Can Watermarks Survive Translation? On the Cross-lingual Consistency of Text Watermark for Large Language Models

Zhiwei He[1*] **Binglin Zhou**[1*] **Hongkun Hao**[1] **Aiwei Liu**[3]
Xing Wang[2†] **Zhaopeng Tu**[2] **Zhuosheng Zhang**[1] **Rui Wang**[1†]
[1]Shanghai Jiao Tong University [2]Tencent AI Lab [3]Tsinghua University
[1]{zwhe.cs,zhoubinglin,zhangzs,wangrui12}@sjtu.edu.cn
[2]{brightxwang,zptu}@tencent.com [3]liuaw20@mails.tsinghua.edu.cn
[Official]:https://github.com/zwhe99/X-SIR [Toolkit]:https://github.com/THU-BPM/MarkLLM

Abstract

Text watermarking technology aims to tag and identify content produced by large language models (LLMs) to prevent misuse. In this study, we introduce the concept of "*cross-lingual consistency*" in text watermarking, which assesses the ability of text watermarks to maintain their effectiveness after being translated into other languages. Preliminary empirical results from two LLMs and three watermarking methods reveal that current text watermarking technologies lack consistency when texts are translated into various languages. Based on this observation, we propose a Cross-lingual Watermark Removal Attack (CWRA) to bypass watermarking by first obtaining a response from an LLM in a pivot language, which is then translated into the target language. CWRA can effectively remove watermarks, decreasing the AUCs to a random-guessing level without performance loss. Furthermore, we analyze two key factors that contribute to the cross-lingual consistency in text watermarking and propose X-SIR as a defense method against CWRA.

1 Introduction

Large language models (LLMs) like GPT-4 (OpenAI, 2023) have demonstrated remarkable content generation capabilities, producing texts that are hard to distinguish from human-written ones. This progress has led to concerns regarding the misuse of LLMs, such as the risks of generating misleading information, impersonating individuals, and compromising academic integrity (Chen and Shu, 2023; Ai et al., 2024; Yuan et al., 2024; Xia et al., 2024). As a countermeasure, text watermarking technology for LLMs has been developed, aiming at tagging and identifying the content produced by LLMs (Kirchenbauer et al., 2023a; Liu et al., 2023). Generally, a text watermarking algorithm embeds a message within LLM-generated content that is imperceptible to human readers, but can be detected algorithmically. By tracking and detecting text watermarks, it becomes possible to mitigate the abuse of LLMs by tracing the origin of texts and ascertaining their authenticity.

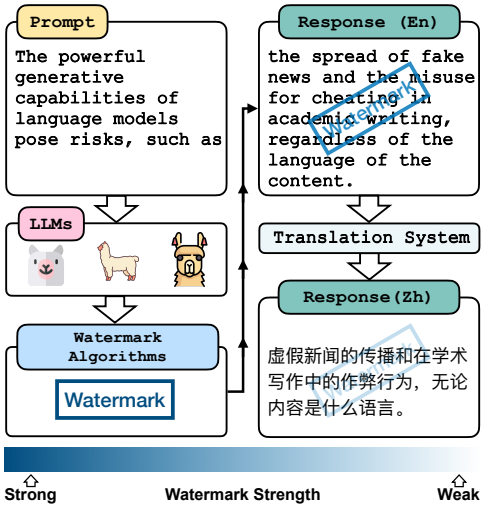


Figure 1: Illustration of watermark dilution in a cross-lingual environment. Best viewed in color.

The robustness of watermarking algorithms, i.e., the ability to detect watermarked text even after it has been modified, is important. Recent works have shown strong robustness under text rewriting and copy-paste attacks (Liu et al., 2024b; Yang et al., 2023b). However, these watermarking techniques have been tested solely within monolingual contexts. In practical scenarios, watermarked texts might be translated (He et al., 2024a,b, 2022; Jiao et al., 2023; Liang et al., 2023), raising questions about the efficacy of text watermarks across languages (see Figure 1). For example, a malicious user could use a watermarked LLM to produce fake news in English and then translate it into Chinese. Obviously, the deceptive impact persists regardless of the language, but it is uncertain whether the watermark would still be detectable after such a

[*]Equal Contribution. Work done during Zhiwei's internship at Tencent AI Lab.
[†]Rui Wang and Xing Wang are co-corresponding authors.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4115–4129

translation. To explore this question, we introduce the concept of *cross-lingual consistency* in text watermarking, aiming to characterize the ability of text watermarks to preserve their strength across languages. Our preliminary results on 2 LLMs $\times$ 3 watermarks reveal that current text watermarking technologies lack consistency across languages.

In light of this finding, we propose the **C**ross-lingual **W**atermark **R**emoval **A**ttack (**CWRA**) to highlight the practical implications arising from deficient cross-lingual consistency. When performing CWRA, the attacker begins by translating the original language prompt into a pivot language, which is fed to the LLM to generate a response in the pivot language. Finally, the response is translated back into the original language. In this way, the attacker obtains the response in the original language and bypasses the watermark with the second translation step. CWRA outperforms re-writing attacks, such as re-translation and paraphrasing (Liu et al., 2023), as it decreases the AUCs to a random-guessing level and achieves the highest text quality.

To resist CWRA, we propose a defense method that improves the cross-lingual consistency of current LLM watermarking. Our method is based on two critical factors. The first is the **cross-lingual semantic clustering of the vocabulary**. Instead of treating each token in the vocabulary as the smallest unit when ironing watermarks, as done by KGW (Kirchenbauer et al., 2023a), our method considers a cluster of tokens that share the same semantics across different languages as the smallest unit of processing. In this way, the post-translated token will still carry the watermark as it would fall in the same cluster as before translation. The second is **cross-lingual semantic robust vocabulary partition**. Inspired by Liu et al. (2024b), we ensure that the partition of the vocabulary are similar for semantically similar contexts in different languages. Despite its limitations, our approach (named X-SIR) substantially elevates the AUCs under the CWRA, paving the way for future research.

Our contributions are summarized as follows:

- **Evaluation** (§ 3): We reveal the deficiency of current text watermarking technologies in maintaining cross-lingual consistency.
- **Attack** (§ 4): Based on this finding, we propose CWRA that successfully bypasses watermarks without degrading the text quality.
- **Defense** (§ 5): We identify two key factors for improving cross-lingual consistency and propose X-SIR as a defense method against CWRA.

2 Background

2.1 Language Model

A language model (LM) M has a defined set of tokens known as its vocabulary $\mathcal{V}$. Given a sequence of tokens $\boldsymbol{x}^{1:n} = (x^1, x^2, \ldots, x^n)$, which we refer to as the *prompt*, the model M computes the conditional probability of the next token over $\mathcal{V}$ as $P_M(x^{n+1}|\boldsymbol{x}^{1:n})$. Therefore, text generation can be achieved through an autoregressive decoding process, where M sequentially predicts one token at a time, forming a *response*. Such an LM can be parameterized by a neural network, such as Transformer (Vaswani et al., 2017), which is called neural LM. Typically, a neural LM computes a vector of logits $\boldsymbol{z}^{n+1} = M(\boldsymbol{x}^{1:n}) \in \mathbb{R}^{|\mathcal{V}|}$ for the next token based on the current sequence $\boldsymbol{x}_{1:n}$ via a neural network. The probability of the next token is then obtained by applying the softmax function to these logits: $P_M(x^{n+1}|\boldsymbol{x}^{1:n}) = \text{softmax}(\boldsymbol{z}^{n+1})$.

2.2 Watermarking for LMs

In this work, we consider the following watermarking methods. All of them embed the watermark by modifying logits during text generation and detect the presence of the watermark for any given text.

KGW (Kirchenbauer et al., 2023a) sets the groundwork for LM watermarking. Ironing a watermark is delineated as the following steps:

(1) compute a hash of $\boldsymbol{x}^{1:n}$: $h^{n+1} = H(\boldsymbol{x}^{1:n})$,
(2) seed a random number generator with h^{n+1} and randomly partitions $\mathcal{V}$ into two disjoint lists: the *green* list $\mathcal{V}_g$ and the *red* list $\mathcal{V}_r$,
(3) adjust the logits $\boldsymbol{z}^{n+1}$ by adding a constant bias δ ($\delta > 0$) for tokens in the green list:

$$\forall i \in \{1, 2, \ldots, |\mathcal{V}|\},$$
$$\tilde{\boldsymbol{z}}_i^{n+1} = \begin{cases} \boldsymbol{z}_i^{n+1} + \delta, & \text{if } v_i \in \mathcal{V}_g, \\ \boldsymbol{z}_i^{n+1}, & \text{if } v_i \in \mathcal{V}_r. \end{cases} \quad (1)$$

As a result, watermarked text will statistically contain more *green tokens*, an attribute unlikely to occur in human-written text. When detecting, one can apply step (1) and (2), and calculate the z-score as the watermark strength of $\boldsymbol{x}$:

$$s = (|\boldsymbol{x}|_g - \gamma|\boldsymbol{x}|)/\sqrt{|\boldsymbol{x}|\gamma(1-\gamma)}, \quad (2)$$

where $|\boldsymbol{x}|_g$ is the number of green tokens in $\boldsymbol{x}$ and $\gamma = \frac{|\mathcal{V}_g|}{|\mathcal{V}|}$. The presence of the watermark can be determined by comparing s with a threshold.

Unbiased watermark (**UW**) views the process of adjusting the logits as applying a Δ function:

$\tilde{\boldsymbol{z}}^{n+1} = \boldsymbol{z}^{n+1} + \Delta$, and designs a Δ function that satisfies:

$$\mathbb{E}\left[\tilde{P}_M\right] = P_M, \tag{3}$$

where $\tilde{P}_M$ is the probability distribution of the next token after logits adjustment (Hu et al., 2023).

Semantic invariant robust watermark (SIR) shows the robustness under re-translation and paraphrasing attack (Liu et al., 2024b). Its core idea is to assign similar Δ for semantically similar prefixes. Given prefix sequences $\boldsymbol{x}$ and $\boldsymbol{y}$, SIR adopts an embedding model E to characterize their semantic similarity and trains a watermark model that yields Δ with the main objective:

$$\mathcal{L} = |\text{Sim}(E(\boldsymbol{x}), E(\boldsymbol{y})) - \text{Sim}(\Delta(\boldsymbol{x}), \Delta(\boldsymbol{y}))|, \tag{4}$$

where $\text{Sim}(\cdot,\cdot)$ denotes similarity function. Furthermore, $\forall i \in \{1, 2, \ldots, |\mathcal{V}|\}$, Δ_i is trained to be close to $+1$ or -1. Therefore, SIR can be seen as an improvement based on KGW, where $\Delta_i > 0$ indicating that v_i is a green token. The original implementation of SIR uses C-BERT (Chanchani and Huang, 2023) as the embedding model, which is English-only. To adopt SIR in the cross-lingual scenario, we use a multilingual S-BERT (Reimers and Gurevych, 2019)[1] instead.

3 Cross-lingual Consistency of Text Watermark

In this section, we define the concept of cross-lingual consistency in text watermarking and answer three research questions (RQ):

- **RQ1**: To what extent are current watermarking algorithms consistent across different languages?
- **RQ2**: Do watermarks exhibit better consistency between similar languages than between distant languages?
- **RQ3**: Does semantic invariant watermark (SIR) exhibit better cross-lingual consistency than others (KGW and UW)?

3.1 Definition

We define cross-lingual consistency as the ability of a watermark, embedded in a text produced by an LLM, to retain its strength after the text is translated into another language. We represent the original strength of the watermark as a random variable, denoted by S (Appendix A), and its strength after translation as $\hat{S}$. To quantitatively assess this consistency, we employ the following two metrics.

[1] paraphrase-multilingual-mpnet-base-v2

Pearson Correlation Coefficient (PCC) We use PCC to assess linear correlation between S and $\hat{S}$:

$$\text{PCC}(S, \hat{S}) = \frac{\text{cov}(S, \hat{S})}{\sigma_S \sigma_{\hat{S}}}, \tag{5}$$

where $\text{cov}(S, \hat{S})$ is the covariance and σ_S and $\sigma_{\hat{S}}$ are the standard deviations. A PCC value close to 1 suggests consistent trends in watermark strengths across languages.

Relative Error (RE) Unlike PCC, which captures consistency in trends, RE is used to assess the magnitude of deviation between S and $\hat{S}$:

$$\text{RE}(S, \hat{S}) = \mathbb{E}\left[\frac{|\hat{S} - S|}{|S|}\right] \times 100\%. \tag{6}$$

A lower RE indicates that the watermark retains strength close to its original value after translation, signifying great cross-lingual consistency.

3.2 Experimental Setup

Setup We sampled a subset of 500 English prompts from the mc4 dataset (Raffel et al., 2019)[2], and generated responses from the LLM using the text watermarking methods described in § 2.2. The default decoding method was multinomial sampling, and both the prompts and the LLM-generated responses were in English. To evaluate the cross-lingual consistency, these watermarked responses were translated into four languages using `gpt-3.5-turbo-0613`[3]: Chinese (Zh), Japanese (Ja), French (Fr), and German (De). Notably, English shares greater similarities with French and German, in contrast to its significant differences from Chinese and Japanese.

Models For the LLMs, we adopt:

- BAICHUAN-7B (Baichuan., 2023): an LLM trained on 1.2 trillion tokens. It offers bilingual support for both Chinese and English officially. We also found that its vocabulary covers Japanese tokens.
- LLAMA-2-7B (Touvron et al., 2023): trained on 2 trillion tokens and only provides support for English officially. Its vocabulary also contains tokens for European languages, such as German and French.

[2] https://huggingface.co/datasets/mc4
[3] https://platform.openai.com/docs/models

Method	PCC ↑					RE (%) ↓				
	En→Zh	En→Ja	En→Fr	En→De	Avg.	En→Zh	En→Ja	En→Fr	En→De	Avg.
BAICHUAN-7B										
KGW	0.074	0.039	0.034	0.056	0.051	87.30	**91.51**	97.78	**95.35**	92.99
UW	0.178	0.139	0.135	0.099	0.138	96.92	96.62	98.32	98.60	97.62
SIR	**0.371**	**0.294**	**0.244**	**0.226**	**0.284**	**75.11**	95.36	**91.83**	99.29	**90.40**
LLAMA-2-7B										
KGW	0.113	0.108	0.117	0.065	0.101	**85.08**	88.15	91.41	95.45	90.02
UW	0.206	0.206	**0.210**	**0.139**	0.190	102.08	94.35	94.38	97.83	97.16
SIR	**0.267**	**0.259**	0.186	0.137	**0.212**	91.51	**73.56**	**91.39**	**78.79**	**83.81**

Table 1: Comparison of cross-lingual consistency between different text watermarking methods (KGW, UW, and SIR). **Bold** entries denote the best result among the three methods.

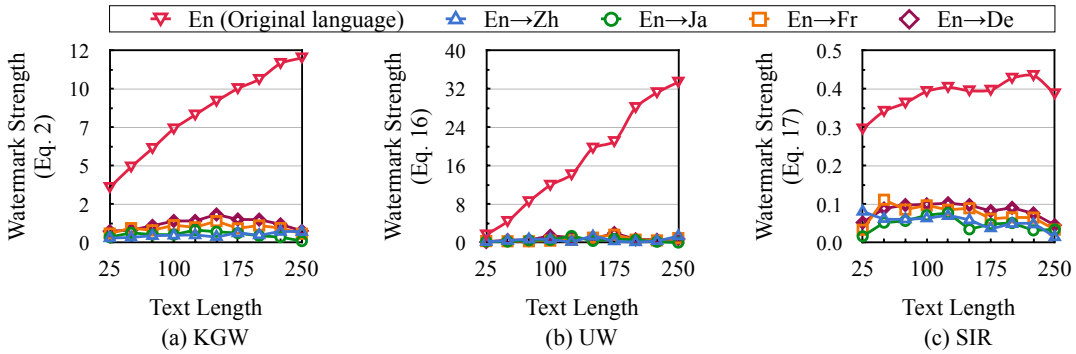


Figure 2: Trends of watermark strengths with text length before and after translation. These are the average results of BAICHUAN-7B and LLAMA-2-7B. Given the distinct calculations for watermark strengths of the three methods, the y-axis scales vary accordingly.

3.3 Results

Table 1 presents the main results. We forced the model to generate 200 tokens in response following the setting of Kirchenbauer et al. (2023a). With response length restriction lifted, Figure 2 illustrates the trend of watermark strengths with text length.

Results for RQ1 We reveal a notable deficiency in the cross-lingual consistency of current watermarking methods. Among all the settings, the PCCs are generally less than 0.3, and the REs are predominantly above 80%. Furthermore, Figure 2 visually demonstrates that the watermark strengths of the three methods exhibit a significant decrease after translation. These results suggest that current watermarking algorithms struggle to maintain effectiveness across language translations.

Results for RQ2 None of the three watermarking methods exhibits such a characteristic that its cross-lingual consistency between similar languages is significantly better than distant ones. This means that even if two languages have similar structures or shared words, it is still difficult for watermarks to transfer between them, which poses a big challenge to the cross-lingual consistency of watermarking.

Results for RQ3 Overall, SIR indeed exhibits superior cross-lingual consistency compared to KGW and UW. It achieves the best average results across the two models and two metrics. When using BAICHUAN-7B, SIR notably outperforms other methods in terms of PCCs for all target languages. This finding highlights the importance of semantic invariance in preserving watermark strength across languages, which we will explore more in § 5. Despite its superiority, SIR still presents a notable reduction in watermark strength in cross-lingual scenarios, as evidenced by Figure 2c.

4 Cross-lingual Watermark Removal Attack

In the previous section, we focus on scenarios where the response of LLM is translated into other languages. However, an attacker typically expects a response from the LLM in the same language as the prompt while removing watermarks. To bridge this gap, we introduce the Cross-lingual Watermark Removal Attack (CWRA) in this section, constituting a complete attack process and posing a more significant challenge to text watermarking than paraphrasing and re-translation attacks.

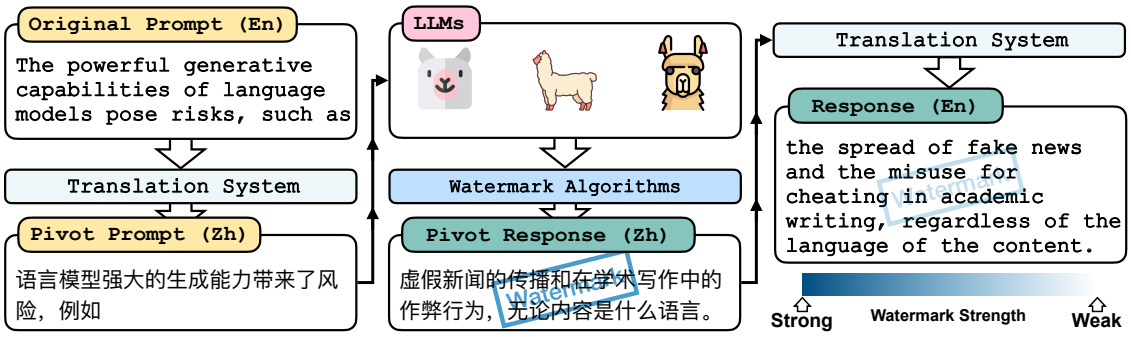


Figure 3: An example pipeline of CWRA with English (En) as the original language and Chinese (Zh) as the pivot language. When performing CWRA, the attacker not only wants to remove the watermark, but also gets a response in the original language with high quality. Its core idea is to wrap the query to the LLM into the pivot language.

Figure 3 shows the process of CWRA. Instead of feeding the original prompt into the LLM, the attacker initiates the attack by translating the prompt into a pivot language named the pivot prompt. The LLM receives the pivot prompt and provides a watermarked response in the pivot language. The attacker then translates the pivot response back into the original language. This approach allows the attacker to obtain the response in the original language. Due to the inherent challenges in maintaining cross-lingual consistency, the watermark would be effectively eliminated during the second translation step.

4.1 Setup

To assess the practicality of attack methods, we consider two downstream tasks: text summarization and question answering. We adopt MultiNews (Fabbri et al., 2019) and ELI5 (Fan et al., 2019) as test sets, respectively. Both datasets are in English and require long text output with an average output length of 198 tokens. We selected 500 samples for each test set that do not exceed the maximum context length of the model and performed zero-shot prompting on BAICHUAN-7B. For CWRA, we select Chinese as the pivot language and compare the following two methods:

- **Paraphrase**: rephrasing the response into different wording while retaining the same meaning.
- **Re-translation**: translating the response into the pivot language and back to the original language.

The paraphraser and translator used in all attack methods are `gpt-3.5-turbo-0613` to ensure consistency across the different attack methods.

4.2 Results

Figure 4 exhibits ROC curves of three watermarking methods under different attack methods.

CWRA vs Other Attack Methods CWRA demonstrates the most effective attack performance, significantly diminishing the AUCs and the TPRs. For one thing, existing watermarking techniques are not designed for cross-lingual contexts, leading to weak cross-lingual consistency. For another thing, strategies such as Re-translation and Paraphrase are essentially semantic-preserving text rewriting. Such strategies tend to preserve some n-grams from the original response, which may still be identifiable by the watermark detection algorithm. In contrast, CWRA reduces such n-grams due to language switching.

SIR vs Other Watermarking Methods Under the CWRA, SIR exhibits superior robustness compared to other watermarking methods. The AUCs for KGW and UW under CWRA plummet to 0.61 and 0.54, respectively, approaching the level of random guessing. In stark contrast, the AUC for the SIR method stands significantly higher at 0.67, aligning with our earlier observations regarding cross-lingual consistency in the RQ3 of § 3.3.

Text Quality As shown in Table 2, these attack methods not only preserve text quality, but also bring slight improvements in most cases due to the good translator and paraphraser. Among the compared methods, CWRA stands out for its superior performance. Considering that the same translator and paraphraser were used across all methods, we speculate that this is because the BAICHUAN-7B model used in our experiments performs even better in the pivot language (Chinese) than in the original language (English). This finding implies that a potential attacker could strategically choose a pivot language at which the LLM excels to perform CWRA, thereby achieving the best text quality while removing the watermark.

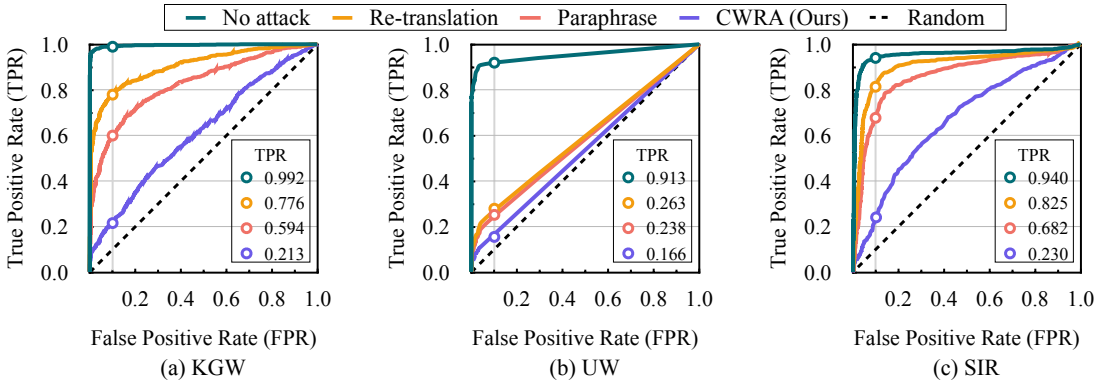


Figure 4: ROC curves for KGW, UW, and SIR under various attack methods: Re-translation, Paraphrase and CWRA. We also present TPR values at a fixed FPR of 0.1. This is the overall result of text summarization and question answering. Figure 9a and Figure 9b display results for each task.

Attack \ WM	KGW			UW			SIR		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Text Summarization									
No attack	14.24	2.68	12.99	13.65	1.68	12.38	13.34	1.79	12.43
Re-translation	14.11	2.43	12.89	13.89	1.77	12.63	13.63	1.98	12.61
Paraphrase	15.10	2.49	13.69	14.72	1.95	13.31	15.56	2.11	14.14
CWRA (Ours)	**18.98**	**3.63**	**17.33**	**15.88**	**2.31**	**14.25**	**17.38**	**2.67**	**15.79**
Question Answering									
No attack	**19.00**	2.18	16.09	11.70	0.49	9.57	16.95	1.35	14.91
Re-translation	18.62	2.32	16.39	12.98	1.30	11.16	16.90	1.80	15.12
Paraphrase	18.45	2.24	**16.47**	14.38	1.37	13.07	17.17	1.79	**15.54**
CWRA (Ours)	18.23	**2.56**	16.27	**15.20**	**1.88**	**13.45**	**17.47**	**2.22**	15.53

Table 2: Comparative analysis of text quality impacted by different watermark removal attacks.

5 Improving Cross-lingual Consistency

Up to this point, we have observed the challenges associated with text watermarking in cross-lingual scenarios. In this section, we first analyze two key factors essential for achieving cross-lingual consistency. Based on our analysis, we propose a defense method against CWRA.

5.1 Two Key Factors of Cross-lingual Consistency

KGW-based watermarking methods fundamentally depend on the partition of the vocabulary, i.e., the red and green lists, as discussed in § 2.2. Therefore, cross-lingual consistency aims to achieve the following goal:

> *The green tokens in the watermarked text will still be recognized as green tokens after being translated into other languages.*

With this goal in mind, we start our analysis with a toy case in Figure 5:

1. We define two simple languages. Language 1 (Lang1) consists of hollow tokens: □, ☆ and ♡. Language 2 (Lang2) consists of solid tokens: ■, ★ and ♥. Tokens with the same shape are semantically equivalent.
2. Given □ as the prefix, a watermarked LM selects ☆ from [☆, ★, ♡, ♥][4] as the next token. Due to watermarking, ☆ is a green token.
3. A machine translator (MT) then translates the entire sentence "□ ☆" into Lang2: "■ ★".

The question of interest is: what conditions must the vocabulary partition satisfy so that the token ★, the semantic equivalent of ☆, is also included in the green list?

Figure 5(a) illustrates a successful case, where two key factors exists:

1. **Cross-lingual semantic clustering of the vocabulary**: semantically equivalent tokens must be in the same partition, either green or red lists.
2. **Cross-lingual semantic robust vocabulary partition**: for semantically equivalent prefixes in different languages: □ and ■, the partitions

[4]We suppose that this LM is bilingual and omit the prefix tokens □ and ■ for simplicity.

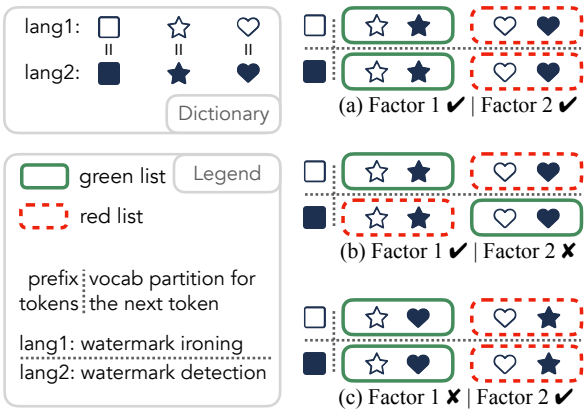


Figure 5: Three cases of attempts to maintain cross-lingual consistency. Cross-lingual consistency can only be achieved if ★ is in the green list when watermark detection. **Factor 1**: semantically equivalent tokens should be in the same list (either red or green). In these cases, ☆ = ★, and ♡ = ♥. **Factor 2**: the vocabulary partitions for semantically equivalent prefixes (□ and ■) should be the same.

of the vocabulary must be the same.

Both Figure 5(b) and Figure 5(c) satisfy only one of the two factors, thus failing to recognize ★ as a green token and losing cross-lingual consistency.

5.2 Defense Method against CWRA

We now improve the SIR so that it satisfies the two factors described above. In Figure 5, the tokens of the two languages are equivalent in one-to-one correspondence. In practice, we relax this semantic equivalence into semantic similarity.

As discussed in § 2.2, SIR uses the Δ function to represent vocabulary partition ($\Delta \in \mathbb{R}^{|\mathcal{V}|}$), where $\Delta_i > 0$ indicating that v_i is a green token. Based on Eq. 4, SIR has already optimized for Factor 2 when using a multilingual embedding model. For prefixes $\boldsymbol{x}$ and $\boldsymbol{y}$, the similarity of their vocabulary partitions for the next token should be close to their semantic similarity:

$$\mathrm{Sim}(\Delta(\boldsymbol{x}), \Delta(\boldsymbol{y})) \approx \mathrm{Sim}(E(\boldsymbol{x}), E(\boldsymbol{y})), \quad (7)$$

where E is a multilingual embedding model.

Based on SIR, we focus on Factor 1, i.e., cross-lingual semantic clustering of the vocabulary. Formally, we define semantic clustering as a partition $\mathcal{C}$ of the vocabulary $\mathcal{V}$: $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{|\mathcal{C}|}\}$, where each cluster $\mathcal{C}_k$ consists of semantically similar tokens. Instead of assigning biases for each token in $\mathcal{V}$, we adapt the Δ function so that it yields biases to each cluster in $\mathcal{C}$, i.e., $\Delta \in \mathbb{R}^{|\mathcal{C}|}$. Thus, the process of adjusting the logits should be:

$$\begin{aligned} &\forall i \in \{1, 2, \ldots, |\mathcal{V}|\}, \\ &\tilde{\boldsymbol{z}}_i^{n+1} = \boldsymbol{z}_i^{n+1} + \Delta_{C(i)}, \end{aligned} \quad (8)$$

where $C(i)$ indicates the index of v_i's cluster within $\mathcal{C}$. By doing so, if token v_i and v_j are semantically similar, they will receive the same bias on logits:

$$C(i) = C(j) \implies \Delta_{C(i)} = \Delta_{C(j)}. \quad (9)$$

In other words, if v_i and v_j are translations of each other, they will fall into the same list. Algorithm 1 shows the procedure of semantic clustering. To obtain such a semantic clustering $\mathcal{C}$, we treat each token in $\mathcal{V}$ as a node, and add an edge (v_i, v_j) whenever (v_i, v_j) corresponds to an entry in a bilingual dictionary $\mathcal{D}$. Therefore, $\mathcal{C}$ is all the connected components of this graph.

Algorithm 1 Constructing semantic clusters

Require: the vocabulary of the LM $\mathcal{V}$, a bilingual dictionary $\mathcal{D}$
Ensure: Semantic clusters $\mathcal{C}$

```
Initialize a graph G = (V, ∅)
for each entry (v_i, v_j) in D do
    if v_i ∈ V and v_j ∈ V then
        Add an edge (v_i, v_j) to G
C = ∅
for each connected component C_k in G do
    C = C ∪ {C_k}
return C
```

Setup We name this method as X-SIR, implemented it with an unified external dictionary $\mathcal{D}$ that covers English (En), Chinese (Zh), Japanese (Ja), French (Fr), and German (De). Following the settings of § 3.2 and § 4.1, we apply X-SIR on BAICHUAN-7B, analyze it in the following and detail its limitations in § 8. We also present more results on other LLMs in Appendix C.

Cross-lingual consistency & ROC curves Figure 6 shows the cross-lingual consistency of SIR and X-SIR when the original responses are translated into other languages. Considering both PCC and RE, X-SIR notably improves the cross-lingual consistency over SIR when the target language is Zh or Ja. However, for De and Fr, X-SIR improves only the PCC but not RE, which we regard as a limitation and discuss in § 8. Further, Figure 7

evaluates both methods' watermark detection performance under cross-lingual scenarios. Whether directly translating the responses into other languages (Figure 7a) or using CWRA (Figure 7b), X-SIR substantially enhances the watermark detection performance, with an average increase in TPR by 0.448 and AUC by 0.221. These findings validate the two key factors of cross-lingual consistency that we identified in § 5.1.

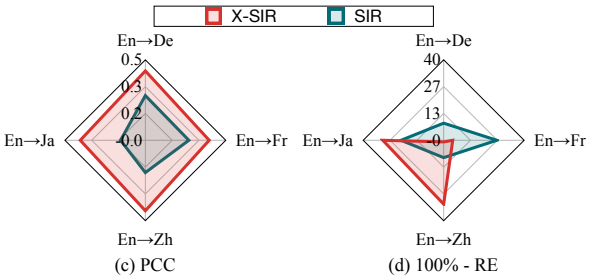


Figure 6: Cross-lingual consistency in terms of PCC and RE. We follow the same setting as § 3.2 and plot 100%-RE for visualization.

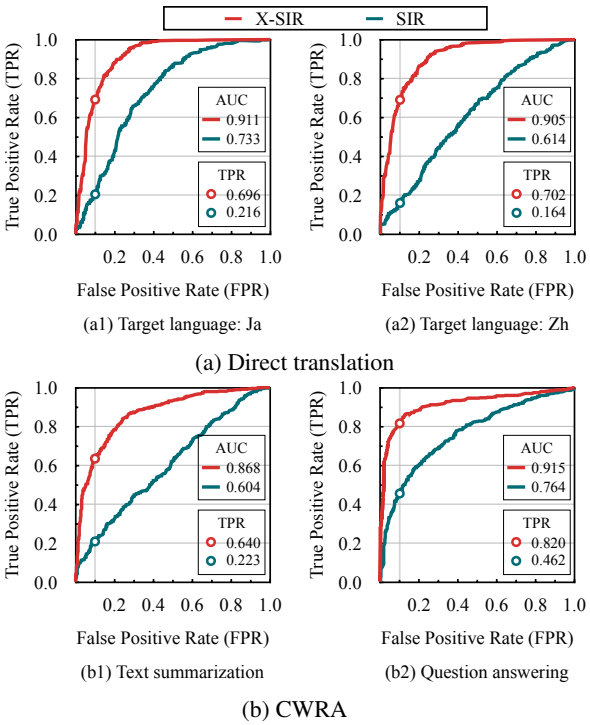


Figure 7: Watermark detection performance under cross-lingual scenarios. (a) Direct translation: the original English responses are translated into Ja or Zh following the setting in § 3.2. (b) CWRA: the full CWRA process as we did in § 4, where English is the original language and Chinese is the pivot language.

No attack & Paraphrase attack The next question is whether the addition of semantic clustering of the vocabulary to X-SIR will affect the original detection performance of SIR. As shown in Figure 8, when there is no attack, both methods perform comparably. However, under the paraphrase attack, X-SIR performs slightly better than SIR. This is reasonable because paraphrasing can be regarded as a special kind of "translation" within the same language and X-SIR improves such intra-lingual consistency.

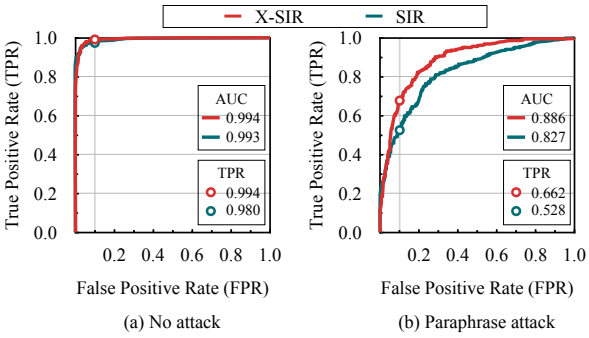


Figure 8: Watermark detection performance under no attack and paraphrase attack.

Text quality As shown in Table 3, X-SIR achieves better text quality than SIR in text summarization, and comparable performance on question answering, meaning the semantic clustering of vocabulary will not negatively affect text quality.

Method	ROUGE-1	ROUGE-2	ROUGE-L
Text Summarization			
SIR	13.34	1.79	12.43
X-SIR	**15.65**	**2.04**	**14.29**
Question Answering			
SIR	**16.95**	1.35	**14.91**
X-SIR	16.77	**1.39**	14.07

Table 3: Effects of X-SIR and SIR on text quality.

6 Related Work

6.1 LLM Watermarking

Text watermarking aims to embed a watermark into a text and detect the watermark for any given text. Currently, text watermark method can be classified into two categories (Liu et al., 2023): watermarking for existing text and watermarking for generated text. In this work, we focus on the latter, which is more challenging and has more practical applications.

This type of watermark method usually can be illustrated as the watermark ironing process (modifying the logits of the LLM during text generation) and watermark detection process (assess the presence of watermark by a calculated watermark strength score). Kirchenbauer et al. (2023a) introduces KGW, the first watermarking method for LLMs. Hu et al. (2023) proposes UW without affecting the output probability distribution compared to KGW. Liu et al. (2024b) introduces SIR, a watermarking method taking into account the semantic information of the text, which shows robustness to text re-writing attacks. Liu et al. (2024a) proposes the first unforgeable and publicly verifiable watermarking algorithm for LLMs. SemStamp (Hou et al., 2023) is another semantic-related watermarking method and it generate watermarked text at sentence granularity instead of token granularity. Tu et al. (2023) introduces WaterBench, the first comprehensive benchmark for LLM watermarks.

6.2 Watermark Robustness

A good watermarking method should be robust to various watermarking removal attacks. However, current works on watermarking robustness mainly focus on single-language attacks, such as paraphrase attacks. For example, Kirchenbauer et al. (2023b) evaluates the robustness of KGW against paraphrase attacks as well as copy-paste attacks and proposes a detect trick to improve the robustness to copy-paste attacks. Zhao et al. (2023) employs a fixed green list to improve the robustness of KGW against paraphrase attacks and editing attacks. Chen et al. (2023) proposes a new paraphrase robust watermarking method "XMark" based on "text redundancy" of text watermark. Lu et al. (2024) proposes an entropy-based text watermarking detection method that achieves better detection performance in low-entropy scenarios.

7 Conclusion

This work aims to investigate the cross-lingual consistency of watermarking methods for LLMs. We first characterize and evaluate the cross-lingual consistency of current watermarking techniques, revealing that current watermarking methods struggle to maintain their watermark strengths across different languages. Based on this observation, we propose the cross-lingual watermark removal attack (CWRA), which significantly challenges watermark robustness by efficiently eliminating watermarks without compromising text quality. Through the analysis of two primary factors that influence cross-lingual consistency, we propose X-SIR as a defense strategy against CWRA. Despite its limitations, this approach greatly improves watermark detection performance under cross-lingual scenario and paves the way for future research. Overall, this work completes a closed loop in the study of cross-lingual consistency in watermarking, including: evaluation, attacking, analysis, and defensing.

Since our first release, the X-SIR algorithm has been integrated into MarkLLM (Pan et al., 2024), an open-source toolkit for LLM Watermarking.

8 Limitations

X-SIR relies on the semantic clustering described in Algorithm 1 which only considers tokens shared by the vocabulary $\mathcal{V}$ of the model and the external dictionary $\mathcal{D}$. This design results in the following limitations:

- **Language coverage**: X-SIR only supports languages supported by the model. However, in a real scenario, an attacker can choose the original and the pivot language at will. This limitation has revealed by Figure 6, where X-SIR brings a much more significant boost to Ja & Zh than to De & Fr since the vocabulary of the LLM (BAICHUAN-7B) supports Zh & Ja better.
- **Vocab coverage**: Since the external dictionary $\mathcal{D}$ only contains whole words, word units can not be clustered in Algorithm 1 and will left as isolated nodes. Consequently, X-SIR's performance might be compromised if the tokenizer favors finer-grained token segmentation.

X-SIR does not solve the issue of cross-lingual consistency but sets the stage for future research.

Acknowledgements

This paper is mainly supported by the Tencent AI Lab Fund (RBFR2024002). Zhiwei and Rui are partially supported by the National Natural Science Foundation of China (62176153) and the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102, as the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University). Zhuosheng is partially supported by the Joint Funds of the National Natural Science Foundation of China (Grant No. U21B2020).

References

Yiming Ai, Zhiwei He, Ziyin Zhang, Wenhong Zhu, Hongkun Hao, Kai Yu, Lingjun Chen, and Rui Wang. 2024. Is cognition and action consistent or not: Investigating large language model's personality. *arXiv preprint arXiv:2402.14679*.

Baichuan. 2023. A large-scale 7b pretraining language model developed by baichuan-inc.

Sachin Chanchani and Ruihong Huang. 2023. Composition-contrastive learning for sentence embeddings. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15836–15848, Toronto, Canada. Association for Computational Linguistics.

Canyu Chen and Kai Shu. 2023. Can llm-generated misinformation be detected? *arXiv preprint arXiv:2309.13788*.

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.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Zhiwei He, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2024a. Exploring Human-Like Translation Strategy with Large Language Models. *Transactions of the Association for Computational Linguistics*, 12:229–246.

Zhiwei He, Xing Wang, Wenxiang Jiao, Zhuosheng Zhang, Rui Wang, Shuming Shi, and Zhaopeng Tu. 2024b. Improving machine translation with human feedback: An exploration of quality estimation as a reward model. *arXiv preprint arXiv:2401.12873*.

Zhiwei He, Xing Wang, Rui Wang, Shuming Shi, and Zhaopeng Tu. 2022. Bridging the data gap between training and inference for unsupervised neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6611–6623, Dublin, Ireland. Association for Computational Linguistics.

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.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Wenxiang Jiao, Jen-tse Huang, Wenxuan Wang, Zhiwei He, Tian Liang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. ParroT: Translating during chat using large language models tuned with human translation and feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15009–15020, Singapore. 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*, volume 202 of *Proceedings of Machine Learning Research*, 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.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Aiwei Liu, Leyi Pan, Xuming Hu, Shuang Li, Lijie Wen, Irwin King, and Philip S. Yu. 2024a. An unforgeable publicly verifiable watermark for large language models. In *The Twelfth International Conference on Learning Representations*.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2024b. A semantic invariant robust watermark for large language models. In *The Twelfth International Conference on Learning Representations*.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Lijie Wen, Irwin King, and Philip S Yu. 2023. A survey of text watermarking in the era of large language models. *arXiv preprint arXiv:2312.07913*.

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

OpenAI. 2023. Gpt-4 technical report.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. 2023. Waterbench: Towards holistic evaluation of watermarks for large language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Tian Xia, Zhiwei He, Tong Ren, Yibo Miao, Zhuosheng Zhang, Yang Yang, and Rui Wang. 2024. Measuring bargaining abilities of llms: A benchmark and a buyer-enhancement method. *arXiv preprint arXiv:2402.15813*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023a. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Xi Yang, Kejiang Chen, Weiming Zhang, Chang Liu, Yuang Qi, Jie Zhang, Han Fang, and Nenghai Yu. 2023b. Watermarking text generated by black-box language models.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, Rui Wang, and Gongshen Liu. 2024. R-judge: Benchmarking safety risk awareness for llm agents. *arXiv preprint arXiv:2401.10019*.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable robust watermarking for ai-generated text.

A Details of Watermarking Methods

As discussed in § 6.1, we focus on the watermarking methods for large language models (LLMs). A watermarking method can be divided into two processes: the watermark ironing process and the watermark detection process. In ironing process, the watermark is embedded into the text by modifying the logits of the LLM during text generation. In detection process, the watermark detector calculates the **watermark strength score** s to assess the presence of watermark. s is a scalar value to indicate the strength of the watermark in the text. For any given text, we can calculate its watermark strength score s based on detection process of the watermarking method. A higher s indicates that the text is more likely to contain watermark. In the opposite, a lower s indicates that the text is less likely to contain watermark. Every watermarking method has its own way to ironing the watermark and calculate the watermark strength score s. We detail KGW, UW and SIR in the following sections.

A.1 KGW

In § 2.2, we introduce the processes of watermark ironing and watermark detection in the KGW method. Here, we detail the experimental settings employed for KGW. KGW uses a hash function H to compute the hash of the previous k tokens. In this work, we adhere to the experimental setting reported by Kirchenbauer et al. (2023a), employing the hash function **minhash** with $k = 4$. The ratio of green token lists $\mathcal{V}_g$ to the total word list $\mathcal{V}$ is set at $\gamma = 0.25$. Additionally, the constant bias δ is fixed at 2.0.

A.2 UW

The ironing process in UW is analogous to that in KGW; however, the two differ in their respective function of modifying the logits. Here is the detail watermark ironing process in UW:

(1) compute a hash of $\boldsymbol{x}^{1:n}$: $h^{n+1} = H(\boldsymbol{x}^{1:n})$, and use h^{n+1} as seed generating a random number $p \in [0, 1)$.

(2) determine the token t satisfies:

$$p \in \left[\sum_{i=1}^{t-1} P_{Mi}(x^{n+1}|\boldsymbol{x}^{1:n}), \sum_{i=1}^{t} P_{Mi}(x^{n+1}|\boldsymbol{x}^{1:n}) \right) \tag{10}$$

(3) set $P_{Mi}(x^{n+1}|\boldsymbol{x}^{1:n}) = 0$ for $i \neq t$ and $P_{Mt}(x^{n+1}|\boldsymbol{x}^{1:n}) = 1$.

Then we get the adjusted probability of next token $\tilde{P}_M(x^{n+1}|\boldsymbol{x}^{1:n})$.

The detection process calculates a maximin variant Log Likelihood Ratio (LLR) of the detected text to assess the watermark strength score. Log Likelihood Ratio (LLR) is defined as:

$$r_i = \frac{\tilde{P}_M(x^i|\boldsymbol{x}^{1:i-1})}{P_M(x^i|\boldsymbol{x}^{1:i-1})} \tag{11}$$

The total score is defined as:

$$R = \frac{\tilde{P}_M(\boldsymbol{x}^{\boldsymbol{a+1:n}}|\boldsymbol{x}^{1:a})}{P_M(\boldsymbol{x}^{\boldsymbol{a+1:n}}|\boldsymbol{x}^{1:a})} \tag{12}$$

Where $\boldsymbol{x}^{\boldsymbol{1:a}}$ is prompt and $\boldsymbol{x}^{\boldsymbol{a+1:n}}$ is the detected text. Let

$$P_i = P_M(x^i|\boldsymbol{x}^{\boldsymbol{1:i-1}}) \tag{13}$$

$$Q_i = \tilde{P}_M(x^i|\boldsymbol{x}^{\boldsymbol{1:i-1}}) \tag{14}$$

$$R_i = (r_i(x_1), r_i(x_2), \cdots, r_i(x_{|\mathcal{V}|})) \tag{15}$$

Where $r_i(x_k)$ is the LLR of token x_k at position i. UW use a maximin variant LLR (Hu et al., 2023) to avoid the limitation of the origin LLR. The calculating process of maximin variant LLR can be formulated as follows:

$$\max_{R_i} \min_{Q_i' \in \Delta_{\mathcal{V}}, TV(Q_i', Q_i) \leq d} \langle Q_i', R_i \rangle, \quad \text{s.t. } \langle P_i, \exp(R_i) \rangle \leq 1 \tag{16}$$

Where $\Delta_{\mathcal{V}}$ is the set of all probability distributions over the symbol set $\mathcal{V}$, and TV is the total variation distance, d is a hyperparameter to control TV, and $\langle \cdot, \cdot \rangle$ is the inner product. UW utilizes the maximin variant LLR to calculate the watermark strength score.

In the experiments, we follow the experiment settings of original paper, using the previous 5 tokens to compute the hash and set $d = 0$.

A.3 SIR

As introduced in § 2.2, the ironing process in SIR assigns a watermark bias, Δ_i, to every token v_i.

For any given text, the watermark detector calculates the mean watermark bias to determine the watermark strength score. Consider the detected text represented as $\boldsymbol{x} = (x^1, \ldots, x^N)$. The watermark strength score, s, can be expressed by the following equation:

$$s = \frac{\sum_{n=1}^{N} \Delta_{I(x^n)}(\boldsymbol{x}^{1:n-1})}{N}, \tag{17}$$

where $I(x^n)$ indicates the index of the token x^n within the vocabulary $\mathcal{V}$, and $\Delta_{I(x^n)}(\boldsymbol{x}^{1:n-1})$ denotes the watermark bias for token x^n at position n. Given that the watermark bias satisfies the unbiased property, $\sum_{t\in\mathcal{V}} \Delta_{I(t)} = 0$, the expected detection score for normal text is 0. Consequently, the detection score for watermarked text should exceed 0 significantly.

B Full Result of Watermark Removal Attack

Figure 9 presents the full results of watermark removal attacks. It is equivalent to Figure 4, but presents results on text summarization and question answering separately.

C Full Result of X-SIR

Figure 10 presents full watermark detection performance (ROC curves, AUCs and TPRs) of X-SIR. We tested it on the following LLMs:

- BAICHUAN-7B (Baichuan., 2023)[5]
- BAICHUAN2-7B-BASE (Yang et al., 2023a)[6]
- LLAMA-2-7B (Touvron et al., 2023)[7]
- MISTRAL-7B-V0.1 (Jiang et al., 2023)[8].

We tested it under the following attack methods:

- No attack
- Paraphrase attack (§ 4.1)
- Direct translation (En→XX): translating original English responses to other languages.

We also plot the performance of SIR as baselines.

[5] https://huggingface.co/baichuan-inc/Baichuan-7B

[6] https://huggingface.co/baichuan-inc/Baichuan2-7B-Base

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

[8] https://huggingface.co/mistralai/Mistral-7B-v0.1

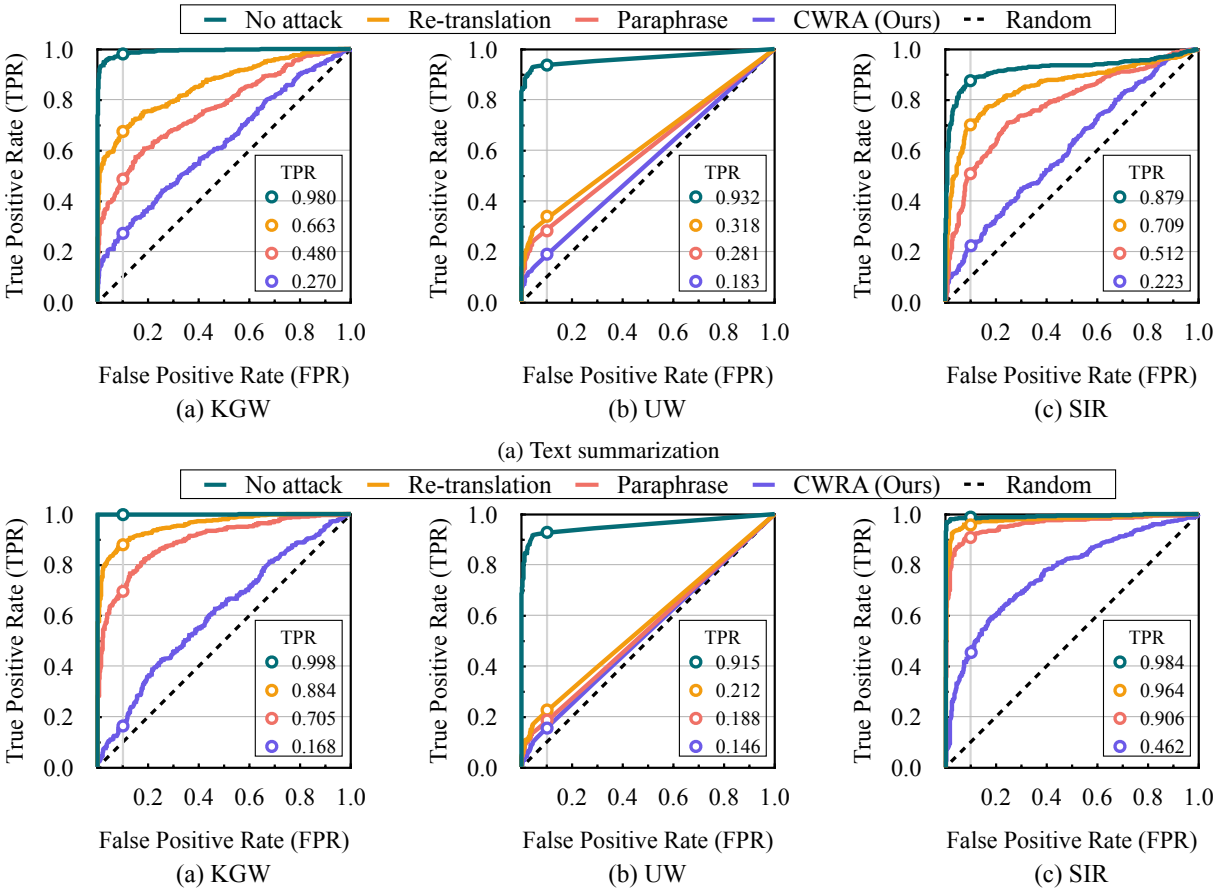
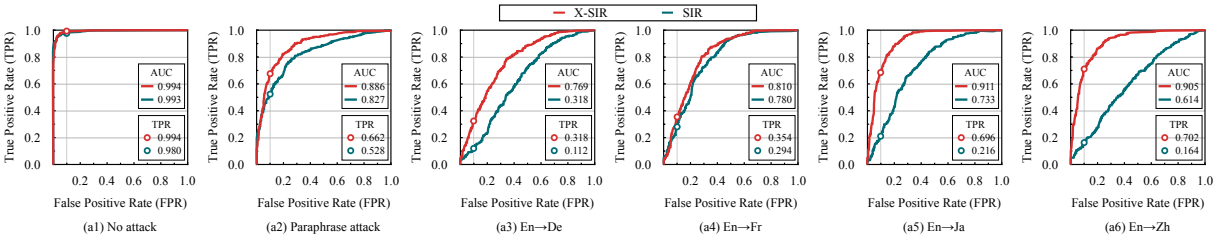
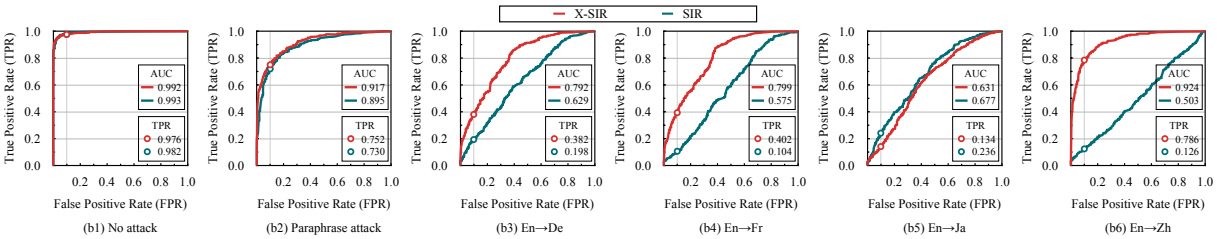


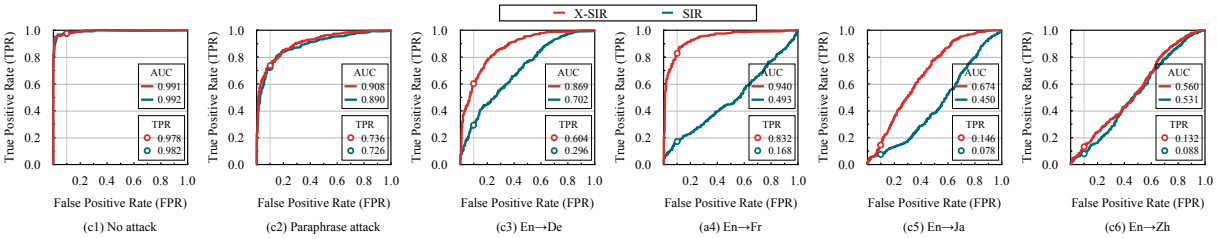
Figure 9: ROC curves for KGW, UW and SIR under various attack methods: Re-translation, Paraphrase and CWRA. We also present TPR values at a fixed FPR of 0.1.



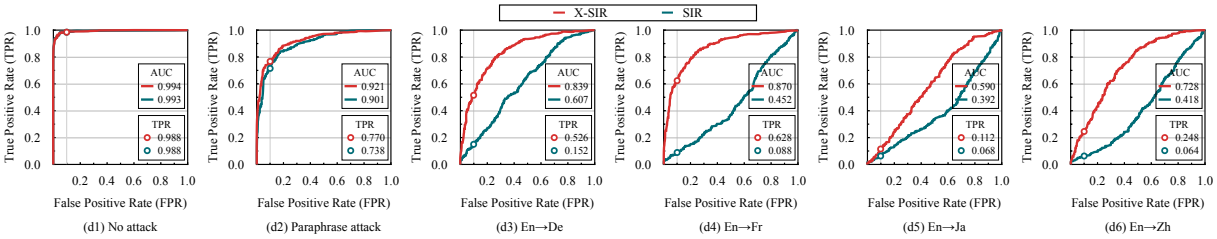
(a) BAICHUAN-7B



(b) BAICHUAN2-7B-BASE



(c) LLAMA-2-7B



(d) MISTRAL-7B-V0.1

Figure 10: ROC curves for X-SIR and SIR under various attack methods: no attack, paraphrase and direct translation. We also present AUC and TPR values at a fixed FPR of 0.1.