The Mark Fades: Adaptive Evolutionary Paraphrase-based Attack against LLM Watermarks

Yusheng Zhao[2*†], **Jian Zhao**[1†‡], **Tianle Zhang**[1], **Feng Wei**[3], **Xuelong Li**[1‡]
Institute of Artificial Intelligence (TeleAI), China Telecom[1]
University of Science & Technology of China (USTC)[2]
Artificial Intelligence Institute, CAICT[3]

Abstract

While LLM watermarking is pivotal for identifying machine-generated content, existing paraphrase-based attacks struggle to achieve an optimal balance between watermark removal efficacy and the text quality. To address this limitation, We propose TSAPA, a training-free evolutionary framework that formulates watermark removal as a constrained multi-objective optimization problem. By leveraging genetic algorithms to navigate the Pareto front, TSAPA utilizes a Pseudo-Log-Likelihood (PLL)-guided mutation strategy to precisely target and modify watermark-carrying tokens. Extensive experiments on Qwen3 series (1.7B/8B/32B) across diverse watermarking schemes demonstrate that TSAPA achieves an attack success rate (ASR) exceeding 90% while maintaining superior text semantic fidelity, significantly outperforming baseline methods. This work exposes critical vulnerabilities in current watermark techniques and provides a novel perspective for their robust evaluation.

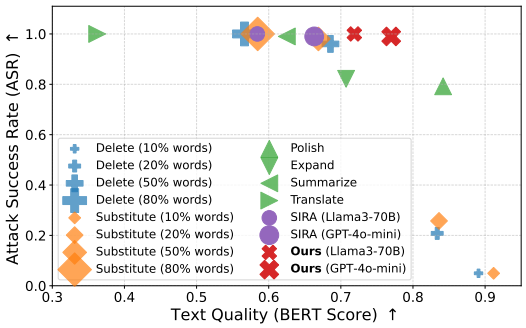


Figure 1: Superior trade-off between Attack Success Rate (ASR) and Text Quality achieved by our method. Evaluated on the article continuation task (Qwen3-32B, EXP scheme), our method (red crosses) significantly outperforms baselines by achieving near 100% ASR while maintaining a high BERTScore (0.73–0.77), surpassing the semantic fidelity of baseline attack methods.

1 Introduction

The rise of Large Language Models (LLMs) has necessitated the tracking of machine-generated text (MGT) for both intellectual property protection and content safety, thereby elevating MGT detection (Mitchell et al., 2023) to a paramount research challenge. Text watermarking has emerged as a prominent solution, embedding imperceptible statistical signals within model outputs to facilitate reliable identification (Kirchenbauer et al., 2023a; Christ et al., 2024). However, the practical utility of any watermarking scheme hinges on its robustness against adversarial manipulation (Kirchenbauer et al., 2023b; Sadasivan et al., 2023). Existing removal attacks have shown that watermarks can be disrupted via techniques ranging from elementary text editing (He et al., 2024; Piet et al., 2025; Zhang et al., 2025) to sophisticated LLM-based paraphrasing (Krishna et al., 2023; Kirchenbauer et al., 2023b; Cheng et al., 2025). Yet, these attacks face an unavoidable dilemma: the trade-off between removal efficacy and the preservation of text quality (Sadasivan et al., 2023). Subtle rewriting often leaves the watermark intact, whereas more aggressive modifications tend to degrade the semantic coherence and linguistic nuance essential to the text's utility.

In this work, we **first** systematically investigate the vulnerabilities of various watermarking schemes through a hierarchical stress test. Our evaluation moves from basic, randomized edits, such as word-level deletion and substitutions, to advanced LLM-driven paraphrasing techniques, including polishing, summarizing, expansion and one-round-back translation. Based on those investigations, we formulate the watermark removal as a constrained multi-objective optimization prob-

*Work done during an internship at TeleAI.
†These authors contributed equally.
‡Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 9428–9445

lem. **Consequently**, we propose the Training-free Self-Adaptive Paraphrase-based Attack (TSAPA). By employing a genetic algorithm, TSAPA iteratively navigates the Pareto front, evolving a candidate population to maximize watermark erasure while preserving semantic fidelity. **Lastly**, extensive experiments are conducted on the Qwen3 series (1.7B, 8B, and 32B). As illustrated in Fig. 1, empirical results demonstrate that TSAPA significantly surpasses baseline methods, establishing a superior trade-off between attack success and text quality.

The primary highlights of this work are summarized as follows:

- TSAPA functions as a **blind**, **black-box adversarial framework** that formulates watermark removal as a **multi-objective optimization** problem. It simultaneously optimizes naturalness, semantic similarity, and lexical diversity through a guided mutation mechanism. By leveraging token-level pseudo-log-likelihood (PLL), the framework precisely isolates and perturbs tokens suspected of carrying watermarking signals.

- TSAPA exhibits **broad generalizability**, proving empirically effective across distinct watermarking schemes and model scales. Extensive evaluations confirm that it achieves **a superior trade-off between removal efficacy and text quality** relative to existing paraphrase-based watermark removal attacks.

2 Related work

LLM watermarking schemes. LLM watermarking embeds hidden signals into generated text to facilitate source attribution. Specifically, during the next-token sampling process, the vocabulary is pseudo-randomly partitioned, and the probabilities of a designated token subset are subtly amplified. This process implants a statistical pattern, imperceptible to human readers yet algorithmically detectable. Existing schemes are generally categorized into two categories, one is the the KGW Family (Kirchenbauer et al., 2023a; Liu et al., 2023b; Wu et al., 2023; Zhao et al., 2024; Lu et al., 2024), while the other is the so-called Christ Family (Aaronson and Kirchner, 2022; Kuditipudi et al., 2023; Christ et al., 2024). Details of the watermarking schemes referenced in this paper are provided in Appendix A.2.

Paraphrase-based watermark removal attacks. Research on adversarial strategies against text watermarking has primarily advanced along two distinct lines. The first line studies paraphrasing based removal under black box or limited feedback settings. The second line analyzes gray box or informed attackers that aim to reverse engineer or systematically disable a specific watermark family.

Paraphrasing represents a formidable threat to watermarking integrity. Moderate, semantics-preserving rewriting by systems such as DIPPER (Krishna et al., 2023) can significantly degrade detection accuracy. While provider-side retrieval-based defenses have been proposed as countermeasures, their efficacy is contingent on provider implementation. More advanced attacks exploit statistical vulnerabilities; for instance, SIRA (Cheng et al., 2025) achieves near-total watermark removal by selectively rewriting high-entropy tokens. Similarly, adaptive attackers have demonstrated the ability to train small open-source models to paraphrase against known watermarks (Diaa et al., 2024), underscoring the necessity for more robust defense mechanisms.

In the gray-box setting, access to token probabilities facilitates potent reverse-engineering attacks. Methods like De-mark (Chen et al., 2024) can fully reconstruct the internal logic of a watermark, enabling complete removal. Furthermore, ostensibly beneficial features like public verifiers can be subverted into attack oracles (Pang et al., 2024). By providing feedback signals, these endpoints can guide sophisticated removal and spoofing strategies, highlighting a fundamental trade-off between public accessibility and security.

Distinct from training-intensive or gray-box strategies that rely on model internals (Diaa et al., 2024; Chen et al., 2024), TSAPA operates as a strictly training-free, black-box framework. Unlike broad paraphrasing methods like DIPPER (Krishna et al., 2023) or SIRA (Cheng et al., 2025), we formulate removal as a constrained multi-objective optimization problem. By utilizing PLL-guided genetic algorithms to precisely target watermark-carrying tokens, TSAPA successfully navigates the Pareto front, achieving a superior balance between attack efficacy and semantic fidelity.

3 Problem formulation

This section formalizes watermark removal via paraphrasing under a strict black-box threat model.

We define the core watermarking components, characterize the adversary's constraints, and establish the formal criteria for a successful attack.

3.1 The Watermark System

An LLM watermark scheme, denoted as $\mathcal{W}$, is comprised of a pair of algorithms: a watermarked generator $\mathcal{G}_{\mathcal{W}}$ and a watermark detector $\mathcal{D}$. The definitions see in Appendix A.1.

3.2 Threat Model

We assume a strict **black-box** adversary whose only resource is a corpus $\mathcal{C} = \{x_{w,1}, x_{w,2}, \ldots, x_{w,n}\}$ of authentic watermarked texts sampled from $\mathcal{G}_{\mathcal{W}}$. The adversary **lacks access to**: (i) the scheme or other forms of $\mathcal{G}_{\mathcal{W}}$ used to generate watermarked text; (ii) the key $sk \in \mathcal{K}$ which is used for watermark generation; and (iii) the query access to $\mathcal{D}$ used for scoring text whether it is watermarked or not. Unlike gray-box settings, this prevents iterative refinement of the attack via oracle feedback.

Adversary's paraphrase. The adversary's sole capability is text manipulation via a paraphrasing model $\mathcal{A}$, which transforms watermarked text into $x^p \leftarrow \mathcal{A}(x^w)$. Under the adversary's full control, $\mathcal{A}$ can be optimized using the corpus $\mathcal{C}$ to learn and neutralize the statistical regularities inherent in the watermarked samples.

Definition 3.1 (Utility-Preserving Removal)**.** An paraphrase-based attack $\mathcal{A}$ is (δ_s, δ_q)**-successful** if the paraphrased text $x^p \leftarrow \mathcal{A}(x^w)$ satisfies:

1. Watermark removal: $\mathcal{D}(x^p, sk) < \delta_s$, where δ_s is the detection threshold.

2. Utility preservation: $|Q(x^p, x^w)| \geq \delta_q$, where Q quantifies semantic similarity relative to original watermarked text x^w.

Definition 3.2 (Watermark System Vulnerability)**.** A watermark system $\mathcal{W}$ is $(\epsilon, (\delta_s, \delta_q))$**-vulnerable** to black-box attacks if a feasible adversary $\mathcal{A}$ exists such that the probability of a (δ_s, δ_q)-successful attack is non-negligible:

$$\mathbb{P}_{p, x^w \leftarrow \mathcal{G}_{\mathcal{W}}(p, sk)} [Sg] \geq \epsilon, \tag{1}$$

where the detective signal is defined as $Sg := \mathcal{D}(\mathcal{A}(x^w), sk) < \delta_s \wedge Q(\mathcal{A}(x^w), x^w) \geq \delta_q$.

Our research investigates whether watermark systems exhibit this vulnerability and seeks to construct an $\mathcal{A}$ that empirically demonstrates it.

4 Our method

In this section, we detailed the formalization and implementation of hierarchical stress test based on text paraphrase and our TSAPA attack. We first present the process of stress test (Sec. 4.1), then we develop the details of the attack method (Sec. 4.2).

4.1 Hierarchical paraphrase stress test

We first establish a comprehensive benchmark of diverse text paraphrase attacks. This benchmark serves as a stress test, probing watermark vulnerability from simple base text edits to sophisticated LLM-based paraphrasing. Refer to Appendix B.1 for the omitted details.

Base edit methods. Simple base edits assess baseline resilience via **word-level deletions, insertions, and substitutions**. While potentially degrading text flow, these stochastic modifications measure a watermark's ability to survive localized tampering. Attack strength is tuned by the proportion of modified words.

LLM-based paraphrasing. According to different style paraphrase, we consider LLMs as text paraphrasers, to conduct **text polishing, expansion, summarization, and back-translation**, respectively. The four paraphrases differ from each other in their **length-adjustment strategies** and **transformation objectives**. Each attack is tuned via a 0.1–0.9 strength parameter within the prompt, allowing for fine-grained control over the rewrite, ranging from minor lexical edits to profound structural and semantic reconfiguration.

4.2 TSAPA attack scheme

We propose TSAPA, an iterative evolutionary framework that treats watermark removal as a discrete multi-objective optimization problem.

The overview. As shown in Fig. 2, the process starts with **population initialization**. In each generation, a **fitness function** evaluates every individual (paraphrased text) based on its ability to remove watermarks while preserving text quality. The algorithm then performs **selection**, choosing fitter individuals to produce the next generation via **semantic crossover** and **guide mutation** based on PLL socre. The overall algorithmic flow is summarize in Alg. 1. We model the **fitness function** maximization as the multi-objective optimization.

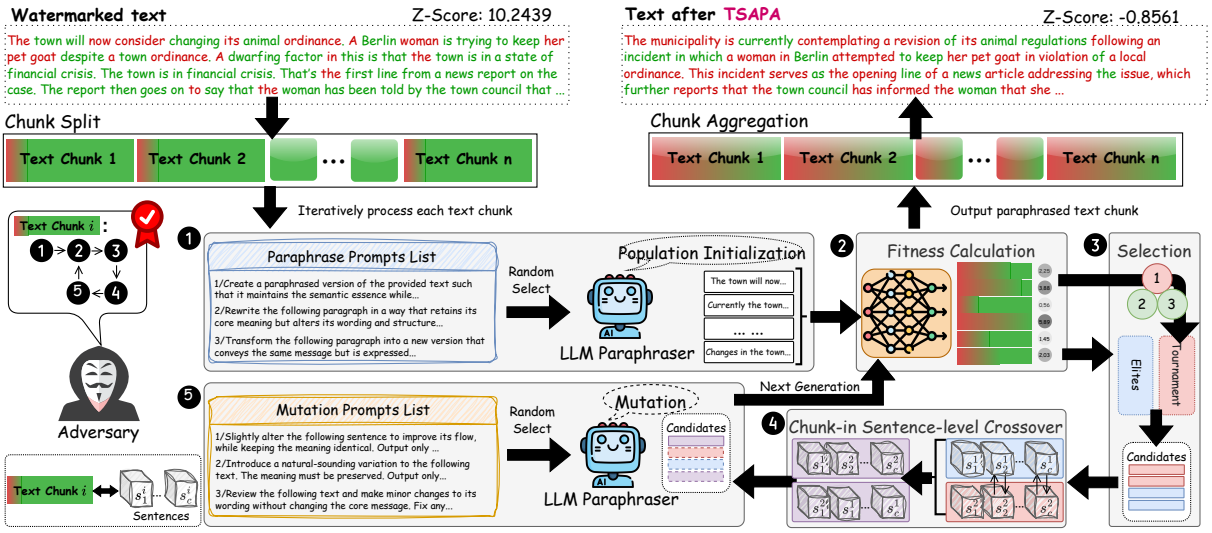


Figure 2: Overview of the workflow of TSAPA attack scheme. The watermark scheme takes KGW as an example.

Population initialization. The process begins with the generation of an initial population of candidate paraphrases from the original text (see Alg. 2). To ensure diversity, this population, denoted as $\mathcal{P}$, is generated using an LLM guided by a varied set of paraphrasing prompts (detailed in Appendix C.2). Throughout the evolution, the population size remains fixed at P.

Algorithm 1 Genetic algorithm for paraphrase-based

Require: A watermarked text x^w
Ensure: The best paraphrased text x^p

```
Split the text x^w into several chunks with approximately equal length
Initialize empty chunk set C ← ∅
for each chunk x^c in chunk set of x^w do
    Initialize population P from x^c   ▷ Alg. 2
    while termination criteria not met do
        Calculate fitness score for each individual in P   ▷ Alg. 3
        Select population according to fitness score   ▷ Alg. 4
        Apply genetic policy on P and gain a new population P ← P'   ▷ Alg. 5 & 6
    end while
    Append paraphrased x^{c_p} ∈ P with best fitness score into chunk set C ← C ∪ x^{c_p}
end for
Obtain paraphrased text x^p from chunk set C
return paraphrased text
```

Multi-objective optimization formulation. We here involve two primary function, that is Removal Efficacy f_{atk} and Semantic Fidelity f_{fid}. (See in Alg. 3)

The **first objective Removal Efficacy** serves as a proxy for the detection signal $\mathcal{D}(x^p, sk)$ under the black-box threat model (Sec. 3.2). Pseudo-log-likelihood score (PLL) (Salazar et al., 2019) is introduced here to quantify text naturalness, where the motivation is detailed in Appendix B.2. The branch of PLL score of a chunk of tokenized text x^c is given by

$$S_{PLL}(x^c) = \frac{\sum_{i=1}^{|x^c|} \log P_{MLM}(t_i | x^c_{\setminus i})}{|x^c|}, \quad (2)$$

where $P_{MLM}(t_i|x^c_{\setminus i})$ indicates that a token t_i is replaced by [MASK] and predicted using all in-context tokens $x^c_{\setminus i} = (t_1, \cdots, t_{i-1}, t_{i+1}, \cdots, t_{|x^c|})$ by a special masked language model (MLM).

The other is to evaluate the text structural diversity by calculating the n-gram frequency (normally $n = 3$), denoted as $S_{DIV_{str}}$,

$$S_{DIV_{str}}(x^c) = \frac{|G_n(x^c)|}{|N_n(x^c)|}, \quad (3)$$

where $G_n(x^c) = \{(w_i, w_{i+1}, \cdots, w_{i+n-1})\}$ is the set of all n-gram in the text x^c and $N_n(x^c)$ is total number of n-gram in the text x^c, i.e., $N_n(x^c) = |x^c| - n + 1$. Another part is the lexical divergence

$$S_{DIV_{lex}}(x^c) = 1 - \text{Self-BLEU}(x^c, x^w). \quad (4)$$

Hence, the final attack score is a weighted ensemble

$$f_{atk}(x^c) = w_1 \cdot \hat{S_{PLL}}(x^c) + w_2 \cdot S_{DIV_{str}}(x^c) + w_3 \cdot S_{DIV_{lex}}(x^c), \tag{5}$$

where $\hat{S_{PLL}}(x^c)$ is the min-max normalization of $S_{PLL}(x^c)$ and w_1, w_2, w_3 are hyperparameters.

The **second objective Semantic Fidelity** ensures that the resulting text satisfies the utility constraint $|Q(x^p, x^w)| \geq \delta_q$. We define $f_{fid}(x)$ via a non-monotonic utility function

$$f_{fid}(x) = \begin{cases} s(x) - \frac{3}{2}(\tau_l - s(x)) & s(x) < \tau_l \\ s(x) & \tau_l \leq s(x) \leq \tau_h \\ \tau_h - 2(s(x) - \tau_h) & s(x) > \tau_h \end{cases} \tag{6}$$

where $s(x) = \cos(\mathbf{E}(x), \mathbf{E}(x^w))$ denotes the cosine similarity between the latent embeddings of the candidate x and the source x^w. For alignment, we also denote $S_{SIM} = s(x)$.

For a watermarked text x^w, the adversary $\mathcal{A}$ seeks an optimal paraphrased x^p in search space $\mathcal{X}$ by maximizing a vector-valued objective function $\mathbf{F}(x^p) = [f_{atk}(x^p), f_{fid}(x^p)]^\top$:

$$\mathcal{A}(x^w) = \arg\max_{x^p \in \mathcal{X}} \mathbf{F}(x^p). \tag{7}$$

Selection with constraint handling. The search in $\mathcal{X}$ for the optimal attack follows a modified NSGA-II framework (Deb et al., 2002). We first rank the population $\mathcal{P}$ using a constraint-dominance relation $\prec_c$, which prioritizes linguistic validity $\mathcal{M}(x) \in \{0, 1\}$ alongside the objective vector $\mathbf{F}(x)$. Formally, for any two candidates $x_i, x_j \in \mathcal{P}$, we define the constraint-dominance relation $x_i \prec_c x_j$ if:

1. $\mathcal{M}(x_i) > \mathcal{M}(x_j)$;
2. $\mathcal{M}(x_i) = \mathcal{M}(x_j)$ and $\mathbf{F}(x_i)$ Pareto-dominates $\mathbf{F}(x_j)$;
3. $\mathcal{M}(x_i) = \mathcal{M}(x_j)$, $\mathbf{F}(x_i) = \mathbf{F}(x_j)$, and $d(x_i) > d(x_j)$.

where $d(x) = \sum_{k \in \{atk, fid\}} \frac{f_k(x_{next}) - f_k(x_{prev})}{f_k^{max} - f_k^{min}}$ is the *crowding distance* in the objective space to maintain population diversity. This sorting partitions $\mathcal{P}$ into a hierarchy of non-dominated fronts $\{R_1, R_2, \ldots, R_n\}$, ensuring that the search first converges to the feasible region before refining Pareto optimality.

Parent selection is performed via a tournament mechanism. The K best-performing individuals are retained as elites. The remaining $P - K$ slots are filled by repeatedly sampling T individuals from $\mathcal{P}$ at random; in each instance, the individual with the highest fitness score is chosen for the subsequent generation.

Semantic crossover via LLM-based synthesis. Given two parent candidates x^{p_1} and x^{p_2} selected from the candidate pool, the operator (Detailed in Alg. 5) first performs sentence-level multi-point crossover. The parent texts are first segmented into sentences. A predefined number of random crossover points are then chosen, and the sentence segments between these points are exchanged between the parents at the probability of crossover rate r_c, such that parent texts generate two offspring x_a and x_b. The operator then generates a synthesis of two offspring $x \leftarrow \text{LLM}(x_a, x_b)$, with appropriate prompts for semantic fusion (see in Appendix C.3).

Guided mutation via PLL-based identification. As presented in Alg. 6, we adopt LLM-based guided mutation via PLL-based identification. The process is executed in two stages:

- **Suspicion identification**: We utilize the previous MLM to compute the PLL for each token t_i in the candidate text x. We identify a set of high-risk tokens W_{sus} that exhibit the most pronounced statistical anomalies:

$$T_{sus} = \text{Top-k}\left(\{w_i \in x | \log P_{MLM}(t_i | x_{\setminus i})\}\right) \tag{8}$$

  where the Top-k selection targets tokens with the lowest relative ranks in the token dictionary.

- **Targeted remediation**: Rather than random replacement, we utilize T_{sus} as explicit negative constraints for the mutation process. Through specialized mutation guided prompt (see in Appendix C.4), the LLM is guided to restructure the text surrounding these suspicious regions, replacing flagged signatures with synonyms and altered syntax.

Upon the termination of the evolutionary process, the adversary identifies a single optimal attack x^* by locating the Knee Point of the primary front R_1. We select the individual that minimizes

the Euclidean distance to the ideal utopia point $\mathbf{1}$ in the normalized objective space

$$x^* = \arg\min_{x \in R_1} \sqrt{(1 - \bar{f}_{atk}(x))^2 + (1 - \bar{f}_{fid}(x))^2} \tag{9}$$

This knee point selection ensures the final attack achieves a robust balance between watermark removal and semantic fidelity, satisfying the (δ_s, δ_q)-success criteria.

5 Experiments

We present experiment setup in Sec. 5.1, the stress test and overview benchmarking in Sec. 5.2 and ablation experiments in Sec. 5.3.

5.1 Experiment setup

Tasks and datasets. Along with prior watermark work (Kirchenbauer et al., 2023a,b; Zhao et al., 2024) and paraphrased-based watermark removal attacks research (Cheng et al., 2025; Zhang et al., 2025), we focus on on article continuation task. The task benchmark is constructed based on MarkLLM (Pan et al., 2024), using the `C4` dataset (Raffel et al., 2020). Specifically, we randomly select 101 samples from the dataset to serve as prompts of such task, and using the corresponding natural non-watermarked texts as the ground truth samples.

Watermark schemes. We select KGW (Kirchenbauer et al., 2023a), EWD (Lu et al., 2024), DIP (Wu et al., 2023), EXP (Aaronson and Kirchner, 2022), SWEET (Lee et al., 2023), distortionary version of SynthID (Dathathri et al., 2024), Unbiased (Hu et al., 2023) and UPV (Liu et al., 2023a) as representative LLM watermark schemes. Hyperparameters follow the default configurations in MarkLLM (Pan et al., 2024).

Watermark models. We generate watermarked texts based on `Qwen3` Series (Yang et al., 2025), including `Qwen3-1.4B`, `Qwen3-8B` and `Qwen3-32B`.

Attack models. We implement paraphrase-based watermark removal attacks via open-sourced models and commercial LLM APIs, including `Llama-3.3-70b Instruct` (Grattafiori et al., 2024) and `gpt-4o-mini` (OpenAI, 2024).

Baseline paraphrase-based attacks. We compare our `TSAPA` scheme against various paraphrase-based watermark removal attacks, including word-level base edit method such as deletion, substitutions and insertion (He et al., 2024; Piet et al., 2025; Zhang et al., 2025), LLM-based paraphrasing (Liu et al., 2023b; Krishna et al., 2023), and SIRA (Cheng et al., 2025). For SIRA, we use RoBERTa (Liu et al., 2019) as the self-information calculator, and `Llama-3.3-70b Instruct` and `gpt-4o-mini` as paraphraser.

Hyperparapmeters setting of our method. For our method, the population size $P = 40$, the attack generation is dafault as 5, elite size $K = 15$, tournament group size $T = 3$ and crossover rate $r_c = 0.75$. For the evolutionary process, we set hyperparameters $w_1 = 0.6$, $w_2 = 0.2$ and $w_3 = 0.2$ for f_{atk}, and fix $\tau_h = 0.95$ and $\tau_l = 0.75$ for f_{fid}.

Evaluation metrics. We adopt three standard metrics from the LLM watermarking literature to evaluate the attacks. We define each as follows:

- **Attack Success Rate (ASR)**. we measure the **direct removal success**, defined as the percentage of originally watermarked samples that the detector classifies as non-watermarked post-attack. Higher ASR directly indicates the performance of attacks.
- **FPR@lowFPR**. we report the detector's **True Positive Rate (TPR)** at very low False Positive Rates (FPRs) of 1%. A potent attack should significantly reduce the TPR at these strict decision thresholds, which reflects the robustness of the attack.
- **Text Quality**. The perceptual and linguistic integrity of the text before and after attack, measured using metrics including **BERT score (BERTS)** (Zhang et al., 2019), **RougeL-F1 (ROUGEL)** (Lin, 2004), **BLEU** (Papineni et al., 2002) and **perplexity rate (PPL)**. For these metrics, higher BERTS, ROUGEL and indicate better content preservation and lexical fidelity with the original watermark text, while a lower PPL signifies greater fluency.

5.2 Experiment results

Hierarchical paraphrase stress test. We evaluate the vulnerability of LLM watermark via hierarchical paraphrase attacks, from base edits to LLM-based paraphrasing, by systematically increasing attack strength varied in $\{0.1, 0.2, 0.5, 0.8\}$. Fig. 3 illustrates a clear trade-off across all attack methods, as the attack strength increases, the ASR improves, but this gain is accompanied by a precipitous decline in BERTScore, indicating a severe

Table 1: Performance comparison between our proposed genetic-optimization framework and various baseline attacks (base Edits, LLM Paraphrasing, and SIRA) across four mainstream watermarking schemes (KGW, EXP, EWD, and UPV) using the `Qwen3-32B` model. The color intensity of the cells indicates performance, where green signifies the best performance and red signifies the worst for each respective metric.

Attack Method		KGW					EXP				
		ASR↑	BERTS↑	ROUGEL↑	BLEU↑	PPL↓	ASR↑	BERTS↑	ROUGEL↑	BLEU↑	PPL↓
Delete	10%	0.00%	0.8813	0.9482	78.29	10.10	4.95%	0.8909	0.9479	78.27	6.37
	20%	0.99%	0.8203	0.8892	58.57	18.24	20.79%	0.8336	0.8892	58.73	11.54
	50%	15.84%	0.6649	0.6669	15.51	96.33	96.04%	0.6856	0.6668	16.01	65.25
	80%	97.03%	0.5570	0.3352	0.32	429.30	100.00%	0.5624	0.3346	0.32	328.88
Substitute	10%	0.00%	0.9072	0.9033	79.38	17.83	4.95%	0.9118	0.9037	79.66	11.46
	20%	0.00%	0.8288	0.8063	61.80	46.05	25.74%	0.8361	0.8065	62.40	29.89
	50%	29.70%	0.6595	0.5244	25.46	189.45	98.02%	0.6689	0.5283	25.67	142.39
	80%	92.08%	0.5733	0.2778	7.37	237.57	100.00%	0.5855	0.2918	7.30	174.26
LLM Paraphrase	Polish	22.77%	0.8483	0.6852	43.45	5.56	79.21%	0.8418	0.6515	41.69	5.45
	Expand	51.49%	0.7127	0.5227	29.30	4.67	82.18%	0.7073	0.4709	28.07	4.18
	Summarize	100.00%	0.6283	0.1899	0.77	11.11	99.01%	0.6251	0.1852	0.62	11.80
	Translate	99.01%	0.3454	0.4654	28.91	6.76	100.00%	0.3622	0.4315	26.89	5.78
SIRA	gpt-4o-mini	100.00%	0.6539	0.2614	6.43	6.67	99.01%	0.6631	0.2655	6.52	6.04
	Llama3.3-70b	100.00%	0.4827	0.1196	0.61	8.06	100.00%	0.5842	0.2061	3.46	8.32
Ours	gpt-4o-mini	92.08%	0.7629	0.5249	18.32	10.81	99.01%	0.7698	0.5204	18.73	9.38
	Llama3.3-70b	93.00%	0.7344	0.4349	21.90	8.41	100.00%	0.7187	0.4032	18.61	7.85

Attack Method		EWD					UPV				
		ASR↑	BERTS↑	ROUGEL↑	BLEU↑	PPL↓	ASR↑	BERTS↑	ROUGEL↑	BLEU↑	PPL↓
Delete	10%	0.99%	0.8819	0.9479	78.38	8.42	51.49%	0.8821	0.9479	78.72	8.16
	20%	1.98%	0.8233	0.8896	58.64	15.30	56.44%	0.8236	0.8896	59.45	14.59
	50%	63.37%	0.6661	0.6677	15.78	82.90	79.21%	0.6736	0.6674	16.48	77.85
	80%	100.00%	0.5559	0.3342	0.35	414.21	79.21%	0.5664	0.3353	0.41	372.44
Substitute	10%	1.98%	0.9061	0.9030	79.42	15.09	56.44%	0.9096	0.9040	79.98	14.32
	20%	2.97%	0.8303	0.8065	8.75	39.00	65.35%	0.8335	0.8067	62.92	36.80
	50%	91.09%	0.6579	0.5238	25.41	168.94	86.14%	0.6632	0.5232	26.93	158.21
	80%	100.00%	0.5747	0.2807	7.71	211.35	87.13%	0.5847	0.2825	9.15	198.95
LLM Paraphrase	Polish	48.51%	0.8495	0.6937	45.69	5.00	95.05%	0.8303	0.6450	41.74	5.41
	Expand	68.32%	0.7244	0.5614	33.12	4.34	100.00%	0.7039	0.4920	29.61	4.21
	Summarize	100.00%	0.6339	0.1975	0.90	10.23	100.00%	0.6150	0.1830	0.69	11.87
	Translate	100.00%	0.3421	0.4641	31.50	5.99	93.07%	0.3621	0.4716	29.75	5.46
SIRA	gpt-4o-mini	99.01%	0.6616	0.2711	7.01	7.09	100.00%	0.6434	0.2479	5.75	8.17
	Llama3.3-70b	100.00%	0.4907	0.1315	0.66	7.57	100.00%	0.4788	0.1136	0.60	7.72
Ours	gpt-4o-mini	97.03%	0.7741	0.5375	20.35	9.37	100.00%	0.7589	0.5055	18.52	10.15
	Llama3.3-70b	94.74%	0.7418	0.4414	23.99	8.05	100.00%	0.7304	0.4276	22.97	7.94

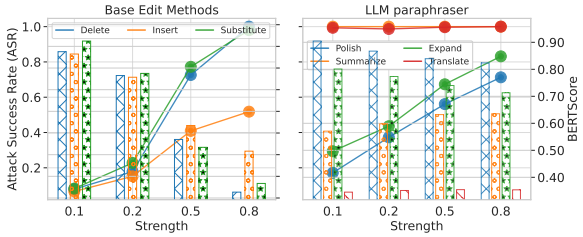


Figure 3: Hierarchical paraphrase stress test of baseline attack methods across varying attack strength. Lines represent the Attack Success Rate (ASR) and bars represent the semantic fidelity measured by BERTScore for both base edits and LLM paraphrase.

loss of text quality. This demonstrates the limitations of non-optimized attack strategies, which fail to maintain high text quality when attempting to achieve high erasure rates, thereby motivating our multi-objective genetic approach to find a more balanced Pareto frontier.

Benchmark against other methods. We benchmark our method against other methods introduced in 5.1 on watermark model `Qwen3-32B` with KGW, EXP, EWD and UPV schemes, respectively. We evaluate attack efficacy and paraphrased text quality simultaneously. The results in Tab. 1 reveal that while high-strength base edits and SIRA can achieve near-perfect ASR, they suffer from a

Table 2: ASR of TSAPA against various watermark schemes on Qwen3 models of varying scales (1.7B, 8B, and 32B), using gpt-4o-mini as the base paraphraser.

Watermark scheme	Watermark models		
	Qwen3-1.7B	Qwen3-8B	Qwen3-32B
DIP	100.00%	99.01%	100.00%
EWD	100.00%	100.00%	97.03%
EXP	100.00%	97.03%	99.01%
KGW	93.07%	95.05%	92.08%
SWEET	100.00%	100.00%	99.01%
SynthID	100.00%	100.00%	100.00%
Unbiased	100.00%	100.00%	100.00%
UPV	100.00%	99.01%	100.00%

catastrophic drop in text quality. In contrast, our method consistently achieves the most robust trade-off across all scenarios, maintaining a high ASR (above 92%) while outperforming others in preserving the original text's meaning and naturalness.

5.3 Ablation study

The impact of the size of watermark model. We evaluate robustness of TSAPA by attacking eight watermarking schemes using Qwen3 (1.7B, 8B and 32B) as source generators. Tab. 2 shows TSAPA consistently achieves near-perfect ASR (over 97%, often 100%) across all configurations. These results demonstrate that TSAPA is invariant to model capacity, suggesting that increasing parameter size of watermark model does not inherently strengthen defense against paraphrase-based attack.

How the attack behavior changes with the growing attack generation? We conducted an ablation study to evaluate the impact of the optimization process by varying the number of attack generations from 1 to 9, while maintaining the population size and fitness scaling factors consistent with our core genetic algorithm configuration. As illustrated in Fig. 4, the Attack Success Rate (ASR) exhibits a clear upward trajectory toward saturation near 100.00%, while TPR@1%FPR decreases significantly from approximately 0.47 to 0.10. These results demonstrate that our iterative multi-objective optimization successfully removes the statistical watermark signals over time, progressively identifying adversarial candidates that effectively circumvent the detector while maximizing attack efficacy.

How does optimization facilitate the attack efficacy-text quality tradeoff? We conduct an ablation study to evaluate the contribution of each fitness component by comparing our multi-objective framework against three single-objective variants. These were assess using the PLL score S_{PLL} (Eq. 2), semantic similarity S_{sim} (from Eq. 6) and lexical diversity S_{DIV} (Eq. 3 and 4), with gpt-4o-mini attacking the KGW-watermarked Qwen3-32B system. As shown in Tab. 3, single-objective optimizations yield polarized results: optimizing S_{SIM} lone preserves high fidelity but fails the attack, while optimizing S_{DIV} achieves perfect ASR at the cost of catastrophic linguistic collapse. In contrast, our full method identifies the optimal Pareto region, securing a robust ASR of 93.07% while maintaining superior text quality (BERTS is 0.7629 and PPL is 7.03).

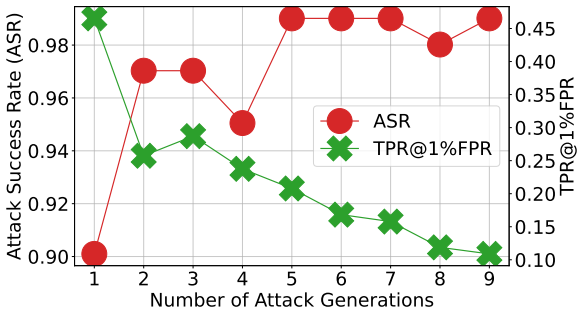


Figure 4: Watermark removal efficacy (ASR and TPR with FPR set to 1%) across optimization generations. Watermark scheme is KGW, while watermark model is Qwen3-32B and attack model is gpt-4o-mini.

Table 3: Ablation study of multi-objectives in the genetic optimization of the TSAPA (attack model: gpt-4o-mini; watermark model: Qwen3-32B). By comparison, the best results are highlighted in **boldface** while the worst are distinguished in **red**.

Method	ASR ↑	BERTS ↑	ROUGEL ↑	PPL ↓
S_{PLL} Only	98.02%	0.7241	0.3611	87.19
S_{SIM} Only	**2.97%**	**1.0000**	**1.0000**	16.15
S_{DIV} Only	**100.00%**	**0.6512**	**0.2334**	**537.26**
Ours (Full)	93.07%	0.7629	0.5249	**7.03**

6 Conclusion

This paper presents a systematic study of paraphrase-based attacks against LLM text watermarking and introduces TSAPA, an evolutionary attack framework. By formulating watermark removal as a multi-objective optimization process, TSAPA achieves a superior trade-off between removal efficacy and paraphrased text quality.

Limitation

Genetic algorithms are notoriously iterative. The evolutionary process involves a high volume of LLM calls. Despite this, TSAPA remains a training-free framework, avoiding the significant computational overhead and data requirements associated with training adaptive attack models. However, the inference latency inherent in the multi-generational evolution renders our method more suitable for offline document processing rather than real-time streaming applications.

Additionally, while our chunk-based processing enables scalability, it optimizes text segments in isolation. This localization may occasionally compromise global discourse coherence or cross-chunk logical flow, as the optimization objectives are calculated without long-range context.

Finally, the attack's upper bound is intrinsically linked to the capability of the adversary's paraphraser; utilizing a weaker LLM than the source model may limit the generation of high-quality candidates that satisfy strict semantic fidelity constraints.

Acknowledgments

We used language models only for polishing English writing and vibe coding, while all ideas, figures, main algorithms, experiments, results and interpretations are our own.

References

Scott Aaronson and H. Kirchner. 2022. Watermarking GPT outputs. https://www.scottaaronson.com/talks/watermark.ppt. PowerPoint presentation.

Ruibo Chen, Yihan Wu, Junfeng Guo, and Heng Huang. 2024. De-mark: Watermark removal in large language models. *arXiv preprint arXiv:2410.13808*.

Yixin Cheng, Hongcheng Guo, Yangming Li, and Leonid Sigal. 2025. Revealing weaknesses in text watermarking through self-information rewrite attacks. In *Forty-second International Conference on Machine Learning (ICML'25)*.

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

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.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

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

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

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 3558–3567.

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*.

Zhiwei He, Binglin Zhou, Hongkun Hao, Aiwei Liu, Xing Wang, Zhaopeng Tu, Zhuosheng Zhang, and Rui Wang. 2024. Can watermarks survive translation? on the cross-lingual consistency of text watermark for large language models. *arXiv preprint arXiv:2402.14007*.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2023. Unbiased watermark for large language models. *arXiv preprint arXiv:2310.10669*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A watermark for large language models. In *International Conference on Machine Learning (ICML'23)*, pages 17061–17084.

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

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems (NeurIPS'23)*, 36:27469–27500.

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

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

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Aiwei Liu, Leyi Pan, Xuming Hu, Shu'ang Li, Lijie Wen, Irwin King, and Philip S Yu. 2023a. An unforgeable publicly verifiable watermark for large language models. *arXiv preprint arXiv:2307.16230*.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2023b. A semantic invariant robust watermark for large language models. *arXiv preprint arXiv:2310.06356*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

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.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2024-07-18.

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

Qi Pang, Shengyuan Hu, Wenting Zheng, and Virginia Smith. 2024. Attacking llm watermarks by exploiting their strengths. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

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

Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. 2025. Markmywords: Analyzing and evaluating language model watermarks. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML'25)*, pages 68–91.

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

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

Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. 2019. Masked language model scoring. *arXiv preprint arXiv:1910.14659*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. 2023. A resilient and accessible distribution-preserving watermark for large language models. *arXiv preprint arXiv:2310.07710*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zhaoxi Zhang, Xiaomei Zhang, Yanjun Zhang, He Zhang, Shirui Pan, Bo Liu, Asif Qumer Gill, and Leo Yu Zhang. 2025. Character-level perturbations disrupt llm watermarks. *arXiv preprint arXiv:2509.09112*.

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

A Preliminaries

A.1 LLM watermark scheme

Watermarked Generator ($\mathcal{G}_{\mathcal{W}}$). The generator is an LLM whose token distribution is subtly modified during decoding to embed a statistical signal. Given a prompt p and a secret key $sk \in \mathcal{K}$ (where $\mathcal{K}$ is the key space), $\mathcal{G}_{\mathcal{W}}$ produces a watermarked text output x^w. This stochastic process is denoted as:

$$x^w \leftarrow \mathcal{G}_{\mathcal{W}}(p, sk). \tag{10}$$

The underlying embedding mechanism $\mathcal{E}$, which is parameterized by sk, is responsible for manipulating the logit scores of the vocabulary at each generation step to favor the selection of certain tokens (e.g., a "green list") over others (e.g., a "red list").

Watermark Detector ($\mathcal{D}$). The detector is a statistical test designed to identify the signal embedded by $\mathcal{G}_{\mathcal{W}}$. Given a candidate text x' and the secret key sk, the detector $\mathcal{D}$ computes a test statistic, typically a z-score or a p-value, to determine the likelihood that x' was generated by $\mathcal{G}_{\mathcal{W}}$ using key sk. For analytical clarity, we model the detector as returning a binary decision based on a predefined significance level α:

$$\mathcal{D}(x', sk) \rightarrow \{1, 0\}, \tag{11}$$

where 1 signifies that the watermark is detected (i.e., the null hypothesis of the text being unwatermarked is rejected) and 0 signifies its absence.

A.2 Scheme implementations of LLM watermark

We evaluate the paraphrase-based watermark removal attack on various watermark schemes, including KGW (Kirchenbauer et al., 2023a), EWD (Lu et al., 2024), DIP (Wu et al., 2023), EXP (Aaronson and Kirchner, 2022), SWEET (Lee et al., 2023), distortionary version of SynthID (Dathathri et al., 2024), Unbiased (Hu et al., 2023) and UPV (Liu et al., 2023a).

- KGW (Kirchenbauer et al., 2023a): This scheme partitions the vocabulary into green and red lists based on the hash of preceding tokens, biasing the sampling process to favor green tokens. It effectively embeds a statistical signal without drastically altering the output distribution. The hyperparameters are set to $\gamma = 0.5$ and $\delta = 3.0$, with a prefix length of $h = 1$.

- EWD (Lu et al., 2024): It adapts the standard logit-biasing approach by incorporating entropy-based weighting during the detection phase. This strategy assigns higher significance to tokens generated in high-entropy contexts to enhance robustness against removal attacks. We configure the parameters as $\gamma = 0.5$, $\delta = 2.0$, and $h = 1$.

- DIP (Wu et al., 2023): Instead of directly shifting logits, this scheme employs a distribution-preserving reweighting function to adjust token probabilities. This ensures the watermarked text maintains the original language model's statistical properties while still embedding a detectable signature. The settings are $\gamma = 0.5$, $\alpha = 0.45$, and a prefix length of $h = 5$.

- EXP (Aaronson and Kirchner, 2022): This scheme utilizes a sampling trick based on the Gumbel distribution to select tokens that maximize a specific score derived from a secret key. It allows for perfect sampling where the watermarked distribution matches the original model distribution theoretically. We use a prefix length of $h = 4$ and a sequence length of 500.

- SWEET (Lee et al., 2023): SWEET selectively applies the watermark bias only when the entropy of the probability distribution exceeds a specific threshold. This selective application preserves the quality of text in low-entropy scenarios, such as code generation or factual statements. The parameters are $\gamma = 0.5$, $\delta = 2.0$, $h = 1$, and an entropy threshold of 0.9.

- SynthID (Dathathri et al., 2024): This scalable watermarking solution uses a tournament-based sampling mechanism or specialized scoring functions to embed information into the generated content. It balances generation quality and detection robustness through configurable n-gram dependencies. The configuration uses an n-gram length of 5 and a context history size of 1024.

- Unbiased (Hu et al., 2023): This method applies a reweighting mechanism designed to en-

Table 4: Benchmarking Methods for Paraphrase-based Attacks

Paraphrase-based Attack	Description
Base Edit Methods	
Insertion	At the word level, $X\%$ existing words of the total word count are randomly inserted back into the text at a random position. The attack strengths ($X\%$) are set to: 10%, 20%, 50%, and 80%.
Deletion	At the word level, $X\%$ of words are randomly removed from the input text. The attack strengths ($X\%$) are set to: 10%, 20%, 50%, and 80%.
Substitution	At the word level, $X\%$ of words are replaced with other randomly chosen existing words from the text. The attack strengths ($X\%$) are set to: 10%, 20%, 50%, and 80%.
LLM Paraphraser	
Polishing	Prompt the LLM to re-paraphrase the text, and maintain text length before and after polishing.
Summarizing	Prompt the LLM to condense the text, reducing its length.
Expanding	Prompt the LLM to elaborate on the original content, adding details and increasing text length.
(One-round-back) Translation	The text is translated from English to an intermediate language and then back to English using the LLM. In this work, the workflow is fixed as `{EN->ZH->EN}`.

sure that the expectation of the watermarked output remains mathematically identical to the unwatermarked baseline. This "unbiased" property prevents the watermark from introducing unwanted statistical drift. We utilize the gamma type with $n\_grid = 10$ and a prefix length of $h = 5$.

- UPV (Liu et al., 2023a): UPV employs a model-based framework that utilizes a trained generator network to partition tokens and a detector network for watermark identification. Unlike heuristic methods, it relies on these neural components to create a binary prediction for detection. The hyperparameters are set to $\gamma = 0.5$, $\delta = 2.0$, and $h = 1$.

The implementation and hyperparameters setting are based on MarkLLM (Pan et al., 2024).

B Additional details of our method

B.1 Hierarchical stress test based on text paraphrase

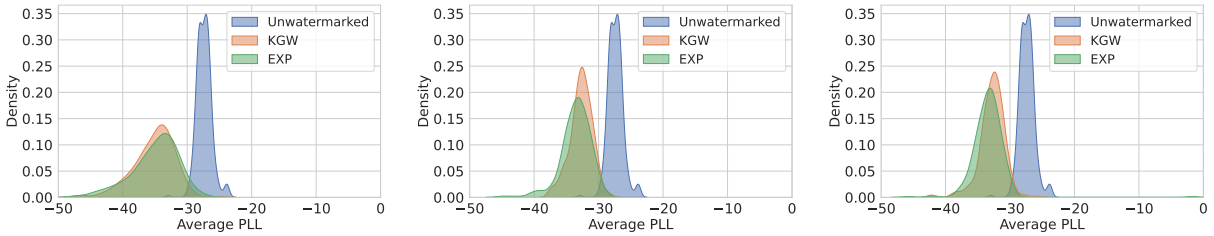
The benchmarking stress test methods are summarized in Tab. 4.

Base edits methods. Base edits represent the most fundamental type of attack, testing the basic resilience of the watermark through direct, often random, modifications to the text. While these attacks may not always preserve the full semantic integrity of the original text, they are highly effective for assessing whether the watermark signal is sufficiently distributed and can withstand localized tampering. Our implementation supports granular control over the edit level (especially world-level) and attack strength (percentage of elements affected).

LLM-based paraphrase. These attacks leverage powerful LLMs to rewrite the text, aiming to preserve core semantics while significantly altering surface-level lexical and syntactic structures, thereby removing the original statistical watermark. The prompts used for LLM paraphrasers can be seen in Sec. C.1.

B.2 The motivation of S_{PLL}

The mechanism of PLL scoring As illustrated in Fig. 6, for a chunk of tokenized text x^c, a token t_i is replaced by `[MASK]` and predicted using all in-context tokens $x^c_{\setminus i} =$



(a) Qwen3-1.7B (b) Qwen3-8B (c) Qwen3-32B

Figure 5: **Average pseudo-log-likelihood score (PLL)** of unwatermarked natural texts and watermarked LLM-generated texts by the KGW and EXP schemes on article continuation task. The MLM is RoBERTa-Large. The watermark model is Qwen3-1.7B, Qwen3-8B and Qwen3-32B, respectively.

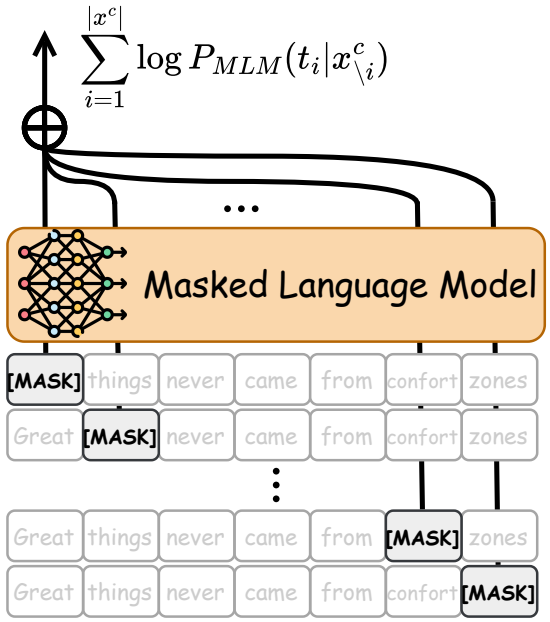


Figure 6: An example on PLL scoring of the text using MLM.

$(t_1, \cdots, t_{i-1}, t_{i+1}, \cdots, t_{|x^c|})$ by a typical masked language model (MLM), such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). Hence, the PLL score is defined as

$$\mathbf{PLL}(x^c) := \sum_{i=1}^{|x^c|} \log P_{MLM}(t_i|x^c_{\setminus i}). \quad (12)$$

Since the lengths of the texts are not the consistently same, the PLL score represented as the chunk of the text is averaged per token, i.e.,

$$S_{PLL}(x^c) = \mathbf{PLL}(x^c)/|x^c|. \quad (13)$$

Intuitively, a high logit value (i.e., ranking near the top of the vocabulary distribution) suggests that the token is natural and contextually coherent. Conversely, a lower logit may indicate artificial manipulation, such as watermark-induced bias.

Distribution difference between watermarked texts and natural non-watermarked text on PLL score. Fig. 5 illustrates the density distributions of the average PLL scores for texts generated by Qwen3 models of varying scales (1.7B, 8B, and 32B). We compare unwatermarked natural texts against those watermarked texts with KGW and EXP schemes. The distributions are virtually non-overlapping, with watermarked texts consistently exhibiting significantly lower PLL scores. This pronounced gap demonstrates that the statistical artifacts introduced by watermarking are readily detectable by a general MLM, confirming that PLL can be used to build a simple yet robust proxy watermark detector.

B.3 Algorithmic description of TSAPA scheme

The process of population initialization is described as Alg. 2.

Algorithm 2 Population Initialization

Require: Watermarked text x^w, size P, LLM M, Prompt set $\mathcal{S}_{pmt}$
Ensure: Population $\mathcal{P} = \{x_k^p\}_{k=1}^P$

```
P ← ∅
for k ← 1 to P do
    p ← RandomSample(S_pmt)        ▷ Ensure diversity via varied prompts
    P ← P ∪ {M(x^w, p)}
end for
return P
```

The multi-objective fitness calculation (Alg. 3), the selection with constraint handling (Alg. 4), the semantic crossover via LLM-based synthesis (Alg. 5), and the guided mutation via PLL-based identification (Alg. 6) are described respectively as follows.

Algorithm 3 Calculate Multi-Objective Fitness

Require: Candidates x^c, x^w, Models $M_{MLM}, \mathbf{E}$, Weights $\mathbf{w}$, Thresholds τ_l, τ_h
Ensure: Fitness vector $\mathbf{F} = [f_{atk}, f_{fid}]^\top$

Step 1: Removal Efficacy (f_{atk})
$\hat{S}_{PLL} \leftarrow \text{Normalize}(S_{PLL}(x^c))$ ▷ PLL via Eq. 2 with Min-Max normalization
$S_{DIV_{str}} \leftarrow |G_n(x^c)|/|N_n(x^c)|$ ▷ Structural diversity, Eq. 3
$S_{DIV_{lex}} \leftarrow 1 - \text{Self-BLEU}(x^c, x^w)$ ▷ Lexical divergence, Eq. 4
$f_{atk} \leftarrow w_1 \cdot \hat{S}_{PLL} + w_2 \cdot S_{DIV_{str}} + w_3 \cdot S_{DIV_{lex}}$
Step 2: Semantic Fidelity (f_{fid})
$s \leftarrow \cos(\mathbf{E}(x^c), \mathbf{E}(x^w))$ ▷ Cosine similarity of embeddings
if $s < \tau_l$ **then**
    $f_{fid} \leftarrow s - 1.5(\tau_l - s)$ ▷ Penalty for low similarity
else if $s > \tau_h$ **then**
    $f_{fid} \leftarrow \tau_h - 2(s - \tau_h)$ ▷ Penalty for over-similarity
else
    $f_{fid} \leftarrow s$ ▷ Utility within $[\tau_l, \tau_h]$
end if
return $\mathbf{F} = [f_{atk}, f_{fid}]^\top$

Algorithm 4 Selection with Constraint Handling

Require: Population $\mathcal{P}$, Objectives $\mathbf{F}$, Constraints $\mathcal{M}$, Hyperparameters P, K, T
Ensure: Next Generation $\mathcal{P}'$

Step 1: Constraint-dominance Ranking
Define $x_i \prec_c x_j$ if: (1) $\mathcal{M}(x_i) > \mathcal{M}(x_j)$;
(2) $\mathcal{M}(x_i) = \mathcal{M}(x_j)$ and $\mathbf{F}(x_i)$ Pareto-dominates $\mathbf{F}(x_j)$;
(3) Equal $\mathcal{M}, \mathbf{F}$ and $d(x_i) > d(x_j)$ ▷ $d(x)$ is crowding distance
Partition $\mathcal{P}$ into fronts $\{R_1, R_2, \ldots, R_n\}$ based on $\prec_c$
Step 2: Elitism & Tournament
$\mathcal{P}' \leftarrow$ Top K individuals from $\{R_1, \ldots, R_n\}$ ▷ Retain elites
while $|\mathcal{P}'| < P$ **do**
    $\mathcal{T} \leftarrow \text{RandomSample}(\mathcal{P}, T)$ ▷ Tournament of size T
    $x_{best} \leftarrow$ Individual in $\mathcal{T}$ with highest rank by $\prec_c$
    $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{x_{best}\}$
end while
return $\mathcal{P}'$

Algorithm 5 Semantic Crossover via LLM-based Synthesis

Require: Parent candidates x^{p_1}, x^{p_2}, Crossover rate r_c, LLM M_{syn}, Fusion prompt p_{fus}
Ensure: Offspring x_{new}

if $\text{Random}(0, 1) \geq r_c$ **then**
    return x^{p_1} ▷ Skip crossover
end if
Step 1: Structural Multi-point Crossover
$S_1, \leftarrow \text{SentenceTokenize}(x^{p_1})$
$S_2 \leftarrow \text{SentenceTokenize}(x^{p_2})$
Select random cut points $\mathcal{I}$ based on $\min(|S_1|, |S_2|)$
$x_a, x_b \leftarrow$ Generate two intermediates by exchanging sentence segments of S_1, S_2 at points $\mathcal{I}$
Step 2: Semantic Synthesis
$x_{new} \leftarrow \text{LLM}(x_a, x_b, p_{fus})$ ▷ Synthesize coherent offspring via LLM
return x_{new}

Algorithm 6 Guided Mutation via PLL-based Identification

Require: Candidate set $\mathcal{P}$, Mutation rate r_m, MLM, LLM, Prompt p_{mut}
Ensure: Mutated population $\mathcal{P}_{next}$

$\mathcal{P}_{next} \leftarrow \emptyset$
for each candidate x in $\mathcal{P}$ **do**
    if $\text{Random}(0, 1) < r_m$ **then**
        Step 1: Suspicion Identification
        Compute token-level PLL: $S_{PLL} \leftarrow \{\log P_{MLM}(t_i | x_{\setminus i}) \mid \forall t_i \in x\}$
        $T_{sus} \leftarrow$ Top-k(x) based on lowest relative ranks in S_{PLL}
        Step 2: Targeted Remediation
        ▷ Guide LLM to restructure x using T_{sus} as negative constraints
        $x' \leftarrow \text{LLM}(x, T_{sus}, p_{mut})$
        $\mathcal{P}_{next} \leftarrow \mathcal{P}_{next} \cup \{x'\}$
    else
        $\mathcal{P}_{next} \leftarrow \mathcal{P}_{next} \cup \{x\}$
    end if
end for
return $\mathcal{P}_{next}$

C Prompts

This section details the prompts employed throughout the attack scheme: those for the LLM-based paraphrase stress test (Sec. C.1), for population initialization (Sec. C.2), and for the semantic crossover and guided mutation operations (Sec. 5 and Sec. C.4).

C.1 Prompts for LLM-based paraphrase

LLM-based paraphrase Prompt Collections

Prompt for Text Polishing

You are a professional writing assistant. Your task is to polish the following text according to a specified "rewriting_strength". The "rewriting_strength" is a scale from 0.1 to 0.9, dictating the extent of your changes.

- **0.1-0.3 (Proofreading):** Minimal changes. Primarily corrects grammar, spelling, punctuation, and obvious typos. The original sentence structure and wording are preserved as much as possible.
- **0.4-0.6 (Standard Polish):** Moderate rewriting. Improves sentence flow, clarity, and word choice by using better synonyms and rephrasing some sentences. The core meaning and style are maintained.
- **0.7-0.9 (Deep Polish):** Substantial rewriting. Significantly enhances the text's style, tone, and impact. This involves restructuring sentences, varying sentence length, and employing more sophisticated or persuasive language, while strictly preserving the original's core message.

Please polish the following text with a rewriting strength of {strength_level}.
You are required to keep the length of the text generally consistent. Do not add any new information or your own opinions. Output only the polished text.
— *Original Text: {text_input}*

Prompt for Text Summarization

You are an expert summarization assistant. Your task is to summarize the following text based on a specified "detail_level".
The "detail_level" is a scale from 0.1 (most abstract) to 0.9 (most detailed).

- **0.1-0.3 (High-Level Abstract):** Highly abstractive. Condenses the text to its absolute core thesis, main finding, or final conclusion. Ideal for a title or one-sentence summary.
- **0.4-0.6 (Key Points Summary):** A balanced approach. Extracts the main arguments and key supporting points while omitting secondary details, anecdotes, and most examples. This provides a standard, functional summary.
- **0.7-0.9 (Detailed Summary):** More extractive and comprehensive. Includes the main ideas, key arguments, and also incorporates crucial supporting details, essential examples, or necessary context to give the reader a fuller understanding.

Please summarize the following text with a detail level of {detailed_level}.
Do not add any new information or your own opinions. The final output must only be the summarized text and its token length must not exceed {MAX_LENGTH}.
— *Original Text: {text_input}*

Prompt for Text Expansion

You are a creative writing assistant. Your task is to expand the following text based on a specified "expansion_strength".
The "expansion_strength" is a scale from 0.1 (minor addition) to 0.9 (major development).

- **0.1-0.3 (Minor Elaboration):** Minimal expansion. Adds descriptive adjectives, adverbs, or short clarifying phrases to the existing sentences to make them slightly more vivid or precise without changing the structure.
- **0.4-0.6 (Moderate Development):** Adds new, complete sentences that provide context, examples, or reasons to

support the original statements. This fleshes out the existing ideas.

- **0.7-0.9 (Substantial Development):** Major expansion. Builds upon the original text by developing its ideas into full paragraphs. This may involve introducing and explaining related subtopics, providing detailed evidence, or building a more comprehensive argument based on the initial premise.

Please expand the following text with an expansion strength of {strength_level}.
Do not add new information or opinions that are fundamentally unrelated to the original text. Output only the expanded text. The final text's token length must be at least {MIN_LENGTH}.
— *Original Text: {text_input}*

Prompt for Text Translation
You are a highly skilled translation assistant, capable of adjusting the tone and formality of your translations. Your task is to translate the following text into {target_language} according to a specified "formality_level". The "formality_level" is a scale from 0.1 (most informal) to 0.9 (most formal). Please adhere to this guide:

- **0.1-0.3 (Informal):** For social media posts, personal blogs, chat messages, or marketing copy targeting a young audience. Use casual language, common idioms, and simpler sentence structures. Contractions and colloquialisms are acceptable if appropriate for the {target_language}.
- **0.4-0.6 (Neutral/Standard):** For news articles, reports, product descriptions, and general professional communication. Use clear, direct, and standard language that is widely accessible. This is the default tone for most content.
- **0.7-0.9 (High Formal):** For legal documents, academic papers, official announcements. Use precise, technical vocabulary, complex sentence structures, and avoid any colloquialisms or contractions. This corresponds to a very formal, official tone.

Please translate the following text with a formality level of {formality_level}.
Do not add any new information or your own opinions. Output only the translated text.
— *Original Text: {text_input}*

C.2 Prompts for population initiation

Prompts Collection $\mathcal{S}_{pmt}$ for Population Initiation

Prompt #1
You are an English paraphrasing tool. Your task is to rewrite the following English text. The output must be in English and must preserve the original meaning perfectly. Output only the rewritten text and nothing else. Here is the text:

Prompt #2
Rewrite the following English text on a line-by-line basis. Your response must be in English and should maintain the original line structure as much as possible. Ensure the core meaning of each line is preserved. Output only the paraphrased text. Here is the text:

Prompt #3
Act as a professional English copy-editor. Your objective is to rewrite the provided text to enhance its clarity and flow while preserving all original information and its approximate length. Your response must be entirely in English. Do not add any new information or personal opinions. Output only the final, edited text. Here is the text:

Prompt #4
Your task is to rephrase the following English text. The goal is to minimize n-gram overlap with the original text while ensuring the meaning is perfectly preserved. Use diverse vocabulary and sentence structures. The output must be in English and coherent. Output only the resulting text. Here is the text:

Prompt #5
Transform the following English text into a simpler and more accessible version. The core message must remain unchanged, but the language should be easier to understand. The output must be in English. Output only the simplified text. Here is the text:

Prompt #6
Rewrite the following English text to adopt a more formal and professional tone. All key information must be retained. The output must be in English. Avoid colloquialisms and use precise terminology. Output only the paraphrased text. Here is the text:

Prompt #7
Paraphrase the following English text by significantly altering its sentence structure. You can combine short sentences or break down long ones, but the core meaning must be fully preserved. Your response must be in English. Output only the structurally altered text. Here is the text:

Prompt #8
You are a writing assistant. Rephrase the provided English text to make it entirely unique in its expression, as if written by a different author. The new version must convey the same information but use a different style and vocabulary. The output must be in English. Output only the unique version of the text. Here is the text:

Prompt #9
Rewrite the following English text with the goal of enriching its vocabulary. Replace common words with more precise or descriptive synonyms where appropriate, while keeping the original meaning intact. The output must be in English. Output only the revised text. Here is the text:

Prompt #10
You are an English paraphrasing model. Rewrite the text provided below the separator. Your output must be in English. The meaning must be identical to the original. You must only output the final text. Here is the text:

Prompt #11
Completely restructure the following text. Change the active/passive voice, replace at least 40% of the core verbs, and ensure the sentence flow is radically different from the original while keeping the meaning. Output only the result.

Prompt #12
Rewrite this from the perspective of a different author. Use unique vocabulary and distinct syntactic patterns. The information must stay the same, but the 'fingerprint' of the text must change. Output only the result.

C.3 Prompts for semantic crossover

Prompt p_{fus} for Semantic Crossover

Synthesize the following two versions into a third, unique version. Do not pick one over the other; merge their best ideas into a completely new expression that retains the original meaning. Output only the merged text.

C.4 Prompt for guided mutation

Prompt p_{mut} for Guided Mutation

You MUST rewrite the following text. The following words are detected as 'AI patterns' and MUST be removed or replaced with creative synonyms: *{bad_words}*. Do not just swap words; restructure the entire sentence around these parts to ensure a natural but different flow. Output only the revised text.

D Additional Experiment Results

Generalization across QA task. Addition to the article continuation task, we have conducted additional experiments on the ELI5 (Fan et al., 2019) and ASQA (Stelmakh et al., 2022) Question Answering (QA) tasks. QA is notably more challenging for paraphrase attacks because factual correctness and entity preservation are much stricter

Table 5: Generalization of TSAPA across Question Answering tasks.

QA Dataset	ASR ↑	BLEU ↑	ROUGEL ↑	BERTS ↑
ELI5	100.00%	6.81	0.2818	0.6805
ASQA	95.05%	22.64	0.4655	0.7840

than in open-ended continuation. Specifically, we generated watermarked texts using `Qwen3-1.7B` with KGW schemes. Then we applied TSAPA, with`gpt-4o-mini` as base paraphraser, to remove watermarks and evaluated watermark removal efficacy and semantic consistency performance, compared to original watermarked texts. Preliminary results are summarized in Tab. 5, shown that TSAPA maintains near-perfect ASR ($> 95\%$) while preserving high semantic utility. This demonstrates that our method successfully generalizes to information-seeking and instruction-constrained tasks.

Robustness with weaker paraphrasers. To test if the search saturates or fails with limited candidate quality, we replaced `gpt-4o-mini` with much weaker, open-source models (`Qwen3-1.7B` and `Qwen3-8B`) as the attack models against a `Qwen3-1.7B` (KGW) watermark on the ASQA dataset. Results are shown in Tab. 6. The performance remains stable. Even when the paraphraser's capacity ceiling is drastically lowered (e.g., to a 1.7B model), TSAPA still achieves $> 94\%$ ASR with good semantic fidelity. This proves that the evolutionary mechanism, rather than just the underlying LLM's strength, is the primary driver of the attack's success.

Table 6: Robustness of TSAPA with different base paraphrasers.

Base Paraphraser	ASR ↑	BERTS ↑	ROUGEL ↑	BLEU ↑
`Qwen3-1.7B`	94.06%	0.7801	0.4588	22.62
`Qwen3-8B`	95.05%	0.7725	0.4460	20.81
`gpt-4o-mini`	95.05 %	0.7840	0.4655	22.64

Ablation on sentence-level crossover. To test whether the sentence-level crossover mechanism is necessary in `TSAPA`, we conducted an ablation study where we replaced the semantic crossover (w/ crossover) with standard single-parent LLM mutation (w/o crossover). The results are shown in Tab. 7, removing the crossover yields a marginal 3% gain in ASR but causes catastrophic degradation in text utility. Without crossover to fuse and stabilize semantic features, PPL explodes from a highly 2.84 to an incoherent 47.93.

Table 7: Ablation study on the sentence-level crossover mechanism in `TSAPA`.

Methods	ASR ↑	PPL ↑	BLEU ↑	ROUGEL ↑	BERTS ↑
w/ crossover	95.04%	**2.84**	**20.52**	**0.5332**	**0.7791**
w/o crossover	**98.01%**	47.93	13.75	0.3794	0.7275

E Data Statement

We utilize the `C4` dataset (Raffel et al., 2020). It is a colossally cleaned version of the web crawl corpus, created by applying extensive filtering heuristics to the April 2019 snapshot of Common Crawl. The dataset is distributed under the ODC-BY license. We confirm that our usage adheres to the license terms and is consistent with its intended use for research purposes. Regarding documentation, `C4` consists of English-language text sourced from diverse public web domains.