SCTI: Self-Calibrated Trident Identification of Black-Box LLM Watermarks

Zhixiong Nan[1], **Haoyu Lu**[1], **Tao Xiang**[1,*], **Yiwei Wang**[2]
[1]Chongqing University [2]Southeast University
nanzx@cqu.edu.cn, 202514131128@stu.cqu.edu.cn
txiang@cqu.edu.cn, wangyw.evan@gmail.com

Abstract

Black-box watermarked LLM identification has become an important task for watermark auditing. The core challenge arises from inherent black-box constraints that deny access to logits, detector keys, model parameters, and internal watermark settings. For this reason, existing methods have not conducted in-depth exploration on this task, leaving prominent limitations: **Fixed-Reference Calibration** and **Family-Specific Tests**. First, existing works use fixed null reference to serve as a baseline for judging the presence of watermarks in LLM outputs, which might misidentify a watermarked LLM as unwatermarked because the null reference is sensitive to the prompt pair, queried model, and sampling configuration. Second, existing statistical-tests based methods adopt distinct statistical features and judgment criteria for different LLM watermark families, making them less suitable when the deployed watermark algorithm is unknown. To remedy the above two limitations, this work proposes **SCTI**, a Self-Calibrated Trident Identification framework for black-box watermarked LLM identification. Specifically, to handle the first limitation, **SCTI** constructs empirical null distributions from the queried responses to estimate the generalizable reference instead of relying on a fixed reference. Meanwhile, to address the second limitation, **SCTI** is configured with a Trident-View Consistency measuring mechanism, which enables a unified pipeline to avoid designing separate tests for individual watermark family. Experimental results verify that **SCTI** outperforms representative black-box watermark identification baselines across diverse LLMs and watermark algorithms.

1 Introduction

LLM watermarking embeds statistical traces into generated texts [1, 2, 3], which has emerged as a practical mechanism for provenance tracking, attribution, and misuse mitigation in deployed LLM services. Accordingly, auditing whether an unknown LLM uses watermarking becomes important for transparency and provenance analysis, especially when watermarks are used to mitigate misuse or trace generated content [1, 2, 3, 4]. Most watermarking studies focus on text-level verification, where the detector knows the watermarking rule or the secret key [1, 5, 6, 7, 8]. However, in many real deployments, an external auditor may not know whether an LLM uses watermarking or not. The auditor can only submit prompts and collect generated responses to infer whether a watermarking mechanism is deployed, which is termed as black-box watermark identification. This paper handles the problem of black-box watermark identification.

Recently, the emerging field of black-box watermarked LLM identification has attracted some research attention [4, 9, 10, 11]. Nevertheless, this direction remains largely underexplored, with only a limited number of existing studies. Representative methods include Water-Probe [4], which uses

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

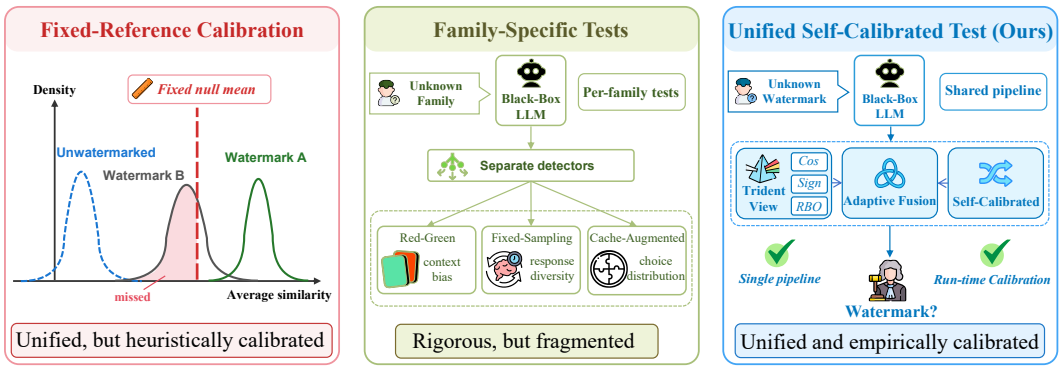


Figure 1: **Positioning of black-box watermark detectors.** Existing routes are either heuristically calibrated or fragmented into family-specific tests. Our method targets a unified pipeline for diverse watermarks under the unknown-watermark setting.

crafted prompts under fixed simulated keys to identify consistent effects caused by watermarking. Gloaguen et al. [10] propose an alternative approach by developing rigorous statistical tests tailored to different watermarking families.

These studies have substantially promoted the progress of black-box watermark identification. Nevertheless, two critical limitations remain to be addressed. **1) Fixed-Reference Calibration.** As illustrated in the left panel of Fig. 1, a unified detector with a fixed reference is sensitive to the prompt pair, queried model, and sampling configuration. Therefore, a fixed calibration rule may misidentify a watermarked LLM as unwatermarked. **2) Family-Specific Tests.** As shown in the middle panel, statistical tests can provide rigorous decisions, but they require different statistics and decision rules for different watermark families. This is not applicable to the scenarios that the auditor does not know the specifically-deployed watermark algorithms.

To address the limitation of fixed-reference calibration, a practical black-box watermark detector should adopt an adaptive calibration mechanism rather than relying on fixed reference criteria. Meanwhile, it should keep a unified pipeline across different watermark algorithms, without assuming any specific watermark algorithm is known in advance. To this end, we propose **SCTI**, a self-calibrated framework for black-box LLM watermark identification, to simultaneously address the above two limitations. Specifically, to handle the first limitation, we propose ***PNC (Permutation Null Calibration)*** to build a self-calibrated reference. Instead of relying on a fixed reference, ***PNC*** estimates the null reference by permuting the pairings between responses from two prompts. For the second limitation, we propose ***TriVC (Trident-View Consistency)*** that measures trident-view profile consistency in a unified pipeline to avoid the separate test for the individual watermark family. We also propose an adaptively-weighted fusion module (i.e., ***AdaFuse***) to adaptively fuse the calibrated view scores, reducing the influence of unstable views on the final decision.

Overall, as illustrated in the right panel of Fig. 1, **SCTI** is a unified self-calibrated identification framework with trident-view consistency measuring mechanism and adaptive fusion scheme. **SCTI** is evaluated on multiple LLMs and various watermarking algorithms. The experiment results demonstrate that **SCTI** achieves high-confidence identification across various watermarking algorithms. Furthermore, diagnostic experiments validate the effectiveness of ***TriVC***, ***PNC***, and ***AdaFuse***.

The contributions of this paper are as follows: **(1)** We propose **SCTI**, a unified self-calibrated framework for black-box watermarked LLM identification, which presents superiority over baselines. **(2)** We propose ***PNC*** that realizes self-calibration, solving the limitation of fixed-reference calibration in existing works. **(3)** We propose ***TriVC*** with trident-view consistency measuring mechanism, which does not need to assume that the specific watermarking algorithm is known.

2 Related Work

Black-box Watermark Detection Methods. Earlier methods [9] study watermark identification under stronger assumptions, such as access to output logits or an unwatermarked counterpart of the model. Water-Probe [4] identifies watermarked LLM by using crafted prompts and comparing

cross-prompt consistency under repeated sampling. Gloaguen et al. [10] develop statistical black-box methods for detecting LLM watermarks, using family-specific tests for Red-Green, Fixed-Sampling, and Cache-Augmented schemes. Recent third-party verification work studies a different setting. TTP-Detect [11] aims to verify whether a given text is watermarked without relying on secret keys or provider-side detectors, while our work focuses on service-level identification of whether a queried LLM carries a watermark mechanism.

Watermarking Algorithms Relevant to Our Benchmarks. This paper focuses on generation-time watermarking, where the watermark is embedded by modifying the token generation process. Unlike post-processing algorithms that modify text after generation, generation-time watermarking changes the token generation process itself by adjusting the next-token distribution or the sampling rule according to a watermark key [4, 2, 3]. Following Water-Probe [4], we organize the watermarking algorithms according to how the watermark key is determined, including N-Gram based watermarking and Fixed-Key-List based watermarking.

In N-Gram watermarking, each decoding-step key is derived from the recent token context and a master key. Therefore, the same local prefix tends to induce the same effective key and repeatable distributional changes. N-Gram watermarking algorithm includes KGW [1], KGW-Min and KGW-Skip [12], AAR [6], DiPmark [8], γ-Reweight [7], and WaterBag [4]. WaterBag follows this key-dependence structure but increases randomness with multiple master keys and their inversions. In Fixed-Key-List watermarking, generation follows a predefined key or pseudo-random vector list, so the signal depends on the list position and segment rather than the preceding N-gram alone. EXP-Edit and ITS-Edit [5] belong to this type. Since the two types induce repeated key conditions differently, we report them separately while applying the same black-box identification pipeline in **SCTI**.

3 Methodology

3.1 Preliminary and Formulation

Theoretical basis of watermarked LLM identification. The essence of LLM text generation lies in the sequential and continuous selection of the next token at each inference step. For LLM watermarking, watermark providers leverage a **secret key** to deliberately manipulate the token sampling probability distribution during generation, thereby embedding imperceptible watermark signals into model outputs. Consequently, watermarked LLMs exhibit highly consistent token selection patterns when responding to similar prompts, which fundamentally enables the identification of watermarked LLMs.

Core idea of watermarked LLM identification. Based on the above theoretical basis, watermarked LLM identification could make use of similar prompts (termed as **matched prompt pair**) to trigger token generation logic of LLMs. During the generation procedure, a group of keys (termed as **simulated fixed keys**) is artificially preset to simulate a secret key. Simulated fixed keys constrain the token selection rules during model inference to eliminate errors brought by random prompt diversity, so as to construct **paired profiles** by recording responses of LLM to matched prompt pair with the constraint of simulated fixed keys. Paired profiles summarize how the key changes the output behavior for the prompts. A watermarked LLM is expected to produce similar profiles under the same simulated keys. Therefore, watermark identification could be realized by testing whether the paired profiles exhibit consistent changes.

Formulation. This paper studies watermarked LLM identification in the black-box setting. In this setting, the detector does not observe logits, keys, family labels, or watermark parameters of LLMs. It only issues a prompt p and receives a response y. The objective is, through repeated prompt-response interaction, distinguishing unwatermarked model and watermarked model.

3.2 Overview

As illustrated in Fig. 2, **SCTI** addresses black-box watermark identification through a four-stage framework: ***Paired Prompt-Key Sampling (PPKS)***, ***Trident-View Consistency (TriVC)***, ***Permutation Null Calibration (PNC)***, and ***Adaptive Fusion (AdaFuse)***. The overall idea of our method is to realize watermark identification by analyzing how the simulated fixed keys change the output of the matched prompt pair. First, given a matched prompt pair (p_1, p_2), ***PPKS*** uses a finite set of simulated fixed keys (e.g., k_1 and k_2) to obtain response banks (e.g., $\mathrm{Bank}(p_1, k_1)$) via repeated sampling and converts

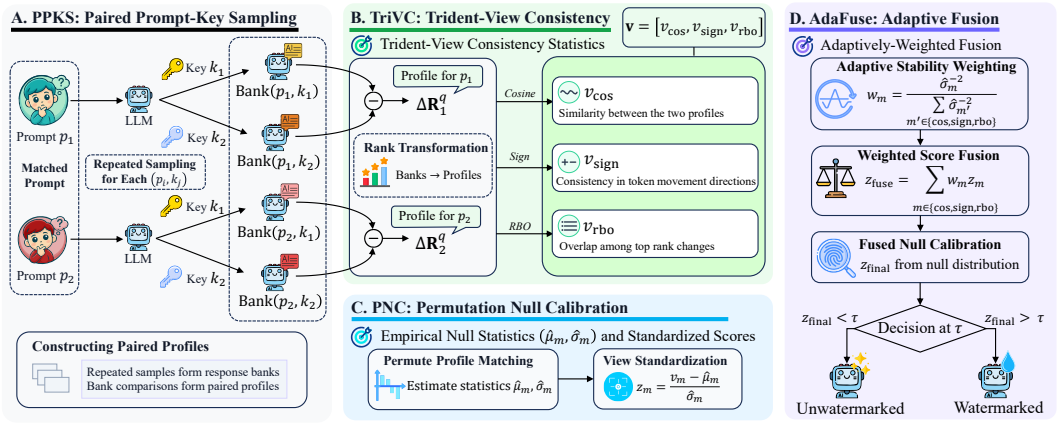


Figure 2: Overview of **SCTI**. ***Panel A (PPKS)*** estimates response banks and forms paired profiles. ***Panel B (TriVC)*** analyzes trident-view consistency of paired profiles. ***Panel C (PNC)*** estimates empirical null distributions. ***Panel D (AdaFuse)*** combines the standardized scores to compute the final fused score z_{final} for identification.

the banks into paired profiles $(\Delta\mathbf{R}_1^q, \Delta\mathbf{R}_2^q)$. Second, ***TriVC*** measures the consistency of paired profiles from three complementary views, obtaining trident-view consistency $\mathbf{v} = \left[v_{\cos},\, v_{\text{sign}},\, v_{\text{rbo}}\right]$. Third, ***PNC*** estimates an empirical null distribution for each consistency statistic and converts $\mathbf{v}$ into the calibrated trident-view consistency $\mathbf{z} = \left[z_{\cos},\, z_{\text{sign}},\, z_{\text{rbo}}\right]$. Finally, ***AdaFuse*** combines the standardized scores into a single fused score z_{final} for watermark identification.

3.3 PPKS: Paired Prompt-Key Sampling

PPKS builds on the principle of Water-Probe [4]. The key idea is to use two prompts with similar output distributions to test whether their response distributions change in a consistent way. Specifically, we define a matched prompt pair (p_1, p_2) and a finite simulated key set $\mathcal{K} = \{k_1, \ldots, k_M\}$. For each prompt p_i, $i \in \{1, 2\}$ and each simulated key $k_j \in \mathcal{K}$, repeated sampling produces n responses:

$$\text{Bank}(p_i, k_j) = \{y_{i,j}^{(1)}, \ldots, y_{i,j}^{(n)}\}, \tag{1}$$

where $y_{i,j}^{(w)}$ denotes the w-th response to prompt p_i under key k_j. $\text{Bank}(p_i, k_j)$ is tokenized to estimate a token distribution $\hat{\mathbf{p}}_i^{(k_j)}$ over the vocabulary $\mathcal{V}$, where $\mathcal{V}$ denotes the tokenizer vocabulary. Following Water-Probe [4], we convert each $\hat{\mathbf{p}}_i^{(k_j)}$ into a rank vector $\mathbf{r}_i^{(k_j)}$. The rank vectors represent token-frequency variation across simulated keys. For a matched prompt pair, we compare the effect of the same two simulated keys on both prompts. The corresponding key pairs are defined as:

$$\mathcal{Q} = \{(k_a, k_b) : 1 \le a < b \le M\}. \tag{2}$$

For each key pair $q = (k_a, k_b) \in \mathcal{Q}$, the profile for prompt p_i is the rank change between the keys:

$$\Delta\mathbf{R}_i^q = \mathbf{r}_i^{(k_a)} - \mathbf{r}_i^{(k_b)}, \qquad i \in \{1, 2\}. \tag{3}$$

The paired profiles $(\Delta\mathbf{R}_1^q, \Delta\mathbf{R}_2^q)$ summarize the distribution changes induced by the simulated key pair on the two matched prompts. These paired profiles form the input to ***TriVC***.

3.4 TriVC: Trident-View Consistency

To avoid designing separate tests for different watermark families, ***TriVC*** measures consistency between paired profiles from global and local perspectives, rather than relying on a statistic tied to a specific watermark algorithm. This requires examining both global and local structures in the paired profiles. Global structure indicates whether two matched prompts respond to the same key pair in a similar overall direction. Local structure is also important because the profile $\Delta\mathbf{R}_i^q$ may concentrate on a smaller subset of tokens. Such local changes can appear as consistent movement directions or as overlap among the tokens whose ranks differ most between simulated keys. However, a single

consistency statistic does not always capture the full cross-prompt effect of a watermark key pair. Therefore, ***TriVC*** measures the consistency between paired profiles from three complementary views: a global cosine view, a local sign view, and a local Rank-Biased Overlap (RBO) view.

Cosine View. For each key pair $q = (k_a, k_b) \in \mathcal{Q}$ defined in Eq. 2, we obtain paired profiles $\Delta\mathbf{R}_1^q$ and $\Delta\mathbf{R}_2^q$. The cosine view measures global alignment over the vocabulary $\mathcal{V}$:

$$s_{\cos}^q = \frac{\langle \Delta\mathbf{R}_1^q, \Delta\mathbf{R}_2^q \rangle}{\|\Delta\mathbf{R}_1^q\|_2 \, \|\Delta\mathbf{R}_2^q\|_2}. \tag{4}$$

A larger $s_{\cos}^q$ indicates that paired profiles have a similar overall direction across the vocabulary $\mathcal{V}$.

Sign View. Sign view focuses on whether each token changes in the same direction in the two profiles. For key pair $q = (k_a, k_b)$, $\Delta\mathbf{R}_{i,t}^q$ denotes the change of token t's rank for prompt p_i when the simulated key changes from k_b to k_a. To filter out tokens that remain unchanged in both profiles, the active token set as:

$$\mathcal{I}^q = \{t \in \mathcal{V} : \Delta\mathbf{R}_{1,t}^q \neq 0 \text{ or } \Delta\mathbf{R}_{2,t}^q \neq 0\}. \tag{5}$$

The sign view then compares the direction of those changes:

$$s_{\text{sign}}^q = \frac{1}{|\mathcal{I}^q|} \sum_{t \in \mathcal{I}^q} \mathbf{1}\left[\text{sign}\left(\Delta\mathbf{R}_{1,t}^q\right) = \text{sign}\left(\Delta\mathbf{R}_{2,t}^q\right)\right], \tag{6}$$

where $\mathbf{1}[\cdot]$ is the indicator function, and $\text{sign}(x) \in \{-1, 0, 1\}$ returns the sign of x. If $\mathcal{I}^q = \varnothing$, we set $s_{\text{sign}}^q = 0$. This view captures consistent movement directions even when the two profiles differ in the magnitude of their rank changes.

RBO View. Global alignment can hardly handle the cases where the two profiles agree mainly on a small set of tokens with large rank changes. Therefore, the rank-biased overlap (RBO) view focuses on tokens with the significant rank changes. For each prompt p_i, $\mathcal{L}_i^q$ denotes the ranked list of K tokens with the largest values of $|\Delta\mathbf{R}_{i,t}^q|$, where K is the list length used for this local comparison. The corresponding RBO statistic is:

$$s_{\text{rbo}}^q = \text{RBO}\left(\mathcal{L}_1^q, \mathcal{L}_2^q\right). \tag{7}$$

RBO compares two ranked lists and assigns larger weight to overlap near the top of the lists [13]. This view checks whether the two profiles share a similar set of tokens with significant rank changes.

For each view $m \in \{\cos, \text{sign}, \text{rbo}\}$, the average value over all key pairs $q \in \mathcal{Q}$ is computed:

$$v_m = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} s_m^q. \tag{8}$$

The resulting trident-view consistency vector is:

$$\mathbf{v} = \left[v_{\cos},\, v_{\text{sign}},\, v_{\text{rbo}}\right]. \tag{9}$$

The three views capture complementary aspects of cross-prompt consistency. Cosine summarizes global alignment over all tokens. Sign checks whether token ranks change in the same direction across profiles. RBO checks whether significant rank changes involve a similar set of tokens. Therefore, ***TriVC*** characterizes cross-prompt consistency from both global and local perspectives, rather than relying on a single similarity statistic or designing a separate statistic for each watermark family.

3.5 PNC: Permutation Null Calibration

The trident-view consistency $\mathbf{v}$ produced by ***TriVC*** needs a null reference that contributes to a watermark identification decision. However, a fixed null reference is unreliable in the black-box setting, since the null distributions of $v_{\cos}$, v_{sign}, and v_{rbo} change with the prompt pair, the queried model, and the sampling configuration. Therefore, ***PNC*** addresses this issue by estimating empirical null distributions for each view through permutation of the profile matching, where reassigned profile pairs are used to recompute the consistency statistics.

For each view $m \in \{\cos, \text{sign}, \text{rbo}\}$, ***PNC*** calibrates the view statistic v_m defined in Eq. 8. The statistic v_m is computed from the consistency statistics s_m^q over all key pairs $q \in \mathcal{Q}$. To estimate its

empirical null distribution, ***PNC*** samples B random permutations of the key pair set $\mathcal{Q}$. The b-th permutation is denoted by π_b, and $\pi_b(q)$ denotes the key pair reassigned to q under this permutation.

Under permutation π_b, $\Delta\mathbf{R}_1^q$ remains associated with the original key pair q, while the p_2 profile is taken from the reassigned key pair $\pi_b(q)$. The consistency statistic recomputed from this reassigned profile pair is denoted by s_m^{q,π_b}. It is noted that if $\pi_b(q) = q$, the profile matching for q remains unchanged and does not provide a permuted comparison. Therefore, only the key pairs with changed matchings are collected as $\mathcal{Q}_b$, and $v_m^{(b)}$ is obtained by averaging the recomputed statistics over them:

$$\mathcal{Q}_b = \{q \in \mathcal{Q} : \pi_b(q) \neq q\}, \qquad v_m^{(b)} = \frac{1}{|\mathcal{Q}_b|} \sum_{q \in \mathcal{Q}_b} s_m^{q,\pi_b}. \tag{10}$$

Repeating the construction of $v_m^{(b)}$ over all sampled permutations yields B permuted view statistics. These permuted view statistics form the empirical null distribution for view m. ***PNC*** then estimates the null mean and variance as:

$$\widehat{\mu}_m = \frac{1}{B} \sum_{b=1}^{B} v_m^{(b)}, \qquad \widehat{\sigma}_m^2 = \frac{1}{B-1} \sum_{b=1}^{B} \left(v_m^{(b)} - \widehat{\mu}_m \right)^2. \tag{11}$$

The standard deviation $\widehat{\sigma}_m$ is obtained as the square root of $\widehat{\sigma}_m^2$. The original view statistic v_m in Eq. 8 is then standardized by this empirical null distribution:

$$z_m = \frac{v_m - \widehat{\mu}_m}{\widehat{\sigma}_m}, \qquad m \in \{\cos, \text{sign}, \text{rbo}\}, \tag{12}$$

where z_m is the standardized score of view m.

Collecting the three standardized scores forms the calibrated trident-view consistency vector $\mathbf{z}$:

$$\mathbf{z} = \left[z_{\cos},\ z_{\text{sign}},\ z_{\text{rbo}} \right]. \tag{13}$$

The calibrated trident-view consistency vector $\mathbf{z}$ serves as the standardized input to ***AdaFuse***.

3.6 AdaFuse: Adaptive Fusion

In ***PNC***, the trident-view consistency $\mathbf{v}$ is standardized as $\mathbf{z}$. A natural fusion choice is to assign equal weight to the three scores in $\mathbf{z}$ and sum them directly. However, ***PNC*** validates that the views might present variability under the empirical null distributions. The null variance $\widehat{\sigma}_m^2$ measures how much the raw statistic of view m varies under its empirical null distribution. A view with larger null variance is less stable, while a view with smaller null variance provides a more stable signal. Therefore, equal-weight averaging dilutes a stable view or imposes excessive influence on a volatile view. ***AdaFuse*** addresses this issue by assigning larger weights to views with smaller null variance.

Following the stability principle, ***AdaFuse*** uses an inverse variance weighting rule. For each view $m \in \{\cos, \text{sign}, \text{rbo}\}$, we define the weight w_m as follows:

$$w_m = \frac{\widehat{\sigma}_m^{-2}}{\sum_{m' \in \{\cos, \text{sign}, \text{rbo}\}} \widehat{\sigma}_{m'}^{-2}}, \qquad m \in \{\cos, \text{sign}, \text{rbo}\}, \tag{14}$$

where m indexes the current view, while m' denotes the view indexes for summation operation. Since w_m is proportional to the inverse null variance, a view with smaller $\widehat{\sigma}_m^2$ receives a larger weight. After obtaining the weights, the fused score is calculated as a weighted sum:

$$z_{\text{fuse}} = \sum_{m \in \{\cos, \text{sign}, \text{rbo}\}} w_m z_m. \tag{15}$$

This score summarizes the three standardized views while reducing the influence of views with larger null variance. Since the final decision depends on z_{fuse}, this fused score is compared with the values obtained by applying the same fusion rule after permutation. ***AdaFuse*** therefore applies the same weighting rule to the permuted view statistics. For each sampled permutation π_b, the permuted view statistics $v_m^{(b)}$ are standardized and fused as:

$$z_{\text{fuse}}^{(b)} = \sum_{m \in \{\cos, \text{sign}, \text{rbo}\}} w_m \frac{v_m^{(b)} - \widehat{\mu}_m}{\widehat{\sigma}_m}, \tag{16}$$

where $z_{\text{fuse}}^{(b)}$ denotes the fused score computed after applying the same fusion rule under the b-th sampled permutation. Repeating this procedure over the B sampled permutations yields B fused null scores, which form the empirical null distribution used to calibrate z_{fuse}. The fused null mean and variance are estimated as follows:

$$\widehat{\mu}_{\text{fuse}} = \frac{1}{B}\sum_{b=1}^{B} z_{\text{fuse}}^{(b)}, \qquad \widehat{\sigma}_{\text{fuse}}^2 = \frac{1}{B-1}\sum_{b=1}^{B}\left(z_{\text{fuse}}^{(b)} - \widehat{\mu}_{\text{fuse}}\right)^2. \tag{17}$$

The final fused score is formulated as follows:

$$z_{\text{final}} = \frac{z_{\text{fuse}} - \widehat{\mu}_{\text{fuse}}}{\widehat{\sigma}_{\text{fuse}}}. \tag{18}$$

A larger z_{final} indicates stronger evidence against the null hypothesis $\mathcal{H}_0$, where the queried LLM is unwatermarked. Given criterion τ, **SCTI** identifies the LLM as watermarked when $z_{\text{final}} > \tau$, and reports no watermark otherwise.

4 Experiments

4.1 Setup

Tested LLMs and Watermarking Algorithms The tested LLMs include Qwen3-1.7B, Qwen3-4B, Qwen3-8B, and Qwen3-14B from Qwen3 family [14], as well as Llama-3-8B, Llama-3.1-8B, and Llama-3.2-3B [15] from Llama family [16]. N-Gram based and Fixed-Key-List based watermarking schemes are tested. For N-Gram based watermarking, we test KGW [1] ($\gamma = 0.5, \delta = 2$), AAR [6] ($N = 1$), KGW-Min [12] (window size $= 4$), KGW-Skip [12] (window size $= 3$), DiPmark [8] ($\alpha = 0.45$), γ-Reweight [7], and WaterBag [4]. For Fixed-Key-List based watermarking, we test EXP-Edit and ITS-Edit [5]. Details of these algorithms are provided in Appendix A.

Baselines. Watermark identification has been studied in different settings, including secret-key detectors for generated text [1, 5] and publicly verifiable watermarking schemes [17, 18, 19]. These methods either verify generated texts or rely on verification logic tied to the watermarking scheme, which differs from our black-box identification setting. In this setting, the detector queries the LLM and determines whether it carries a generation-time watermarking algorithm. Therefore, **SCTI** is compared with Water-Probe [4] and Gloaguen et al. [10], two representative baselines for black-box watermark identification. Water-Probe [4] detects watermarked LLMs by using crafted prompts and measuring cross-prompt consistency under repeated sampling. Gloaguen et al. [10] detect watermarked LLMs by querying black-box models and applying different statistical tests for different watermarking algorithms.

4.2 Main Results

Tab. 1 presents the comparison results under the unwatermarked (i.e., the column **Non**) condition and different watermarking conditions (i.e., the columns **N-Gram** and **Fixed-Key-List**). Following the confidence convention of Water-Probe [4], a score between 4 and 10 is considered as the moderate confidence that quantifies the degree of deviation from the unwatermarked distribution, and a score larger than 10 indicates the high confidence. Therefore, a reliable watermarked LLM identification method should yield the scores less than 4 in the unwatermarked condition and higher than 10 in the watermarked condition. For intuitive visualization, dark/light purple is used to indicate high-confidence/moderate-confidence watermark identification in the watermarking conditions, and an uncolored cell is utilized to denote that non-watermark is identified.

In Tab. 1, " $-$ " denotes that the corresponding test is not applicable to this watermark setting. We can observe that the baselines do not consistently perform well in various conditions. For example, on the WaterBag watermark, Water-Probe-v2 [4] exhibits lower confidence on several models. In contrast, **SCTI** achieves high-confidence identification for all tested LLMs across N-Gram and Fixed-Key-List schemes. At the same time, **SCTI** produces the scores less than 4 in the unwatermarked condition, without misidentifying unwatermarked models as watermarked. These results verify the superiority of our method over baselines, demonstrating the robustness and generalizability of our method across diverse watermarking schemes on different LLMs. The superiority is mainly contributed to the design of **SCTI**. Instead of relying on a single consistency statistic, **SCTI** compares paired profiles through complementary views that capture watermark-induced differences at multiple levels.

Table 1: Identification scores for black-box watermark identification across LLMs and watermark families. Dark purple indicates high-confidence watermark identification, Light purple indicates moderate-confidence watermark identification, while no color indicates no watermark identified.

LLM	Method	Non	N-Gram							Fixed-Key-List	
			KGW	AAR	KGW-Min	KGW-Skip	DiPmark	γ-Reweight	WaterBag	EXP-Edit	ITS-Edit
Qwen3-1.7B	Gloaguen et al. [10]	<4	>10	-	-	-	-	-	-	-	
	Water-Probe-v2 [4]	-1.93	37.06	205.24	20.41	18.48	22.62	66.19	3.88	26.97	132.49
	SCTI (Ours)	-1.08	43.25	85.56	13.45	37.37	17.67	13.81	33.82	20.42	18.29
Llama-3.2-3B	Gloaguen et al. [10]	<4	>10	-	-	-	-	-	-	-	
	Water-Probe-v2 [4]	-12.42	29.91	96.14	50.00	19.91	80.04	67.07	3.30	34.99	7702.12
	SCTI (Ours)	1.05	102.25	38.44	122.27	113.48	94.31	78.92	50.02	151.17	151.70
Qwen3-4B	Gloaguen et al. [10]	<4	>10	-	-	-	-	-	-	-	
	Water-Probe-v2 [4]	-10.40	22.33	19.10	35.83	7.91	20.87	17.64	6.35	5.31	214.78
	SCTI (Ours)	-11.62	101.20	26.27	120.39	74.05	51.62	48.50	62.28	72.43	114.11
Llama-3-8B	Gloaguen et al. [10]	<4	>10	-	-	-	-	-	-	-	
	Water-Probe-v2 [4]	-0.71	16.79	15.88	22.64	22.55	37.45	53.15	8.49	21.73	153.85
	SCTI (Ours)	-1.29	101.43	41.86	126.66	87.3	132.54	113.97	40.83	158.98	50.50
Llama-3.1-8B	Gloaguen et al. [10]	<4	>10	-	-	-	-	-	-	-	
	Water-Probe-v2 [4]	-64.31	25.24	104.77	10.03	49.23	38.47	81.36	9.45	31.35	12701.95
	SCTI (Ours)	-6.31	108.19	43.75	134.93	106.17	118.47	109.64	82.46	146.06	1127.96
Qwen3-8B	Gloaguen et al. [10]	<4	>10	-	-	-	-	-	-	-	
	Water-Probe-v2 [4]	-1.07	15.99	30.21	7.38	5.94	9.01	9.64	3.73	8.73	105.29
	SCTI (Ours)	-8.75	30.54	43.97	37.70	28.58	27.76	26.49	20.44	20.36	58.47
Qwen3-14B	Gloaguen et al. [10]	<4	>10	-	-	-	-	-	-	-	
	Water-Probe-v2 [4]	-0.12	7.27	26.97	9.28	13.57	15.78	33.6	2.96	6.20	218.52
	SCTI (Ours)	0.02	50.66	13.42	40.15	71.17	47.55	45.47	34.50	30.40	129.49

4.3 Diagnostic Experiments

Diagnostic Experiments on *TriVC*. ***TriVC*** is designed to measure cross-prompt consistency from three views. To verify its effectiveness, we conduct four experiments: **1)** only the cosine view is enabled; **2)** only the sign view is enabled; **3)** only the RBO view is enabled; **4)** all three views are enabled. The experimental results are reported in Tab. 2. More results are provided in Appendix D.

Tab. 2 reveals that each individual view captures different aspects of the watermark effect. For example, the cosine view is effective on AAR for Qwen3-1.7B and on DiPmark for Llama3.2-3B, exhibiting that global profile alignment is able to capture cross-prompt consistency in some watermarking algorithms. The sign view achieves high-confidence identification on KGW and KGW-Min, indicating that directional agreement in token rank changes is informative. The RBO view is also informative in several watermarking algorithms, such as KGW and ITS-Edit, which verify that tokens with significant rank changes can provide additional evidence. By integrating three views of consistency, ***TriVC*** captures not only global profile alignment but also local evidence of movement directions and token rank changes, which eventually enables the model to estimate the low-confidence scores in the unwatermarked condition and high-confidence in the watermarked condition.

Table 2: Ablation study on ***TriVC*** view variants across LLMs and watermark schemes.

LLM	Method	Non	N-Gram							Fixed-Key-List	
			KGW	AAR	KGW-Min	KGW-Skip	DiPmark	γ-Reweight	WaterBag	EXP-Edit	ITS-Edit
Qwen3-1.7B	Cosine	-3.25	16.93	100.35	4.10	29.59	14.31	9.82	19.65	23.04	7.11
	Sign	1.65	55.00	95.01	16.93	50.67	18.20	18.88	37.83	25.17	19.56
	RBO	-1.18	34.97	59.10	8.05	27.07	14.68	6.79	28.55	16.40	17.76
	TriVC	-1.08	43.25	85.56	13.45	37.37	17.67	13.81	33.82	20.42	18.29
Llama-3.2-3B	Cosine	-3.62	60.54	29.81	90.81	49.18	86.11	71.48	24.64	104.15	116.14
	Sign	17.48	120.53	44.21	134.21	135.03	92.72	79.34	72.14	169.15	154.64
	RBO	2.73	80.55	33.18	94.59	89.50	76.53	62.13	31.82	133.12	140.13
	TriVC	1.05	102.25	38.44	122.27	113.48	94.31	78.92	50.02	151.17	151.70

Fig. 3 further explains why ***TriVC*** is more stable than the single view configuration. The figure illustrates pairwise spearman correlations of cosine, sign, and RBO across four watermark algorithms on Qwen3-14B. For clarity, we show representative cases here and provide more correlation results in Appendix D. Correlation coefficients ranging from 0 to 1 demonstrate that these views are relevant yet non-interchangeable. We can observe that all correlation results are positive, verifying that the three views capture correlated watermark signals. Therefore, ***TriVC*** effectively enriches the consistency measurement between paired profiles.

Diagnostic Experiments on *PNC*. ***PNC*** is designed to calibrate raw view statistics with an empirical null distribution. To examine its efficacy, ***PNC*** is compared with two fixed reference variants on the cosine view: **1)** the reference mean is fixed at $\mu = 0$, which corresponds to the ideal null reference; **2)** the reference mean is fixed at $\mu = 0.1$, following the fixed reference setting in Water-Probe [4].

Table 3: ***PNC*** calibration ablation for the cosine view across LLMs and watermark schemes.

LLM	Method	Non	N-Gram							Fixed-Key-List	
			KGW	AAR	KGW-Min	KGW-Skip	DiPmark	γ-Reweight	WaterBag	EXP-Edit	ITS-Edit
Qwen3-1.7B	Fixed $\mu = 0$	0.19	24.37	98.62	18.91	30.26	20.11	15.05	20.47	37.24	11.06
	Fixed $\mu = 0.1$	-24.39	8.61	74.38	4.26	11.82	10.42	7.52	15.61	15.71	7.55
	PNC	-3.25	16.93	100.35	4.10	29.59	14.31	9.82	19.65	23.04	7.11
Llama-3.2-3B	Fixed $\mu = 0$	10.33	61.57	29.95	91.33	61.88	89.10	70.64	31.70	115.72	115.59
	Fixed $\mu = 0.1$	-37.98	21.83	24.99	37.83	19.14	51.45	41.87	8.46	66.71	73.52
	PNC	-3.62	60.54	29.81	90.81	49.18	86.11	71.48	24.64	104.15	116.14

The results are reported in Tab. 3. We can observe that the calibration mechanism with fixed reference mean fails to achieve stable performance. Taking Llama-3.2-3B as an example, when $\mu = 0$, the score under the unwatermarked setting remains high, although this reference assumes an ideal null reference. When $\mu = 0.1$, the score under the unwatermarked setting becomes lower, but the watermarked scores are suppressed, especially on WaterBag. In contrast, ***PNC*** keeps the low unwatermarked score and preserves effective identification across watermarked settings. These results verify the effectiveness of empirical null calibration. The reason is that the permutation process constructs the null distribution from the same prompt pair, queried LLM, and sampling configuration. By reassigning the profile matching, ***PNC*** estimates the consistency level that can arise from the current samples without relying on the original matched profiles. Therefore, the final score measures how much the observed consistency exceeds the reference range, rather than relying on a fixed null mean.



Figure 3: Pairwise spearman correlations among the cosine, sign, and RBO views on Qwen3-14B.

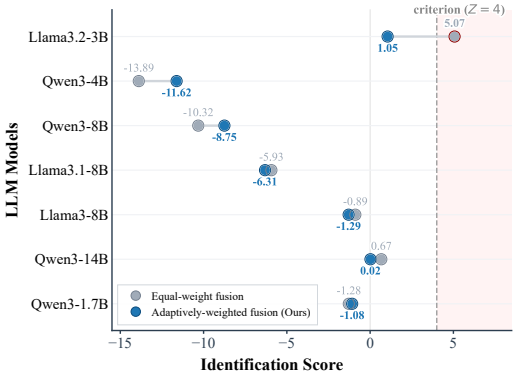


Figure 4: Comparison between equal-weight fusion and ***AdaFuse*** in the unwatermarked setting.

Diagnostic Experiments on *AdaFuse*. ***AdaFuse*** is designed to fuse the consistency from different views through dynamic weighting, which is supposed to be more reliable than the fusion strategy adopting fixed equal weights. Therefore, we further conduct comparative experiments with equal-weight fusion where the three standardized view scores are averaged with the same weight. This experiment is conducted in the unwatermarked setting, since a reliable fusion rule should avoid misidentifying unwatermarked models as watermarked. Without such mitigation, high scores on watermarked LLMs alone are not sufficient to show reliable identification.

Fig. 4 illustrates the scores of equal-weight fusion and ***AdaFuse***. Each gray point denotes the equal-weight fusion score for an unwatermarked LLM, while each blue point denotes the ***AdaFuse*** score. For a reliable fusion, the score under the unwatermarked setting should remain below the criterion 4, corresponding to the left side of dashed line. Equal-weight fusion is a simple and useful fusion strategy in many cases. However, it is unreliable when one view produces an unstable score. Taking Llama-3.2-3B as an example, equal-weight fusion crosses the identification criterion and misidentifies unwatermarked models as watermarked, while ***AdaFuse*** remains below the criterion and avoids this false identification. It demonstrates that ***AdaFuse*** provides a more stable result, which benefits from that ***AdaFuse*** assigns smaller weights to views with larger null variance. Therefore, the final score is less affected by unstable views in the unwatermarked setting.

5 Conclusion

This paper studies the problem of black-box LLM watermark identification. We propose **SCTI**, a self-calibrated framework that identifies watermark presence by comparing paired prompt responses

under simulated fixed keys. Specially, **SCTI** is equipped with ***PNC*** to solve the limitation of fixed-reference calibration in existing methods. Meanwhile, ***TriVC*** is also designed to avoid the limitation of family-specific tests. Experiments across multiple LLMs and watermarking algorithms verify the superiority of **SCTI**.

Acknowledgments

We would like to express our gratitude to the anonymous NeurIPS 2026 reviewers (Reviewer FogK, 3Fod, and 3L7c) and Area Chair HLLi for their valuable feedback and suggestions that helped improve this paper.

Funding. This research received no external funding or third-party support.

Competing Interests. The authors declare no competing interests.

References

[1] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, "A watermark for large language models," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds. PMLR, 2023, pp. 17 061–17 084. [Online]. Available: https://proceedings.mlr.press/v202/kirchenbauer23a.html

[2] A. Liu, L. Pan, Y. Lu, J. Li, X. Hu, X. Zhang, L. Wen, I. King, H. Xiong, and P. S. Yu, "A survey of text watermarking in the era of large language models," *ACM Comput. Surv.*, vol. 57, no. 2, pp. 47:1–47:36, 2025. [Online]. Available: https://doi.org/10.1145/3691626

[3] T. Sander, P. Fernandez, A. Durmus, M. Douze, and T. Furon, "Watermarking makes language models radioactive," *arXiv preprint arXiv:2402.14904*, 2024.

[4] A. Liu, S. Guan, Y. Liu, L. Pan, Y. Zhang, L. Fang, L. Wen, P. S. Yu, and X. Hu, "Can watermarked llms be identified by users via crafted prompts?" in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=ujpAYpFDEA

[5] R. Kuditipudi, J. Thickstun, T. Hashimoto, and P. Liang, "Robust distortion-free watermarks for language models," *Trans. Mach. Learn. Res.*, vol. 2024, 2024. [Online]. Available: https://openreview.net/forum?id=FpaCL1MO2C

[6] S. Aaronson and H. Kirchner, "Watermarking gpt outputs," 2022, https://www.scottaaronson.com/talks/watermark.ppt.

[7] Z. Hu, L. Chen, X. Wu, Y. Wu, H. Zhang, and H. Huang, "Unbiased watermark for large language models," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=uWVC5FVidc

[8] Y. Wu, Z. Hu, H. Zhang, and H. Huang, "Dipmark: A stealthy, efficient and resilient watermark for large language models," *CoRR*, vol. abs/2310.07710, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2310.07710

[9] L. Tang, G. Uberti, and T. Shlomi, "Baselines for identifying watermarked large language models," *CoRR*, vol. abs/2305.18456, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.18456

[10] T. Gloaguen, N. Jovanovic, R. Staab, and M. T. Vechev, "Black-box detection of language model watermarks," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=E4LAVLXAHW

[11] Z. Wang, Y. Ren, Y. Cao, F. Fang, X. Li, and L. Guo, "Rethinking LLM watermark detection in black-box settings: A non-intrusive third-party framework," *CoRR*, vol. abs/2603.14968, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2603.14968

[12] J. Kirchenbauer, J. Geiping, Y. Wen, M. Shu, K. Saifullah, K. Kong, K. Fernando, A. Saha, M. Goldblum, and T. Goldstein, "On the reliability of watermarks for large

language models," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=DEJIDCmWOz

[13] W. Webber, A. Moffat, and J. Zobel, "A similarity measure for indefinite rankings," *ACM Trans. Inf. Syst.*, vol. 28, no. 4, pp. 20:1–20:38, 2010. [Online]. Available: https://doi.org/10.1145/1852102.1852106

[14] Q. Team, "Qwen3 technical report," *CoRR*, vol. abs/2505.09388, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.09388

[15] Meta AI, "Llama-3.2-3b," https://huggingface.co/meta-llama/Llama-3.2-3B, 2024.

[16] L. Team, "The llama 3 herd of models," *CoRR*, vol. abs/2407.21783, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2407.21783

[17] A. Liu, L. Pan, X. Hu, S. Li, L. Wen, I. King, and P. S. Yu, "An unforgeable publicly verifiable watermark for large language models," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=gMLQwKDY3N

[18] J. Fairoze, S. Garg, S. Jha, S. Mahloujifar, M. Mahmoody, and M. Wang, "Publicly-detectable watermarking for language models," *IACR Commun. Cryptol.*, vol. 1, no. 4, p. 31, 2024. [Online]. Available: https://doi.org/10.62056/ahmpdkp10

[19] H. Duan, L. Xiang, and X. Zhang, "Pvmark: Enabling public verifiability for LLM watermarking schemes," *CoRR*, vol. abs/2510.26274, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.26274

[20] A. B. Hou, J. Zhang, T. He, Y. Wang, Y. Chuang, H. Wang, L. Shen, B. V. Durme, D. Khashabi, and Y. Tsvetkov, "Semstamp: A semantic watermark with paraphrastic robustness for text generation," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, K. Duh, H. Gómez-Adorno, and S. Bethard, Eds. Association for Computational Linguistics, 2024, pp. 4067–4082. [Online]. Available: https://doi.org/10.18653/v1/2024.naacl-long.226

[21] A. B. Hou, J. Zhang, Y. Wang, D. Khashabi, and T. He, "k-semstamp: A clustering-based semantic watermark for detection of machine-generated text," in *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, ser. Findings of ACL, L. Ku, A. Martins, and V. Srikumar, Eds., vol. ACL 2024. Association for Computational Linguistics, 2024, pp. 1706–1715. [Online]. Available: https://doi.org/10.18653/v1/2024.findings-acl.98

[22] Z. Ye, G. Dey, C. Christodoulopoulos, C. Peris, A. Ramakrishna, W. Ruan, A. Galstyan, K. Chang, R. Gupta, and N. Mehrabi, "SWAN: semantic watermarking with abstract meaning representation," in *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, M. Liakata, V. P. Moreira, J. Zhang, and D. Jurgens, Eds. Association for Computational Linguistics, 2026, pp. 36 304–36 315. [Online]. Available: https://doi.org/10.18653/v1/2026.acl-long.1681

[23] N. Jovanovic, R. Staab, and M. T. Vechev, "Watermark stealing in large language models," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. A. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR / OpenReview.net, 2024, pp. 22 570–22 593. [Online]. Available: https://proceedings.mlr.press/v235/jovanovic24a.html

[24] R. Chen, Y. Wu, J. Guo, and H. Huang, "De-mark: Watermark removal in large language models," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds., vol. 267. PMLR / OpenReview.net, 2025. [Online]. Available: https://proceedings.mlr.press/v267/chen25bq.html

[25] N. Jovanovic, R. Staab, and M. T. Vechev, "Watermark stealing in large language models," in *Forty-first International Conference on Machine Learning, ICML 2024,*

Vienna, Austria, July 21-27, 2024, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. A. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds. PMLR / OpenReview.net, 2024, pp. 22 570–22 593. [Online]. Available: https://proceedings.mlr.press/v235/jovanovic24a.html

[26] Q. Wu and V. Chandrasekaran, "Bypassing LLM watermarks with color-aware substitutions," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 8549–8581. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.464

[27] Z. Zhang, X. Zhang, Y. Zhang, L. Y. Zhang, C. Chen, S. Hu, A. Gill, and S. Pan, "Large language model watermark stealing with mixed integer programming," *CoRR*, vol. abs/2405.19677, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2405.19677

[28] Q. Pang, S. Hu, W. Zheng, and V. Smith, "Attacking LLM watermarks by exploiting their strengths," *CoRR*, vol. abs/2402.16187, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.16187

[29] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi, "Can ai-generated text be reliably detected?" *CoRR*, vol. abs/2303.11156, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2303.11156

[30] J. Piet, C. Sitawarin, V. Fang, N. Mu, and D. A. Wagner, "Mark my words: Analyzing and evaluating language model watermarks," *CoRR*, vol. abs/2312.00273, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2312.00273

[31] L. Pan, A. Liu, Z. He, Z. Gao, X. Zhao, Y. Lu, B. Zhou, S. Liu, X. Hu, L. Wen, I. King, and P. S. Yu, "Markllm: An open-source toolkit for LLM watermarking," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - System Demonstrations, Miami, Florida, USA, November 12-16, 2024*, D. I. H. Farías, T. Hope, and M. Li, Eds. Association for Computational Linguistics, 2024, pp. 61–71. [Online]. Available: https://doi.org/10.18653/v1/2024.emnlp-demo.7

[32] S. Dathathri, A. See, S. Ghaisas, P. Huang, R. McAdam, J. Welbl, V. Bachani, A. Kaskasoli, R. Stanforth, T. Matejovicova, J. Hayes, N. Vyas, M. A. Merey, J. Brown-Cohen, R. Bunel, B. Balle, A. T. Cemgil, Z. Ahmed, K. Stacpoole, I. Shumailov, C. Baetu, S. Gowal, D. Hassabis, and P. Kohli, "Scalable watermarking for identifying large language model outputs," *Nat.*, vol. 634, no. 8035, pp. 818–823, 2024. [Online]. Available: https://doi.org/10.1038/s41586-024-08025-4

[33] C. Gu, X. L. Li, P. Liang, and T. Hashimoto, "On the learnability of watermarks for language models," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=9k0krNzvlV

[34] Y. Liang, J. Xiao, W. Gan, and P. S. Yu, "Watermarking techniques for large language models: a survey," *Artif. Intell. Rev.*, vol. 59, no. 2, p. 74, 2026. [Online]. Available: https://doi.org/10.1007/s10462-025-11474-6

Appendix

A Details of Watermarking Algorithms **14**

B Details of Baselines **15**

B.1 Water-Probe [4] 15

B.2 Gloaguen et al. [10] 15

C Additional Experimental Details **16**

C.1 Sampling and Evaluation Configuration 16

C.2 Hyperparameters of **SCTI** 16

C.3 Watermarking Algorithm Configuration 16

D Additional Experimental Results **16**

D.1 Extended Diagnostic Results on ***TriVC*** 16

D.2 Extended Diagnostic Results on ***PNC*** 17

D.3 Extended Correlation Heatmaps of ***TriVC*** Views 18

D.4 Compute Resources 18

E Broader Impacts, Security Scope, and Deployment Context **18**

F Limitations **19**

A Details of Watermarking Algorithms

We provide a brief description of the watermarking algorithms used in our experiments. These algorithms can be broadly divided into two categories: N-Gram based watermarking and Fixed-Key-List based watermarking. N-Gram based methods determine the watermark key from the preceding tokens, while Fixed-Key-List based methods use a pre-defined key list during generation.

KGW [1]. The Key-based Green-list Watermarking (KGW) algorithm partitions the vocabulary into a red list and a green list according to a secret key and the preceding token context. During generation, the logits of green-list tokens are increased, so the watermarked model is more likely to sample green tokens. The watermark can then be detected by checking whether the generated text contains a statistically unusual preference for green-list tokens.

KGW-Min [12]. KGW-Min is a variant of KGW that changes how the context is used to determine the watermark partition. Instead of using only the immediate previous token, it selects the minimum token ID within a window of preceding tokens to construct the red and green lists. This design still follows the KGW principle of promoting green-list tokens, but uses a different context rule to define the watermark key.

KGW-Skip [12]. KGW-Skip is another KGW variant that derives the watermark partition from a skipped context position within a window of preceding tokens. In our implementation, the partition is determined by the left-most token ID in the selected context window. Similar to KGW, the generated text is watermarked by increasing the probability of green-list tokens during sampling.

AAR [6]. The AAR watermark uses a pseudo-random vector generated from the preceding token context to guide token selection. At each generation step, the pseudo-random vector interacts with the model probability distribution and selects a token through exponential minimum sampling. This method is designed to embed a detectable signal while preserving the generated distribution in expectation.

γ-Reweight [7]. The γ-Reweight watermark first shuffles the token probability vector using a seed determined by the preceding N-Gram context. It then discards a fixed portion of the shuffled probability mass and reweights the remaining part before sampling. Because the shuffle is determined by the context-dependent key, repeated generation under the same effective key can introduce a consistent watermark pattern.

DiPmark [8]. DiPmark is a distribution-preserving watermark that also relies on a context-dependent random shuffle of the token probability vector. After shuffling, it removes an α portion of the probability vector and amplifies the remaining probabilities by $1/(1-\alpha)$. Compared with standard reweighting, DiPmark uses this controlled reweighting strategy to preserve text quality while keeping the watermark statistically detectable.

WaterBag [4]. WaterBag is designed to improve the imperceptibility of N-Gram based watermarking by increasing the randomness of watermark key selection. Instead of using a single master key, it samples from a bag of master keys and their inverse keys during generation. The inverse keys are constructed so that the average effect of paired keys is close to the original model distribution. As a result, WaterBag makes repeated sampling under the same effective key more difficult, which provides a stronger test case for black-box watermark identification.

EXP-Edit [5]. EXP-Edit is a Fixed-Key-List based watermarking algorithm built on exponential minimum sampling. It constructs a fixed list of pseudo-random vectors and randomly selects a starting position in this list for each generation. The vectors are then used sequentially to guide token sampling. During detection, the generated text is aligned with the pseudo-random vector list, and edit distance is used to measure whether the text carries the watermark signal.

ITS-Edit [5]. ITS-Edit follows the same Fixed-Key-List framework as EXP-Edit, but uses inverse transform sampling instead of exponential minimum sampling. It also relies on a fixed pseudo-random vector list and a randomly selected starting position during generation. The watermark is detected by checking whether the generated token sequence is statistically consistent with the hidden key sequence.

Beyond the token-level watermarking algorithms, sentence-level watermarking methods have also been explored, including SemStamp [20], k-SemStamp [21], and SWAN [22]. These methods incor-

porate watermark signals into sentence-level semantic representations, providing a complementary direction to token-level watermarking.

B Details of Baselines

This appendix provides additional details on the two black-box watermark identification baselines used in our main comparison. Both baselines only require query access to the target LLM and do not assume access to model parameters, logits, or watermark keys. Water-Probe [4] provides a unified prompt-based identification pipeline, while Gloaguen et al. [10] provide statistical tests designed for different watermark families. Since the two methods report their evidence in different forms, we also describe how their outputs are presented in our main tables.

Related black-box settings also include watermark stealing and watermark removal [23, 24]. Unlike these methods, which respectively aim to infer hidden watermark rules or suppress watermark signals, **SCTI** focuses on identifying watermark presence without reconstructing the underlying watermark mechanism.

B.1 Water-Probe [4]

Water-Probe [4] identifies whether an LLM is watermarked by constructing crafted prompts and comparing the consistency of output changes across prompts under repeated sampling. The key intuition is that, if a generation-time watermark is present, repeated samples under the same effective watermark key tend to produce consistent differences across related prompts. In contrast, for an unwatermarked LLM, such cross-prompt consistency should remain weak.

In our experiments, we use Water-Probe-v2 as the main Water-Probe baseline, since it is designed to handle both N-Gram based watermarking and Fixed-Key-List based watermarking, which matches the two watermark categories evaluated in our benchmark. For each tested LLM and watermark setting, Water-Probe-v2 returns a detection score in z-score form. Following the original Water-Probe convention, larger z-scores indicate stronger evidence that the queried LLM is watermarked. Therefore, the Water-Probe entries in our main table are directly reported as the z-scores produced by Water-Probe-v2.

For presentation, we use the same confidence convention as Water-Probe. A score between 4 and 10 is treated as moderate-confidence watermark identification, and a score above 10 is treated as high-confidence watermark identification. Scores below 4 are treated as insufficient evidence for identifying a watermark. This convention is used only to make the table easier to read; the numerical scores are still reported directly.

B.2 Gloaguen et al. [10]

Gloaguen et al. [10] study black-box watermark detection through statistical tests. Different from Water-Probe, their method is not a single unified pipeline for all watermarking algorithms. Instead, it provides separate tests for different watermark families, including Red-Green, Fixed-Sampling, and Cache-Augmented watermarks.

In our main comparison, we use the Red-Green test from Gloaguen et al. as the statistical-test baseline and apply it to the KGW setting. This choice is aligned with the watermark mechanism, because KGW belongs to the Red-Green watermark family.

The Red-Green test queries the target LLM with prompts that separate prefix and context effects. It then estimates the model's choice behavior from repeated black-box responses and tests whether the observed changes contain context-dependent bias, which is the characteristic signal of a Red-Green watermark. The original test reports a p-value through a Monte Carlo permutation test, where a smaller p-value corresponds to stronger evidence for the presence of a Red-Green watermark.

Since our main tables report detection scores rather than p-values, we convert the p-value from the Red-Green test into a normal-equivalent score for presentation consistency. This conversion preserves the evidence direction: smaller p-values become larger scores, so all methods in the table follow the same convention that larger values indicate stronger watermark evidence. When a Monte Carlo

p-value is reported as zero due to finite numerical resolution, we clip it at the smallest resolvable value before conversion.

C Additional Experimental Details

This appendix provides additional details of the experimental setup, including the sampling configuration, hyperparameters of **SCTI**, and watermarking algorithm configurations used in our experiments.

C.1 Sampling and Evaluation Configuration

For all main experiments, we use the same matched prompt pair and simulated-key protocol across watermarking algorithms. The sampling temperature is set to 1.0, following the main setting used by Water-Probe [4]. For each prompt and simulated key, we collect 10^4 sampled responses.

All experiments are conducted under the black-box response-only setting. The detector only observes the generated responses of the queried LLM and does not access model logits, model parameters, detector keys, or internal watermark configurations. For open-source models, logits are used only when generating watermarked responses according to the specified watermarking algorithms; the identification procedure itself uses only sampled responses.

C.2 Hyperparameters of SCTI

The main hyperparameters of **SCTI** are fixed across all models and watermarking algorithms. For permutation null calibration, we set the number of random permutations to $B = 256$. This value provides a stable empirical null distribution while keeping the computational cost manageable. For the RBO view in ***TriVC***, the ranked list length is set to $K = 50$. That is, for each prompt p_i, L_i^q contains the top 50 tokens with the largest rank changes under key pair q. The RBO persistence parameter is set to $p = 0.9$, which assigns larger weights to overlap near the top of the ranked lists.

C.3 Watermarking Algorithm Configuration

We evaluate **SCTI** on both N-Gram based and Fixed-Key-List based watermarking algorithms. For N-Gram based watermarking, we test KGW [1], KGW-Min and KGW-Skip [12], AAR [6], DiPmark [8], γ-Reweight [7], and WaterBag [4]. For Fixed-Key-List based watermarking, we test EXP-Edit and ITS-Edit [5]. The detailed descriptions of these watermarking algorithms are provided in Appendix A.

For KGW, we use $\gamma = 0.5$ and $\delta = 2.0$. For KGW-Min, the context window size is set to 4, and for KGW-Skip, the context window size is set to 3. For AAR, we use prefix length of 1. For DiPmark, we set $\alpha = 0.45$. For γ-Reweight, we follow the unbiased reweighting configuration used in the corresponding implementation. For EXP-Edit and ITS-Edit, the key-list length is set to 420, following the Water-Probe evaluation setting [4].

For WaterBag, we use KGW as the base watermarking rule and increase key randomness through a bag of master keys and their inverse keys. The parameters are $\gamma = 0.5$, $\delta = 2.0$, prefix length 4, and bag size 4. We do not tune these parameters for individual LLMs or watermark families.

D Additional Experimental Results

This section provides additional experimental results that complement the diagnostic experiments in the main paper. First, we report the ***TriVC*** view ablation across evaluated LLMs and watermark algorithms. Second, We provide the ***PNC*** calibration ablation for the cosine view. Third, we include the complete pairwise correlation heatmaps for the three ***TriVC*** views on Qwen3-14B.

D.1 Extended Diagnostic Results on *TriVC*

Tab. 4 reports an extended ablation of ***TriVC*** view variants across representative LLMs and watermark schemes. Besides the cosine view, sign view, and RBO view, the table also includes two-view combinations to examine whether partial fusion is sufficient. The results show that different views contribute useful but uneven evidence. For example, the sign view often gives strong watermarked

scores, but it can also raise the unwatermarked score, such as on Llama-3.2-3B and Qwen3-14B. Cosine and RBO provide complementary evidence, but they may be weaker on some watermark schemes. Two-view combinations improve over single views in several cases, yet they still do not consistently balance watermark identification and unwatermarked control. By integrating all three views, ***TriVC*** keeps the unwatermarked scores low while preserving strong evidence across both N-Gram and Fixed-Key-List based watermarking algorithms. These results support the use of trident-view consistency rather than relying on a single view or a partial view combination.

Table 4: Ablation study on ***TriVC*** view variants across LLMs and watermark schemes.

LLM	Method		N-Gram							Fixed-Key-List	
		Non	KGW	AAR	KGW-Min	KGW-Skip	DiPmark	γ-Reweight	WaterBag	EXP-Edit	ITS-Edit
Qwen3-1.7B	Cosine	-3.25	16.93	100.35	4.10	29.59	14.31	9.82	19.65	23.04	7.11
	Sign	1.65	55.00	95.01	16.93	50.67	18.20	18.88	37.83	25.17	19.56
	RBO	-1.18	34.97	59.10	8.05	27.07	14.68	6.79	28.55	16.40	17.76
	Cosine + Sign	-0.83	36.53	97.35	11.53	41.64	17.37	15.41	29.63	21.73	14.45
	Cosine + RBO	-2.31	27.48	84.61	6.35	28.52	14.58	9.06	25.62	16.36	13.64
	Sign + RBO	0.72	47.62	81.74	15.27	36.26	17.85	12.97	34.81	17.83	20.64
	TriVC	-1.08	43.25	85.56	13.45	37.37	17.67	13.81	33.82	20.42	18.29
Llama-3.2-3B	Cosine	-3.62	60.54	29.81	90.81	49.18	86.11	71.48	24.64	104.15	116.14
	Sign	17.48	120.53	44.21	134.21	135.03	92.72	79.34	72.14	169.15	154.64
	RBO	2.73	80.55	33.18	94.59	89.50	76.53	62.13	31.82	133.12	140.13
	Cosine + Sign	5.72	98.74	39.86	118.70	95.31	98.28	80.56	52.54	146.72	143.24
	Cosine + RBO	-0.71	83.36	35.31	98.48	81.34	83.19	69.83	64.26	125.92	133.42
	Sign + RBO	8.53	114.49	41.72	126.41	126.24	93.50	74.90	58.11	161.47	156.72
	TriVC	1.05	102.25	38.44	122.27	113.48	94.31	78.92	50.02	151.17	151.70
Qwen3-8B	Cosine	-10.09	23.25	38.40	23.27	16.29	25.18	25.40	6.47	16.93	43.10
	Sign	-12.41	41.62	47.10	50.70	38.94	33.86	31.27	27.66	22.90	67.58
	RBO	-6.72	24.68	37.44	30.00	23.87	22.18	20.64	17.87	17.83	55.20
	Cosine + Sign	-10.54	32.54	44.19	37.95	28.54	30.49	29.37	20.49	19.84	56.19
	Cosine + RBO	-6.72	24.19	38.54	27.45	21.53	24.92	23.97	15.69	17.72	50.44
	Sign + RBO	-7.64	33.09	44.08	41.62	31.49	27.37	26.04	25.55	20.89	62.25
	TriVC	-8.75	30.54	43.97	37.70	28.58	27.76	26.49	20.44	20.36	58.47
Qwen3-14B	Cosine	-3.47	29.91	9.29	40.96	51.66	44.38	39.20	22.66	30.29	100.22
	Sign	5.14	65.77	14.51	42.80	81.20	47.68	49.97	40.65	34.41	145.34
	RBO	-0.72	42.04	12.13	34.00	29.53	41.47	39.03	30.67	26.28	138.82
	Cosine + Sign	0.38	48.53	12.54	45.60	70.32	48.25	47.61	33.26	33.82	124.86
	Cosine + RBO	-1.58	37.21	11.64	37.63	68.19	45.36	42.75	28.26	28.95	120.59
	Sign + RBO	1.85	56.28	13.01	40.47	74.31	47.31	47.22	37.23	30.81	140.83
	TriVC	0.02	50.66	13.42	40.15	71.17	47.55	45.47	34.50	30.40	129.49

D.2 Extended Diagnostic Results on *PNC*

Tab. 5 provides the calibration ablation for ***PNC*** on the cosine view. We compare ***PNC*** with two fixed-reference variants, where the reference mean is set to $\mu = 0$ or $\mu = 0.1$. These two variants examine whether a manually fixed null reference is sufficient across different LLMs and watermark algorithms.

The results show that fixed-reference calibration is sensitive to the selected reference mean. When $\mu = 0$, watermarked scores remain high, but the unwatermarked score can also be large, such as 10.33 on Llama-3.2-3B and 9.65 on Qwen3-14B. This indicates that an ideal null reference does not always remove the consistency induced by the prompt pair, queried model, and sampling process. When $\mu = 0.1$, the unwatermarked scores are reduced, but several watermarked scores are also suppressed, especially on WaterBag and some Fixed-Key-List settings. In contrast, ***PNC*** constructs the null reference from the current observed samples through permutation. Therefore, the calibrated score is compared with the variation produced under the same test configuration, which keeps unwatermarked scores low while preserving effective identification in most watermarked settings.

Table 5: ***PNC*** calibration ablation for the cosine view across LLMs and watermark schemes.

LLM	Method		N-Gram							Fixed-Key-List	
		Non	KGW	AAR	KGW-Min	KGW-Skip	DiPmark	γ-Reweight	WaterBag	EXP-Edit	ITS-Edit
Qwen3-1.7B	Fixed $\mu = 0$	0.19	24.37	98.62	18.91	30.26	20.11	15.05	20.47	37.24	11.06
	Fixed $\mu = 0.1$	-24.39	8.61	74.38	4.26	11.82	10.42	7.52	15.61	15.71	7.55
	PNC	-3.25	16.93	100.35	4.10	29.59	14.31	9.82	19.65	23.04	7.11
Llama-3.2-3B	Fixed $\mu = 0$	10.33	61.57	29.95	91.33	61.88	89.10	70.64	31.70	115.72	115.59
	Fixed $\mu = 0.1$	-37.98	21.83	24.99	37.83	19.14	51.45	41.87	8.46	66.71	73.52
	PNC	-3.62	60.54	29.81	90.81	49.18	86.11	71.48	24.64	104.15	116.14
Qwen3-8B	Fixed $\mu = 0$	-7.49	33.86	48.24	30.01	27.15	25.57	26.19	7.61	29.27	45.81
	Fixed $\mu = 0.1$	-24.57	18.42	30.78	13.15	9.14	15.09	16.12	5.12	9.62	32.73
	PNC	-10.09	23.25	38.40	23.27	16.29	25.18	25.40	6.47	16.93	43.10
Qwen3-14B	Fixed $\mu = 0$	9.65	30.74	10.20	42.37	57.54	46.57	45.23	23.61	30.74	130.19
	Fixed $\mu = 0.1$	-13.71	15.86	7.36	25.43	36.03	31.47	29.04	7.65	14.98	75.82
	PNC	-3.47	29.91	9.29	40.96	51.66	44.38	39.20	22.66	30.29	100.22

D.3 Extended Correlation Heatmaps of *TriVC* Views

Fig. 5 provides the pairwise Spearman correlation heatmaps among the cosine, sign, and RBO views on Qwen3-14B across watermark algorithms. The off-diagonal correlations are consistently positive, showing that the three views respond to related watermark-induced changes. However, these correlations remain moderate and vary across schemes, with values roughly ranging from 0.24 to 0.53. This indicates that the views are not redundant. Therefore, the heatmaps further support the design of ***TriVC***, which combines global profile alignment, local directional agreement, and overlap among tokens with large rank changes.

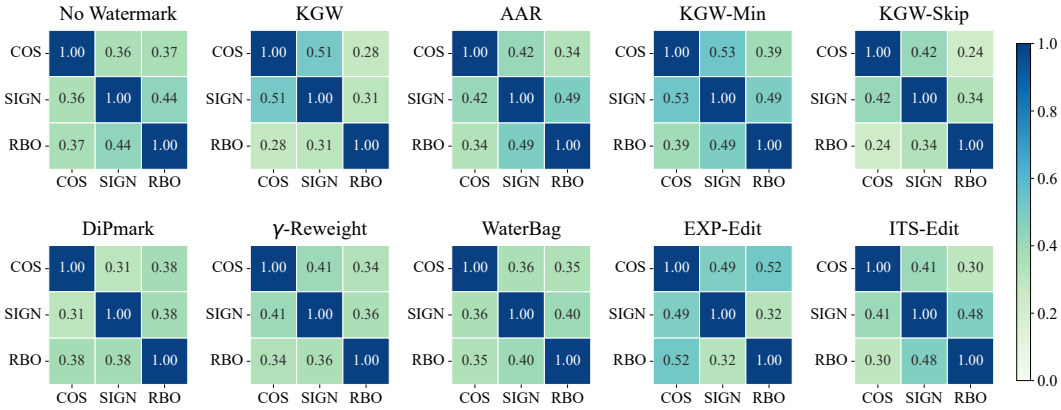


Figure 5: Pairwise Spearman correlations among the cosine, sign, and RBO views on Qwen3-14B across watermark schemes.

D.4 Compute Resources

The main computational cost of our experiments comes from local LLM inference for collecting prompt-response samples. For the largest evaluated model, Qwen3-14B, we used two NVIDIA RTX A6000 GPUs, with approximately 25GB of GPU memory used on each GPU during inference. Generating the prompt-response samples for one complete Qwen3-14B detection run took approximately 0.5 hours.

E Broader Impacts, Security Scope, and Deployment Context

Broader Impacts and Security Scope. This work studies black-box watermark identification, where an external auditor queries an unknown LLM and estimates whether a generation-time watermarking mechanism is deployed. This capability has positive societal value because it can improve transparency around watermark deployment, support independent auditing, and help researchers evaluate whether watermarking remains identifiable in realistic query-only settings. At the same time, watermark identification also has potential misuse risks. Prior work [25] has shown that information about a watermark can make removal, spoofing, or scrubbing attacks easier. Other attacks weaken watermark evidence by exploiting token-level watermark structure, editing generated text, or recovering green-list information [26, 27, 28, 29]. Therefore, **SCTI** should be interpreted as a black-box auditing method, not as a method for recovering secret keys, removing watermarks, or producing adversarial text.

Relation to Watermark Evaluation and Deployment. The setting studied in this paper is different from standard watermark evaluation and deployment-oriented watermark design. Benchmark and toolkit efforts such as Mark My Words [30] and MarkLLM [31] evaluate or implement watermarking algorithms, covering aspects such as detectability, robustness, quality impact, and practical usability. Production-oriented watermarking systems such as SynthID-Text [32] study how to embed and detect watermarks at scale while preserving generation quality. Publicly verifiable watermarking schemes further change the verification interface by building public verification support into the watermarking mechanism itself [17, 18, 19]. These directions are complementary to **SCTI**: they mainly address how

to design, deploy, verify, or evaluate watermarking schemes, while **SCTI** asks whether watermark deployment can be identified from sampled responses alone.

Future Extensions. The current experiments focus on representative N-Gram and Fixed-Key-List based generation-time watermarking algorithms. Future work can extend the benchmark to learned watermarking and production-oriented watermarking schemes [33, 32]. Another useful direction is to evaluate black-box identification under stronger deployment variations, including adaptive decoding policies, rewriting-based post-processing, and adversarial attempts to hide watermark evidence [29, 26, 28, 34]. These extensions would help clarify when watermark deployment is identifiable in practice and when stronger watermark designs are needed.

F Limitations

SCTI is designed for black-box identification of generation-time watermarking mechanisms. We assume that the auditor can repeatedly query the same LLM service through a normal prompt-response interface, and that the service returns genuine outputs under a stable sampling configuration. **SCTI** does not require access to logits, parameters, secret keys, or internal watermark settings. However, it also does not aim to remove watermarks, recover watermark keys, or verify whether a given text is watermarked. These tasks require different evidence and are outside the scope of this work.

SCTI relies on repeated sampling from open-ended prompts to estimate profiles. Therefore, it is less suitable for settings where the API returns heavily filtered responses, because such outputs provide limited variation for measuring cross-prompt consistency. The query cost may also increase when the watermark uses highly random key selection or when the watermark effect is weak under the tested prompts. Although ***PNC*** reduces the influence of prompt-specific and model-specific variation through empirical null calibration, it cannot create watermark evidence when the queried outputs do not contain enough observable variation.

Our experiments are mainly conducted in controlled open-source settings, where the watermark labels and watermarking algorithms are known for evaluation. For closed-source services, the true watermark deployment status is usually unavailable. Therefore, a low **SCTI** score should be interpreted as no detected watermark evidence under the tested query configuration, rather than a formal proof that the service is unwatermarked. Future work could evaluate **SCTI** under broader API conditions, adaptive service providers, and watermark designs that intentionally reduce repeated consistency across queries.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions and scope are accurately made in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a dedicated limitations section in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose all the information needed to reproduce the main experimental results of the paper in the experimental section and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide code in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We disclose all the information needed to reproduce the main experimental results of the paper in the experimental section and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main experimental results report calibrated identification scores with confidence criteria based on empirical null distributions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the usage in appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We report this in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all related assets including models, data, code and baselines.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide code and experimental scripts in our repository.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development does not use LLMs as an important, original, or non-standard methodological component; LLMs are only the queried target models in our watermark identification experiments.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.