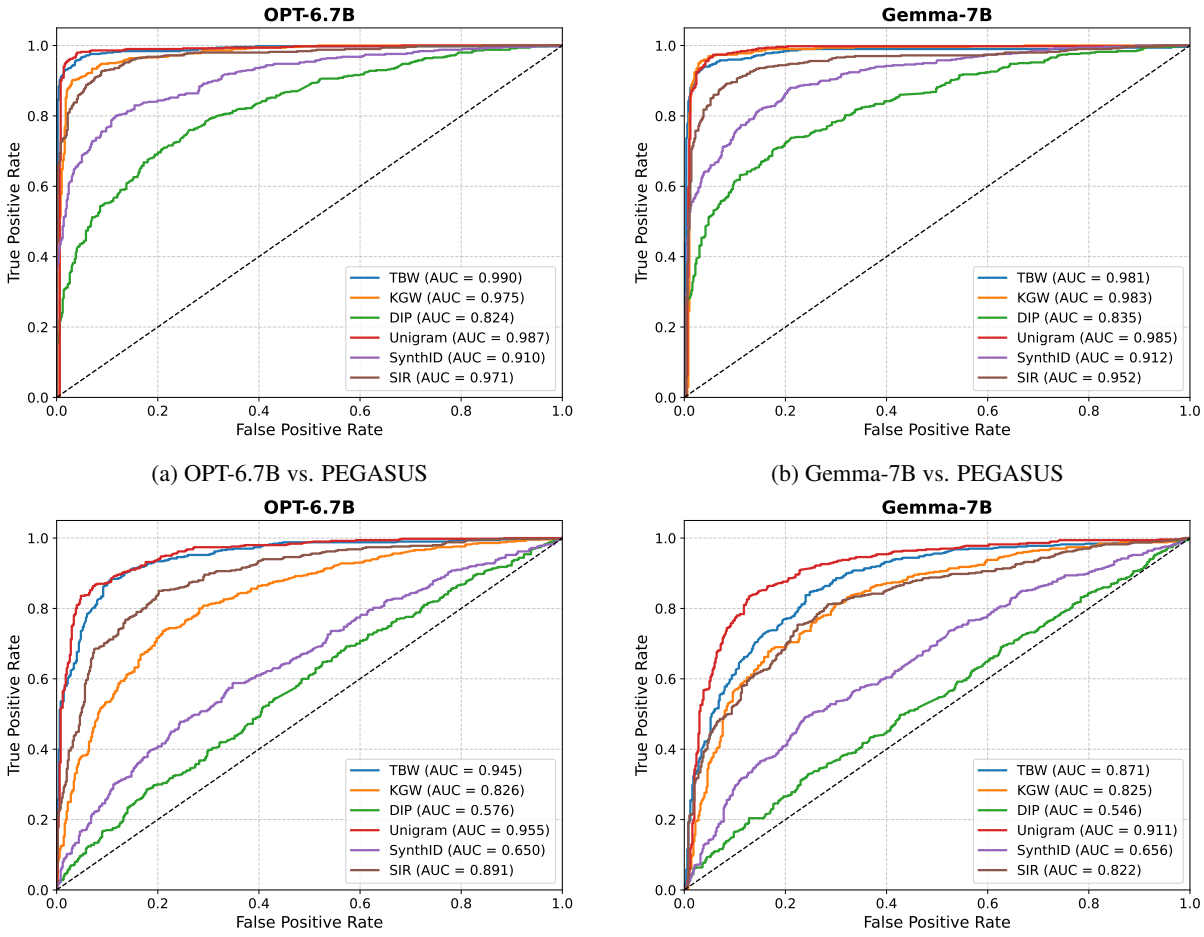
TBW at $\delta = 3.0$ achieves the highest ROC-AUC and TPR@1%FPR under both no-attack and DIPPER paraphrasing conditions, while being 6-13$\times$ faster than the semantic baselines. Perplexity remains comparable to PostMark and only moderately higher than k-SemStamp, whose lower perplexity comes at the cost of substantially higher generation time and lower detection performance. These results confirm that TBW's in-generation approach provides a fundamentally different and practically advantageous trade-off compared to post-processing semantic watermarks as it avoids the additional LLM inference passes required by rewriting-based methods while achieving stronger robustness and detection at a fraction of the computational cost.

### E.7 TBW Robustness ROC Curves

Figure 8 presents the ROC curves and corresponding AUC values for the evaluated watermarking methods. For OPT-6.7B, our method achieves comparable robustness to Unigram, demonstrating strong detection performance. For GEMMA-7B, we observe a slight reduction in robustness; however, this trade-off comes with improved text quality. The difference in AUC between our method and Unigram is minimal, approximately 4%, highlighting the balance between robustness and text

| Attack | Method | ROC-AUC | Best F1 Score | TPR@1% FPR | TPR@10% FPR |
|---|---|---|---|---|---|
| No Attack | TBW | **0.999** | **0.990** | **0.980** | **1.000** |
| | ITS-EDIT | 0.043 | 0.667 | 0.000 | 0.000 |
| PEGASUS | TBW | **0.959** | **0.939** | **0.800** | **0.920** |
| | ITS-EDIT | 0.417 | 0.667 | 0.000 | 0.100 |
| DIPPER | TBW | **0.929** | **0.875** | **0.575** | **0.840** |
| | ITS-EDIT | 0.519 | 0.667 | 0.020 | 0.040 |

Table 9: Comparison of robustness metrics between TBW and ITS-Edit on OPT-6.7B, evaluated using PEGASUS and DIPPER paraphrasers where **bold** indicates better scores.




(a) OPT-6.7B vs. PEGASUS

(b) Gemma-7B vs. PEGASUS

(c) OPT-6.7B vs. DIPPER

(d) Gemma-7B vs. DIPPER

Figure 8: ROC curves comparing watermark methods on OPT-6.7B and Gemma-7B against PEGASUS (top) and DIPPER (bottom) paraphrasing attacks.

| Method | Time | PPL | No Attack | | DIPPER | |
|---|---|---|---|---|---|---|
| | (s/sample) | (mean ± std) | ROC-AUC | TPR@1% | ROC-AUC | TPR@1% |
| TBW ($\delta$=2.0) | 5.23 ± 1.93 | 12.36 ± 3.64 | 0.971 | 0.879 | 0.694 | 0.222 |
| TBW ($\delta$=3.0) | 5.08 ± 2.01 | 12.46 ± 4.80 | **0.986** | **0.949** | **0.802** | **0.357** |
| k-SemStamp | 65.93 ± 75.03 | **9.16 ± 2.85** | 0.960 | 0.750 | 0.743 | 0.150 |
| PostMark | 29.61 ± 17.56 | 12.43 ± 2.90 | 0.968 | 0.920 | 0.780 | 0.210 |

Table 10: Comparison of TBW against semantic black-box watermarking methods on OPT-6.7B over 100 C4 prompts. **Bold** indicates best results. TPR@1% denotes TPR at 1% FPR.

quality.

### E.8 False Positive Rate (FPR) on Human-Written Text

#### E.8.1 Watermarking Methods FPR

To evaluate the practical viability of watermarking schemes, we assess false positive rates (FPR) on human-written content using the same experimental parameters as described in §5.1. We evaluated 1,000 human-written C4 samples (200 tokens) using TBW, KGW, DiP, Unigram, and SynthID detection schemes on both OPT-6.7B (vocabulary size: 50,272) and GEMMA-7B (vocabulary size: 256,000).

The results from Table 11 demonstrate that TBW maintains practical false positive rates across both models, with notably better performance on GEMMA-7B (0.20% FPR) compared to OPT-6.7B (0.70% FPR). This vocabulary-dependent behavior aligns with our hypothesis that larger vocabularies enable more precise semantic token partitioning, reducing false classifications of human-written content. While TBW shows slightly higher FPR on smaller vocabulary models, this trade-off is offset by the significant robustness advantages demonstrated in our main evaluation.

#### E.8.2 TBW Threshold FPR Sensitivity

To understand how the similarity threshold $\tau$ (used in Algorithm 1 for token-to-topic mapping) affects false positive rates, we conducted a threshold sensitivity analysis on GEMMA-7B using the same 1,000 human-written C4 samples (200 tokens). As the threshold increases, fewer tokens meet the semantic similarity criteria for topic assignment, resulting in more tokens being distributed via the round-robin mechanism.

Table 12 results demonstrate a clear threshold effect with very low thresholds ($\tau \leq 0.2$) lead to unacceptably high false positive rates (>30%), as most semantically aligned tokens meet the similarity criteria among topic lists, resulting in overly broad topic partitions that reduce watermark specificity. The optimal range appears to be $\tau \in [0.3, 0.9]$, where FPR remains below 2%, with $\tau \in [0.4, 0.9]$ achieving FPR below 1%. Our chosen threshold of $\tau = 0.7$ (highlighted in bold) provides a reasonable balance between semantic coherence and false positive control.

### E.9 Detection Attacks

Building on our threat model in §3.2, we also consider attacks that attempt to *detect* the presence of a watermark directly, rather than removing or obfuscating it through text manipulation. In this setting, the adversary does not seek to degrade or rephrase the text but instead aims to distinguish between watermarked and non-watermarked outputs based on statistical signals. One can argue that this corresponds to testing the "stealthiness" of the watermark. However, given that our topic-based watermarking scheme preserves text quality comparable to non-watermarked outputs, the watermark remains effectively stealthy in practice.

For completeness, we evaluate our method against non-watermarked text using two classes of detection strategies: (i) statistical hypothesis tests over generated output and (ii) supervised machine learning classification trained to separate watermarked and non-watermarked corpora. We conduct experiments across our two representative models (GEMMA-7B and OPT-6.7B) and evaluate at bias strengths $\delta = 2.0$ and $\delta = 3.0$. These values match those used in our main evaluations of text quality and robustness, allowing us to study detection performance under the same operating points and to compare trade-offs across quality, robustness, and detectability. While these experiments provide additional perspective, they rely on assumptions such as access to large paired corpora of watermarked

| Model | Method | FPR (%) | False Positives | Detection Score | Threshold |
|---|---|---|---|---|---|
| OPT-6.7B | **TBW** | 0.70 | 7/1000 | $0.437 \pm 1.664$ | 4.75 |
| | KGW | 0.00 | 0/1000 | $-0.112 \pm 1.033$ | 4.0 |
| | DiP | 0.20 | 2/1000 | $-0.016 \pm 0.504$ | 1.513 |
| | Unigram | 0.00 | 0/1000 | $-0.205 \pm 1.268$ | 4.0 |
| | SynthID | 0.40 | 4/1000 | $0.500 \pm 0.007$ | 0.52 |
| GEMMA-7B | **TBW** | 0.20 | 2/1000 | $-0.200 \pm 1.405$ | 4.75 |
| | KGW | 0.00 | 0/1000 | $-0.194 \pm 1.103$ | 4.0 |
| | DiP | 0.10 | 1/1000 | $-0.019 \pm 0.502$ | 1.513 |
| | Unigram | 0.00 | 0/1000 | $-0.459 \pm 1.252$ | 4.0 |
| | SynthID | 0.60 | 6/1000 | $0.500 \pm 0.007$ | 0.52 |

Table 11: False positive rate (FPR) analysis on 1,000 human-written C4 samples (200 tokens) across different watermarking schemes and language models. Detection thresholds follow the defaults from respective original implementations.

| Threshold ($\tau$) | FPR | False Positives | Detection Score |
|---|---|---|---|
| 0.1 | 33.2 | 332/1000 | $0.346 \pm 4.830$ |
| 0.2 | 33.2 | 332/1000 | $0.365 \pm 4.876$ |
| 0.3 | 1.60 | 16/1000 | $0.543 \pm 2.012$ |
| 0.4 | 0.10 | 1/1000 | $-0.217 \pm 1.591$ |
| 0.5 | 0.20 | 2/1000 | $0.318 \pm 1.467$ |
| 0.6 | 0.30 | 3/1000 | $-0.375 \pm 1.521$ |
| **0.7** | **0.60** | **6/1000** | $\mathbf{0.261 \pm 1.795}$ |
| 0.8 | 0.40 | 4/1000 | $0.232 \pm 1.455$ |
| 0.9 | 0.90 | 9/1000 | $0.216 \pm 1.912$ |

Table 12: Threshold sensitivity analysis for TBW on GEMMA-7B using 1,000 human-written C4 samples. Lower thresholds result in less semantically coherent topic partitions.

and non-watermarked outputs that rarely hold in deployed systems, and thus fall outside the scope of practical adversaries targeted by our primary threat model.

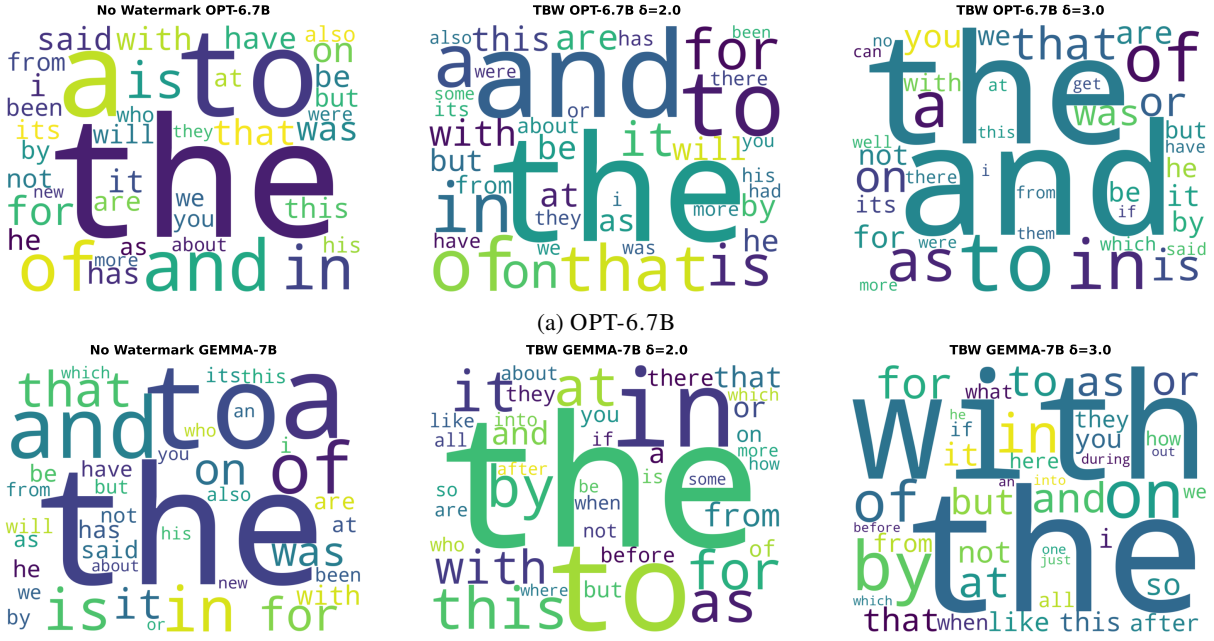
### E.9.1 Statistical Tests

We probe whether simple distributional cues can separate watermarked from non-watermarked outputs by comparing bag-of-words statistics over large corpora. From an attacker's perspective, the goal would be to identify watermark presence based on observable shifts in word frequencies. In particular, because our scheme biases topic-specific tokens, one might expect that watermarked outputs exhibit disproportionately higher rates of such tokens compared to non-watermarked outputs.

To test this, for each model and bias strength ($\delta \in \{2.0, 3.0\}$) we generate 1,000 watermarked and 1,000 non-watermarked completions using the same prompts as in our main experiments: the first 100 tokens of each C4 sample serve as the prompt, and the model produces $200 \pm 5$ tokens for analysis. To remain tokenizer-agnostic and focus on surface lexical signals, we lowercase, strip punctuation, split on whitespace, and compute corpus-level word counts. From these counts we derive the top-$N$ normalized frequencies and visualize the relative mass using word clouds in Figure 9.

Across all models and bias strengths, the most frequent tokens are dominated by common function words such as "the," "and," "to," and "that." While certain settings show minor variation (e.g., the word "with" appears slightly more often in GEMMA-7B TBW at $\delta = 3.0$), these tokens are also prevalent in the corresponding non-watermarked corpora, and no systematic elevation of topic-specific tokens is observed. This outcome is precisely what we expect from a stealthy watermark where the lexical distribution remains consistent with non-watermarked text even under naive frequency analysis.

We note that for Gemma, the watermarked outputs exhibit slightly flatter token distributions compared to the corresponding non-watermarked outputs. While this effect is subtle and not readily visible without corpus-level aggregation, it is consistent across bias strengths and may indicate a mild redistribution of lexical mass. Nonetheless, in realistic deployments, attackers would not have access to paired corpora of watermarked and non-watermarked text from the same model, and thus could not easily identify subtle differences even if they existed. The closest analogue is Google's evaluation of SynthID-Text, where generations from the same model included both watermarked and non-watermarked outputs without disclosure to




(a) OPT-6.7B

(b) GEMMA-7B

Figure 9: Word clouds of the top-40 normalized token frequencies for OPT-6.7B (top) and GEMMA-7B (bottom). From left to right, each row shows outputs from: (i) non-watermarked generations, (ii) TBW with bias $\delta = 2.0$, and (iii) TBW with bias $\delta = 3.0$. Across both models and bias strengths, the distributions are dominated by common function words (e.g., "the," "and," "to"), with no systematic elevation of topic-specific tokens.

users (SynthID-Team, 2024; Dathathri et al., 2024). However, since TBW preserves the same dominant distribution of function words across conditions, such comparisons likewise fail to yield clear evidence of watermark presence. In short, statistical tests do not provide a practical detection vector against TBW.

### E.9.2 Machine Learning Classification

We next test whether a supervised classifier can distinguish between watermarked and non-watermarked text. For each model and bias strength ($\delta \in \{2.0, 3.0\}$), we collect 1,000 non-watermarked completions from unique prompts and 1,000 watermarked completions from a disjoint set of prompts. As in our main experiments, the first 100 tokens of each C4 sample are used as input, and the model generates $200 \pm 5$ tokens for analysis. These 2,000 samples form the training set, where we assign a label of 0 for non-watermarked text and 1 for watermarked text. For evaluation, we use 500 unseen prompts from the same pool to generate 500 non-watermarked and 500 watermarked samples, forming a balanced test set. We train a BERT-BASE (Devlin et al., 2019) classifier on the training data using the hyperparameters reported in Table 13, and evaluate on the held-out test set.

| Hyperparameter | Value |
|---|---|
| Epochs | 3 |
| Batch size (per device) | 16 |
| Learning rate | 5e-5 |
| Optimizer | AdamW |

Table 13: Fine-tuning hyperparameters for BERT-BASE used in the classification experiments.

Table 14 reports accuracy and F1 scores for OPT-6.7B and GEMMA-7B under $\delta \in \{2.0, 3.0\}$. For OPT, classification accuracy reaches 0.80-0.83 with F1 scores of 0.75-0.80, indicating that supervised models can learn a moderately reliable separation between watermarked and non-watermarked corpora. In contrast, GEMMA achieves near-perfect accuracy and F1 ($\approx 0.99$) across both bias strengths. We hypothesize this stems from the slightly flattened token distributions observed in the statistical analysis, which BERT-BASE is able to exploit effectively.

Despite these results, the assumptions behind this evaluation remain unrealistic for real-world adversaries. Training the classifier requires thousands of labeled watermarked and non-watermarked samples, whereas deployed systems consistently apply

| Language Model | Bias $\delta$ | Accuracy | F1 |
|---|---|---|---|
| OPT-6.7B | 2.0 | 0.80 | 0.75 |
| OPT-6.7B | 3.0 | 0.83 | 0.80 |
| GEMMA-7B | 2.0 | 0.99 | 0.99 |
| GEMMA-7B | 3.0 | 0.99 | 0.99 |

Table 14: Classification performance of BERT-BASE trained to distinguish watermarked from non-watermarked text. Results are reported as Accuracy and F1 on a held-out test set of 1,000 samples (500 watermarked, 500 non-watermarked).

watermarking and never reveal non-watermarked outputs. In principle, an attacker could attempt to scrape known non-watermarked text online and compare it against API outputs, but such a classifier would primarily be distinguishing across models or domains rather than detecting watermark presence. This approach has been attempted in prior work (Islam et al., 2023; Bafna et al., 2024), but it misses the purpose of watermarking to provide a lightweight provenance signal that avoids constant retraining of detectors as model outputs evolve. We further discuss the practicality and assumptions of these detection-based attacks, and why they remain outside our primary threat model, in the following section.

#### E.9.3 Discussion and Implications

This section considers the broader implications of our detection-based evaluations. While our statistical and machine-learning tests provide insight into the distributional behavior of TBW, they do not translate into practical adversarial strategies under real deployment conditions.

As discussed previously, an attacker would almost never have access to both watermarked and non-watermarked outputs from the same model. The most feasible alternative is that an attacker could continuously query a deployed API and examine lexical distributions for anomalies. However, our word cloud analysis (Figure 9) shows that the most frequent tokens remain dominated by common function words (e.g., "the," "and," "to"), with no systematic inflation of topic-specific tokens. This suggests that distributional probing would not expose the watermark in practice.

Similarly, supervised classification attacks (Table 14) assume the availability of thousands of labeled samples of both watermarked and non-watermarked text. In deployed settings, such paired corpora are inaccessible, since watermarking is applied uniformly and non-watermarked outputs are never exposed to end users. At best, an adversary could attempt to scrape known non-watermarked text online and compare it against API outputs. Yet in this case, the classifier would be distinguishing differences across models or domains, not detecting the presence of a watermark. The distinction is important as watermarking aims to provide a provenance signal robust to model drift, not to enable adversaries to retrain detectors whenever models evolve linguistically.

One might argue that detection attacks could become more feasible in open-source ecosystems where both watermarked and non-watermarked versions of a model are released (e.g., through repositories like Hugging Face (Hugging Face, 2025)). While this is a theoretical concern, watermarking is most critical in contexts of malicious misuse, academic dishonesty, or provenance tracking which are cases that overwhelmingly involve flagship proprietary systems rather than community-maintained open-source checkpoints. In practice, deployed systems do not expose both versions of a model to end users, and thus the adversarial assumptions behind detection-based attacks are rarely satisfied.

In summary, our analysis reinforces that detection attacks are neither representative of realistic adversarial capabilities nor an effective threat to TBW. By contrast, the practical adversaries captured in our primary threat model by those performing text degradation or semantic rephrasing are explicitly addressed by our scheme, and our evaluation demonstrates that TBW maintains robustness under these conditions.

### E.10 Maximum $z$-Score Detection: Detailed Analysis

Given the superior performance of the maximum $z$-score detection method in Table 3, we provide detailed performance metrics and distribution analysis. This evaluation uses 500 watermarked and 500 non-watermarked samples generated under our standard experimental configuration.

#### E.10.1 Performance Metrics

Table 15 reports standard classification metrics at the default operating threshold for maximum $z$-score detection. Results confirm exceptional performance across accuracy, precision, recall, F1-score, and ROC-AUC, with GEMMA-7B achieving the

| Language Model | Accuracy | Precision | Recall | F1 score | ROC–AUC | FPR |
|---|---|---|---|---|---|---|
| OPT-6.7B | 0.921 | 0.866 | 0.996 | 0.927 | 0.996 | 0.154 |
| GEMMA-7B | 0.994 | 0.988 | 1.000 | 0.994 | 1.000 | 0.012 |

Table 15: Classification performance of the maximum $z$-score detection method on 500 watermarked and 500 non-watermarked samples. The method achieves consistently high accuracy, precision, recall, and ROC-AUC, with low false positive rates (FPR).

highest scores in all categories. The method maintains extremely low false positive rates (0.154 for OPT-6.7B and 0.012 for GEMMA-7B) while sustaining near-perfect recall, making it well-suited for deployment scenarios where false accusations of AI generation must be minimized. Figure 10 presents the corresponding ROC curves, illustrating the near-perfect separation between watermarked and non-watermarked content.

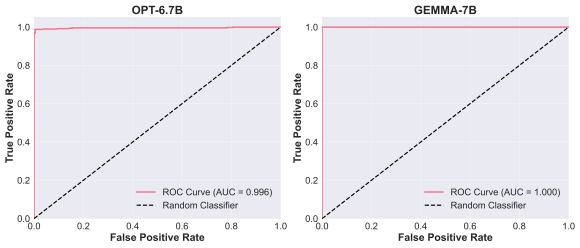



Figure 10: ROC curves for maximum $z$-score detection on OPT-6.7B and GEMMA-7B. Both models achieve AUC values of 0.996 and 1.000, indicating near-perfect separation between watermarked and non-watermarked content.

### E.10.2 $z$-Score Distribution Analysis

As discussed in Table 15, the false positive rate (FPR) for OPT-6.7B is higher than for GEMMA-7B, despite both achieving near-perfect recall. To better understand this behavior, we examine the $z$-score distributions for the maximum $z$-score detection method. Figure 11 shows that while both models exhibit clear bimodal separation between watermarked and non-watermarked text, the non-watermarked distribution for OPT-6.7B lies slightly closer to the detection threshold than for GEMMA-7B. This overlap explains the higher FPR observed for OPT-6.7B.

Importantly, this analysis indicates that adjusting the decision threshold upward would reduce false positives for OPT-6.7B without significantly impacting recall, as the watermarked distribution remains well-separated from the non-watermarked

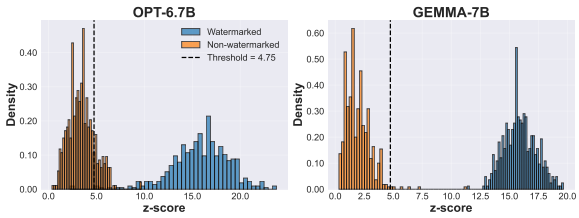



Figure 11: $z$-score distributions for the maximum $z$-score detection method on OPT-6.7B and GEMMA-7B. Both show clear separation between watermarked and non-watermarked text, with OPT-6.7B's closer overlap explaining its higher false positive rate.

one. For GEMMA-7B, the separation is even more pronounced, producing extremely low FPR values under the same threshold. These findings highlight the flexibility of the maximum $z$-score method with strong separation allowing operators to tune detection thresholds to meet application-specific trade-offs between false positives and false negatives. Our sensitivity analysis (F.1) further demonstrates that this separation is maintained across watermark strengths.

## F Ablation Studies

### F.1 TBW Strength Sensitivity

To provide deeper insights into the trade-offs between watermark detectability and text quality, we analyze how varying the bias parameter $\delta$ affects both detection performance and lexical diversity. This analysis addresses key questions about parameter selection and the practical implications of different watermark strengths for deployment scenarios.

**Experimental Setup.** We conduct this evaluation using OPT-6.7B, which serves as a *worst-case evaluation setting* due to its relatively small vocabulary size (50,272 tokens) compared to GEMMA-7B (256,000 tokens). As demonstrated throughout our evaluation, TBW's performance scales positively with vocabulary size, so any robustness and diversity preservation observed here represents a

conservative lower bound for larger models.

We evaluate 100 samples from the C4 dataset with 100-token prompts, generating $200 \pm 5$ tokens. We test three similarity thresholds $\tau \in \{0.3, 0.5, 0.7\}$ across five bias strengths $\delta \in \{0.0, 1.0, 2.0, 5.0, 10.0\}$. Multiple $\tau$ values are included to validate that TBW's semantic partitioning remains stable under both looser and stricter similarity constraints. To quantify lexical diversity, we report Distinct-N metrics ($N = 1, 2, 3, 4$), which measure the ratio of unique n-grams to total n-grams. Higher Distinct-N values indicate greater lexical variety, while lower values suggest increased repetition and potential quality degradation.

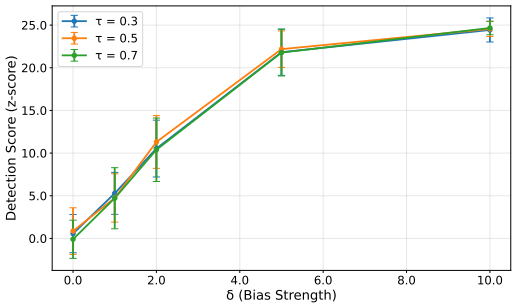



Figure 12: Detection score ($z$-score) vs. bias strength $\delta$ across similarity thresholds. Higher $\delta$ yields stronger watermark signals, with detection saturating around $\delta = 5.0$.

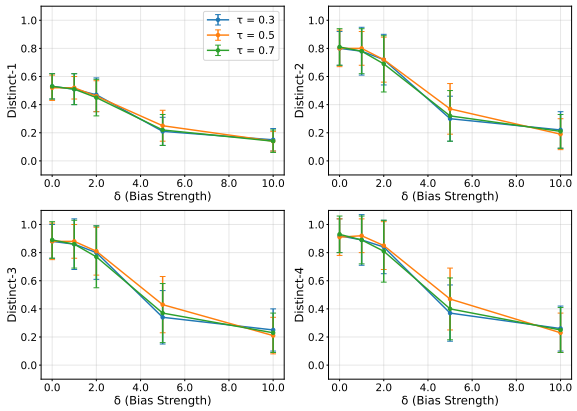



Figure 13: Lexical diversity (Distinct-N) vs. bias strength $\delta$. Moderate $\delta$ values maintain diversity, while very high strengths lead to increased repetition.

**Detection-Quality Trade-off.** Figures 12 and 13 illustrate the core trade-off in watermark strength. As $\delta$ increases, detection scores rise monotonically across all $\tau$ values, reaching $z$-scores above 24 at $\delta = 10.0$ (Figure 12). However, higher strengths gradually reduce lexical diversity (Figure 13), with the most notable changes occurring between $\delta = 2.0$ and $\delta = 5.0$. This pattern underscores the importance of moderate watermark strengths for balancing detectability and output quality. Notably, $\tau$ has minimal influence on either detection performance or diversity, indicating that semantic alignment can be optimized independently of other quality metrics. Additional visualizations covering the full parameter space are provided in F.2.

**Practical Recommendations.** For most deployment scenarios, we recommend $\delta = 2.0$-$3.0$, which sits just before the diversity inflection point in Figure 13 and well past the detection saturation zone in Figure 12. In this range, TBW consistently achieves strong detection performance ($z$-score $> 10$) while preserving high lexical variety (Distinct-N $> 0.7$). A similarity threshold of $\tau = 0.7$ provides optimal semantic coherence without sacrificing detection capability.

**Overall Comparison.** Across all metrics, TBW strikes the most favorable balance relative to prior watermarking schemes. In terms of accuracy, TBW's maximum $z$-score detection consistently achieves near-perfect ROC-AUC ($> 0.99$), remaining competitive with more expensive multi-pass schemes. On robustness, TBW substantially outperforms lightweight methods such as SynthID and KGW under both lexical perturbation and full-text paraphrasing, closing much of the gap to heavyweight approaches (e.g., EXP, ITS-Edit) without their quality degradation. Finally, in efficiency, TBW introduces negligible generation overhead, matching production-ready systems while delivering robustness gains that prior efficient schemes cannot achieve. Taken together, these results underscore TBW's practicality for real-world deployment.

### F.2 Parameter Space Analysis

Figures 14 and 15 provide complementary views of the complete parameter space. The scatter plots in Figure 14 reveal that as detection scores increase with higher $\delta$ values, lexical diversity consistently decreases across all Distinct-N metrics, with threshold values showing negligible variation. This relationship is further confirmed by the heat map in Figure 15, which illustrates that lower bias strengths ($\delta \leq 2.0$) maintain high Distinct-N scores ($> 0.7$) across all thresholds, while aggressive watermarking ($\delta \geq 5.0$) results in substantial quality degradation (Distinct-N $< 0.5$). The vertical consistency in the heat map reinforces that similarity threshold

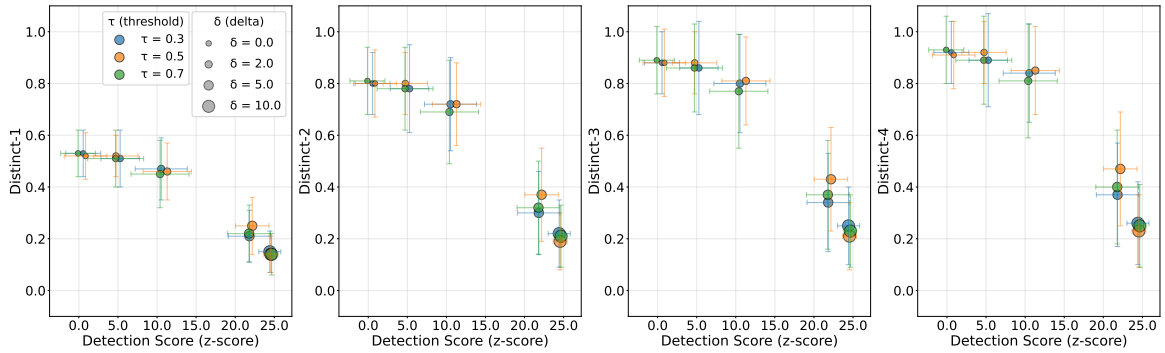



Figure 14: Distinct-N scores vs. detection scores across parameter combinations. The relationship demonstrates consistent quality degradation as watermark strength increases, with minimal threshold influence.

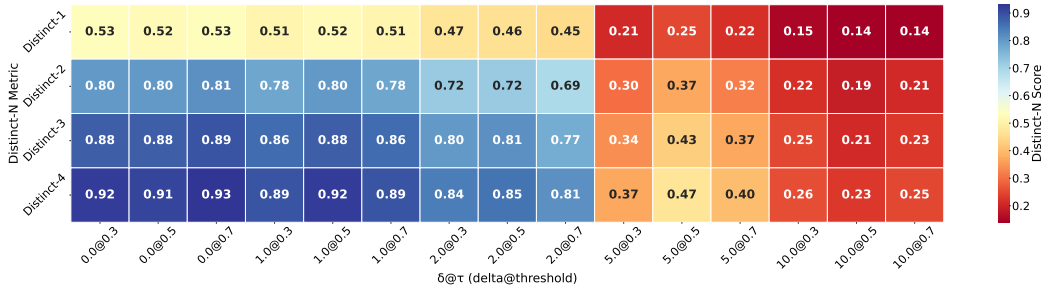



Figure 15: Comprehensive heat map of Distinct-N metrics across bias strength $\delta$ and similarity threshold $\tau$ combinations.

$\tau$ can be selected based on semantic considerations without significantly affecting lexical diversity.

### F.3 Task-Specific Evaluation on Summarization

To evaluate TBW's performance on downstream tasks beyond general text generation, we conduct a targeted experiment on scientific document summarization. This evaluation addresses a critical question for practical deployment on whether topic-based watermarking maintains effectiveness when applied to domain-specific, constrained generation tasks that require both semantic coherence and factual accuracy.

#### F.3.1 Experimental Setup

We use the TLDR dataset (Cachola et al., 2020), which consists of arXiv abstracts paired with reference one-sentence summaries, providing a summarization task within well-defined scientific domains. For this evaluation, we define five domain-specific topics aligned with the dataset's content: {`physics`, `chemistry`, `mathematics`, `biology`, and `computer science`}.

Using OPT-6.7B, we condition the model on scientific abstracts and prompt it to generate single-sentence summaries of $50 \pm 5$ tokens. This constrained generation setting tests whether TBW can maintain watermark detectability while preserving task-specific quality in a domain where precision and semantic accuracy are paramount. We evaluate summarization quality using ROUGE-1, ROUGE-2, and ROUGE-L metrics, which measure n-gram overlap and longest common subsequence matching between generated and reference summaries, providing standard benchmarks for summarization performance. We employ our standard similarity threshold $\tau = 0.7$ aligned with our main evaluations.

#### F.3.2 Quality and Robustness Results

Table 16 presents the summarization quality comparison between watermarked and non-watermarked outputs. TBW summaries demonstrate superior performance across all ROUGE metrics compared to non-watermarked baselines.

The consistent quality improvements across all ROUGE metrics suggest that TBW's topic-aware token selection provides benefits for domain-specific tasks. We attribute these improvements to TBW's semantic biasing mechanism, which guides token selection toward domain-appropriate vocab-

| Metric | TBW | Non-Watermarked |
|---|---|---|
| ROUGE-1 | $\mathbf{0.190 \pm 0.111}$ | $0.158 \pm 0.098$ |
| ROUGE-2 | $\mathbf{0.044 \pm 0.073}$ | $0.018 \pm 0.035$ |
| ROUGE-L | $\mathbf{0.131 \pm 0.075}$ | $0.112 \pm 0.070$ |

Table 16: Summarization quality comparison on TLDR dataset using OPT-6.7B. Bold indicates better performance.

ularies during generation. By favoring tokens semantically aligned with scientific topics, the watermarking process promotes technical terminology and conceptual relationships that better match reference summary patterns.

| Detection Metric | Value |
|---|---|
| AUC-ROC | 0.946 |
| TPR@1%FPR | 0.860 |
| TPR@10%FPR | 0.880 |

Table 17: TBW detection performance on scientific summarization task using OPT-6.7B.

Table 17 demonstrates that these quality improvements do not compromise watermark detectability. TBW maintains robust detection performance with AUC-ROC of 0.946 and high TPR values at stringent false positive thresholds, confirming reliable detection capabilities suitable for production deployment on specialized tasks.

### F.4 Scaling the Number of Topics

This section provides full experimental details and additional visualizations for the topic scalability analysis presented in §5.6.

#### F.4.1 Experimental Setup.

Unless noted here, we follow the protocol used in the main experiments. We use GEMMA-7B because its larger vocabulary is more representative of deployed LLMs than OPT-6.7B, and because scaling $K$ increases partition granularity. We sample 100 prompts from C4, taking the first 100 tokens as the input prompt and generating $200 \pm 5$ tokens. For watermarking we apply a logit bias $\delta = 2.0$ and a semantic threshold $\tau = 0.5$ when constructing topic token lists. These settings are relaxed relative to the main experiments to compensate for smaller per-topic coverage at larger $K$ (i.e., admitting more tokens per topic while avoiding over-biasing). Detection uses the maximum-$z$ scheme from the main paper. For text quality we report BERTScore F1 (Zhang et al., 2020b), following prior work (Zhang et al., 2024).

We adopt generic, broad-coverage topic inventories designed to be distinct rather than near-synonymous, starting from the four topics used in the main experiments:

- $K = 4$: {`animals`, `technology`, `sports`, `medicine`}
- $K = 8$: $K$=4 + {`politics`, `entertainment`, `education`, `finance`}
- $K = 16$: $K$=8 + {`science`, `law`, `food`, `travel`, `environment`, `religion`, `fashion`, `history`}
- $K = 32$: $K$=16 + {`art`, `military`, `gaming`, `literature`, `parenting`, `space`, `transportation`, `psychology`, `agriculture`, `housing`, `cryptocurrency`, `architecture`, `economics`, `fitness`, `relationships`, `mythology`}

As in the main method, tokens are assigned to a topic if their embedding similarity exceeds $\tau$; unassigned tokens are round-robined to balance coverage.

#### F.4.2 Results

**Text quality vs. $K$.** Using the same setup, we assess BERTScore F1 as a text-quality metric (Zhang et al., 2024). Figure 16 shows no degradation as $K$ increases with scores remaining flat with low variability. This indicates that relaxing $\tau$ to 0.5, in order to maintain reasonable per-topic coverage, together with a moderate $\delta$=2.0, preserves fluency and semantics.

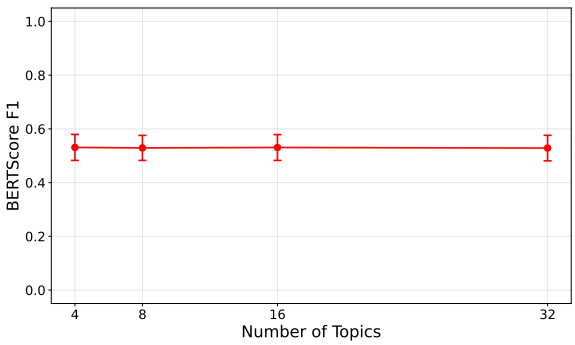



Figure 16: Text quality vs. number of topics. BERTScore F1 remains flat as $K$ increases, indicating no quality degradation. Error bars denote $\pm$s.d.

**Joint view quality vs. strength.** To visualize the trade-off directly, Figure 17 scatters per-sample $z$-score against BERTScore F1 across all $K$. Most points cluster in $z \in [8, 12]$ and F1$\in [0.50, 0.60]$, with the $K = 32$ points shifted modestly lower in $z$ (consistent with Figure 5) but without a quality penalty. Larger $K$ thus modestly weakens the detectable signature while keeping quality stable.

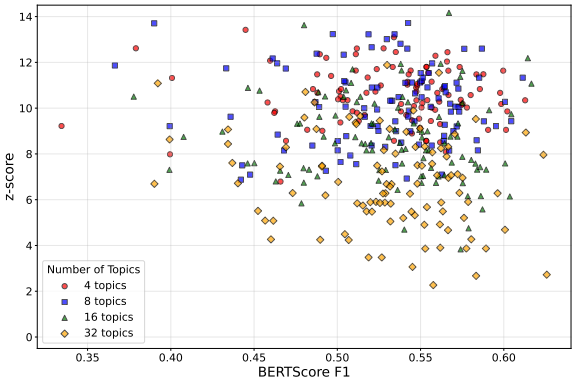



Figure 17: Detection vs. quality trade-off. Per-sample $z$-score (max-$z$) vs. BERTScore F1 across all $K$. Points cluster in $z \in [8, 12]$, F1$\in [0.50, 0.60]$; $K = 32$ shifts modestly lower in $z$ without affecting F1.

## G Attack Configuration Details

### G.1 Paraphrasing Attacks

**PEGASUS.** We employ PEGASUS (Zhang et al., 2020a), a sequence-to-sequence model pre-trained on unlabeled text for abstractive summarization. For paraphrasing evaluation, we configure the model with `num_sequences=1` and `num_beams=3` to generate single paraphrased outputs with controlled beam search diversity.

**DIPPER.** We utilize DIPPER (Krishna et al., 2023), a paraphrase generation model built by fine-tuning T5 for controllable text rewriting. Following the model's control code specification (Krishna, 2023), we set lexical diversity to `lexical=20` and syntactic order diversity to `order=40`, corresponding to high lexical and syntactic variation (L80-O60 in the original paper's notation).

### G.2 Lexical Perturbation Attacks

We implement combination attacks that uniformly distribute perturbations across three edit types: insertion, deletion, and substitution. For each perturbation percentage $p$, we calculate the total number of edits as $\lfloor p \times \frac{n}{100} \rfloor$ where $n$ is the text length, then allocate edits equally across operations (insertion: $\lfloor \text{total}/3 \rfloor$, deletion: $\lfloor \text{total}/3 \rfloor$ , substitution: remainder).

**Random Perturbations.** Words are selected uniformly at random from the text. Insertions use high-frequency terms from the Reuters corpus (Lewis et al., 2004), deletions remove arbitrary tokens, and substitutions replace words with random high-frequency alternatives.

**Targeted Perturbations.** Following the intuition that watermarks may concentrate in semantically important tokens, we target words with Part of Speech (POS) tags indicating nouns, verbs, adjectives, and adverbs (tags beginning with 'N', 'V', 'J', 'R'). Substitutions utilize WordNet (Fellbaum, 1998) synonyms when available, while insertions and deletions preferentially operate on this linguistically important subset.

## H Watermarked Text Examples

To provide insight into the practical impact of TBW on text generation quality, we present representative examples comparing watermarked and non-watermarked outputs across both evaluated models in Table 18. These samples demonstrate that TBW maintains semantic coherence and natural language flow while embedding detectable watermark signals. Examples are selected from the C4 evaluation dataset using our standard experimental configuration ($\delta = 3.0$, $\tau = 0.7$).

| idx | Model | Prompt | Real Completion | No Watermark | Watermarked |
|---|---|---|---|---|---|
| 27 | OPT-6.7B | [...] is for those updated engines to be able to comply with pollution standards.<br>The Environment and Public Works panel advanced the measure to the floor of the Senate by voice vote. Carper, the panel's ranking Democrat, | had initially authored the bill more than a decade ago with former Ohio Sen. George Voinovich (R).<br>"With today's vote, we're one step closer to making sure this bipartisan tradition, imagined and inc [...truncated] | said the bill would provide an incentive for more than $1 billion in investments in diesel engine retrofits and would help reduce harmful emissions and improve air quality. The bill would expand the DERA program to include [...truncated] | described DERA as a "crucial issue that has failed to get sufficient attention from our colleagues in recent years despite significant environmental and public health concerns." He explained that about 1 billion tons of [...truncated] |
| 177 | OPT-6.7B | [...] start building a diesel version of the Grand Cherokee next year, its first diesel passenger vehicle in several years.<br>That confirms a statement by Chrysler CEO Sergio Marchionne from last April that the popular Jeep | crossover would get a diesel option, which he reiterated in October. The new clean-diesel sport-utility vehicle was buried in an announcement that Chrysler would add a third production shift at its Jeep[...truncated] | Grand Cherokee would get a diesel version.<br>In April, he told reporters "I'm very confident that we're going to build a diesel Grand Cherokee." Diesels have been slow to take off in the U.S. But the need for [...truncated] | Grand Cherokee, which accounts for nearly 30 percent of the U.S. market, will have diesel capability when they launch, and that they already have plans to add diesel four-door Jeeps next year, with more diesel [...truncated] |
| 456 | GEMMA-7B | [...] of minimalism.<br>For prints isn't something expected from this storied house of minimalism. Monochrome is more its style. Those prints were subtle yet eye-opening on many levels. They showed Mr. Costa was again | pushing the boundaries of minimalism, this time by proposing a riveting, inspired kind of glam minimalism. | thinking outside the box, and not just in terms of the printed pieces but in the silhouettes and the new mixing of materials.<br>The designer created a collection that reflected a New York City woman's [...truncated] | taking risks at what could become his signature look (I say signature as the man always seems to hit just the right button). The bold color against these lush materials like herm wool or cashmere looked luxurious but at [...truncated] |
| 233 | GEMMA-7B | [...], it was never feasible. In brief, the report cited by Axios suggested that the only way to truly secure the next generation of wireless networks, on which critical infrastructure like self-driving cars will rely, against snooping by China | and others, would be for the government to build that network itself. There are several things wrong with this idea. You probably thought of a couple before you even got to this sentence. That would[...truncated] | is for the government to build its own network. The idea is that the government would operate the network, rather than private companies, and place it under strict security protocols. The idea was put forth, according to the report[...truncated] | or other hostile powers is if the federal government operates the service itself (there were some suggestions as well to use military bases as anchors for these connections.) As you can imagine, given how many people[...truncated] |

Table 18: Text generation examples comparing non-watermarked and TBW watermarked outputs across OPT-6.7B and GEMMA-7B models with $\delta = 3.0$, $\tau = 0.7$.