$\mathcal{B}^4$: A Black-Box ScruBBing Attack on LLM Watermarks

Baizhou Huang [♣♡*] **Xiao Pu** [◇*†] **Xiaojun Wan** [♣♡]
[♣]Wangxuan Institute of Computer Technology, Peking University
[♡]State Key Laboratory of General Artificial Intelligence
[◇]University of California, Santa Barbara
`{hbz19,wanxiaojun}@pku.edu.cn` `xiao_pu@ucsb.edu`

Abstract

Watermarking has emerged as a prominent technique for LLM-generated content detection by embedding imperceptible patterns. Despite supreme performance, its robustness against adversarial attacks remains underexplored. Previous work typically considers a grey-box attack setting, where the specific type of watermark is already known. Some methods even necessitates knowledge about details of hyperparameters. Such prerequisites are unattainable in real-world scenarios. Targeting at a more realistic black-box threat model with fewer assumptions, we here propose $\mathcal{B}^4$, a **B**lack-**B**ox scru**BB**ing attack on LLM watermarks. Specifically, we formulate the watermark scrubbing attack as a constrained optimization problem by capturing its objectives with two distributions: a *Watermark Distribution* and a *Fidelity Distribution*. The optimization problem can be approximately solved using two proxy distributions. Experimental results across 12 different settings demonstrate the superior performance of $\mathcal{B}^4$ compared with other baselines. [1]

1 Introduction

The rapid advancement of large language models (LLMs) has demonstrated their unimaginable potentials across various applications. Systems such as ChatGPT (OpenAI, 2023) are now seamlessly integrated into many aspects of daily life. Despite benefits, the extensive deployment of LLMs has sparked serious concerns regarding potential misuse, such as large-scale disinformation, automated spamming, and social media manipulation, thereby threatening academic integrity and intellectual property rights (Bender et al., 2021; Liu et al., 2023b). Consequently, detecting LLM-generated content has emerged as a crucial focus in the discourse on LLM safety and responsible deployment (Mitchell et al., 2023; Pu et al., 2023; Yang et al., 2023).

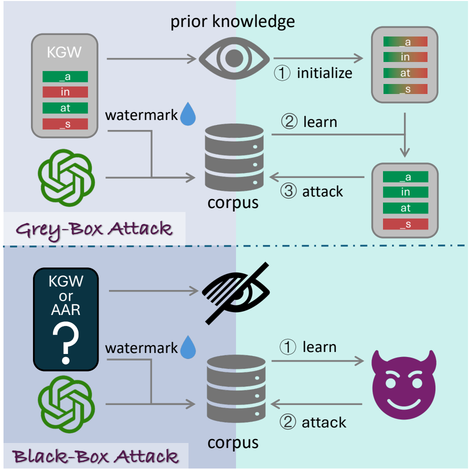

Figure 1: Difference between grey-box and black-box threat models. The left part in purple represents the victim and the right part in green represents the attacker. (Top) Prior work used prior knowledge for parametrization, e.g. the green-list partition of vocabulary in KGW, which makes watermark stealing easier. (Bottom) Under a more realistic black-box setting, the problem becomes much more challenging.

Watermarking stands out as a prominent technique for detecting LLM-generated text. It injects a hidden pattern invisible to human into generated contents of a specific LLM (Kirchenbauer et al., 2023a). By altering the original distribution of LLMs to a specific watermark distribution during each decoding step (Kuditipudi et al., 2023; Zhao et al., 2023; Hu et al., 2023) , all model generated contents can be statistically distinguished through hypothesis testing between the watermarked and the original distributions. This approach achieves a high detection rate and can be easily deployed, with only a negligible cost in the quality of generated content.

Despite its supreme performance, the robustness

[*] Equal contribution.
[†] Work done during internship in Peking University.
[1] The code is available in `https://github.com/skpig/B4-watermark-attack.git`.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 9113–9126
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

of watermarking methods against adversarial attacks remains underexplored. Among which, the *scrubbing attack* (Jovanović et al., 2024) presents a notable challenge in practical settings: if an adversary can successfully paraphrase LLM-generated contents into another semantically similar but watermark-free form, the effectiveness of the watermark will be heavily compromised.

Wu and Chandrasekaran (2024) and Jovanović et al. (2024) explored this question by proposing different watermark scrubbing methods. However, these approaches typically require strong prior knowledge about the specific type of watermark algorithm used in the victim LLM, even the specific hyperparameter of the algorithm (e.g. the context window size in KGW). For example, Wu and Chandrasekaran (2024) proposed a method based on the assumption that the victim LLM is protected with KGW watermarking algorithm (Kirchenbauer et al., 2023a), so that they can parameterize their attack model based on the green-red list partition. Such prerequisites are virtually unattainable in real-world scenarios, especially when we can only access the victim LLM via an API interface. The stringent constraint on prior information makes these attacking methods impractical for real-world applications.

In this work, we consider a more realistic black-box threat model with the sole prior knowledge of watermark existence. Under this attack setting, we further propose $\mathcal{B}^4$ (**B**lack-**B**ox scru**BB**ing attack), a universal watermark attack method, specially designed for the black-box threat model. With less assumption about the victim LLM, this method is more practical for real-life usage, allowing us to accurately explore the robustness of current watermark techniques.

The intuition of $\mathcal{B}^4$ is rooted in two fundamental objectives of an ideal watermark scrubbing attack: the adversarial texts should both exhibit minimal watermark patterns to evade from detection (*Efficacy*) and preserve the semantic information of the original texts (*Fidelity*). Drawing on this insight, we propose to capture these dual properties with distance to two distributions respectively: a *Watermark Distribution* and a *Fidelity Distribution*. Consequently, we formulate the task of watermark scrubbing attack into a constrained optimization problem. The local optimal of this problem can be solved by approximating these distributions with two proxies. Specifically, we steer a much smaller proxy model to approximate the *Watermark Distribution* of the victim LLM through distillation, and apply a general paraphrase model to obtain a distribution similar to the original texts for the *Fidelity Distribution*.

Compared to baseline watermark srubbing methods, our proposed $\mathcal{B}^4$ improves the attack success rate by up to 68.13% across different victim LLMs and watermarking method settings. Since there is an inherent trade-off between attack efficacy and semantic fidelity, we further demonstrate the superior performance of $\mathcal{B}^4$ over baselines using Pareto fronts: $\mathcal{B}^4$ removes text watermarks more efficiently under the same fidelity constraints, while also distorting less semantic information to achieve the same level of attack.

In summary, our contributions include: (1) formalizing the watermark scrubbing attack as a constrained optimization problem for the first time, and (2) developing a black-box watermark scrubbing method based on this formulation, which demonstrates superior performance compared to previous approaches.

2 Preliminary

2.1 LLM Watermarking

We begin by introducing LLM watermarking. Given any prompt $\mathbf{x}$, an LLM generates a sequence of output tokens $\mathbf{y}_i \sim P(\cdot|\mathbf{x}, \mathbf{y}_{<i})$ in an auto-regressive manner. Watermarking modifies this distribution into a distorted distribution, $P_w(\cdot|\mathbf{x}, \mathbf{y}_{<i})$, embedding hidden patterns associated with a secret key. Detecting watermarked text can thus be framed as a hypothesis testing problem, with the alternative hypothesis positing that the candidate text is sampled from the altered distribution. Typically, this detection process involves accumulating per-token statistics s_i to conduct a one-tailed significance test.

Initial research on LLM watermarking (Aaronson, 2023; Kirchenbauer et al., 2023a) demonstrated promising results, inspiring numerous follow-up works that introduced diverse watermarking algorithms (Kirchenbauer et al., 2023b; Kuditipudi et al., 2023; Zhao et al., 2023; Hu et al., 2023; Liu et al., 2023a; Hou et al., 2023). In this paper, we focus on four prominent watermarking techniques: KGW(Kirchenbauer et al., 2023a), AAR (Aaronson, 2023), Unigram (Zhao et al., 2023) and EXP (Kuditipudi et al., 2023). We provide a detailed description of these watermarking methods in Appendix A.

2.2 Watermark Attack Threat Model

An ideal watermarking algorithm should not only be able to precisely distinguish watermarked texts, but also be resilient to adversarial attacks. The watermark scrubbing attack can be viewed as a specialized form of post-hoc paraphrasing, where the original *Watermarked Sample* $\mathbf{y}^w$ is transformed into *Adversarial Sample* $\mathbf{y}$. It should retain similar semantics (*Fidelity*) while evading detection by removing watermark patterns (*Efficacy*).

In this work, our threat model consists of two main actors: the victim and the attacker. The *victim* is a proprietary large language model, protected by a watermarking algorithm and accessible only through an API service interface. The *attacker* is able to query the victim model with different prompts through the interface to obtain corresponding responses. We only make the minimal assumption that the attacker is informed with the existence of watermarks but possesses no prior knowledge of the specific type of the watermarking algorithm used. This stands in contrast to previous work that commonly relies on strong assumptions, such as the details of hyperparameters (Wu and Chandrasekaran, 2024; Jovanović et al., 2024) or an oracle detector (Pang et al., 2024), which severely restrict the range of attackable watermarking algorithms.

Our threat model is more closely aligned with real-world scenarios – Most leading companies such as OpenAI and Anthropic provide online API-based services under similar settings. By adhering to this realistic threat model, our investigation will provide more meaningful insights into the robustness of current watermarking techniques, fostering a deeper understanding of their vulnerabilities.

3 Method

In this section, we introduce $\mathcal{B}^4$, a universal watermark attacking method under the black-box threat model settings. We begin by explaining the intuition behind the method, followed by formalizing watermark scrubbing attack as an optimization problem. Next, we detail the numerical solving process of the formulated problem via two proxy distributions. Finally, we conclude with a discussion of potential approximation errors, along with a proposed adjustment for alleviation. We present the pseudocodes of our method in Appendix B.

3.1 Scrubbing as an Optimization Problem

As mentioned in Section 2.1, various watermarking algorithms differ in their watermark injection and detection mechanisms, which often causes a scrubbing attack effective against one algorithm but failing against others. Nonetheless, all watermarking techniques fundamentally involve altering the token distribution, regardless of the specific parametrization or detection statistics used. Thus, we can generalize the watermarking process as a transformation to a watermark distribution, $P_w(\mathbf{y})$.

Suppose an ideal adversarial sample $\mathbf{y}$ of watermark attack is drawn from a probability distribution $Q(\cdot|\mathbf{y}^w)$, conditioned on a watermarked sample $\mathbf{y}^w$. We can then characterize the objectives of watermark scrubbing attack using mathematical expressions. Fidelity reflects the similarity between $\mathbf{y}$ and $\mathbf{y}^w$, which can be expressed by the divergence from an implicit *Fidelity Distribution* $P_f(\mathbf{y}|\mathbf{y}^w) \propto \text{SIM}(\mathbf{y}, \mathbf{y}^w)$, where $\text{SIM}(\cdot,\cdot)$ is a true similarity measure between two texts. Efficacy, on the other hand, requires to minimize the likelihood that a watermark detector successfully identifies the adversarial sample as being watermarked. One sufficient condition to achieve this is to maximize the distance from the *Watermark Distribution* P_w, since the watermark detection is a hypothesis testing with P_w as the distribution under the alternative hypothesis. Therefore, to find the optimal solution Q can be formalized into the following constrained optimization problem:

$$\min_{Q(\cdot|\mathbf{y}^w)} -D_{\text{KL}}(Q(\cdot|\mathbf{y}^w)||P_w(\cdot))$$
$$s.t. D_{\text{KL}}(Q(\cdot|\mathbf{y}^w)||P_f(\cdot|\mathbf{y}^w)) \leq \epsilon, \quad (1)$$

where ϵ is a hyperparameter controlling the degree of allowed semantic deviation from the original watermarked sample. Since the problem satisfies the Slater Constraint Qualification (Bertsekas, 1999, Proposition 3.3.9), the local minimal are guaranteed to satisfy the Karush-Kuhn-Tucker (KKT) conditions. Solving the KKT conditions for the problem above immediately yields the following corollary:

Corollary 1. *The local minimum point Q^* has the form of*

$$Q^*(\mathbf{y}|\mathbf{y}^w) = \frac{1}{Z} {P_f}^{\frac{1}{1-\lambda^*}}(\mathbf{y}|\mathbf{y}^w) {P_w}^{-\frac{\lambda^*}{1-\lambda^*}}(\mathbf{y}), \quad (2)$$

where $\lambda^* \in (0,1)$ is the corresponding Lagrange multiplier satisfying $D_{\text{KL}}(Q^*||P_f) = \epsilon$, and Z is the normalizing constant.

The corresponding Lagrange multiplier λ^* can be solved using the Newton-Raphson Method.

3.2 Approximated Solution with Proxy Distributions

To solve the optimization problem above, we also need to parameterize both P_w and P_f. However, both of them are inherently intractable. Given the lack of prior knowledge regarding the watermarking algorithm behind the API, obtaining the accurate form of P_w is impossible. Therefore, we leverage model distillation (Hinton et al., 2015) to learn a proxy, denoted as $\hat{P_w}$ [2]. In specific, we sample from the golden watermark distribution by querying the victim LLM for multiple times. Those responses form a training corpus $\mathcal{D}$ to train a local language model p_θ by minimizing the Negative Log-Likelihood (NLL) loss,

$$\mathcal{L}(\theta) = -\sum_{\mathbf{y}\in\mathcal{D}} \log \hat{P_w}(\mathbf{y};\theta) = -\sum_{\mathbf{y}\in\mathcal{D};i} \log p_\theta(\mathbf{y}_i|\mathbf{y}_{<i})$$

The choice of the local model can vary among numerous open-source LLMs available. Empirically, we find that a proxy watermark distribution can be learned surprisingly well from a moderately sized corpus.

As for the fidelity distribution P_f, we simply apply a paraphrase model p_ϕ as the proxy fidelity distribution, with the watermarked sample $\mathbf{y}^w$ serving as context, i.e. $\hat{P_f}(\mathbf{y}|\mathbf{y}^w;\phi) = \prod_i p_\phi(\mathbf{y}_i|\mathbf{y}_{<i}, \mathbf{y}^w)$. Now that we are able to parameterize the local optimal Q^*, transforming Equation 2 into:

$$Q^*(\mathbf{y}_i|\mathbf{y}_{<i}, \mathbf{y}^w) = \frac{\hat{P_f}^{\frac{1}{1-\lambda^*}}(\mathbf{y}_i|\mathbf{y}_{<i}, \mathbf{y}^w; \phi)}{\hat{P_w}^{\frac{\lambda^*}{1-\lambda^*}}(\mathbf{y}_i|\mathbf{y}_{<i}; \theta)} \quad (3)$$

This allows us to sample adversarial texts in an auto-regressive manner[3].

3.3 Approximation Error Adjustment (AEA)

While proxy distributions provide a practical approximation for solving the optimization problem, they may diverge from the golden distributions in certain regions of sample space. This issue is particularly severe for the proxy watermark distribution $\hat{P_w}$, due to the inherent limitations of sampling-based model distillation.

Unlike logit-based distillation (Hinton et al., 2015), which offers a holistic guidance over the entire space, sampling-based distillation (Kim and Rush, 2016) applies a one-peak correction on certain regions around each observed sample by the NLL loss. When the training data is sparse, the student model may struggle to generalize to unobserved regions of the teacher watermark distribution P_w. For instance, in the Unigram algorithm, green list tokens generally have higher output probabilities. However, due to the randomness of sampling, some of them may remain unobserved, leading to significant discrepancies between their probabilities in the proxy distribution $\hat{P_w}$ and the golden distribution P_w. These approximation errors may critically affect our method, given that the optimization objective involves minimizing KL divergence, a global measure over the entire sample space.

To address this issue, we introduce an adjustment mechanism to refine the approximation errors by simply excluding those under-fitting regions from the calculation of the KL divergence objective. We identify these regions, Σ_u^i, which is a subset of vocabulary Σ, by comparing the proxy distributions before and after distillation at each decoding step, given the context $\mathbf{y}_{<i}$:

$$\Sigma_u^i = \{v \in \Sigma : |p_\theta(v|\mathbf{y}_{<i}) - p_{\theta_{ini}}(v|\mathbf{y}_{<i})| < \mu\},$$

where θ_{ini} is the initialized weights before distillation, and μ is a pre-defined threshold[4]. Eventually, we can incorporate the approximation error adjustment into Equation 3 to obtain the adjusted attack distribution:

$$Q^*(\mathbf{y}_i|\mathbf{y}_{<i}, \mathbf{y}^w) = \begin{cases} \hat{P_f}(\mathbf{y}_i|\mathbf{y}_{<i}, \mathbf{y}^w;\phi) & \text{if } \mathbf{y}_i \in \Sigma_u^i, \\ \frac{\hat{P_f}^{\frac{1}{1-\lambda^*}}(\mathbf{y}_i|\mathbf{y}_{<i}, \mathbf{y}^w;\phi)}{\hat{P_w}^{\frac{\lambda^*}{1-\lambda^*}}(\mathbf{y}_i|\mathbf{y}_{<i};\theta)} & \text{otherwise.} \end{cases} \quad (4)$$

Besides the above adjustment, the proposed $\mathcal{B}^4$ is also compatible with all existing decoding strategies, such as nucleus sampling (Holtzman et al., 2020). In this paper, we apply both top-50 and 10-beam search to ensure high-quality outputs.

4 Experiments

4.1 Experimental Settings

4.1.1 Victim Models and Target Datasets

In order to simulate different scenarios in the real world, we consider three different sizes of victim LLMs from two well-known model families, including Llama-2-7B (Touvron et al., 2023a)[5],

[2] Throughout this work, proxy distributions are denoted using the hat notation ($\hat{\ }$).

[3] We ignore the normalizing constant for simplicity.

[4] We apply the uniform probability $1/|\Sigma|$ as threshold.

[5] https://huggingface.co/meta-llama/Llama-2-7b

Llama-3-70B (Touvron et al., 2023b)[6] and Qwen2-72B (Yang et al., 2024)[7]. We implement four watermarking algorithms mentioned in Section 2.1, i.e. KGW, Unigram, EXP and AAR. We follow the experimental setting in previous work (Kirchenbauer et al., 2023a,b) to utilize a random subset of C4-Realnewslike dataset (Raffel et al., 2020) as prompts to query these watermarked models for 100 responses in the length of 200 tokens. These responses construct the watermarked sample dataset to evaluate different attack methods.

4.1.2 Baselines

Given the black-box threat model setting, we consider three watermark attack baselines in our experiments. **Base** directly utilizes an LLM to paraphrase the watermarked sample via prompting. We apply the identical paraphraser ϕ used by $\mathcal{B}^4$ for fair comparison. Recursive paraphrasing (**RP**) applies a paraphraser to paraphrase the watermarked sample in chunk-level for multiple times (Sadasivan et al., 2023) . We also use the identical paraphraser ϕ for fair comparison. **DIPPER** (Krishna et al., 2023) is a T5-XXL-based paraphraser specifically tuned for evading AI-generation detection.

4.1.3 Metrics

As stated in Section 2.2, *Fidelity* and *Efficacy* serve as two goals of our watermark scrubbing method. For fidelity, we follow Krishna et al. (2023); Jovanović et al. (2024) to use P-SP (Wieting et al., 2022) to reflect the semantic similarity between the texts before and after attacks. For efficacy, we use ROC-AUC (Hanley and McNeil, 1982) to evaluate the watermark strength after attacks. One should be noted that there is an inherent trade-off between these two metrics: under smaller fidelity constraint, the valid semantics space is more limited for modification to remove the watermark patterns within the original text, thereby inevitably leading to lower efficacy. Due to this, instead of comparing different methods within a single dimension, we aim to identify an attacking method that is optimal in terms of both fidelity and efficacy. We borrow the concept of Pareto front from economics as an evaluation method, which consists of a group of optimal states in multi-objective scenarios. In practice, we apply each method to generate diverse adversarial samples under different fidelity constraints by grid searching its hyperparameter space. Take $\mathcal{B}^4$ as an example, we vary ϵ, the fidelity constraint threshold defined in Problem 1, to obtain diverse data points with different fidelity and efficacy, thereby forming a Pareto front curve as shown in Figure 2. The hyperparameter space of other baselines are presented in Appendix C. For a more intuitive comparison, we also include the numerical AUC@f metrics, which indicates the best ROC-AUC score that an attacking method can achieve under a specific fidelity constraint f.

	Unigram		KGW		AAR		EXP	
Method	@70	@80	@70	@80	@65	@70	@75	@80
	Llama-3-70B							
$\mathcal{B}^4$	**30.19**	80.91	**49.11**	**66.12**	**75.69**	**80.60**	**69.42**	**75.23**
DIPPER	81.92	88.46	68.00	81.14	82.53	87.37	77.68	80.68
RP	59.77	**79.45**	67.14	76.88	80.34	-	70.69	82.92
Base	72.98	83.57	67.56	81.54	83.98	89.04	73.88	78.70
	Llama-2-7B							
$\mathcal{B}^4$	**68.67**	**89.41**	**62.38**	**69.24**	**73.81**	**82.69**	**53.84**	**69.37**
DIPPER	87.36	93.13	77.97	81.83	86.84	89.16	66.79	79.33
RP	86.91	93.51	71.67	74.06	80.89	82.99	73.57	77.85
Base	82.80	91.48	73.26	78.45	86.25	88.22	67.00	75.71
	Qwen2-72B							
$\mathcal{B}^4$	**26.77**	**54.04**	**47.53**	**70.19**	**60.24**	**71.15**	**58.49**	**68.66**
DIPPER	80.15	91.89	72.28	84.95	84.34	90.96	76.62	78.39
RP	65.16	79.47	72.64	78.67	82.36	85.93	74.34	74.34
Base	84.00	84.00	75.72	75.72	83.03	87.05	76.27	76.27

Table 1: AUC@f (↓) of watermark scrubbing attack methods against different victim LLMs protected by different watermark algorithms. The best performance under each setting are **Bolded**. Dash ('-') indicates an unattainable fidelity threshold for a specific method. Note that the selected threshold f varies due to the varying performance of attack methods against different watermarking algorithms.

4.1.4 Model Choice for Proxy Distributions

Our proposed $\mathcal{B}^4$ requires a local available LLM θ to fit the watermark distribution P_w and a paraphraser ϕ as $\hat{P}_f$. The choice of θ is diverse. We consider two possible scenarios in the real life regarding to whether we have access to a model belonging to the same family of the victim LLM. Some leading companies such as Google and Mistral, provides API service of large-scale models, while also open-sourcing some small-scale models with the same architecture. In such scenario, we can surely leverage the small size model as θ for better distillation performance. Therefore, we apply Qwen2-0.5B to fit P_w of Qwen2-72B to explore our method under this setting. In contrast, we apply the Gemma-2-2B-it and Gemma-2-2B to fit P_w of Llama-2-7B and Llama-3-70B respectively.

[6] https://huggingface.co/meta-llama/Meta-Llama-3-70B

[7] https://huggingface.co/Qwen/Qwen2-72B

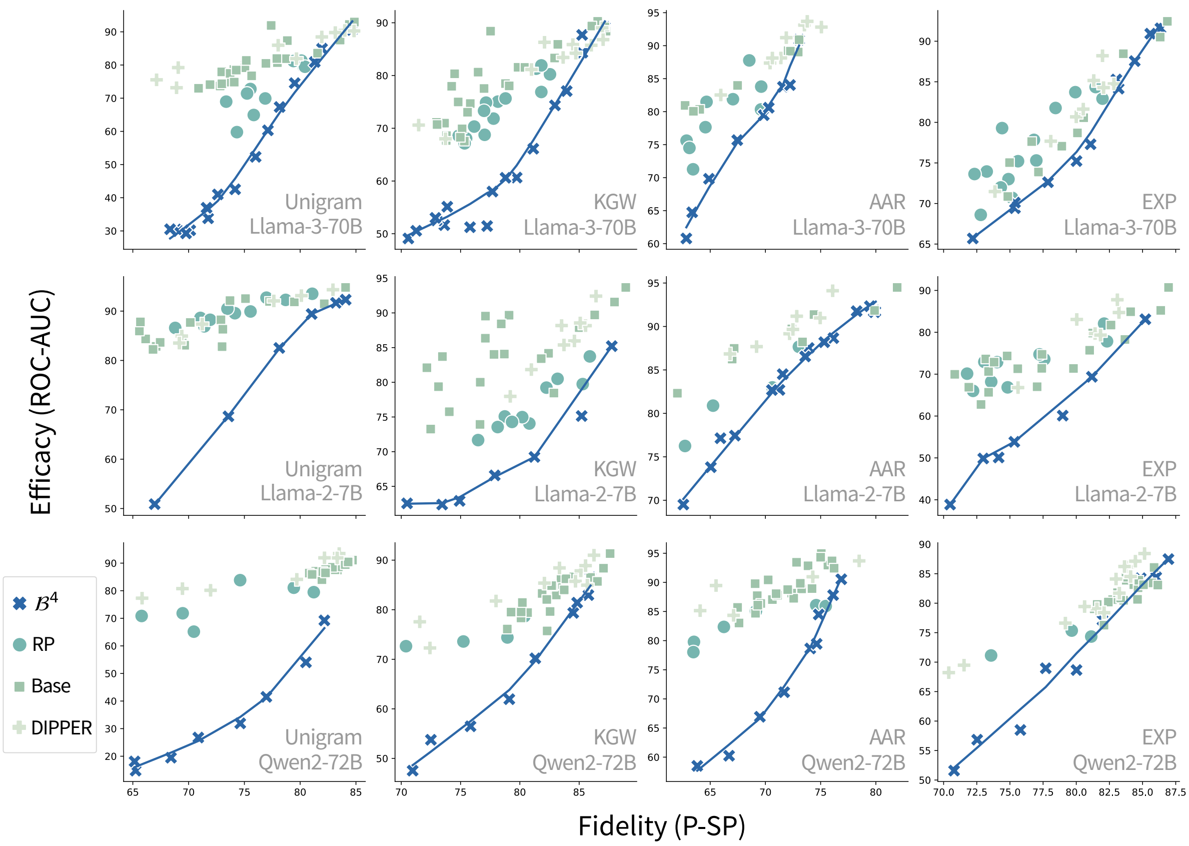

Figure 2: Performance of watermark scrubbing attack methods against different victim LLMs protected by different watermark algorithms. The y-axis indicates Efficacy, measured by ROC-AUC (↓), while the x-axis indicates Fidelity, measured by P-SP (↑). Each data point represents one watermarking algorithm with one specific group of hyperparameters. We draw the Pareto front of $\mathcal{B}^4$ by LOWESS (Jacoby, 2000).

Victim	θ	ϕ
Llama-2-7B	Gemma-2-2B	Gemma-2-2B-it
Llama-3-70B	Gemma-2-2B	Gemma-2-9B-it
Qwen2-72B	Qwen2-0.5B	Qwen2-7B-it

Table 2: Selection of the proxy watermark distribution model θ and the proxy fidelity distribution model ϕ against different victim LLMs.

The diversity of proxy distributions aims at proving the model-agnostic of our proposed $\mathcal{B}^4$. The specific choices of θ and ϕ are listed in Table 2. More experimental details are presented in Appendix C.

4.2 Main Results

The primary results of our experiments are illustrated in Figure 2, with a detailed numerical comparison provided in Table 1, where we selected several specific fidelity thresholds. Data points with less than 60% P-SP or more than 95% ROC-AUC were excluded, as they reflect either poor fidelity or weak efficacy.

Trade-off between fidelity and efficacy. Our experimental results across various watermarks and victim models reveal a clear trade-off between fidelity and efficacy, following a distinctive curve that moves from the upper right to the lower left, which aligns with our intuition from Section 4.1.3. Furthermore, it is worth noting that all attack methods struggle to remove watermarks when the fidelity constraints are stringent. This also aligns with our intuition. The lexical space for modification is rather limited under stringent fidelity constraint. As a result, the output spaces of different attack methods overlap significantly, leading to comparable performance in terms of efficacy.

$\mathcal{B}^4$ outperforms baselines across all settings. Table 1 shows that $\mathcal{B}^4$ consistently achieves the best performance: under the same fidelity constraints, B4 is more effective in scrubbing watermark, reflected by lower ROC-AUCs. The superior performance of $\mathcal{B}^4$ is even more evident in Figure 2, where the Pareto front for $\mathcal{B}^4$ consistently lies below the data points of other methods.

Robustness of different watermarking methods. An important goal of adversarial attacks is to assess the robustness of various watermarking algorithms. We observe that the other three attack baselines generally struggle to effectively remove watermarks and have difficulty differentiating the robustness of different watermarking techniques. However, with its superior efficacy, $\mathcal{B}^4$ is able to expose the differences among various watermarking schemes. Our results indicate that $\mathcal{B}^4$ is most efficient in scrubbing Unigram, followed by EXP and KGW, while AAR is the most difficult to remove. This may be related to the learnability of the watermarking patterns. For example, Unigram is easier to learn because it maintains a fixed green-red word list, whereas learning AAR watermark requires learning the sampling distribution pattern based on a context window, making it more challenging.

Comparing different models. In Section 4.1.4, we introduced two proxy watermark distribution model settings for $\mathcal{B}^4$: using smaller models from either the same or a different family as the victim model. Our experiments reveal that using small models from the same family (e.g., attacking Qwen2-72B with Qwen2-0.5B) serves as a more effective proxy. This finding aligns with our intuition, as models from the same family share an identical tokenizer and training data, enabling them to more easily capture the watermarked patterns during distillation, instead of struggling with the domain shifts.

4.3 Further Analysis

4.3.1 Ablation Study

Here we explore the effect of hyperparameter ϵ and the approximation error adjustments (AEA), as shown in Figure 3.

Effect of ϵ. With the growth of ϵ, semantic fidelity decreases while attacking efficacy increases smoothly. This phenomenon aligns with the role of ϵ as the level of fidelity constraint in Problem 1, further justifying our optimization problem.

Effect of AEA. It is evident that approximation error adjustment consistently yields performance improvements to $\mathcal{B}^4$, which becomes more significant under lower fidelity constraints. Notably, without of AEA, an unusual phenomenon arises when attacking Unigram-protected Llama-3-70B: as ϵ increases to a large value, the curve deviates abnormally, shifting towards the upper left, indicating both diminished fidelity and efficacy. But this abnormal phenomenon disappears after the adjustment is applied, with fidelity and efficacy exhibiting a normal trade-off relationship. It suggests that the Q^* calculated without AEA is no longer local optimal, which further implies the approximation error discussed in Section 3.3 may have greater influence when the constraint in Problem 1 is loose. These findings again underscore the necessity of our proposed adjustments.

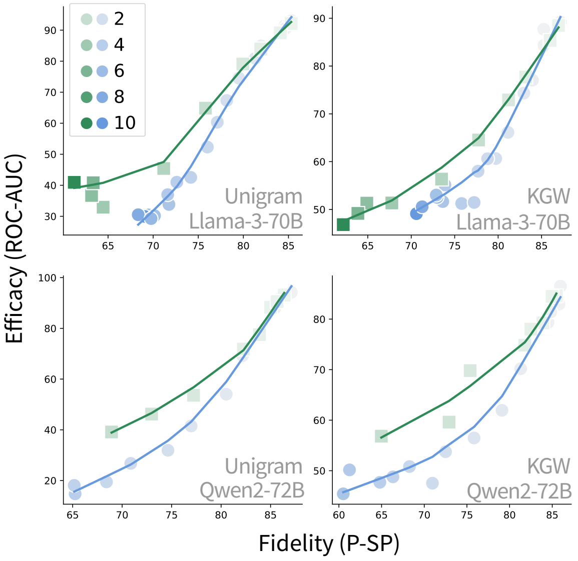

Figure 3: Ablation of ϵ and AEA. The blue dots and lines denote performance of $\mathcal{B}^4$ with AEA while the green denotes that without AEA. The value of ϵ is indicated by the shading of dots, with darker colors representing larger ϵ.

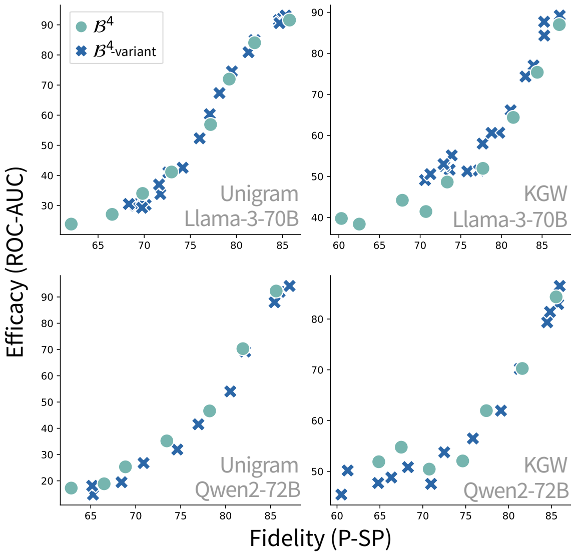

Figure 4: Comparison between $\mathcal{B}^4$ and $\mathcal{B}^4$-variant.

Training Data	Qwen2-72B KGW	Qwen2-72B Unigram	Llama-3-70B KGW	Llama-3-70B Unigram
100,000	99.17	100.00	96.79	98.83
150,000	98.97	100.00	99.02	98.44
200,000	99.15	100.00	98.57	99.01
250,000	99.24	99.98	98.57	99.46

Table 3: ROC-AUC (↑) of proxy models fitting to Unigram/KGW watermarked Qwen2-72B and Llama-3-70B with different sizes of watermarked training dataset.

4.3.2 $\mathcal{B}^4$-Variant for Speeding Up

As discussed in Section 3.1, Lagrange multiplier λ^* can be determined by solving the equation $D_{\mathrm{KL}}(Q^*||P_f) = \epsilon$. In other words, λ^* is implicitly dependent on the hyper-parameter ϵ via an implicit function $\lambda^* = \Lambda(\epsilon)$. Therefore, instead of parameterize ϵ, we can parameterize λ^*, bypassing the need for solving the equation. We refer to this speedup approach as $\mathcal{B}^4$-variant, and we compare its performance to the original parametrization version in Figure 4. Empirical results show that the speedup variant performs on par with the original $\mathcal{B}^4$, offering a comparable solution with reduced complexity.

4.3.3 On the Scale of Training Corpus

We apply 200,000 training samples to distill the proxy watermark distribution $\hat{P}_w$, which is at a cost of about \$90[8]. Here we further discuss how the size of training dataset effects the performance of our method. We start by exploring this question: how many training samples are sufficient for a proxy model to learn the watermark distribution? We utilize Gemma-2-2B and Qwen2-0.5B as proxy models to fit the watermark distribution of Llama-3-70B and Qwen2-72B, respectively, on different scales of watermarked training corpus. As shown in Table 3, when the training corpus shrinks from 250,000 to 100,000, we don't see a noticeable ROC-AUC drop of proxy models — With only 100,000 watermarked samples, a small proxy model is already able to learn the watermark distribution pretty well.

We present the performance of our $\mathcal{B}^4$ method under different training dataset settings in Figure 5. We find that a dataset of 200,000 watermarked samples are sufficient for training, and 100,000 is also enough for efficient scrubbing attacks.

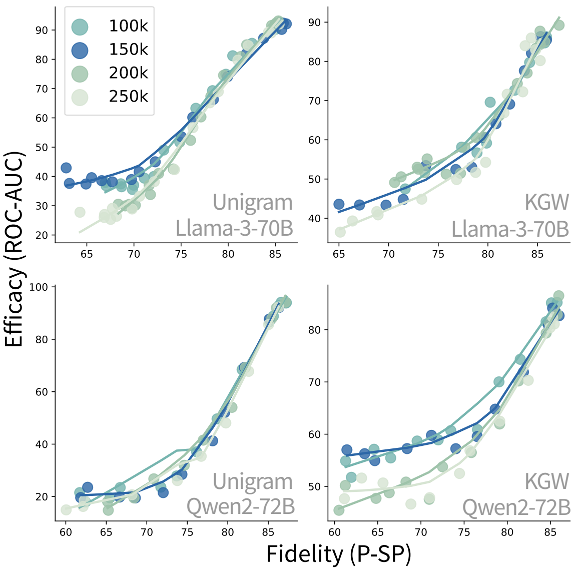

Figure 5: Performance of $\mathcal{B}^4$ with different size of training corpus for approximating watermark distribution.

5 Related Work

LLM Watermarking. Early foundational work in LLM watermarking include KGW (Kirchenbauer et al., 2023a) and AAR (Aaronson, 2023), both of which have significantly influenced subsequent research. Following KGW, various methods have been proposed to enhance performance by introducing innovations such as alternative hash functions (Kirchenbauer et al., 2023b; Hou et al., 2023; Liu et al., 2023a; Ren et al., 2023), heuristic partitioning strategies (Li et al., 2023; Chen et al., 2023), multi-bit message embedding (Wang et al., 2023), and more robust hypothesis testing techniques (Fernandez et al., 2023).

Watermark Scrubbing Attack. Several previous studies have explored adversarial attacks against watermarking algorithms (Sadasivan et al., 2023; Pang et al., 2024). Both Jovanović et al. (2024) and Zhang et al. (2024) focus on stealing the green-list vocabulary in KGW by analyzing token frequency statistics, which requires prior knowledge of the context window size for accurate estimation. Zhang et al. (2024) further formalize the task of stealing the green-list vocabulary into a mixed-integer programming problem, though this method relies on access to the full vocabulary and tokenizer, making it difficult to apply in real-world black-box settings. Another line of work aims to develop black-box scrubbing attacks. Both Zhang et al. (2023) and Sadasivan et al. (2023) propose recursive paraphrasing pipelines, while Krishna et al.

[8] The price is calculated based on OpenAI's ChatGPT API pricing (gpt-3.5-turbo-0125).

(2023) introduce DIPPER, a paraphraser specifically tuned to evade AI-generated text detection. Among all, our work is most closely related to the findings of Gu et al. (2024), who demonstrate that model distillation is an effective technique for watermark spoofing. This insight has motivated the design of our proxy watermark distribution.

6 Conclusion

In this work we research into the watermark scrubbing attack method within the black-box setting, a practically significant but under-studied field. Without needing to know details or even the type of watermarking method used, we derive the format of local-optimal adversarial sample distribution by approximating a fidelity distribution and a watermark distribution. Our proposed attack approach, $\mathcal{B}^4$, can effectively remove the watermark pattern without distorting the original semantic fidelity, demonstrating its superior performance over all baseline models across a wide range of victim settings.

Acknowledgments

This work was supported by Beijing Science and Technology Program (Z231100007423011) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We appreciate the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

Limitations

In this work, we apply the basic distillation technique to approximate the watermark distribution. However, the watermark distillation is quite different from normal distillation, since we aims at learning a specific token pattern, instead of learning knowledge. We do not discuss a distillation technique specific designed for watermark distillation due to space limitations. We will explore this further in future work.

Ethical Statements

This work explores adversarial attacks against proprietary LLM watermarking protections. While we recognize the potential implications, we are unaware of any current real-world deployments of watermarking techniques that could be impacted by our methods. Therefore, we believe the risks of malicious applications are limited at this time, and our research cannot be exploited in practice under present conditions. Nonetheless, we emphasize that our primary goal is to advance the understanding of watermark robustness, contributing to the development of more secure and resilient systems.

References

Scott Aaronson. 2023. Watermarking of large language models. Technical report.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

D.P. Bertsekas. 1999. *Nonlinear Programming*. Athena scientific optimization and computation series. Athena Scientific.

Liang Chen, Yatao Bian, Yang Deng, Shuaiyi Li, Bingzhe Wu, Peilin Zhao, and Kam-fai Wong. 2023. X-Mark: Towards Lossless Watermarking Through Lexical Redundancy. *Preprint*, arxiv:2311.09832.

Gerard Debreu. 1960. Individual choice behavior: A theoretical analysis.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. 2023. Three Bricks to Consolidate Watermarks for Large Language Models. *Preprint*, arxiv:2308.00113.

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

James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. *Preprint*, arXiv:1904.09751.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. SemStamp: A Semantic Watermark with Paraphrastic Robustness for Text Generation. *Preprint*, arxiv:2310.03991.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2023. Unbiased Watermark for Large Language Models. *Preprint*, arxiv:2310.10669.

William G. Jacoby. 2000. Loess:: a nonparametric, graphical tool for depicting relationships between variables. *Electoral Studies*, 19(4):577–613.

Nikola Jovanović, Robin Staab, and Martin Vechev. 2024. Watermark Stealing in Large Language Models. *Preprint*, arXiv:2402.19361.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A Watermark for Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17061–17084. PMLR.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2023b. On the Reliability of Watermarks for Large Language Models. *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. *Preprint*, arxiv:2303.13408.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust Distortion-free Watermarks for Language Models. *Preprint*, arxiv:2307.15593.

Yuhang Li, Yihan Wang, Zhouxing Shi, and Cho-Jui Hsieh. 2023. Improving the Generation Quality of Watermarked Large Language Models via Word Importance Scoring. *Preprint*, arxiv:2311.09668.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2023a. A Semantic Invariant Robust Watermark for Large Language Models. *Preprint*, arxiv:2310.06356.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. 2023b. A Survey of Text Watermarking in the Era of Large Language Models. *Preprint*, arxiv:2312.07913.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *Preprint*, arXiv:2301.11305.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Qi Pang, Shengyuan Hu, Wenting Zheng, and Virginia Smith. 2024. Attacking LLM Watermarks by Exploiting Their Strengths. *Preprint*, arXiv:2402.16187.

Xiao Pu, Jingyu Zhang, Xiaochuang Han, Yulia Tsvetkov, and Tianxing He. 2023. On the zero-shot generalization of machine-generated text detectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4799–4808, Singapore. Association for Computational Linguistics.

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.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. 2023. A Robust Semantics-based Watermark for Large Language Model against Paraphrasing. *Preprint*, arxiv:2311.08721.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can AI-Generated Text be Reliably Detected? *Preprint*, arxiv:2303.11156.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Towards Codable Text Watermarking for Large Language Models. *Preprint*, arxiv:2307.15992.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-kirkpatrick. 2022. Paraphrastic representations at scale. In *Proceedings of the 2022 Conference* *on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 379–388, Abu Dhabi, UAE. Association for Computational Linguistics.

Qilong Wu and Varun Chandrasekaran. 2024. Bypassing LLM Watermarks with Color-Aware Substitutions. *Preprint*, arXiv:2403.14719.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023. A Survey on Detection of LLMs-Generated Content. *Preprint*, arxiv:2310.15654.

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. 2023. Watermarks in the Sand: Impossibility of Strong Watermarking for Generative Models. https://arxiv.org/abs/2311.04378v2.

Zhaoxi Zhang, Xiaomei Zhang, Yanjun Zhang, Leo Yu Zhang, Chao Chen, Shengshan Hu, Asif Gill, and Shirui Pan. 2024. Large Language Model Watermark Stealing With Mixed Integer Programming. *Preprint*, arXiv:2405.19677.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable Robust Watermarking for AI-Generated Text. https://arxiv.org/abs/2306.17439v2.

A Preliminary on Watermarking Methods

In this section we introduce the four watermaring methods used in our work.

- KGW (Kirchenbauer et al., 2023a). In each decoding step, KGW utilizes a c-length context window $\mathbf{y}_{i-c:i-1}$ as a random seed to partition the vocabulary Σ into a green list and a red list. The size of the green list is controlled by a hyperparameter γ. A perturbation τ is then applied to the logits of these green-list tokens to shape the watermark distribution.
- AAR (Aaronson, 2023). AAR also applies a c-length context window as a seed to generate a standard Gumbel distribution vector, which is then utilized for Gumbel-max sampling (Debreu, 1960). Essentially, the final watermark distribution becomes a degenerate distribution based on the sampled token.
- Unigram (Zhao et al., 2023). Similar to KGW, Unigram partitions the vocabulary but eliminates the dependency on the context window. Instead, it maintains a fixed green list throughout the generation process, providing increased robustness against adversarial attacks.
- EXP (Kuditipudi et al., 2023). EXP also borrows the idea of Gumbel-max sampling in AAR. However, instead of seeding on the context window, it maintains a global private seed sequence, where each seed corresponding to a specific time step during decoding.

B Pseudocodes for $\mathcal{B}^4$

Algorithm 1 Watermark Distribution Approximation

```
Input: Training corpus $D$, watermark distribution
       $P_w$, initialization weight $\theta_{ini}$
Output: Proxy watermark distribution $\hat{P_w}$
$\theta \leftarrow \theta_{ini}$
foreach epoch do
    foreach $\mathbf{y} \in D$ do
        $\mathcal{L}(\theta) \leftarrow -\sum_i \log p_\theta(\mathbf{y}_i|\mathbf{y}_{<i})$
        $\theta \leftarrow \theta - t\nabla L(\theta)$
    end
end
$\hat{P_w} \leftarrow p_\theta$
return $\hat{P_w}$
```

Algorithm 2 $\mathcal{B}^4$ with Approximation Error Adjustment

```
Input: Watermarked sample $\mathbf{y}^w$, proxy watermark distribution $\hat{P}_w$, proxy model after distillation $\theta$, proxy model
       before distillation $\theta_{ini}$, paraphraser distribution $\hat{P}_f$,
       paraphraser $\phi$, hyperparameter $\mu$
Output: Scrubbing attack distribution $Q^*$
foreach decoding step $i$ do
    $\Sigma_u^i \leftarrow \{v \in \Sigma_i : |p_\theta(v|\mathbf{y}_{<i}) - p_{\theta_{ini}}(v|\mathbf{y}_{<i})| < \mu\}$
    $\lambda^* \leftarrow$ solve_lagrange_multiplier($\hat{P}_{f;\phi}, \hat{P}_{w;\theta}, \Sigma -$
     $\Sigma_u^i$)
    foreach $\mathbf{y}_i \in \Sigma_u^i$ do
        $Q^*(\mathbf{y}_i|\mathbf{y}_{<i}, \mathbf{y}^w) \leftarrow \hat{P}_f(\mathbf{y}_i|\mathbf{y}_{<i}, \mathbf{y}^w; \phi)$
    end
    foreach $\mathbf{y}_i \in \Sigma - \Sigma_u^i$ do
        $Q^*(\mathbf{y}_i|\mathbf{y}_{<i}, \mathbf{y}^w) \leftarrow \frac{\hat{P_f}^{\frac{1}{1-\lambda}}(\mathbf{y}_i|\mathbf{y}_{<i},\mathbf{y}^w;\phi)}{\hat{P_w}^{\frac{\lambda}{1-\lambda}}(\mathbf{y}_i|\mathbf{y}_{<i};\theta)}$
    end
end
return $Q^*$
Function solve_lagrange_multiplier($\hat{P}_{f;\phi}, \hat{P}_{w;\theta}, \Sigma$):
    $\lambda \leftarrow \lambda_0$
    $h(\lambda, v) \leftarrow$ lambda $\lambda, v$ : $\frac{\hat{P_f}^{\frac{1}{1-\lambda}}(v|\mathbf{y}_{<i},\mathbf{y}^w;\phi)}{\hat{P_w}^{\frac{\lambda}{1-\lambda}}(v|\mathbf{y}_{<i};\theta)}$
    $f(\lambda) \leftarrow$ lambda $\lambda$ : $\sum_{v\in\Sigma} h(\lambda, v) \log \frac{h(\lambda,v)}{\hat{P_f}(v)} - \epsilon$
    while $|f(\lambda)| < 10^{-6}$ do
        $\lambda \leftarrow \lambda - \frac{f(\lambda)}{f'(\lambda)}$
    end
    $\lambda^* \leftarrow \lambda$
    return $\lambda^*$
```

C Experimental Details

Watermark algorithms. We implement the four watermarking algorithms mentioned above. For both KGW and Unigram, we apply a common setting with $\gamma = 0.5, \tau = 2$ following Kirchenbauer et al. (2023a). For EXP, we maintain a key sequence of length 256, enough to seed the evaluation datasets of token length 200. For both KGW and AAR, we set the context width c to 1 for better robustness.

Corpus Construction. We randomly select 200,000 samples from the English(en) subset of the C4 dataset and truncate each sample to the first 50 tokens to create the prompting dataset. We then apply each of the four watermark methods, i.e., KGW, Unigram, AAR and EXP, to query the victim models with these 50-token prompts, to generate 200,000 responses of 200-tokens. This results in a training corpus containing responses from a victim LLM embedded with a specific type of watermark.

Training Details. We then fine-tune the proxy models on each training corpus for 5 epochs, with a batch size of 128 sequences, sequence length of 256 tokens. We save the checkpoint of each train-

RP	
paraphrase counts	[1,2,3,4,5]
chunk size	[1,2]
prompt	Instruction 1 in Table D
DIPPER	
lex	[30,35,40,45,50,55,60,65,70]
Base	
beam size	[1, 10]
prompts	12 instructions in Table D
$\mathcal{B}^4$	
ϵ	[0.01,0.1,0.5,1,1.5,2,3,4,5,6,10]
beam size	10
prompt	Instruction 1 in Table D

Table 4: Hyperparamter space of different baselines

ing epoch and only reserve the one with the best validation result. We follow Gu et al. (2024) to set the maximal learning rate to 1e-5, and use cosine learning rate decay with a linear warmup for the first 500 steps. Training a Llama-2-2B proxy model took approximately 9 hours on 4 NVIDIA RTX A6000 48 GPUs, and training a Qwen2-0.5 proxy model took approximately 4 hours on 1 NVIDIA RTX A6000 48 GPU.

Hyperparameter Space. To generate a comprehensive Pareto front, we enumerate several sets of hyperparameters for each baseline method. The hyperparameter spaces for the different methods are listed in Table 4. Since the data points corresponding to these baselines are not uniformly distributed in the Fidelity-Efficacy plot across different experimental settings, we manually add some specific hyperparameter settings where necessary to ensure accurate representation.

D Prompt Instructions

Here we list instructions we used for paraphrasing:

Instructions for Paraphrase

Intruction 1: Paraphrase the following paragraphs line by line. Don't output any other information except the paraphrased texts. This is the text:

Intruction 2: You are an expert copy-editor. Please rewrite the following text in your own voice and paraphrase all sentences. Ensure that the final output contains the same information as the original text and has roughly the same length. Do not leave out any important details when rewriting in your own voice. This is the text:

Intruction 3: As an expert copy-editor, please rewrite the following text in your own voice while ensuring that the final output contains the same information as the original text and has roughly the same length. Please paraphrase all sentences and do not omit any crucial details. Additionally, please take care to provide any relevant information about public figures, organizations, or other entities mentioned in the text to avoid any potential misunderstandings or biases.

Instruction 4: As an expert copy-editor, please rewrite the following text in your own voice while ensuring that the final output contains the same information as the original text and has roughly the same length. Please paraphrase all sentences and do not omit any crucial details. Don't output any other information except the paraphrased texts. This is the text:

Intruction 5: Paraphrase the following paragraphs line by line. Try to keep the similar length to the original paragraphs. Don't output any other information except the paraphrased texts.This is the text:

Intruction 6: As an expert copy-editor, please rewrite the following text in your own voice while ensuring that the final output contains the same information as the original text and has roughly the same length. Please paraphrase all sentences and do not omit any crucial details. Don't output any other information except the paraphrased texts. This is the text:

Intruction 7: Paraphrase the following paragraph such that it preserves the original meaning but uses different phrasing and vocabulary. Ensure that the new version has minimal overlap with the original in terms of common phrases, word sequences, and n-grams. Output should be natural, coherent, and maintain the key information from the source text. Here are the texts:

Intruction 8: Paraphrase the following paragraph such that it preserves the original meaning but uses different phrasing and vocabulary. Ensure that the new version has minimal overlap with the original in terms of common phrases, word sequences, and n-grams. Output should be natural, coherent, and maintain the key information from the source text. Here are the texts:

Intruction 9: Paraphrase the following paragraph such that it preserves the original meaning but has minimal overlap with the original in terms of common phrases, word sequences, and n-grams. Here is the text:

Intruction 10: Paraphrase the following paragraph in your own tone. Ensure that it has minimal overlap with the original in terms of common phrases, word sequences, and n-grams. Here is the texts:

Intruction 11: Rewrite the following paragraph in a way that retains its core meaning but alters its wording and structure. Focus on minimizing shared n-grams and phrases between the original and the rewritten text, while keeping the content clear and coherent. Here are the texts:

Intruction 12: Transform the following paragraph into a new version that conveys the same message but is expressed with different wording and phrasing. Try to keep n-gram overlaps minimal, employing synonyms, rephrased expressions, and varied sentence patterns. Here are the texts:

Intruction 13: Create a paraphrased version of the provided text such that it maintains the semantic essence while minimizing the similarity in wording and n-gram patterns. Focus on using distinct phrases and vocabulary to achieve a high degree of linguistic diversity. Here are the texts:

E Sketch for Optimization Problem Solution

In this section, we present the solution process of the proposed optimization problem 1.

We begin with formally introducing the Slater Constraint Qualification for Convex Inequalities (Bertsekas, 1999, Proposition 3.3.9)

Lemma E.1. *Let x^* be a local minimum of the problem*

$$\begin{aligned} &\min f(x) \\ s.t. \quad & g_i(x) \leq 0, \forall i \\ & h_j(x) = 0, \forall j \end{aligned}$$

,where f and g_i are continuously differentiable. Assume that h_j are linear, g_i are convex and there exists a feasible vector x_0 satisfying

$$g_i(x_0) < 0, \forall i$$

Then x^* satisfies the KKT conditions.

In our proposed optimization problem, the objective function $f = -D_{\mathrm{KL}}(Q||P_w)$ and the inequality $g = D_{\mathrm{KL}}(Q||P_f) - \epsilon$ in Problem 1 involves KL-divergence with the range in $[0, \infty)$. And the equality $h = \sum_{\mathbf{y}} Q(\mathbf{y}) - 1$ is linear. We immediately have the Slater Constraint Qualification satisfied.

Now that we can derive the format of local minimum point Q^* by solving KKT conditions. Based on the Stationarity Condition, for all $\mathbf{y} \in \Sigma^*$, we have

$$\begin{aligned} \frac{\partial L}{\partial Q^*(\mathbf{y})} &= 0 \\ -(\log \frac{Q^*(\mathbf{y})}{P_w(\mathbf{y})} + 1) + \lambda^*(\log \frac{Q^*(\mathbf{y})}{P_f(\mathbf{y}|\mathbf{y}^w)} + 1) &= 0 \\ Q^*(\mathbf{y}) \propto P_f{}^{\frac{1}{1-\lambda^*}}(\mathbf{y}|\mathbf{y}^w) P_w{}^{-\frac{\lambda^*}{1-\lambda^*}}(\mathbf{y}), \end{aligned}$$

which is exactly the statement in Corollary 1.

Further with Complementary Slackness, we have either $\lambda^* = 0$ or $D_{\mathrm{KL}}(Q^*||P_f) = \epsilon$. The former equation indicates a trivial solution, where $Q^* = P_f$. Therefore, we focus on the latter equation, which can be numerically solved via the commonly used Newton-Raphson Method.