Bypassing LLM Watermarks with Color-Aware Substitutions

Anonymous ACL submission

Abstract

Watermarking approaches are proposed to identify if the text being circulated is human or large language model (LLM) generated. The state-of-the-art strategy of Kirchenbauer et al. (2023a) biases the LLM to generate specific "green" tokens. However, determining the robustness of this watermarking method is an open problem. Existing attack methods do not incorporate color information (if a token is green/not) and may fail to evade longer text watermark detection. We propose *Self Color Testing-based Substitution (SCTS)*, the first "color-aware" attack. SCTS gets color information by strategically prompting the watermarked LLM and comparing output frequencies, using which it can determine token colors. It then substitutes green tokens with red ones. In our experiments, SCTS successfully evades watermark detection using fewer edits than related work. Additionally, we show both theoretically and empirically that SCTS can remove the watermark for arbitrarily long watermarked text.

1 Introduction

Large language models (LLMs) are pervasively used in applications related to education (Hadi et al., 2023) and programming (Fan et al., 2023). As with most technologies, however, they have dual use—for generating misinformation (Chen and Shu, 2023) and facilitating academic dishonesty (Lancaster, 2021). Identifying LLM-generated content is a subject of immense academic interest, and a variety of methods have been proposed. Paraphrasing Yang et al. (2023), "detection techniques have witnessed advancements propelled by innovations in zero-shot methods, fine-tuning" etc.

Among these techniques, the most promising is watermarking (Aaronson, 2022), as other techniques are susceptible to a large number of false positives (Aaronson, 2022). This problem is further exacerbated when the LLM's output more closely mirrors human content (Zhang et al., 2023; Ghosal et al., 2023). Watermarking, on the other hand, embeds information into the generation by manipulating the decoding process. For example, the watermark of Kirchenbauer et al. (2023a,b) biases the logits to boost the probability of specific tokens (denoted "green tokens") in contrast to the remaining "red" tokens. The set of green tokens is determined by a hash function, and the security of this watermark is based on the hardness of finding collisions. Despite the emergence of more watermarking schemes (Kuditipudi et al., 2023; Christ et al., 2023), research shows that methods that (slightly) modify logit distributions outperform others without significant output quality degradation (Piet et al., 2023). Humans can not identify if the text is watermarked, and detection requires processing approximately one hundred tokens. Thus, for the remainder of this work, we focus on the Kirchenbauer et al. (2023a) watermark and its variant by Zhao et al. (2023) as exemplars.

We study *if watermarking approaches based on logit perturbation are robust to post-processing distortions.* This is an important question across various applications. For example, it will reliably determine if academic integrity was violated in educational applications: if the watermark is not robust, a student can easily edit the generated watermarking text, and the resulting data will fail verification. While prior works also ask this or similar questions (Sadasivan et al., 2023; Lu et al., 2023), they do so under unreasonable settings, where edits (or post-processing distortions) to watermarked text are unconstrained (i.e., edit large fractions of the generated text with potential degradation to output quality). They use dedicated paraphrasing models (Sadasivan et al., 2023), or design customized prompts for post-hoc output paraphrasing (Lu et al., 2023; Shi et al., 2023). Additionally, they design prompts to generate hard-to-detect text (which is often contextually unrelated), sometimes with low entropy (Lu et al., 2023; Shi et al., 2023).

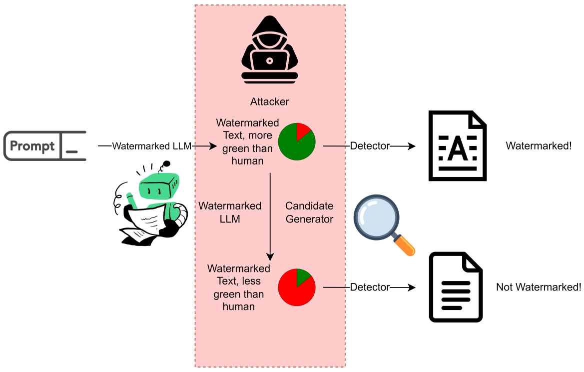

Figure 1: Illustration of the setting for SCTS. The red box indicates the attacker's capability.

We evaluate robustness under a more realistic setup: post-hoc editing involves making a "reasonable" number of edits to maximally retain information generated by the LLM. To do so, we make two observations. First, previous methods are *color-agnostic*: existing approaches that replace text via paraphrasing still preserve some fraction of the original text, resulting in dilution rather than erasure of watermarks (Kirchenbauer et al., 2023b) (i.e., larger number of edits is needed for erasure). Even if they manage to evade detection for certain samples by removing "most" fragments, our experiments demonstrate that they would fail when constrained by a smaller edit budget (§ 6). Second, prior attacks require another high-quality "unwatermarked" LLM for paraphrase generation which generates verbose text, and is not very practical as it can be used for generation in the first place.

Building atop these observations, we propose the **Self Color Testing-based Substitution (SCTS)** attack. This introduces new text fragments that contain fewer green tokens than human-written text with high probability. We prompt the LLM to perform targeted (periodic) generations given input context (refer prompt in § 4) to get color information from the frequency in the generated text. This is used to perform color-aware substitution, and in turn neutralizes the higher number of green list tokens from the preserved fragments. Consequently, SCTS can evade watermark detection for arbitrarily long text segments *within a reasonable edit distance budget, without using an unwatermarked LLM*. Our evaluation compares SCTS and existing representative attack methods over a series of edit distance budgets. We conclude that across various of comparison, our approach is superior in reducing AUROC to less than 0.5 on two LLMs and watermarking schemes. Additionally, and most importantly, our approach is theoretically grounded, with a comprehensive analysis.

In summary, our main contributions are:

1. We theoretically show how existing color-agnostic methods can only "dilute" watermarks, and can not evade detection when the watermarked text is sufficiently long. Empirically, we show that existing methods fail to evade watermarks within reasonable edit distance budgets (§ 3).
2. We propose the first "color-aware" attack method by prompting the LLM for random generation and trying to only replace the green tokens/words with red tokens/words. We then analyze the working mechanism of our attack, and estimate its efficacy and costs under reasonable edit distance constraints (§ 4).
3. We conduct extensive experiments on `vicuna-7b-v1.5-16k` (Zheng et al., 2023) and `Llama-2-7b-chat-hf` (Touvron et al., 2023) with different hashing strategies. We show that under the same edit distance budget, our attack is more effective in evading watermarks than previous methods (§ 6).

2 Background and Related Work

2.1 Background: Text Watermarking

We utilize the work of Kirchenbauer et al. (2023a) as an exemplar approach; it involves perturbing the logits, so as to guide generation towards specific tokens. Their approach works as follows: an input sample $\mathbf{w} = w_{-s+1}, w_{-s+2}, \cdots, w_0$ of s words is tokenized to obtain s' tokens. These are then fed to the the large language model (LLM) which generates T tokens as the response. Note that tokens are generated one by one; more specifically, before the generation of every output token, c tokens (where c is known as the context size) preceding that are used to seed a pseudo-random (hash) function which is used to divide the vocabulary space of size $|V|$[1], dynamically, into a green and red list, where the size of the green list is $\gamma|V|$. Based on reweighting green tokens using an offset δ (more details in §3 of Kirchenbauer et al. (2023a)), the approach prioritizes the selection of a green list

[1]This is the vanilla left hash. A preferred variant is to use the token being generated also as input to the hash, which is called self hash. Self hash is generally empirically more robust with the additional cost to hash more than once using different $c + 1$ grams.

token as the next candidate. This process is repeated to generate all T output tokens. The aim of their approach is to ensure that the number of green tokens in the generation ($||T||_G$) is higher than the color-agnostic expectation. Post-hoc, this is verified using a statistical test:

$$z = \frac{\|T\|_G - \gamma T}{\sqrt{T\gamma(1-\gamma)}}$$

The text is labeled as watermarked if and only if z is greater than some threshold.

2.2 Attacks against Watermarks

Most, if not all attacks involve replacing a subset of words.

Paraphrasing: These attacks aim to replace a group of words with semantically similar counterparts. This can be done directly using a specialized LLM (Krishna et al., 2023; Sadasivan et al., 2023), word-level substitutions (Shi et al., 2023), or translation (to another language and back) (Christ et al., 2023). For example, the recursive paraphrasing (RP) approach (Sadasivan et al., 2023) paraphrases (up to 5 iterations) the watermarked text using a dedicated, unwatermarked paraphrasing model.

Prompts: Another class of attacks involves carefully designing prompts to guide the model to evade detection (Lu et al., 2023), or to guide the LLM to generate low-entropy text which is hard to watermark. As an example of the first category, Lu et al. (2023) propose SICO-Para (SICO), where the LLM is tasked with generating features of human-written text. Using such features and human-written style examples, it can guide the LLM to augment the AI-generated text to be more human-like. SICO alternatively performs sentence and word-level updates, to greedily minimize the probability of detection using a proxy. As such a proxy is usually not a watermark detector and focuses on the semantics, it would not help when attacking watermarks.

In our work, we focus on RP and SICO as representative baselines.

Limitations of Current Approaches: Both these approaches do not utilize knowledge of the context size parameter c in their approach. Additionally, both approaches are *color-agnostic* i.e., meaning that the fragments introduced by the attacker will be independent of the color (i.e., $z \sim 0$) of the large number existing fragments (i.e., $z > 0$) in expectation. This introduces natural tensions. Most importantly, both approaches are ineffective when there are constraints placed on the number of permitted edits. For example, we see that both SICO and RP's AUROC is still ≥ 0.86 under the most relaxed 0.5 normalized edit distance (edit of 50% of words for the watermarked text)[2].

2.3 Threat Model

Recall that robustness of watermarks is defined as their tolerance to edits post-hoc. Prior work (Lu et al., 2023) evaluates robustness by making unrealistic assumptions about the (robustness) adversary's capabilities. They assume that the adversary has access to a version of an LLM without a watermarking algorithm. In our work, we do not make this strong assumption. Like other prior approaches (Zhang et al., 2023; Sadasivan et al., 2023), we assume:

1. API access to the watermarked model, using which we can issue input prompts, and observe the generated responses.
2. The watermarked model is aligned, and capable of following instructions provided.
3. Knowledge of the context size c.
4. No knowledge of other watermarking hyperparameters, like γ, temperature t, and δ.
5. Access to some model (not necessarily an LLM) capable of generating word substitution candidates. This model can be watermarked.

We believe assumption 2 is realistic, given how most models in the status quo are instruction finetuned, and trained using reinforcement learning with human feedback (Lambert et al., 2022). We also stress that assumptions 2 and 3 are not strict.

3 The Building Blocks

We will first show some properties of the watermarking efficacy as a function of output length, and use this to explain why existing color-agnostic attack methods fail for sufficiently long watermarked text. Then, we focus on substitution-based attack and formalize the color-agnostic baseline. But before we begin, we outline the assumptions and definitions we make throughout this section.

Assumption 1: We consider the left hash for ease of exposition. As stated earlier, the left hash is one where the the green list for the current token is

[2] AUROC describes the distinction of the z-scores for (attacked) watermarked text and unwatermarked text and it is more comprehensive than success rate.

obtained by hashing c tokens counting backwards from the previous token.

Assumption 2: Each (generated) token is green with constant probability p in watermarked text, i.i.d, s.t. $\gamma < p < 1$. In the output, the number of c-grams is $T - c$.

Definition 1 ($c+1$-gram)**.** *This is the minimum number of tokens used for color testing. It includes the token whose color is to be checked, and the c tokens before it. The color of a $c+1$-gram refers to the color of the last token, or equivalently, the color result when this $c+1$-gram is sent to the detector.*

Definition 2 (Effective length T_e)**.** *This is the number of $c+1$-grams used for color testing i.e., $T_e = T - c$*

Definition 3 (Detection threshold z_{th})**.** *This is the threshold used in detection. The detector outputs "watermarked" $\iff z > z_{th}$.*

Definition 4 (Critical length T_c)**.** *This is the value of T_e such that $\mathbb{E}[z] = z_{th}$. This is roughly the effective length needed for successful detection, and may or may not exist.*

Definition 5 (Average green probability q)**.** *: This is the average probability for a $c+1$-gram being green in a given sample.*

3.1 Theorem: Watermark Strength

Theorem 1. *$\mathbb{E}[z]$ is proportional to $\sqrt{T_e}$. Assume the colors of $c+1$-grams are independent, we have:*

- *For $q < \gamma$, the probability of detection as "watermarked" converges exponentially to 0 with respect to T_e.*
- *For $q > \gamma$, the probability of being detected as "unwatermarked" converges exponentially to 0 with respect to T_e.*

Furthermore, if the probability of green for different $c+1$-grams is identically and independently distributed (i.i.d.), then:

- *For $q \neq \gamma$, tighter bounds are applicable in comparison to the scenarios described above in the independent case.*
- *For $q = \gamma$, the probability of being detected as "unwatermarked" converges to a constant determined by z_{th}.*

The proof of the above is in Appendix A.

3.2 Why Existing Methods Fail For Long Text

As a warm-up, let us consider text that is not attacked. Applying Theorem 1, where $q = p > \gamma$ (for the i.i.d. case), we can see that the false positive rate converges to 0 exponentially w.r.t. T_e.

Candidate Attack 1. There are attacks that design prompts to guide the watermarked LLM to generate low-entropy text to evade detection. It is hard to incorporate watermarks in such text segments. But even if one can circumvent this challenge, we will see that it may only dilute the watermark.

Assume that the attacker can reduce p to p', such that $p > p' > \gamma$, i.e., the $c+1$-grams are i.i.d green with probability p'. Note still $p' > \gamma$, apply theorem 1, where we set $q = p' > \gamma$ (for the i.i.d. case as before), still the false positive rate converges to 0 exponentially w.r.t. T_e.

Conclusion. This attack would fail for sufficiently long watermarked text, and it can only dilute the watermark.

Candidate Attack 2. Here, the attacker uses color-agnostic post-processing methods to evade detection. Existing paraphrasing-based methods (Lu et al., 2023; Krishna et al., 2023) fall into this category. Since the post-processing is color-agnostic, we assume the newly generated $c+1$-grams are green with i.i.d. probability γ, and the number of such *new $c+1$-grams* is T_{new}. Also note that post-processing attacks such as paraphrasing are statistically likely to leak n-grams or even longer fragments of the original text (Kirchenbauer et al., 2023b). We call the leaked segments *old $c+1$-grams*. These old $c+1$-grams are green with i.i.d. probability p, and the number of such old $c+1$-grams is T_{old}. The existence of two classes of $c+1$-grams suggests that when detecting the watermarking from the attacked text, $T_e = T_{old} + T_{new}$. The ratio of leaked $c+1$-grams is $r_o = \frac{T_{old}}{T_{old}+T_{new}}$ is lower bounded by constant $r > 0$. Finally, the colors for different $c+1$-grams categories are independent.

Definition 6. *We define*

- *$\|T\|_G^{old}$, the random variable for the number of green old $c+1$-grams.*
- *$\|T\|_G^{new}$, the random variable for the number of green new $c+1$-grams.*

So $\|T\|_G^{old} \sim B(T_{old}, p)$, $\|T\|_G^{new} \sim B(T_{new}, \gamma)$, $\|T\|_G = \|T\|_G^{old} + \|T\|_G^{new}$.

Applying the independent case of the Theorem 1 ($q \geq \gamma$), we can see that the probability of evading detection exponentially converges to 0 w.r.t. T_e.

Remark 1. In paraphrasing attacks, we can safely assume that attack makes the text longer.

Remark 2. Even if attack 1 and attack 2 are combined, i.e., the attacker can weaken the watermark and do color-agnostic post-processing, then it would still fail for sufficiently long text. To see this, just let $p = p' > \gamma$ in the above derivation.

Conclusion This attack would fail for long enough generated text when a non-zero ratio of $c+1$-grams is preserved.

4 Our Approach: Self Color Testing

We first introduce some notation. The watermarked sentence (i.e., output of the LLM) is denoted $\mathbf{w} = \{w_1, w_2, \cdots, w_T\}$. In this sentence, the word being substituted is denoted w_b. The candidate substitution is w'_b. Finally, $\mathbf{w}^c_b$ denotes the context of w_b, which determines the green list at w_b. $\mathbf{w}^c_b$ is the c words before w_b (assuming left hash)

Step 1. Self Color Testing: The secret to a successful attack against watermarked LLMs is to have color information. However, obtaining this is not straightforward. We leverage the insight that aligned and instruction fine-tuned models are compliant with user instructions. Thus, we can prompt the LLM (at temperature $t = 0$) to generate seemingly random strings (in a deterministic manner) with customized input prefixes, and infer color information from the frequency of outputs generated (abstracted by the "Count" method). To this end, we focus our discussion on the word level to avoid some encoding issues and incomplete word biases. Algorithm 1 contains all relevant details.

Algorithm 1: Self Color Testing (SCT)

Input: $w_b, w'_b, \mathbf{w}^c_b = \{w_{b-c} \dots w_{b-1}\}, p_{th}$;
Output: Test result in {GR, RG, S}. GR (RG) means w_b is green (red) while w'_b is red (green);
$p = \text{MakePrompt}(w_b, w'_b, \mathbf{w}^c_b)$;
$o = \text{LLM}(p)$;
$c_b, c'_b = \text{Count}(o, \mathbf{w}^c_b w_b), \text{Count}(o, \mathbf{w}^c_b w'_b)$;
$p_t = \chi^2\text{-test}(c_b, c'_b)$;
if $p_t \geq p_{th}$ **then**
  return S;
end
else if $c_b > c'_b$ **then**
  return GR;
else
  return RG;
end

To explain the intuition behind the algorithm, consider the following example prompt for color testing. Here, note that (w_b=`includes`, w'_b=`contains`, and $\mathbf{w}^c_b$=`kernel`)

```
Choose two phrases ( kernel includes, kernel contains ), and generate a long uniformly random string of those phrases separated by ";". Previous phrases should have no influence on future phrases: kernel includes; kernel contains; kernel includes; kernel contains; kernel contains; kernel includes; kernel contains; kernel contains; kernel contains; kernel contains; kernel contains; kernel contains; kernel contains; kernel contains; kernel includes; kernel contains; kernel contains; kernel includes; kernel contains; kernel contains;
```

When the above prompt is processed by the LLM, we check the frequency of the response. Assume the token before "`includes`" and "`contains`" are the same in the encoded "`kernel includes`" and "`kernel contains`". Since the model is deterministic, and the model follows the prompt's term periodicity (which is alternating), the model will always generate the green token if w_b and w'_b are of different colors, so SCT has a perfect recall on GR and RG. For S there can be border cases, but in realistic applications ($t > 0$), it will have the same probability of occurring.

Step 2. SCT Substitution: We use color-testing to test different candidates to ensure that the green token is substituted by a red one. Note: (a) Generate_candidate($\mathbf{w}', i$) generates k substitution candidates (different from the original) for $\mathbf{w}'$ at index i, and (b) Substitute($\mathbf{w}', i, w'_i$) updates the i-th word of $\mathbf{w}'$ with w'_i.

The choice of step size 2 upon successful substitution is heuristic. Consider the left hash $c = 1$ case. Suppose we just substituted w_i with w'_i: this will also change the green list at w_{i+1}. It is reasonable to assume the probability of being green for w_{i+1} is reduced to γ (from p) as we did not check its color before substitution. Consequently, continuing to substitute w_{i+1} is less likely to be successful, as w_{i+1} has a lower probability of being green, which in turn will lead to more computation.

Budget Enforcement: To enforce the budget, we add checks after each replacement.

4.1 Analysis

To simplify our analysis, we assume that each word corresponds to a single token, and that the candi-

Algorithm 2: SCTS Algorithm

Input: $\mathbf{w}$,
Output: $\mathbf{w}'$
Initialization: $i = c$, $\mathbf{w}' = \mathbf{w}$
while $i \le T$ **do**
  $w'_{i,1}, \ldots, w'_{i,k}$ = Generate_candidates($\mathbf{w}'$, i)
  $success$ = False
  for $j = 1$ **to** k **do**
    if $SCT(w_i, w'_{i,j}, \{w_{i-c} \ldots w_{i-1}\})$ == GR **then**
      $\mathbf{w}'$ = Substitute($\mathbf{w}'$, i, $w'_{i,j}$)
      $success$ = True
      break
    end
  end
  if $success$ **then**
    $i \leftarrow i + 2$
    (advance by 2 for economic substitution)
  else
    $i \leftarrow i + 1$
  end
end

date generation process is independent from the red and green list, i.e., every candidate is green with i.i.d probability γ. We focus on $c = 1$ left hash for simplicity.

Expected Green Ratio. Let $q(T_e)$ be the average green probability. When the number of candidates in each substitution attempt $k > 1 + \log_\gamma \frac{1-p}{p(1-\gamma)}$, we have

$$q(T_e) \le \max\{\frac{\gamma}{2}, \frac{\gamma^k p}{1 - p + \gamma^k p}\} < \gamma \quad (1)$$

$$\lim_{T_e \to \infty} q(T_e) = \gamma - \frac{\gamma(1 - p\gamma^{k-1})}{1 + (1 - \gamma^k)p} < \gamma \quad (2)$$

Using similar techniques as theorem 1, we have

$$\log_e \Pr(z > z_{th}) < \quad (3)$$
$$-(\gamma - q(T_e))^2 \left(\sqrt{T_e} + \frac{\sqrt{\gamma(1-\gamma)} z_{th}}{\gamma - q(T_e)} \right)^2$$

for $q(T_e) < \gamma$, which holds for large enough k and/or T_e. Thus, the probability of failing to evade the watermark converges to 0 exponentially. This means that, in expectation, our method can reduce the average green probability to less than γ, and is amenable to arbitrarily long text. More details are presented in Appendix B.

We also analyze the number of LLM calls needed for our approach in Appendix C. Our method is reasonably fast, with $O(1)$ w.r.t. k, and $O(T_e)$ w.r.t. T_e in expectation.

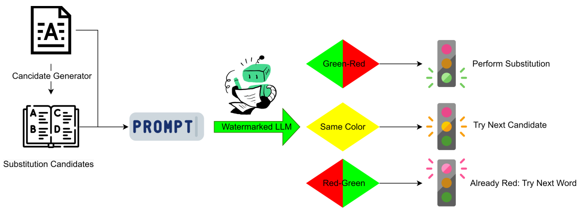

Figure 2: Illustration of one substitution in SCTS for simplicity. Take different actions depending on the frequency in the SCT test.

5 Experimental Setup

5.1 Baseline Methods

Constraints: The constraint we use is word level edit distance normalized by the length of words for the watermarked text. The constraints we use are from 0.05 to 0.5, with intervals of 0.05.

1. RP. To enforce edit-distance constraints, we check if the paraphrased sample is within the constraint. If not, we use the unparaphrased sample as the result. The other settings are as suggested by the official implementation of Sadasivan et al. (2023). RPi denotes i cycles of recursive paraphrasing; $i = \{1, 2, 3, 4, 5\}$.

2. SICO. We focus on the SICO-para variant (Lu et al., 2023). To enforce edit constraints, we add a description of the constraint in the prompt and only include compliant samples. Detailed prompt is in Appendix D. The other settings are kept the same as in Lu et al. (2023). The training samples are also the same as that used in Lu et al. (2023).

3. RB. This is a baseline for substitution without color testing. It uses the same candidate generation but without SCT. This is used to show if the color information from SCT is useful.

5.2 Models

We use `vicuna-7b-v1.5-16k` and `Llama-2-7b-chat-hf` as watermarked models. For the candidate generation model, we use the Hugging Face pipeline using `distilroberta-base` (Sanh et al., 2019). We have $k = 5$ substitution candidates for each word.

5.3 Watermarking Scheme

We consider two schemes.

1. UMD by Kirchenbauer et al. (2023a): Here, we consider both left hash (or Min-LeftHash) and self hash (or Min-SelfHash) (more details in (Kirchenbauer et al., 2023b)), and context size of $c = 1, 2, 4, 8$. As suggested by the authors, we set $z_{th} = 4$, $\gamma = 0.25$, and $\delta = 2$. For efficiency,

we batched the self hash. We consider two models: `vicuna-7b-v1.5-16k` and `Llama-2-7b-chat-hf` .

2. Unigram by Zhao et al. (2023): This is a variant of UMD's SelfHash with $c = 1$, but *whose green list is fixed*. Here, we set $z_{th} = 4$, $\gamma = 0.5$, and $\delta = 2$ (based on their released code [3]). We consider only one model: `vicuna-7b-v1.5-16k` [4].

Other Parameters The p-value threshold for the χ^2 test in SCT is set to 0.01.

More details about datasets and metrics we calculate is presented in Appendix D.

6 Experiment Results

Through our evaluation, we wish to answer the following questions: (1) Is SCTS more effective than previous methods?; (2) What budget is required for a successful attack?; and (3) Does the above hold for different target watermarked models?

We observe that: (1) SCTS is consistently more effective for different values of c, hashing schemes, and watermarking schemes, while preserving output semantics (§ 6.1); (2) a normalized edit distance of $0.25 - 0.35$ is sufficient for SCTS, while other attacks need more (§ 6.2); and (3) attack success is heightened for models which are aligned and instruction fine-tuned (§ 6.3).

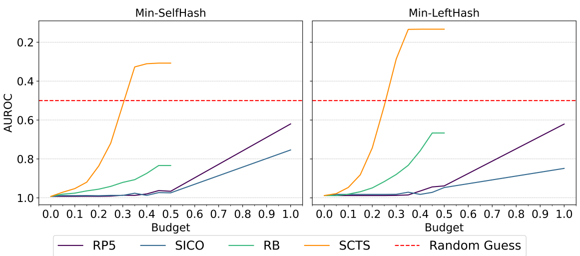

Figure 3: AUROC for `vicuna-7b-v1.5-16k`, 50 samples, UMD watermarking, $c = 4$. The orange curve (SCTS) is consistently and significantly above other baselines, and it is the only one cross 0.5.

For most metrics, we visualize the $c = 4$ case of UMD due to space constraints. Detailed results (for both UMD and Unigram) are present in Appendix E, and highlight the same trends. In particular, we observe that Unigram is more robust than UMD, but still susceptible to SCTS. For RP, we only visualize RP5 as it is the strongest attack.

[3] We reduce z_{th} from 6 to 4 to make ASR more representative

[4] For UMD self hash $c = 1$ and Unigram We slightly modify the SCTS prompt (Appendix D.5), and it will always advance by 1 regardless if the substitution succeeds.

6.1 SCTS' Efficacy

Yes, it is. Figure 3 and 5 shows the performance of the attack on AUROC and detection success respectively over different edit budgets. We can see that SCTS is consistently more effective.

At relatively high budgets like 0.35, SCTS can reduce the AUROC to less than 0.5, which means that the z-score (used for detection) is, on average, more negative. This in turn corresponds to the scenario where $q < \gamma$ in equation 2, despite some of the assumptions we made not holding.

SICO and RP, in contrast, fail to evade detection at most budgets, and are even worse than RB. Even though they work reasonably well when they are unconstrained, there are still a few detectable samples. Also, their AUROCs are still in the range of 0.6 to 0.9, suggesting that the watermark is generally only diluted and can be detected for longer text.

Impact of c and hashing scheme. Both figures show that self hash is generally more robust than left hash, especially for RB and SCTS. This is consistent with the findings in Kirchenbauer et al. (2023b), and also holds for $c = 2, 4, 8$. Also, smaller c is generally more robust from our experiment results in the Appendix E, Table 2, 3, 12, 13, consistent with the findings of Kirchenbauer et al. (2023b). Nevertheless, smaller c comes with a higher risk of leaking the green $c + 1$-grams to an attacker, more loss in generation quality (Kirchenbauer et al., 2023b), lower z and successful detection rates as shown in our experiments.

Semantic Similarity. Another key factor for a successful attack is if it can preserve semantics. From Table 1, notice our method successfully preserves semantics during the attack, with a mean cosine similarity 0.8832 and 0.8732 at 0.4 budget (for left and self hash respectively), which is comparable to RP1. Results for the Unigram watermark are slightly lower, and more details are presented in Table 11 in Appendix E. We believe that semantics will be better preserved if we use more powerful substitute generators, or make larger substitutions (phrases vs. words as we currently do).

SCT Accuracy. The key factor for our success is the accuracy of SCT. From Figure 4, we see that the accuracy for SCT is ≥ 0.5. This is significantly higher than 0.33, a rough random baseline estimate for a three-class classification problem, suggesting the effectiveness of SCT. This is particularly impressive given that Kirchenbauer et al. (2023a)'s

Table 1: Semantic similarity for `vicuna-7b-v1.5-16k`, 50 samples, UMD watermarking, $c = 4$. SCTS successfully preserves semantics.

Hashing	Method	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
Left	RP5	1.0000	0.9979	0.9978	0.9986	0.9922	0.9894	0.9881	0.9711	0.9586	0.9467	0.5436
	SICO	0.9984	0.9988	0.9906	0.9889	0.9884	0.9886	0.9934	0.9910	0.9845	0.9600	0.7104
	RB	0.9862	0.9728	0.9538	0.9394	0.9203	0.9008	0.8854	0.8732	0.8546	0.8546	-
	SCTS	0.9878	0.9743	0.9579	0.9366	0.9198	0.9018	0.8842	0.8832	0.8832	0.8832	-
Self	RP5	0.9998	0.9987	0.9988	0.9916	0.9931	0.9841	0.9709	0.9582	0.9488	0.9416	0.5464
	SICO	0.9979	0.9999	0.9925	0.9880	0.9614	0.9869	0.9811	0.9857	0.9722	0.9674	0.6439
	RB	0.9847	0.9686	0.9507	0.9330	0.9141	0.8964	0.8801	0.8611	0.8459	0.8459	-
	SCTS	0.9850	0.9673	0.9499	0.9285	0.9105	0.8891	0.8754	0.8737	0.8732	0.8732	-

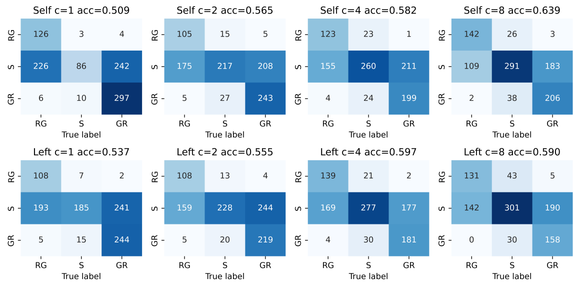

Figure 4: Confusion matrix and accuracy for SCT over 1000 samples for `vicuna-7b-v1.5-16k`, UMD. Accuracies are at least 0.5 for all c and hashing.

coloring is designed to be hard to identify.

6.2 Attack Budget

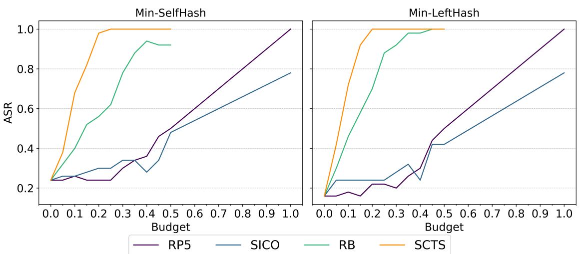

Figure 5: Attack success for `vicuna-7b-v1.5-16k`, 50 samples, UMD, $c = 4$. SCTS (orange) can significantly evade more detection under the same budget than other baselines.

For samples we evaluated, SCTS is successful at evading detection with as low as 0.25 normalized edit distance (i.e., at most 1 in 4 words are replaced). Albeit RB also maximizes attack success at higher budgets, only SCTS is able to consistently reduce AUROC to less than 0.5. The trend is consistent for all values of c considered.

However, note that SCTS saturates in the $0.35 - 0.45$ interval. This is because the current implementation only scans the sentence once when performing substitutions, and the attack success for each attempt can be lower than the theoretical result. Determining techniques to scan the sentence multiple times to increase residual replacement success is subject to future research.

6.3 Different Watermarked Models

Notice that the success of SCT is implicitly connected to the ability of the model to follow instructions. We observed that `Llama-2-7b` (model that is not instruction fine-tuned) would frequently fail to follow the prompt. For example, when the prompt is to let it generate phrases from `kernel includes` and `kernel contains`, it would probably generate several of these phrases and then begin to generate `includes` and `contains` (in words only instead of phrases), whose green lists are different. Consequently, there are insufficient number of samples for the χ^2 test leading to bias in SCT towards `S`. In this case, the success rate for each substitution attempt would drop, and SCTS would saturate at a lower budget, become slower, and less effective. On the other hand, the chat variant `Llama-2-7b-chat-hf` suffers less from this issue, while `vicuna-7b-v1.5-16k` (further fine-tuned from `Llama-2-7b-chat-hf`) is even better. We would argue that as models become more aligned and instruction fine-tuned, they will follow the prompts better in general, and SCTS will be more effective.

7 Conclusions

Our study presents SCTS, an algorithm to evade watermark detection without using external LLMs. We demonstrate that SCTS can effectively eliminate watermarks from long texts using a straightforward algorithm. This approach reveals that specific prompting techniques can uncover and exploit private watermarking information, enabling evasion. We aim to inspire further research on developing more robust and secure watermarking schemes.

8 Discussion and Open Questions

8.1 Limitations

One limitation of SCTS is that its efficiency can be improved, as shown in 6, 7. The color information is also limited to one pair due to an attacker can only prompt the black-box watermarked model. We assume the attacker knows c, while a workaround is possible for future work. SCT accuracy is also not very high so the color information is not that accurate. Lastly, SCTS is currently limited to UMD and its variants like Unigram.

8.2 Open Questions

Can SCTS be faster? One way is to store the color information already found to reduce repetitive color testing, with the risk of the accumulation of incorrect results and the cost of space. Such caching is more practical for small c.

Can one LLM query get more color information? Currently, our color testing can only test one pair in each LLM query, and it can not distinguish if the two words/tokens are both red or green. More candidates for random generation can help with the cost of more undesired factors getting involved, like the increased complexity of the prompt, the reduced frequency count for each candidate, and the more complex cases in hypothesis testing.

Unknown c Besides prompting the model to guess c first, one way to use SCTS in this case is to use a large estimated c. We leave this for future work.

Can the accuracy of SCT be higher? (Tang et al., 2023) shows the complicated behavior when the LLM model is prompted to generate uniform random strings, which are far from uniformity and vary model by model. Such behavior makes our color testing sometimes inaccurate. One possible way is to have multiple variants, like exchanging the position of the new candidate and the old to do a second prompt, to improve accuracy with the cost of more computation.

8.3 Harms

Through this work, we propose an approach to circumvent text watermarking strategies. This has implications towards spreading misinformation and purporting enhanced (nefarious) dual-use of LLMs. We hope that our findings can help design more robust watermarking techniques.

References

Scott Aaronson. 2022. My ai safety lecture for ut effective altruism. Shtetl-Optimized: The blog of Scott Aaronson. Retrieved on January, 13:2024.

Richard Arratia and Louis Gordon. 1989. Tutorial on large deviations for the binomial distribution. *Bulletin of mathematical biology*, 51(1):125–131.

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

Miranda Christ, Sam Gunn, and Or Zamir. 2023. Undetectable watermarks for language models. *arXiv preprint arXiv:2306.09194*.

Steven R Dunbar. 2011. The de moivre-laplace central limit theorem. *Topics in Probability Theory and Stochastic Processes*.

Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M Zhang. 2023. Large language models for software engineering: Survey and open problems. *arXiv preprint arXiv:2310.03533*.

Soumya Suvra Ghosal, Souradip Chakraborty, Jonas Geiping, Furong Huang, Dinesh Manocha, and Amrit Singh Bedi. 2023. Towards possibilities & impossibilities of ai-generated text detection: A survey. *arXiv preprint arXiv:2310.15264*.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints*.

Wassily Hoeffding. 1994. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A watermark for large language models. *arXiv preprint arXiv:2301.10226*.

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

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *arXiv preprint arXiv:2303.13408*.

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

Nathan Lambert, Louis Castricato, Leandro von Werra, and Alex Havrilla. 2022. Illustrating reinforcement learning from human feedback (rlhf). *Hugging Face Blog*. Https://huggingface.co/blog/rlhf.

Thomas Lancaster. 2021. Academic dishonesty or academic integrity? using natural language processing (nlp) techniques to investigate positive integrity in academic integrity research. *Journal of Academic Ethics*, 19(3):363–383.

Ning Lu, Shengcai Liu, Rui He, and Ke Tang. 2023. Large language models can be guided to evade ai-generated text detection. *arXiv preprint arXiv:2305.10847*.

Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. 2023. Mark my words: Analyzing and evaluating language model watermarks. *arXiv preprint arXiv:2312.00273*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2023. Red teaming language model detectors with language models. *arXiv preprint arXiv:2305.19713*.

Leonard Tang, Gavin Uberti, and Tom Shlomi. 2023. Baselines for identifying watermarked large language models. *arXiv preprint arXiv:2305.18456*.

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

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023. A survey on detection of llms-generated content. *arXiv preprint arXiv:2310.15654*.

Hanlin Zhang, Benjamin L Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. 2023. Watermarks in the sand: Impossibility of strong watermarking for generative models. *arXiv preprint arXiv:2311.04378*.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Appendix

A Proof of Theorem 1

Recall

$$z = \frac{\|T\|_G - \gamma T_e}{\sqrt{T_e \gamma(1-\gamma)}}$$

For the expected z, as $\mathbb{E}[\|T\|_G] = qT_e$ so that

$$\mathbb{E}[z] = \frac{q-\gamma}{\sqrt{\gamma(1-\gamma)}}\sqrt{T_e}$$

The $\propto$ relationship is obvious. For T_c, simply let $\mathbb{E}[z] = z_{th}$ and solve for T_e.

A.1 Case 1: i.i.d. case

Let $D(a||q)$ be KL-divergence with base e:

$$D(a||q) \coloneqq a \ln \frac{a}{q} + (1-a) \ln \frac{1-a}{1-q}$$

S1. If $q > \gamma$, then $\mathbb{E}[z] \propto \sqrt{T_e}$ and $T_c = \frac{\gamma(1-\gamma)z_{th}^2}{(q-\gamma)^2}$

Under the assumption, we have $\|T\|_G \sim B(T_e, q)$. From the Chernoff bound (Arratia and Gordon, 1989), for $y \leq T_e q$:

$$\Pr(\|T\|_G \leq y) \leq \exp(-T_e D(\frac{y}{T_e}||q)) \tag{4}$$

Take $y = z_{th}\sqrt{T_e\gamma(1-\gamma)} + \gamma T_e \leq qT_e$ as $T_e \geq T_c$, we have:

$$\Pr(z \leq z_{th}) \leq \exp\left(-T_e D\left(\frac{\sqrt{\gamma(1-\gamma)}z_{th}}{\sqrt{T_e}} + \gamma||q\right)\right) \tag{5}$$

Note that

$$\lim_{T_e \to \infty} \frac{-T_e D(\frac{\sqrt{\gamma(1-\gamma)}z_{th}}{T_e} + \gamma||q)}{T_e} = -D(\gamma||q)$$

$\implies$ Probability of being labeled "not watermarked" converges to 0 exponentially w.r.t. T_e.

S2. If $q = \gamma$, then $\mathbb{E}[z] = 0$, and T_c does not exist.

According to the De Moivre–Laplace theorem (Dunbar, 2011), as $T_e \to \infty$, the distribution of z approaches the standard normal distribution $N(0,1)$. This convergence allows us to use the properties of the standard normal distribution to estimate probabilities related to z. Specifically, the probability of z being detected as watermarked when exceeding a threshold z_{th} can be expressed as:

$$\lim_{T_e \to \infty} P(z > z_{th}) = 1 - \Phi(z_{th})$$

where $\Phi(z_{th})$ is the cumulative distribution function (CDF) of the standard normal distribution.

$\implies$ Probability of being labeled as "watermarked" converges to a positive constant. Typical value $z_{th} = 4$, $1 - \Phi(z_{th}) \approx 0.00003167$.

S3. If $q < \gamma$, then $\mathbb{E}[z] \propto -\sqrt{T_e}$, and T_c does not exist. A symmetric bound as equation 4 is: for $y \leq T_e q$,

$$\Pr(\|T\|_G \geq y) \leq \exp(-T_e D(\frac{k}{T_e}||q))$$

Take $y = z_{th}\sqrt{T_e\gamma(1-\gamma)} + \gamma T_e \geq qT_e$,

$$\Pr(z > z_{th}) \leq \Pr(z \geq z_{th}) \leq \exp\left(-T_e D(\frac{\sqrt{\gamma(1-\gamma)}z_{th}}{\sqrt{T_e}} + \gamma||q)\right)$$

$\implies$ Probability of being labeled "watermarked" converges to 0 exponentially w.r.t. T_e.

A.2 Case 2: Independent case

Assume the $c+1$-grams' colors are independent.

S1. If $q > \gamma$, $\mathbb{E}[z] \propto \sqrt{T_e}$ and $T_c = \frac{\gamma(1-\gamma)z_{th}^2}{(q-\gamma)^2}$

$\|T\|_G$ is a sum of T_e different variables bounded by $[0,1]$. From the Hoeffding Inequality (Hoeffding, 1994), for $t \geq 0$:

$$\Pr(qT_e - \|T\|_G \geq t) \leq \exp\left(-\frac{2t^2}{T_e}\right) \tag{6}$$

For $T_e \geq T_c$, take $t = (q-\gamma)\sqrt{T_e}(\sqrt{T_e} - \sqrt{T_c}) \geq 0$, then:

$$\Pr(z \leq z_{th}) \leq \exp\left(-2(q-\gamma)^2(\sqrt{T_e} - \sqrt{T_c})^2\right) \tag{7}$$

$\implies$ Still, the probability of being labeled as "not watermarked" converges to 0 exponentially w.r.t. T_e.

S2. If $q = \gamma$, $\mathbb{E}[z] = 0$ and T_c does not exist.

S3. If $q < \gamma$, $\mathbb{E}[z] \propto -\sqrt{T_e}$ and T_c does not exist. A symmetric bound as from the Hoeffding Inequality (Hoeffding, 1994) gives, for $t \geq 0$:

$$\Pr(\|T\|_G - qT_e \geq t) \leq \exp(-\frac{2t^2}{T_e}) \tag{8}$$

Take $t = (\gamma - q)(\sqrt{T_e} + \frac{\sqrt{\gamma(1-\gamma)}z_{th}}{\gamma-q})\sqrt{T_e} \geq 0$, and note that $P(z > z_{th}) \leq P(z \geq z_{th})$:

$$\Pr(z > z_{th}) \leq \exp(-2(\gamma-q)^2(\sqrt{T_e} + \frac{\sqrt{\gamma(1-\gamma)}z_{th}}{\gamma-q})^2) \tag{9}$$

$\implies$ Still, the probability of being labeled as "watermarked" converges to 0 exponentially w.r.t. T_e.

B SCTS Efficacy Analysis

Success Rate

For each 2-grams' substitution attempt, the success probability

$$\begin{aligned} p_s &= \Pr(\text{2-gram is green, all } k \text{ candidates are green}) \\ &= \Pr(\text{2-gram is green}) \cdot \Pr(\text{one candidate is green})^k \\ &= p(1-\gamma^k) \end{aligned} \tag{10}$$

Grouping

The grouping is as follows: one 2-gram as a group if it is preserved in SCTS; two adjacent 2-grams as a group if they are not preserved because of the same substitution. We call the first type of groups "old groups" and the second type of groups "new groups". A corner case for one substitution only changes one 2-gram as the word being substituted is the last (could not be the first for SCTS) is not considered.

Expected Green Ratio

From the assumption that candidate generation is independent of color, each group's number of green 2-grams is independent, and $i.i.d.$ within old groups and new groups. The expectation of the ratio of green tokens for these two types of groups is $\frac{\gamma}{2}$ for the new one and p_o for the old one, where

$$\begin{aligned} p_o &= \Pr(\text{2-gram is green} \mid \text{attempt fails}) && (11) \\ &= \frac{\Pr(\text{attempt fails} \mid \text{2-gram is green}) \cdot \Pr(\text{2-gram is green})}{\Pr(\text{attempt fails})} && (12) \\ &= \frac{\gamma^k \cdot p}{1 - p_s} && (13) \\ &= \frac{\gamma^k p}{1 - p + \gamma^k p} && (14) \end{aligned}$$

Note

$$\frac{\gamma^k p}{1 - p + \gamma^k p} < \gamma \iff k > 1 + \log_\gamma \frac{1-p}{p(1-\gamma)},$$

(1) holds as desired

Let the expected number of green 2-grams after SCTS attack be $e(T_e) = q(T_e)T_e$. Then we have:

$$e(T_e) = p_s(e(T_e - 2)) + \gamma) + (1 - p_s)(e(T_e - 1) + p_o) \tag{15}$$

So

$$T_e \cdot q(T_e) = p_s(q(T_e - 2) \cdot (T_e - 2) + \gamma) + (1 - p_s)(q(T_e - 1)(T_e - 1) + p_o) \tag{16}$$

Let $q = \lim_{T_e \to \infty} q(T_e)$, $T_e \to \infty$, then

$$T_e \cdot q = p_s(q \cdot (T_e - 2) + \gamma) + (1 - p_s)(q \cdot (T_e - 1) + p_o) \tag{17}$$

$$q = \frac{p_s \gamma + (1 - p_s) p_o}{1 + p_s} = \gamma - \frac{\gamma(1 - p\gamma^{k-1})}{1 + (1 - \gamma^k)p} < \gamma \tag{18}$$

Success probability bound

Definition 7 (New 2-gram ratio r_n)**.** *Defined as the number of new (to the unattacked watermarked text) 2-grams divided by* T_e*. Then* $0 \leq r_n \leq 1$.

Based on the grouping (B), we have $\frac{r_n T_e}{2}$ new groups and $(1 - r_n)T_e$ old groups. Under the independent assumption, the Hoeffding inequality gives: for $t \geq 0$,

$$\begin{aligned} \Pr(qT_e - \|T\|_G \geq t) &\leq \exp\left(-\frac{2t^2}{\frac{r_n T_e}{2} \cdot 2^2 + (1 - r_n)T_e \cdot 1^2}\right) \\ &= \exp\left(-\frac{2t^2}{(1 + r_n)T_e}\right) \\ &\leq \exp\left(-\frac{t^2}{T_e}\right) \end{aligned} \tag{19}$$

So the only difference from equation 9 is the denominator. When $q < \gamma$ for large enough k and/or T_e, use similar techniques, we have:

$$\Pr(z > z_{th}) \leq \exp((\gamma - q(T_e))^2(\sqrt{T_e} + \frac{\sqrt{\gamma(1-\gamma)} z_{th}}{\gamma - q(T_e)})^2) \tag{20}$$

Note equation 2, the probability of failing to evade watermarking exponentially converges to 0 w.r.t. T_e.

C SCTS LLM Calls

Definition 8 (Number of LLM calls N_{T_e})**.** *The number of LLM calls needed for SCTS on one sample with T_e 2-grams.*

$$\begin{aligned}\mathbb{E}[N_{T_e}] &\leq \max\left\{\frac{1}{2}\left(\frac{1}{\gamma}-\left(\frac{1}{1-(1-\gamma)^k}-1\right)k\right), \frac{1-\gamma^k}{1-\gamma}\right\}T_e \\ &< \max\left\{\frac{1}{2\gamma}, \frac{1}{1-\gamma}\right\}T_e\end{aligned} \tag{21}$$

As $T_e \to \infty$ and $k \to \infty$, the expected number of LLM calls per $c+1$-gram can be estimated. This is crucial for scalability.

$$\lim_{T_e,k\to\infty}\frac{\mathbb{E}[N_{T_e}]}{T_e} = \frac{p\gamma+(1-p)(1-\gamma)}{(1+p)\gamma(1-\gamma)} \tag{22}$$

For example, with $p = 0.5$ and $\gamma = 0.25$, this value is $\frac{16}{9}$.

Also, N_{T_e} will not deviate far from its expectation with high probability.

$$\Pr(|\frac{N_{T_e}-\mathbb{E}[N_{T_e}]}{T_e}| \geq t) \leq \frac{C}{T_e t^2} \tag{23}$$

For $t > 0$ and constant

$$\begin{aligned}C &:= \max\left\{\frac{1}{2}\left(\frac{1-\gamma}{\gamma^2}-\frac{k^2(1-\gamma)^k}{(1-(1-\gamma)^k)^2}\right), \frac{\gamma}{(1-\gamma)^2}-\frac{\gamma^k((2k-1)(1-\gamma)+\gamma^k)}{(1-\gamma)^2}\right\} \\ &< \max\left\{\frac{1-\gamma}{2\gamma^2}, \frac{\gamma}{(1-\gamma)^2}\right\}\end{aligned} \tag{24}$$

C.1 Derivation

The grouping is the same as B.

Definition 9 (N_n)**.** *The number of LLM calls for a new 2-grams group.*

Definition 10 (N_o)**.** *The number of LLM calls for a old 2-grams group.*

We have

$$\Pr(N_n = i) = \frac{\Pr(G_\gamma = i)}{\Pr(G_\gamma \leq k)}, \quad i = 1, 2, \ldots, k \tag{25}$$

where G_γ is a geometric distribution with mean $\frac{1}{\gamma}$. So,

$$\begin{aligned}\mathbb{E}[N_n] &= \mathbb{E}[G_\gamma | G_\gamma \leq k] \\ &= \frac{1}{\gamma} - \left(\frac{1}{1-(1-\gamma)^k}-1\right)k \\ &< \frac{1}{\gamma}\end{aligned} \tag{26}$$

For the old 2-gram group, we have:

$$\begin{aligned}\Pr(N_o = i) &= \Pr(G_{1-\gamma} = i), \quad 1 \leq i \leq k-1 \\ \Pr(N_o = k) &= \gamma^{k-1} \\ &= \Pr(G_{1-\gamma} = k) + \Pr(G_{1-\gamma} > k)\end{aligned} \tag{27}$$

Therefore,

$$\mathbb{E}[N_o] = \frac{1-\gamma^k}{1-\gamma} < \frac{1}{1-\gamma} \tag{28}$$

Note that

$$\begin{aligned}\mathbb{E}[N_{T_e}] &= \left(r_n\frac{\mathbb{E}[N_n]}{2} + (1-r_n)\mathbb{E}[N_o]\right)T_e \\ &\leq \max\left\{\frac{\mathbb{E}[N_n]}{2}, \mathbb{E}[N_o]\right\}T_e\end{aligned} \tag{29}$$

So equation 21 holds.

Expectation limit

Let

$$\mathbb{E}[N_o] = n_o$$

$$p_i^s = \Pr(\text{success at the } i\text{-th time}) = p\gamma^{i-1}(1-\gamma)$$

for notation simplicity.

$$N_{T_e} = \sum_{i=1}^{k} p_i^s(N_{T_e-2} + i) + (1-p_s)(N_{T_e-1} + n_o) \tag{30}$$

Let $T_e \to \infty$, $\lim_{T_e\to\infty}\frac{N_{T_e}}{T_e} = R$

$$T_eR = \sum_{i=1}^{k} p_i^s((T_e-2)R + i) + (1-p_s)((T_e-1)R + n_o)$$

$$R = \frac{\sum_{i=1}^{k} ip_i^s + (1-p_s)n_o}{1+p_s}$$

From this, we can get R for any k, but the math is cumbersome. For simpler math, let $k \to \infty$. Then $p_s = p$, N_o is a geometric distribution with mean $n_o = \frac{1}{\gamma}$,

so

$$\sum_{i=1}^{\infty} ip_i^s = \frac{p}{1-\gamma}$$

Plug in and compute, we have equation 22 hold

Expectation bound by Chebyshev's inequality

By straight-forward compute,

$$\begin{aligned}Var(N_n) &= \frac{1-\gamma}{\gamma^2} - \frac{k^2(1-\gamma)^k}{(1-(1-\gamma)^k)^2} \\ &< \frac{1-\gamma}{\gamma^2}\end{aligned} \tag{31}$$

$$\begin{aligned}Var(N_o) &= \frac{\gamma}{(1-\gamma)^2} - \frac{\gamma^k((2k-1)(1-\gamma)+\gamma^k)}{(1-\gamma)^2} \\ &< \frac{\gamma}{(1-\gamma)^2}\end{aligned} \tag{32}$$

From independence, we have

$$\begin{aligned}&Var(N_{T_e}) \\ =&\left(r_n\frac{Var(N_n)}{2} + (1-r_n)Var(N_o)\right)T_e \\ \leq&\max\left\{\frac{Var(N_n)}{2}, Var(N_o)\right\}T_e \\ =&CT_e\end{aligned} \tag{33}$$

From Chebyshev's inequality, for $\tau > 0$,

$$\Pr(|N_{T_e} - \mathbb{E}[N_{T_e}]| \geq \tau\sqrt{CT_e}) \leq \frac{1}{\tau^2} \tag{34}$$

Namely

$$\Pr(|\frac{N_{T_e} - \mathbb{E}[N_{T_e}]}{T_e}| \geq \tau\sqrt{\frac{C}{T_e}}) \leq \frac{1}{\tau^2} \tag{35}$$

Take $t = \tau\sqrt{\frac{C}{T_e}}$, then equation 23 hold.

D Additional Experimental Details

D.1 Dataset

Following Kirchenbauer et al. (2023a), we use the training set of `C4 dataset's RealNewsLike subset` (Raffel et al., 2020) For each sample, we first check if it is at least 500 tokens. If it is, we keep only the last 500 tokens. For this truncated part, the first 100 tokens are used as the prompt, and the remaining 400 tokens are used as "human-written" text for this prompt. The (watermarked) text is generated by the LLM with watermarked decoding, but keeping other configurations' default as in the original implementation. To avoid corner cases where the watermarked text is too short, we remove samples that are less than 20 words long. The number of samples is set to 50 and 10 for `vicuna-7b-v1.5-16k` and `Llama-2-7b-chat-hf` respectively, for reasons related to computational overheads.

SCT Experiment. For SCT accuracy, we log the first 1000 SCT results, including the SCTS result and ground truth in the same configuration as the main experiments (measuring AUROC, Success Rate, # LLM calls, and Running time) but in another run. This additional experiment is performed for UMD and `vicuna-7b-v1.5-16k` for simplicity.

D.2 Budgets

We take budget $\{0.05, 0.1, 0.15, 0.2, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50\}$. For RP and SICO, we additionally perform unconstrained attacks (budget$= 1$).

D.3 Metrics

For evaluation, we consider the metrics listed below.

1. **Area Under the Receiver Operating Characteristic (AUROC)**. Calculated based on the human-generated text and LLM-generated text. *The lower the score is, the more effective the attack is*, and 0.5 corresponds with the random guess.

2. **Attack success rate.** It is the ratio of samples that are not successfully detected with the default threshold $z_{th} = 4$. *Larger the value, the better the attack.*

3. **# LLM calls.** This is the number of LLM calls, including the calls to the paraphraser for RP and `vicuna-7b-v1.5-16k / Llama-2-7b-chat-hf` calls for other methods. Calls of candidate generator `distilroberta-base` do not count as that model is much smaller and faster. *Smaller the value, the more efficient the attack.*

4. **Running time.** This is used for comparing the speed of different approaches. Note that all experiments were conducted on one NVIDIA H100 80GB HBM3 GPU with Driver Version 535.54.03 and CUDA Version 12.2 on Ubuntu 20.04.6 LTS. We use Python 3.11.5 while Python 3.8.16 is used for SICO and RP. *Smaller the value, more efficient the attack.*

5. **Semantic similarity.** We measure semantic similarity using embeddings generated by the `flan-t5-xxl` sentence encoder. We report the average cosine similarity for this particular metric. *Higher the value, the more information is preserved by the attack.*

6. **Accuracy.** This is the accuracy of the SCT test. *Higher the better.*

7. **Confusion matrix.** This is the confusion matrix associated with the SCT test, modeled as a three-class classification problem. *Closer to the diagonal is the better.*

D.4 SICO Details

D.4.1 Prompt

For the SICO prompt with a budget of 0.5, we modify this part of the prompt

```
Based on the description, rewrite this to P2 style:
```

to

```
Based on the description, rewrite this to P2 style, changing at most 50% of
the words to achieve the goal.
```

For different budgets less than 1 (Unconstrained), the percentage of words that can be changed (`50%`) is updated to reflect the allocated budget accordingly.

D.4.2 Training

For `Llama-2-7b-chat-hf`, we use the prompt from corresponding Unigram's training for reasons that they share the same training data and computational overheads.

D.5 SCTS prompt for UMD Min-SelfHash $c = 1$ and Unigram

Because UMD Min-SelfHash $c = 1$ and Unigram are essentially fixed green list agnostic to context, an adapted prompt example is as follows:

```
Choose two words ( includes, contains), and generate a long uniformly random string of those
words separated by ";". Previous phrases should have no influence on future phrases: includes;
contains; includes; contains; contains; includes; contains; contains; contains; contains;
contains; contains; contains; contains; includes; contains; contains; includes; contains;
contains;
```

E Full Experiment Results

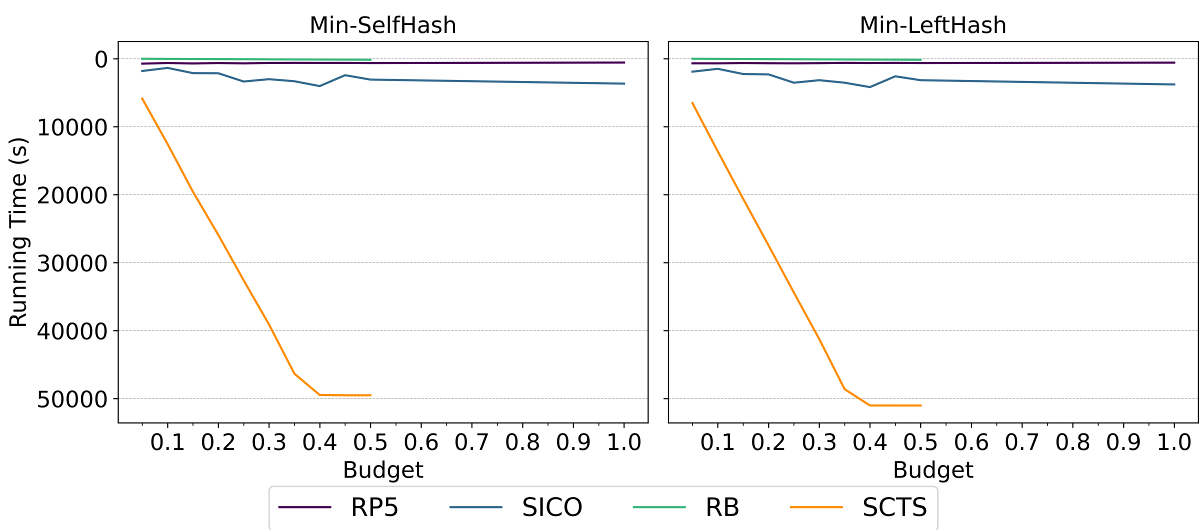

Figure 6: Running time in seconds for vicuna-7b-v1.5-16k, 50 samples, UMD watermarking, $c = 4$. A longer running time is needed for SCTS to perform a color-aware attack.

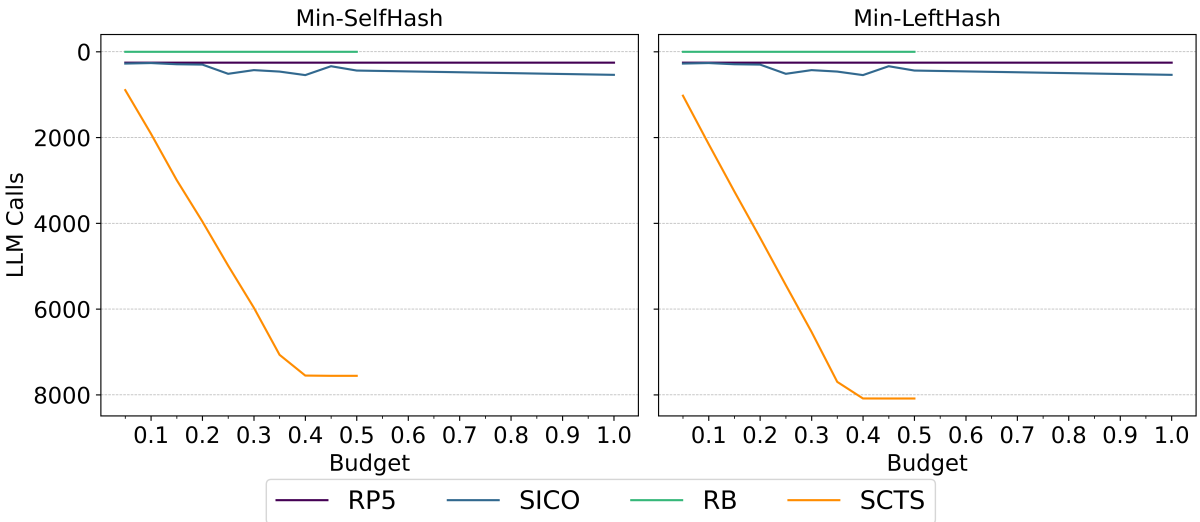

Figure 7: # LLM calls for vicuna-7b-v1.5-16k, 50 samples, UMD watermarking, $c = 4$. The longer running time of SCTS mostly comes from more LLM calls for color information.

Table 2: AUROC for `vicuna-7b-v1.5-16k`, 50 samples, UMD watermarking. SCTS achieves significantly lower AUROC under the same budget compared to other baselines and is the only method cross 0.5, and the trend is consistent over different c and hashing methods.

c	hashing	method	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
1	left	RP1	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9880	0.9984	0.9900	0.8036
1	left	RP2	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9964	0.9984	0.9880	0.9948	0.9872	0.9096
1	left	RP3	0.9984	0.9984	0.9984	0.9984	0.9984	0.9964	0.9964	0.9984	0.9868	0.9948	0.9824	0.7428
1	left	RP4	0.9984	0.9984	0.9984	0.9984	0.9984	0.9964	0.9964	0.9984	0.9868	0.9828	0.9828	0.8236
1	left	RP5	0.9984	0.9984	0.9984	0.9984	0.9984	0.9964	0.9964	0.9984	0.9860	0.9804	0.9800	0.7136
1	left	SICO	0.9984	0.9960	0.9960	0.9960	0.9960	0.9960	0.9960	0.9956	0.9956	0.9852	0.9908	0.8404
1	left	RB	0.9984	0.9892	0.9816	0.9732	0.9600	0.9524	0.9320	0.9024	0.8744	0.8134	0.7488	-
1	left	SCTS	0.9984	0.9804	0.9548	0.9096	0.8168	0.7016	0.5488	0.3480	0.3064	0.3064	0.3064	-
2	left	RP1	0.9888	0.9888	0.9888	0.9888	0.9888	0.9888	0.9888	0.9888	0.9872	0.9724	0.9744	0.7884
2	left	RP2	0.9888	0.9888	0.9888	0.9888	0.9888	0.9888	0.9888	0.9888	0.9872	0.9704	0.9804	0.8836
2	left	RP3	0.9888	0.9888	0.9888	0.9888	0.9888	0.9888	0.9888	0.9888	0.9844	0.9660	0.9632	0.6824
2	left	RP4	0.9888	0.9888	0.9888	0.9888	0.9888	0.9888	0.9872	0.9888	0.9844	0.9656	0.9576	0.7916
2	left	RP5	0.9888	0.9888	0.9888	0.9888	0.9888	0.9888	0.9872	0.9888	0.9844	0.9612	0.9548	0.6433
2	left	SICO	0.9888	0.9824	0.9824	0.9824	0.9824	0.9812	0.9824	0.9800	0.9824	0.9764	0.9844	0.7580
2	left	RB	0.9888	0.9792	0.9536	0.9404	0.9196	0.8968	0.8712	0.8168	0.7728	0.7048	0.6628	-
2	left	SCTS	0.9888	0.9776	0.9256	0.8370	0.6856	0.5202	0.3492	0.1868	0.1720	0.1720	0.1720	-
4	left	RP1	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9700	0.9712	0.9648	0.8298
4	left	RP2	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9730	0.9572	0.9600	0.8298
4	left	RP3	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9876	0.9884	0.9714	0.9476	0.9460	0.7241
4	left	RP4	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9852	0.9662	0.9440	0.9372	0.6731
4	left	RP5	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9876	0.9852	0.9650	0.9440	0.9388	0.6210
4	left	SICO	0.9884	0.9824	0.9824	0.9824	0.9824	0.9824	0.9820	0.9712	0.9824	0.9720	0.9472	0.8488
4	left	RB	0.9884	0.9888	0.9812	0.9688	0.9492	0.9166	0.8800	0.8330	0.7580	0.6668	0.6668	-
4	left	SCTS	0.9884	0.9784	0.9468	0.8818	0.7436	0.5270	0.2862	0.1342	0.1332	0.1332	0.1332	-
8	left	RP1	0.9816	0.9816	0.9816	0.9816	0.9832	0.9816	0.9820	0.9708	0.9756	0.9440	0.9544	0.7816
8	left	RP2	0.9816	0.9816	0.9816	0.9816	0.9832	0.9808	0.9820	0.9680	0.9708	0.9492	0.9584	0.7978
8	left	RP3	0.9816	0.9816	0.9816	0.9816	0.9832	0.9816	0.9788	0.9672	0.9604	0.9384	0.9184	0.5876
8	left	RP4	0.9816	0.9816	0.9816	0.9816	0.9832	0.9816	0.9784	0.9632	0.9608	0.9364	0.9360	0.6848
8	left	RP5	0.9816	0.9816	0.9816	0.9816	0.9832	0.9808	0.9776	0.9616	0.9588	0.9208	0.9332	0.5522
8	left	SICO	0.9816	0.9764	0.9776	0.9776	0.9780	0.9772	0.9752	0.9556	0.9776	0.9572	0.9148	0.7264
8	left	RB	0.9816	0.9736	0.9548	0.9260	0.8880	0.8508	0.7968	0.7188	0.6332	0.5770	0.5770	-
8	left	SCTS	0.9816	0.9552	0.8956	0.7828	0.5662	0.3608	0.1614	0.0918	0.0918	0.0918	0.0918	-
1	self	RP1	0.9988	0.9988	0.9988	0.9988	0.9988	0.9932	0.9988	0.9912	0.9956	0.9952	0.9988	0.9124
1	self	RP2	0.9988	0.9988	0.9988	0.9980	0.9988	0.9932	0.9988	0.9936	0.9968	0.9972	0.9976	0.9560
1	self	RP3	0.9988	0.9988	0.9988	0.9988	0.9988	0.9932	0.9988	0.9952	0.9968	0.9904	0.9880	0.8848
1	self	RP4	0.9988	0.9988	0.9988	0.9988	0.9988	0.9920	0.9988	0.9952	0.9948	0.9936	0.9828	0.9248
1	self	RP5	0.9988	0.9988	0.9988	0.9988	0.9988	0.9920	0.9988	0.9940	0.9940	0.9876	0.9828	0.8434
1	self	SICO	0.9988	0.9984	0.9984	0.9984	0.9976	0.9932	0.9984	0.9984	0.9928	0.9984	0.9916	0.8620
1	self	RB	0.9988	0.9960	0.9964	0.9952	0.9896	0.9888	0.9832	0.9740	0.9680	0.9580	0.9536	-
1	self	SCTS	0.9988	0.9936	0.9824	0.9524	0.9044	0.8480	0.8036	0.7160	0.6266	0.4990	0.3728	-
2	self	RP1	0.9912	0.9912	0.9912	0.9912	0.9912	0.9912	0.9912	0.9912	0.9864	0.9892	0.9832	0.9044
2	self	RP2	0.9912	0.9912	0.9912	0.9912	0.9912	0.9912	0.9912	0.9912	0.9888	0.9772	0.9832	0.8984
2	self	RP3	0.9912	0.9912	0.9912	0.9912	0.9912	0.9912	0.9912	0.9880	0.9888	0.9788	0.9768	0.8044
2	self	RP4	0.9912	0.9912	0.9912	0.9912	0.9912	0.9912	0.9912	0.9844	0.9848	0.9732	0.9744	0.8012
2	self	RP5	0.9912	0.9912	0.9912	0.9912	0.9892	0.9912	0.9912	0.9844	0.9796	0.9640	0.9652	0.6884
2	self	SICO	0.9912	0.9748	0.9748	0.9748	0.9748	0.9740	0.9728	0.9708	0.9748	0.9724	0.9728	0.8836
2	self	RB	0.9912	0.9764	0.9724	0.9656	0.9564	0.9368	0.9288	0.9108	0.8680	0.8268	0.7600	-
2	self	SCTS	0.9912	0.9656	0.9248	0.8812	0.7860	0.6628	0.5140	0.3576	0.3216	0.3216	0.3216	-
4	self	RP1	0.9928	0.9928	0.9928	0.9928	0.9928	0.9928	0.9928	0.9928	0.9892	0.9784	0.9812	0.8552
4	self	RP2	0.9928	0.9924	0.9924	0.9916	0.9916	0.9920	0.9912	0.9924	0.9872	0.9736	0.9804	0.8700
4	self	RP3	0.9928	0.9924	0.9924	0.9908	0.9920	0.9912	0.9900	0.9912	0.9844	0.9660	0.9748	0.6561
4	self	RP4	0.9928	0.9924	0.9924	0.9920	0.9912	0.9912	0.9876	0.9920	0.9832	0.9636	0.9672	0.7444
4	self	RP5	0.9928	0.9924	0.9924	0.9920	0.9924	0.9912	0.9868	0.9876	0.9800	0.9628	0.9664	0.6202
4	self	SICO	0.9928	0.9884	0.9888	0.9884	0.9892	0.9876	0.9876	0.9764	0.9876	0.9732	0.9744	0.7540
4	self	RB	0.9928	0.9812	0.9760	0.9644	0.9556	0.9412	0.9208	0.9068	0.8744	0.8340	0.8340	-
4	self	SCTS	0.9928	0.9712	0.9528	0.9196	0.8364	0.7192	0.5230	0.3272	0.3108	0.3076	0.3076	-
8	self	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9952	0.9900	0.9740	0.7892
8	self	RP2	1.0000	1.0000	1.0000	1.0000	0.9980	1.0000	0.9992	1.0000	0.9984	0.9872	0.9840	0.8148
8	self	RP3	1.0000	1.0000	1.0000	1.0000	0.9972	1.0000	0.9976	0.9984	0.9932	0.9796	0.9464	0.6164
8	self	RP4	1.0000	1.0000	1.0000	1.0000	0.9968	0.9972	0.9980	0.9976	0.9892	0.9724	0.9512	0.7068
8	self	RP5	1.0000	1.0000	1.0000	1.0000	0.9968	0.9972	0.9964	0.9972	0.9868	0.9664	0.9412	0.6388
8	self	SICO	1.0000	0.9976	0.9976	0.9956	0.9976	0.9972	0.9968	0.9956	0.9968	0.9908	0.9656	0.6951
8	self	RB	1.0000	0.9932	0.9912	0.9820	0.9692	0.9560	0.9152	0.8924	0.8552	0.8204	0.8204	-
8	self	SCTS	1.0000	0.9868	0.9484	0.8880	0.7664	0.5556	0.3628	0.2236	0.2236	0.2236	0.2236	-

Table 3: ASR for `vicuna-7b-v1.5-16k`, 50 samples, UMD watermarking. SCTS achieves significantly higher ASR under the same budget compared to other baselines, and the trend is consistent over different c and hashing methods.

c	Hashing	Method	0 (Unattacked)	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
1	left	RP1	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.20	0.28	0.96
1	left	RP2	0.18	0.18	0.18	0.18	0.18	0.22	0.20	0.24	0.26	0.30	0.36	0.94
1	left	RP3	0.18	0.18	0.18	0.18	0.18	0.24	0.26	0.28	0.32	0.36	0.46	1.00
1	left	RP4	0.18	0.18	0.18	0.18	0.18	0.24	0.30	0.26	0.40	0.40	0.58	0.96
1	left	RP5	0.18	0.18	0.18	0.20	0.18	0.24	0.30	0.28	0.44	0.42	0.62	1.00
1	left	SICO	0.18	0.28	0.28	0.28	0.30	0.30	0.32	0.30	0.28	0.40	0.40	0.82
1	left	RB	0.18	0.44	0.60	0.72	0.82	0.90	0.92	0.94	1.00	1.00	1.00	-
1	left	SCTS	0.18	0.56	0.74	0.92	0.96	1.00	1.00	1.00	1.00	1.00	1.00	-
2	left	RP1	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.12	0.14	0.30	0.94
2	left	RP2	0.08	0.08	0.08	0.10	0.08	0.10	0.10	0.08	0.16	0.18	0.36	0.80
2	left	RP3	0.08	0.08	0.08	0.10	0.08	0.10	0.14	0.08	0.20	0.26	0.56	0.98
2	left	RP4	0.08	0.08	0.08	0.10	0.08	0.14	0.18	0.10	0.22	0.36	0.58	0.94
2	left	RP5	0.08	0.08	0.08	0.10	0.08	0.14	0.18	0.10	0.26	0.38	0.60	1.00
2	left	SICO	0.08	0.16	0.18	0.16	0.16	0.18	0.16	0.20	0.16	0.26	0.26	0.78
2	left	RB	0.08	0.32	0.46	0.56	0.72	0.84	0.90	0.96	1.00	1.00	1.00	-
2	left	SCTS	0.08	0.46	0.68	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-
4	left	RP1	0.16	0.16	0.16	0.16	0.16	0.16	0.20	0.16	0.18	0.24	0.38	0.98
4	left	RP2	0.16	0.16	0.16	0.16	0.18	0.18	0.20	0.16	0.22	0.32	0.40	0.86
4	left	RP3	0.16	0.16	0.18	0.16	0.20	0.20	0.22	0.20	0.22	0.36	0.48	1.00
4	left	RP4	0.16	0.16	0.18	0.16	0.22	0.22	0.20	0.26	0.30	0.42	0.48	0.90
4	left	RP5	0.16	0.16	0.18	0.16	0.22	0.22	0.20	0.26	0.30	0.44	0.50	1.00
4	left	SICO	0.16	0.24	0.24	0.24	0.24	0.24	0.28	0.32	0.24	0.42	0.42	0.78
4	left	RB	0.16	0.30	0.46	0.58	0.70	0.88	0.92	0.98	0.98	1.00	1.00	-
4	left	SCTS	0.16	0.42	0.72	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-
8	left	RP1	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.18	0.32	0.40	0.94
8	left	RP2	0.16	0.16	0.16	0.16	0.18	0.22	0.20	0.26	0.32	0.40	0.50	0.88
8	left	RP3	0.16	0.16	0.18	0.18	0.18	0.24	0.26	0.34	0.38	0.52	0.60	1.00
8	left	RP4	0.16	0.16	0.18	0.18	0.20	0.24	0.30	0.38	0.46	0.56	0.66	0.94
8	left	RP5	0.16	0.16	0.18	0.18	0.20	0.24	0.30	0.38	0.46	0.58	0.70	1.00
8	left	SICO	0.16	0.30	0.32	0.32	0.30	0.32	0.34	0.36	0.32	0.40	0.46	0.84
8	left	RB	0.16	0.42	0.56	0.66	0.82	0.94	0.96	0.98	1.00	1.00	1.00	-
8	left	SCTS	0.16	0.54	0.80	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-
1	self	RP1	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.60	0.62	0.64	0.68	0.96
1	self	RP2	0.58	0.58	0.58	0.58	0.58	0.58	0.62	0.60	0.64	0.66	0.74	0.84
1	self	RP3	0.58	0.58	0.58	0.58	0.58	0.62	0.64	0.64	0.72	0.66	0.80	0.98
1	self	RP4	0.58	0.58	0.58	0.58	0.58	0.64	0.64	0.64	0.76	0.72	0.78	0.94
1	self	RP5	0.58	0.58	0.58	0.58	0.60	0.62	0.64	0.66	0.76	0.74	0.82	0.98
1	self	SICO	0.58	0.62	0.62	0.64	0.64	0.64	0.64	0.64	0.62	0.70	0.64	0.90
1	self	RB	0.58	0.64	0.74	0.84	0.88	0.86	0.88	0.90	0.90	0.92	0.94	-
1	self	SCTS	0.58	0.80	0.92	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-
2	self	RP1	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.18	0.16	0.22	0.90
2	self	RP2	0.12	0.12	0.12	0.12	0.12	0.14	0.16	0.20	0.24	0.22	0.24	0.82
2	self	RP3	0.12	0.12	0.12	0.12	0.14	0.20	0.18	0.20	0.30	0.28	0.34	0.96
2	self	RP4	0.12	0.12	0.12	0.12	0.16	0.18	0.20	0.24	0.30	0.30	0.40	0.88
2	self	RP5	0.12	0.12	0.12	0.12	0.16	0.18	0.20	0.26	0.32	0.36	0.50	0.96
2	self	SICO	0.12	0.20	0.18	0.18	0.20	0.22	0.28	0.24	0.18	0.30	0.26	0.54
2	self	RB	0.12	0.24	0.38	0.40	0.52	0.58	0.72	0.80	0.86	0.92	0.98	-
2	self	SCTS	0.12	0.44	0.60	0.72	0.88	0.96	1.00	1.00	1.00	1.00	1.00	-
4	self	RP1	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.26	0.26	0.34	0.92
4	self	RP2	0.24	0.24	0.24	0.24	0.24	0.26	0.24	0.24	0.28	0.28	0.36	0.84
4	self	RP3	0.24	0.24	0.24	0.24	0.24	0.26	0.30	0.24	0.32	0.38	0.38	1.00
4	self	RP4	0.24	0.24	0.24	0.24	0.24	0.24	0.30	0.32	0.30	0.42	0.42	0.96
4	self	RP5	0.24	0.24	0.26	0.24	0.24	0.24	0.30	0.34	0.36	0.46	0.50	1.00
4	self	SICO	0.24	0.26	0.26	0.28	0.30	0.30	0.34	0.34	0.28	0.34	0.48	0.78
4	self	RB	0.24	0.32	0.40	0.52	0.56	0.62	0.78	0.88	0.94	0.92	0.92	-
4	self	SCTS	0.24	0.38	0.68	0.82	0.98	1.00	1.00	1.00	1.00	1.00	1.00	-
8	self	RP1	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.14	0.16	0.22	0.92
8	self	RP2	0.10	0.10	0.10	0.10	0.12	0.10	0.12	0.14	0.18	0.26	0.28	0.80
8	self	RP3	0.10	0.10	0.10	0.10	0.14	0.10	0.14	0.18	0.28	0.30	0.46	0.98
8	self	RP4	0.10	0.10	0.10	0.10	0.14	0.16	0.14	0.22	0.38	0.38	0.46	0.92
8	self	RP5	0.10	0.10	0.10	0.10	0.14	0.18	0.16	0.22	0.40	0.38	0.48	0.98
8	self	SICO	0.10	0.18	0.18	0.16	0.18	0.20	0.24	0.20	0.18	0.22	0.30	0.80
8	self	RB	0.10	0.26	0.34	0.54	0.56	0.70	0.76	0.90	0.90	0.94	0.94	-
8	self	SCTS	0.10	0.36	0.66	0.88	0.96	0.98	0.98	0.98	0.98	0.98	0.98	-

Table 4: Running time (s) for `vicuna-7b-v1.5-16k`, 50 samples, UMD watermarking. A longer running time is needed for SCTS to perform a color-aware attack, and the trend is consistent over different c and hashing methods.

c	hashing	method	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
1	left	RP1	197	178	190	190	185	186	190	200	183	185	200
1	left	RP2	374	363	376	375	367	374	381	387	373	361	357
1	left	RP3	551	545	563	554	554	556	560	573	557	538	485
1	left	RP4	734	730	757	744	729	732	736	747	732	707	595
1	left	RP5	918	914	943	926	908	905	914	912	885	860	678
1	left	SICO	1926	1453	2195	2362	3522	3159	3438	4082	2568	3229	3758
1	left	RB	25	51	76	102	127	152	176	201	225	250	-
1	left	SCTS	7835	15792	24067	32493	40087	48173	55944	60895	61131	61131	-
2	left	RP1	184	186	191	185	183	186	183	185	185	185	190
2	left	RP2	340	342	338	342	328	329	329	332	335	351	355
2	left	RP3	492	493	493	496	476	476	469	478	473	496	479
2	left	RP4	640	635	651	641	619	634	605	615	609	627	589
2	left	RP5	795	786	809	787	777	803	749	748	731	744	681
2	left	SICO	1885	1480	2221	2382	3481	3169	3459	4196	2637	3127	3815
2	left	RB	23	46	69	92	114	136	159	181	202	226	-
2	left	SCTS	7501	15496	24103	32538	41075	49208	56859	60151	60176	60176	-
4	left	RP1	151	148	151	152	157	159	147	147	150	154	155
4	left	RP2	298	281	273	278	290	274	270	292	270	305	305
4	left	RP3	430	410	398	411	411	407	377	423	402	432	426
4	left	RP4	551	558	514	540	547	532	513	524	520	537	516
4	left	RP5	684	689	645	672	687	663	619	642	621	647	585
4	left	SICO	1915	1494	2259	2313	3537	3168	3537	4183	2595	3168	3793
4	left	RB	17	35	51	68	85	101	117	134	150	169	-
4	left	SCTS	6528	13639	20612	27455	34400	41233	48609	51009	51016	51016	-
8	left	RP1	166	166	172	171	170	166	168	165	165	164	164
8	left	RP2	304	295	303	301	308	303	303	299	297	309	313
8	left	RP3	448	433	440	436	440	434	434	437	424	437	439
8	left	RP4	589	568	577	575	566	569	557	560	540	544	548
8	left	RP5	721	705	707	705	692	694	676	670	643	643	637
8	left	SICO	1932	1507	2231	2374	3545	3327	3467	4221	2590	3146	3837
8	left	RB	20	39	59	78	97	115	134	153	171	191	-
8	left	SCTS	7888	16181	26515	34708	42493	50956	58507	60981	60981	60981	-
1	self	RP1	206	192	210	199	198	201	208	202	207	197	210
1	self	RP2	386	370	392	380	366	385	387	375	382	390	403
1	self	RP3	568	559	585	564	534	564	552	551	546	561	549
1	self	RP4	747	741	770	739	705	740	704	701	701	709	679
1	self	RP5	933	919	947	911	863	911	863	845	842	838	783
1	self	SICO	1902	1468	2160	2224	3324	3112	3414	4053	2564	3111	3749
1	self	RB	22	44	66	88	109	130	152	173	194	215	-
1	self	SCTS	6831	13435	19927	26888	33682	40726	47189	53644	60359	67246	-
2	self	RP1	182	176	171	173	177	170	171	175	171	170	172
2	self	RP2	323	326	312	318	310	339	317	320	322	312	332
2	self	RP3	477	474	456	467	453	499	448	458	465	445	460
2	self	RP4	632	614	604	611	587	641	574	592	589	566	571
2	self	RP5	784	757	756	759	726	782	698	709	699	675	659
2	self	SICO	1860	1518	2209	2258	3373	3039	3355	4012	2463	3199	3937
2	self	RB	19	38	57	76	94	113	131	149	167	185	-
2	self	SCTS	7231	14897	22329	30056	37410	44783	52166	56948	57194	57194	-
4	self	RP1	146	142	141	143	146	143	142	142	147	145	144
4	self	RP2	271	264	282	265	269	269	266	272	275	280	285
4	self	RP3	462	395	419	388	395	393	386	399	402	408	399
4	self	RP4	597	514	551	519	543	516	502	516	525	530	494
4	self	RP5	731	642	706	647	690	634	619	629	625	650	571
4	self	SICO	1815	1360	2129	2147	3364	3017	3322	4032	2433	3080	3672
4	self	RB	17	34	50	66	82	97	113	128	143	161	-
4	self	SCTS	5882	12556	19580	25929	32609	39105	46334	49469	49510	49510	-
8	self	RP1	173	170	168	182	165	171	169	167	170	169	170
8	self	RP2	295	297	299	318	296	300	304	305	297	310	328
8	self	RP3	433	445	429	456	421	429	442	431	423	449	466
8	self	RP4	572	582	565	586	552	560	564	551	538	569	640
8	self	RP5	705	709	692	720	680	688	679	662	635	667	742
8	self	SICO	1897	1412	2222	2211	3371	3011	3379	4079	2535	3120	3853
8	self	RB	18	35	53	70	87	104	121	137	154	171	-
8	self	SCTS	7938	17274	25689	34750	43226	51518	59759	62113	62113	62113	-

Table 5: # LLM calls for `vicuna-7b-v1.5-16k`, 50 samples, UMD watermarking. The longer running time of SCTS mostly comes from more LLM calls for color information, and the trend is consistent over different c and hashing methods.

c	hashing	method	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
1	left	RP1	50	50	50	50	50	50	50	50	50	50	50
1	left	RP2	100	100	100	100	100	100	100	100	100	100	100
1	left	RP3	150	150	150	150	150	150	150	150	150	150	150
1	left	RP4	200	200	200	200	200	200	200	200	200	200	200
1	left	RP5	250	250	250	250	250	250	250	250	250	250	250
1	left	SICO	275	263	291	299	514	428	462	544	337	438	538
1	left	RB	0	0	0	0	0	0	0	0	0	0	-
1	left	SCTS	1235	2489	3791	5119	6317	7589	8810	9588	9625	9625	-
2	left	RP1	50	50	50	50	50	50	50	50	50	50	50
2	left	RP2	100	100	100	100	100	100	100	100	100	100	100
2	left	RP3	150	150	150	150	150	150	150	150	150	150	150
2	left	RP4	200	200	200	200	200	200	200	200	200	200	200
2	left	RP5	250	250	250	250	250	250	250	250	250	250	250
2	left	SICO	275	263	291	299	514	428	462	544	337	438	538
2	left	RB	0	0	0	0	0	0	0	0	0	0	-
2	left	SCTS	1206	2495	3871	5221	6585	7884	9110	9637	9641	9641	-
4	left	RP1	50	50	50	50	50	50	50	50	50	50	50
4	left	RP2	100	100	100	100	100	100	100	100	100	100	100
4	left	RP3	150	150	150	150	150	150	150	150	150	150	150
4	left	RP4	200	200	200	200	200	200	200	200	200	200	200
4	left	RP5	250	250	250	250	250	250	250	250	250	250	250
4	left	SICO	275	263	291	299	514	428	462	544	337	438	538
4	left	RB	0	0	0	0	0	0	0	0	0	0	-
4	left	SCTS	1024	2153	3259	4339	5443	6533	7699	8086	8087	8087	-
8	left	RP1	50	50	50	50	50	50	50	50	50	50	50
8	left	RP2	100	100	100	100	100	100	100	100	100	100	100
8	left	RP3	150	150	150	150	150	150	150	150	150	150	150
8	left	RP4	200	200	200	200	200	200	200	200	200	200	200
8	left	RP5	250	250	250	250	250	250	250	250	250	250	250
8	left	SICO	275	263	291	299	514	428	462	544	337	438	538
8	left	RB	0	0	0	0	0	0	0	0	0	0	-
8	left	SCTS	1256	2623	4367	5657	6858	8247	9446	9819	9819	9819	-
1	self	RP1	50	50	50	50	50	50	50	50	50	50	50
1	self	RP2	100	100	100	100	100	100	100	100	100	100	100
1	self	RP3	150	150	150	150	150	150	150	150	150	150	150
1	self	RP4	200	200	200	200	200	200	200	200	200	200	200
1	self	RP5	250	250	250	250	250	250	250	250	250	250	250
1	self	SICO	275	263	291	299	514	428	462	544	337	438	538
1	self	RB	0	0	0	0	0	0	0	0	0	0	-
1	self	SCTS	1036	2037	3029	4086	5120	6190	7179	8162	9184	10232	-
2	self	RP1	50	50	50	50	50	50	50	50	50	50	50
2	self	RP2	100	100	100	100	100	100	100	100	100	100	100
2	self	RP3	150	150	150	150	150	150	150	150	150	150	150
2	self	RP4	200	200	200	200	200	200	200	200	200	200	200
2	self	RP5	250	250	250	250	250	250	250	250	250	250	250
2	self	SICO	275	263	291	299	514	428	462	544	337	438	538
2	self	RB	0	0	0	0	0	0	0	0	0	0	-
2	self	SCTS	1088	2259	3380	4541	5651	6779	7897	8625	8664	8664	-
4	self	RP1	50	50	50	50	50	50	50	50	50	50	50
4	self	RP2	100	100	100	100	100	100	100	100	100	100	100
4	self	RP3	150	150	150	150	150	150	150	150	150	150	150
4	self	RP4	200	200	200	200	200	200	200	200	200	200	200
4	self	RP5	250	250	250	250	250	250	250	250	250	250	250
4	self	SICO	275	263	291	299	514	428	462	544	337	438	538
4	self	RB	0	0	0	0	0	0	0	0	0	0	-
4	self	SCTS	894	1912	2992	3960	4987	5969	7068	7552	7558	7558	-
8	self	RP1	50	50	50	50	50	50	50	50	50	50	50
8	self	RP2	100	100	100	100	100	100	100	100	100	100	100
8	self	RP3	150	150	150	150	150	150	150	150	150	150	150
8	self	RP4	200	200	200	200	200	200	200	200	200	200	200
8	self	RP5	250	250	250	250	250	250	250	250	250	250	250
8	self	SICO	275	263	291	299	514	428	462	544	337	438	538
8	self	RB	0	0	0	0	0	0	0	0	0	0	-
8	self	SCTS	1192	2628	3920	5252	6514	7760	8991	9369	9369	9369	-

Table 6: Semantic similarity for `vicuna-7b-v1.5-16k`, 50 samples, UMD watermarking. SCTS successfully preserves semantics, and the trend is consistent over different c and hashing methods.

c	hashing	method	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
1	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9987	0.9992	0.9928	0.8719
1	left	RP2	1.0000	1.0000	0.9997	0.9987	0.9959	0.9964	0.9905	0.9908	0.9787	0.9780	0.8188
1	left	RP3	1.0000	0.9998	0.9996	0.9981	0.9913	0.9878	0.9838	0.9803	0.9637	0.9559	0.7495
1	left	RP4	1.0000	0.9998	0.9996	0.9976	0.9895	0.9811	0.9801	0.9674	0.9531	0.9355	0.6934
1	left	RP5	1.0000	0.9998	0.9978	0.9973	0.9895	0.9809	0.9788	0.9645	0.9488	0.9198	0.6076
1	left	SICO	0.9989	1.0000	0.9906	0.9800	0.9847	0.9933	0.9958	0.9878	0.9674	0.9831	0.7065
1	left	RB	0.9890	0.9771	0.9639	0.9469	0.9337	0.9195	0.9054	0.8950	0.8863	0.8659	-
1	left	SCTS	0.9885	0.9745	0.9579	0.9426	0.9228	0.9113	0.8961	0.8906	0.8906	0.8906	-
2	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9971	0.9976	0.9832	0.8875
2	left	RP2	1.0000	1.0000	0.9999	0.9973	0.9986	0.9970	0.9967	0.9887	0.9840	0.9632	0.8508
2	left	RP3	1.0000	0.9999	0.9991	0.9954	0.9949	0.9948	0.9884	0.9775	0.9686	0.9360	0.7496
2	left	RP4	1.0000	0.9999	0.9977	0.9952	0.9914	0.9854	0.9858	0.9697	0.9464	0.9135	0.7114
2	left	RP5	1.0000	0.9999	0.9977	0.9940	0.9914	0.9854	0.9854	0.9618	0.9392	0.9070	0.6375
2	left	SICO	0.9996	0.9990	0.9914	0.9854	0.9818	0.9932	0.9870	0.9987	0.9773	0.9801	0.7282
2	left	RB	0.9899	0.9786	0.9639	0.9509	0.9371	0.9231	0.9061	0.8898	0.8766	0.8629	-
2	left	SCTS	0.9887	0.9734	0.9587	0.9414	0.9237	0.9077	0.8926	0.8912	0.8912	0.8912	-
4	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	0.9992	1.0000	0.9984	0.9945	0.9862	0.8981
4	left	RP2	1.0000	0.9981	0.9995	0.9999	0.9986	0.9968	0.9955	0.9913	0.9827	0.9789	0.8256
4	left	RP3	1.0000	0.9979	0.9982	0.9991	0.9940	0.9924	0.9919	0.9834	0.9737	0.9634	0.7331
4	left	RP4	1.0000	0.9979	0.9981	0.9986	0.9924	0.9903	0.9884	0.9736	0.9645	0.9554	0.6564
4	left	RP5	1.0000	0.9979	0.9978	0.9986	0.9922	0.9894	0.9881	0.9711	0.9586	0.9467	0.5436
4	left	SICO	0.9984	0.9988	0.9906	0.9889	0.9884	0.9886	0.9934	0.9910	0.9845	0.9600	0.7104
4	left	RB	0.9862	0.9728	0.9538	0.9394	0.9203	0.9008	0.8854	0.8732	0.8546	0.8546	-
4	left	SCTS	0.9878	0.9743	0.9579	0.9366	0.9198	0.9018	0.8842	0.8832	0.8832	0.8832	-
8	left	RP1	1.0000	1.0000	1.0000	0.9988	1.0000	0.9983	0.9980	0.9964	0.9840	0.9793	0.8792
8	left	RP2	0.9989	0.9974	1.0000	0.9956	0.9978	0.9917	0.9865	0.9789	0.9714	0.9608	0.8153
8	left	RP3	0.9989	0.9971	0.9994	0.9948	0.9898	0.9845	0.9740	0.9675	0.9553	0.9428	0.7462
8	left	RP4	0.9982	0.9970	0.9985	0.9895	0.9897	0.9786	0.9664	0.9592	0.9447	0.9318	0.7082
8	left	RP5	0.9982	0.9970	0.9985	0.9879	0.9863	0.9770	0.9657	0.9578	0.9348	0.9215	0.6324
8	left	SICO	0.9958	0.9997	0.9871	0.9922	0.9801	0.9879	0.9809	0.9858	0.9697	0.9658	0.6553
8	left	RB	0.9859	0.9688	0.9501	0.9343	0.9178	0.9043	0.8874	0.8698	0.8540	0.8540	-
8	left	SCTS	0.9889	0.9697	0.9513	0.9329	0.9166	0.8970	0.8842	0.8842	0.8842	0.8842	-
1	self	RP1	1.0000	1.0000	1.0000	1.0000	0.9999	0.9998	0.9982	0.9947	0.9952	0.9918	0.9154
1	self	RP2	1.0000	0.9999	0.9972	0.9948	0.9970	0.9887	0.9848	0.9818	0.9797	0.9623	0.8838
1	self	RP3	1.0000	0.9994	0.9954	0.9925	0.9923	0.9859	0.9764	0.9685	0.9601	0.9454	0.7861
1	self	RP4	1.0000	0.9990	0.9920	0.9932	0.9900	0.9832	0.9748	0.9613	0.9494	0.9221	0.7580
1	self	RP5	1.0000	0.9990	0.9919	0.9914	0.9900	0.9829	0.9690	0.9566	0.9463	0.9113	0.6610
1	self	SICO	0.9993	0.9986	0.9904	0.9853	0.9745	0.9975	0.9899	0.9891	0.9756	0.9811	0.7040
1	self	RB	0.9836	0.9692	0.9560	0.9425	0.9304	0.9165	0.9033	0.8893	0.8723	0.8531	-
1	self	SCTS	0.9859	0.9700	0.9555	0.9415	0.9258	0.9117	0.8944	0.8788	0.8612	0.8430	-
2	self	RP1	1.0000	1.0000	1.0000	1.0000	0.9992	1.0000	1.0000	0.9959	0.9950	0.9914	0.9019
2	self	RP2	0.9999	0.9998	0.9997	0.9994	0.9976	0.9954	0.9897	0.9851	0.9771	0.9713	0.8937
2	self	RP3	0.9998	0.9998	0.9996	0.9981	0.9957	0.9928	0.9815	0.9722	0.9520	0.9517	0.7993
2	self	RP4	0.9998	0.9998	0.9996	0.9976	0.9938	0.9868	0.9774	0.9609	0.9391	0.9343	0.7701
2	self	RP5	0.9998	0.9998	0.9995	0.9969	0.9918	0.9868	0.9720	0.9600	0.9287	0.9236	0.6896
2	self	SICO	0.9980	0.9990	0.9964	0.9891	0.9743	0.9928	0.9777	0.9834	0.9649	0.9776	0.7597
2	self	RB	0.9872	0.9732	0.9562	0.9392	0.9191	0.9028	0.8854	0.8739	0.8633	0.8501	-
2	self	SCTS	0.9832	0.9621	0.9414	0.9209	0.9047	0.8852	0.8695	0.8687	0.8687	0.8687	-
4	self	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9927	0.9974	0.9921	0.8825
4	self	RP2	0.9999	0.9998	0.9989	0.9969	0.9961	0.9937	0.9890	0.9828	0.9770	0.9749	0.7580
4	self	RP3	0.9998	0.9996	0.9990	0.9938	0.9958	0.9871	0.9799	0.9758	0.9686	0.9638	0.6806
4	self	RP4	0.9998	0.9992	0.9988	0.9916	0.9931	0.9845	0.9761	0.9643	0.9513	0.9537	0.6235
4	self	RP5	0.9998	0.9987	0.9988	0.9916	0.9931	0.9841	0.9709	0.9582	0.9488	0.9416	0.5464
4	self	SICO	0.9979	0.9999	0.9925	0.9880	0.9614	0.9869	0.9811	0.9857	0.9722	0.9674	0.6439
4	self	RB	0.9847	0.9686	0.9507	0.9330	0.9141	0.8964	0.8801	0.8611	0.8459	0.8459	-
4	self	SCTS	0.9850	0.9673	0.9499	0.9285	0.9105	0.8891	0.8754	0.8737	0.8732	0.8732	-
8	self	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9996	0.9970	0.9974	0.9944	0.9064
8	self	RP2	1.0000	0.9995	0.9980	0.9953	0.9929	0.9978	0.9860	0.9816	0.9859	0.9661	0.8792
8	self	RP3	0.9996	0.9995	0.9977	0.9928	0.9919	0.9902	0.9766	0.9741	0.9668	0.9464	0.7893
8	self	RP4	0.9988	0.9995	0.9974	0.9921	0.9853	0.9884	0.9748	0.9626	0.9476	0.9335	0.7644
8	self	RP5	0.9988	0.9995	0.9974	0.9921	0.9848	0.9862	0.9719	0.9603	0.9373	0.9198	0.6658
8	self	SICO	0.9984	1.0000	0.9818	0.9924	0.9796	0.9848	0.9945	0.9688	0.9807	0.9749	0.7222
8	self	RB	0.9900	0.9742	0.9593	0.9431	0.9273	0.9084	0.8917	0.8760	0.8608	0.8608	-
8	self	SCTS	0.9861	0.9706	0.9534	0.9367	0.9197	0.9021	0.8837	0.8837	0.8837	0.8837	-

Table 7: AUROC for `vicuna-7b-v1.5-16k`, 50 samples, Unigram watermarking. We observe that Unigram is more robust than UMD, but still susceptible to SCTS with a consistent trend.

method	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
RP1	0.9852	0.9852	0.9852	0.9852	0.9852	0.9856	0.9852	0.9852	0.9848	0.9844	0.9852	0.8852
RP2	0.9852	0.9852	0.9852	0.9852	0.9852	0.9856	0.9848	0.9844	0.9844	0.9840	0.9852	0.8708
RP3	0.9852	0.9852	0.9852	0.9852	0.9852	0.9856	0.9848	0.9820	0.9844	0.9840	0.9836	0.8700
RP4	0.9852	0.9860	0.9852	0.9852	0.9852	0.9856	0.9848	0.9840	0.9844	0.9840	0.9832	0.8564
RP5	0.9852	0.9860	0.9852	0.9852	0.9852	0.9856	0.9848	0.9840	0.9844	0.9832	0.9832	0.8088
SICO	0.9852	0.9860	0.9860	0.9856	0.9860	0.9860	0.9824	0.9836	0.9848	0.9840	0.9488	0.8160
RB	0.9852	0.9852	0.9856	0.9848	0.9848	0.9836	0.9736	0.9700	0.9668	0.9616	0.9584	-
SCTS	0.9852	0.9820	0.9736	0.9552	0.9132	0.8488	0.7324	0.5932	0.4428	0.3288	0.2604	-

Table 8: ASR for `vicuna-7b-v1.5-16k`, 50 samples, Unigram watermarking. Unigram's unattacked ASR is higher, but still susceptible to SCTS with a consistent trend compared to UMD.

method	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
RP1	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.64	0.64	0.64	1.00
RP2	0.62	0.62	0.62	0.62	0.62	0.62	0.66	0.62	0.66	0.64	0.70	0.92
RP3	0.62	0.62	0.62	0.62	0.62	0.66	0.68	0.66	0.68	0.66	0.72	1.00
RP4	0.62	0.62	0.62	0.62	0.62	0.64	0.68	0.66	0.72	0.74	0.78	0.96
RP5	0.62	0.62	0.62	0.62	0.62	0.64	0.68	0.68	0.76	0.74	0.78	0.98
SICO	0.62	0.66	0.64	0.62	0.66	0.64	0.70	0.66	0.74	0.68	0.80	0.92
RB	0.62	0.66	0.70	0.76	0.82	0.88	0.90	0.90	0.92	0.90	0.94	-
SCTS	0.62	0.76	0.90	0.94	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-

Table 9: Running time in seconds for `vicuna-7b-v1.5-16k`, 50 samples, Unigram watermarking. The trend is consistent from UMD, while the shorter running time is more due to implementation rather than methods per se.

method	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
RP1	92	89	89	90	93	92	90	93	92	92	89
RP2	159	158	158	159	159	172	160	155	151	156	167
RP3	220	221	224	228	226	239	222	215	215	220	234
RP4	287	288	297	290	294	306	282	277	280	279	286
RP5	369	365	366	355	363	371	345	341	339	338	332
SICO	1716	1240	2093	2123	3380	2930	3351	4066	2588	3108	3809
RB	5	10	15	19	24	28	33	37	41	44	-
SCTS	2125	4415	6553	8939	11267	13681	16042	18347	20580	22785	-

Table 10: # LLM calls for `vicuna-7b-v1.5-16k`, 50 samples, Unigram watermarking. The trend is consistent with UMD.

method	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
RP1	50	50	50	50	50	50	50	50	50	50	50
RP2	100	100	100	100	100	100	100	100	100	100	100
RP3	150	150	150	150	150	150	150	150	150	150	150
RP4	200	200	200	200	200	200	200	200	200	200	200
RP5	250	250	250	250	250	250	250	250	250	250	250
SICO	275	263	291	299	514	428	462	544	337	438	538
RB	0	0	0	0	0	0	0	0	0	0	-
SCTS	576	1175	1730	2353	2984	3666	4318	4945	5547	6176	-

Table 11: Semantic similarity for `vicuna-7b-v1.5-16k`, 50 samples, Unigram watermarking. The trend is consistent with UMD with slightly lower values.

method	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
RP1	1.0000	1.0000	1.0000	1.0000	0.9997	1.0000	1.0000	0.9973	0.9976	0.9949	0.8735
RP2	1.0000	1.0000	0.9981	0.9986	0.9974	0.9924	0.9888	0.9884	0.9933	0.9833	0.7835
RP3	1.0000	1.0000	0.9967	0.9986	0.9953	0.9924	0.9838	0.9832	0.9903	0.9751	0.6923
RP4	0.9991	1.0000	0.9966	0.9971	0.9935	0.9914	0.9824	0.9744	0.9809	0.9727	0.5879
RP5	0.9991	1.0000	0.9966	0.9971	0.9927	0.9903	0.9821	0.9713	0.9769	0.9688	0.5247
SICO	0.9980	1.0000	0.9959	0.9883	0.9744	0.9829	0.9841	0.9812	0.9920	0.9556	0.5702
RB	0.9809	0.9664	0.9503	0.9343	0.9175	0.8973	0.8794	0.8603	0.8406	0.8151	-
SCTS	0.9801	0.9666	0.9441	0.9241	0.9047	0.8836	0.8653	0.8434	0.8233	0.8015	-

Table 12: AUROC for `Llama-2-7b-chat-hf`, 10 samples, UMD watermarking. The trend is consistent compared with `vicuna-7b-v1.5-16k` over different c and hashing methods.

c	hashing	method	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
1	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9900	1.0000	0.7700
1	left	RP2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8900
1	left	RP3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9700	1.0000	0.7800
1	left	RP4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9700	0.9900	0.7700
1	left	RP5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9700	0.9900	0.7100
1	left	SICO	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9300
1	left	RB	1.0000	1.0000	0.9700	0.9500	0.8500	0.8300	0.8100	0.8300	0.7900	0.7200	0.6100	-
1	left	SCTS	1.0000	1.0000	0.9400	0.8800	0.8100	0.7600	0.6100	0.4900	0.4900	0.4900	0.4900	-
2	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5800
2	left	RP2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9600	0.9000
2	left	RP3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9600	0.5300
2	left	RP4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9900	0.9600	0.6700
2	left	RP5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9200	0.6900
2	left	SICO	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9300
2	left	RB	1.0000	1.0000	0.9600	0.9500	0.9200	0.8700	0.8300	0.6700	0.5800	0.5600	0.5100	-
2	left	SCTS	1.0000	1.0000	0.9000	0.7700	0.4700	0.3100	0.1500	0.1300	0.1300	0.1300	0.1300	-
4	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9500	0.8900	0.6800
4	left	RP2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9500	0.9200	0.8700
4	left	RP3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9100	0.9500	0.6300
4	left	RP4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9900	0.9500	0.9200	0.7900
4	left	RP5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9400	0.9000	0.4100
4	left	SICO	1.0000	1.0000	1.0000	1.0000	0.9900	1.0000	0.9900	0.9900	1.0000	1.0000	0.9500	0.7900
4	left	RB	1.0000	1.0000	1.0000	0.9900	0.9600	0.9300	0.8400	0.7700	0.6100	0.5800	0.5800	-
4	left	SCTS	1.0000	0.9800	0.8900	0.7400	0.4600	0.1900	0.0800	0.1000	0.1000	0.1000	0.1000	-
8	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9700	1.0000	0.7200
8	left	RP2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9700	0.9700	0.8667
8	left	RP3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9900	1.0000	0.9600	0.9900	0.5556
8	left	RP4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9900	1.0000	0.9500	0.9600	0.6889
8	left	RP5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9900	1.0000	0.9700	0.9600	0.5667
8	left	SICO	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9300	0.9700	0.8200
8	left	RB	1.0000	0.9900	0.9600	0.9300	0.9000	0.8500	0.8200	0.7300	0.6700	0.6300	0.6300	-
8	left	SCTS	1.0000	0.9000	0.8300	0.6900	0.4900	0.2100	0.1100	0.1100	0.1100	0.1100	0.1100	-
1	self	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8700
1	self	RP2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9300
1	self	RP3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9300
1	self	RP4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8700
1	self	RP5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5500
1	self	SICO	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9700	1.0000	0.8300
1	self	RB	1.0000	1.0000	0.9800	0.9700	0.9900	0.9200	0.9100	0.8900	0.8500	0.8800	0.8300	-
1	self	SCTS	1.0000	0.9600	0.9600	0.9000	0.7700	0.6300	0.5300	0.4200	0.3800	0.3000	0.2400	-
2	self	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9700	0.7500
2	self	RP2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9700	0.9500
2	self	RP3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9900	0.9600	0.7100
2	self	RP4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9900	0.9700	0.7900
2	self	RP5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9900	0.9400	0.7400
2	self	SICO	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9300
2	self	RB	1.0000	1.0000	0.9900	0.9800	0.9600	0.8900	0.8300	0.8000	0.7300	0.6400	0.5800	-
2	self	SCTS	1.0000	1.0000	0.9700	0.8900	0.7200	0.5600	0.4700	0.3300	0.3300	0.3300	0.3300	-
4	self	RP1	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9800	0.9500	0.9800	0.7600
4	self	RP2	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9800	0.9500	0.9800	0.8300
4	self	RP3	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9700	0.9600	0.9000	0.6700
4	self	RP4	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9700	0.9600	0.9000	0.7200
4	self	RP5	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9700	0.9500	0.8600	0.6800
4	self	SICO	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900	0.9800	0.9500	0.9100
4	self	RB	0.9900	0.9900	0.9900	0.9600	0.8800	0.8800	0.8800	0.8600	0.7800	0.7500	0.7500	-
4	self	SCTS	0.9900	0.9800	0.8900	0.7500	0.6300	0.4500	0.2500	0.2500	0.2000	0.2000	0.2000	-
8	self	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9700	1.0000	1.0000	0.6600
8	self	RP2	1.0000	1.0000	0.9800	0.9700	0.9800	0.9900	0.9600	0.9900	0.9400	0.9400	0.9300	0.7000
8	self	RP3	1.0000	1.0000	0.9800	0.9700	0.9800	0.9700	0.9600	0.9700	0.9100	0.8300	0.9100	0.5400
8	self	RP4	1.0000	1.0000	0.9900	0.9700	0.9800	0.9700	0.9500	0.9500	0.8600	0.8200	0.8800	0.5400
8	self	RP5	1.0000	1.0000	0.9900	0.9700	0.9800	0.9700	0.9500	0.9500	0.8500	0.8000	0.8700	0.5200
8	self	SICO	1.0000	0.9800	0.9800	0.9800	0.9800	0.9800	0.9800	0.9500	0.9800	0.9700	0.9700	0.7100
8	self	RB	1.0000	0.9500	0.9000	0.8300	0.7800	0.7200	0.6700	0.6700	0.6300	0.5700	0.5700	-
8	self	SCTS	1.0000	0.8700	0.7000	0.6100	0.4100	0.1700	0.0900	0.0900	0.0900	0.0900	0.0900	-

Table 13: ASR for `Llama-2-7b-chat-hf`, 10 samples, UMD watermarking. The trend is consistent compared with `vicuna-7b-v1.5-16k` over different c and hashing methods.

c	Hashing	Method	0 (Unattacked)	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
1	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9000	1.0000	0.0000
1	left	RP2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9000	0.9000	0.9000	0.9000	0.2000
1	left	RP3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9000	0.8000	0.9000	0.8000	0.0000
1	left	RP4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9000	0.9000	0.8000	0.9000	0.6000	0.0000
1	left	RP5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9000	0.9000	0.8000	0.8000	0.6000	0.0000
1	left	SICO	1.0000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.3000
1	left	RB	1.0000	0.5000	0.4000	0.3000	0.2000	0.2000	0.1000	0.0000	0.0000	0.0000	0.0000	-
1	left	SCTS	1.0000	0.5000	0.2000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-
2	left	RP1	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.8000	0.0000
2	left	RP2	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.7000	0.6000	0.2000
2	left	RP3	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.4000	0.6000	0.0000
2	left	RP4	0.9000	0.9000	0.9000	0.9000	0.8000	0.8000	0.9000	0.9000	0.8000	0.4000	0.5000	0.2000
2	left	RP5	0.9000	0.9000	0.9000	0.9000	0.8000	0.8000	0.9000	0.9000	0.8000	0.4000	0.4000	0.0000
2	left	SICO	0.9000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.6000	0.8000	0.5000
2	left	RB	0.9000	0.8000	0.6000	0.2000	0.2000	0.2000	0.2000	0.1000	0.0000	0.0000	0.0000	-
2	left	SCTS	0.9000	0.6000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-
4	left	RP1	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.7000	0.6000	0.6000	0.4000	0.0000
4	left	RP2	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.7000	0.6000	0.4000	0.3000	0.1000
4	left	RP3	0.8000	0.8000	0.8000	0.8000	0.8000	0.7000	0.7000	0.7000	0.6000	0.4000	0.2000	0.0000
4	left	RP4	0.8000	0.8000	0.8000	0.8000	0.8000	0.7000	0.7000	0.7000	0.6000	0.4000	0.2000	0.0000
4	left	RP5	0.8000	0.8000	0.8000	0.8000	0.8000	0.6000	0.7000	0.7000	0.6000	0.4000	0.2000	0.0000
4	left	SICO	0.8000	0.5000	0.5000	0.5000	0.5000	0.5000	0.4000	0.5000	0.4000	0.5000	0.4000	0.2000
4	left	RB	0.8000	0.3000	0.3000	0.3000	0.1000	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	-
4	left	SCTS	0.8000	0.2000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-
8	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9000	0.9000	0.8000	0.1000
8	left	RP2	1.0000	1.0000	1.0000	1.0000	1.0000	0.9000	1.0000	1.0000	0.7000	0.8000	0.7000	0.1100
8	left	RP3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8000	0.8000	0.8000	0.7000	0.0000
8	left	RP4	1.0000	1.0000	0.9000	1.0000	1.0000	0.9000	0.9000	0.8000	0.6000	0.7000	0.5000	0.0000
8	left	RP5	1.0000	1.0000	0.9000	1.0000	1.0000	0.9000	0.9000	0.8000	0.6000	0.7000	0.5000	0.0000
8	left	SICO	1.0000	0.9000	0.8000	0.9000	0.9000	0.9000	0.9000	0.8000	0.8000	0.8000	0.7000	0.3000
8	left	RB	1.0000	0.6000	0.5000	0.3000	0.1000	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	-
8	left	SCTS	1.0000	0.4000	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-
1	self	RP1	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.3000	0.0000
1	self	RP2	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.3000	0.4000	0.4000	0.0000
1	self	RP3	0.4000	0.4000	0.3000	0.4000	0.4000	0.4000	0.4000	0.3000	0.2000	0.4000	0.3000	0.0000
1	self	RP4	0.4000	0.4000	0.3000	0.4000	0.4000	0.4000	0.4000	0.1000	0.2000	0.4000	0.4000	0.0000
1	self	RP5	0.4000	0.4000	0.3000	0.4000	0.4000	0.4000	0.4000	0.1000	0.3000	0.4000	0.3000	0.0000
1	self	SICO	0.4000	0.5000	0.5000	0.5000	0.5000	0.4000	0.4000	0.5000	0.4000	0.2000	0.3000	0.1000
1	self	RB	0.4000	0.2000	0.2000	0.2000	0.1000	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	-
1	self	SCTS	0.4000	0.1000	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-
2	self	RP1	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.8000	0.0000
2	self	RP2	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.9000	0.8000	0.7000	0.8000	0.2000
2	self	RP3	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.7000	0.7000	0.6000	0.4000	0.0000
2	self	RP4	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.7000	0.8000	0.6000	0.6000	0.4000	0.0000
2	self	RP5	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.7000	0.8000	0.6000	0.6000	0.4000	0.0000
2	self	SICO	0.9000	0.8000	0.8000	0.8000	0.8000	0.8000	0.7000	0.7000	0.8000	0.7000	0.5000	0.2000
2	self	RB	0.9000	0.5000	0.2000	0.2000	0.2000	0.1000	0.1000	0.0000	0.0000	0.0000	0.0000	-
2	self	SCTS	0.9000	0.4000	0.2000	0.2000	0.2000	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	-
4	self	RP1	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.9000	0.8000	0.1000
4	self	RP2	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.7000	0.7000	0.4000	0.0000
4	self	RP3	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.9000	0.8000	0.5000	0.5000	0.4000	0.0000
4	self	RP4	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.9000	0.8000	0.5000	0.5000	0.4000	0.0000
4	self	RP5	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.9000	0.8000	0.5000	0.4000	0.4000	0.0000
4	self	SICO	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.6000	0.6000	0.5000
4	self	RB	0.9000	0.6000	0.4000	0.4000	0.3000	0.2000	0.2000	0.2000	0.1000	0.1000	0.1000	-
4	self	SCTS	0.9000	0.5000	0.2000	0.1000	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-
8	self	RP1	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.9000	0.8000	0.9000	0.7000	0.0000
8	self	RP2	0.9000	0.9000	0.8000	0.8000	0.8000	0.8000	0.7000	0.8000	0.4000	0.6000	0.4000	0.0000
8	self	RP3	0.9000	0.9000	0.8000	0.8000	0.8000	0.8000	0.7000	0.7000	0.4000	0.3000	0.3000	0.0000
8	self	RP4	0.9000	0.9000	0.8000	0.8000	0.8000	0.8000	0.7000	0.6000	0.4000	0.3000	0.3000	0.0000
8	self	RP5	0.9000	0.9000	0.8000	0.8000	0.7000	0.8000	0.7000	0.6000	0.3000	0.2000	0.3000	0.0000
8	self	SICO	0.9000	0.5000	0.5000	0.5000	0.5000	0.4000	0.4000	0.3000	0.5000	0.5000	0.4000	0.1000
8	self	RB	0.9000	0.3000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-
8	self	SCTS	0.9000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-

Table 14: Running time in seconds for `Llama-2-7b-chat-hf`, 10 samples, UMD watermarking. The trend is consistent compared with `vicuna-7b-v1.5-16k` over different c and hashing methods.

c	hashing	method	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
1	left	RP1	34	43	35	34	36	36	36	36	37	34	35
1	left	RP2	61	73	61	62	64	65	63	62	65	62	62
1	left	RP3	92	105	87	91	94	92	91	90	91	91	83
1	left	RP4	119	128	118	116	126	122	116	118	120	120	100
1	left	RP5	148	156	148	142	153	150	144	142	149	144	110
1	left	SICO	1477	1082	1727	1809	3146	2726	3117	3733	2269	2904	3504
1	left	RB	5	11	16	21	26	32	37	42	47	52	-
1	left	SCTS	1941	3950	5851	7912	9891	11722	13379	13769	13769	13769	-
2	left	RP1	39	42	40	40	38	41	41	39	41	41	38
2	left	RP2	70	84	69	74	70	76	74	70	71	76	76
2	left	RP3	102	117	96	104	101	106	103	99	104	104	107
2	left	RP4	132	146	124	146	134	137	132	124	131	135	131
2	left	RP5	161	184	157	177	162	167	159	152	155	161	149
2	left	SICO	1483	1095	1716	1810	3158	2733	3118	3736	2268	2928	3511
2	left	RB	5	11	16	22	27	32	37	43	48	53	-
2	left	SCTS	2047	4093	6472	9148	11656	13947	15238	15454	15454	15454	-
4	left	RP1	33	31	30	30	29	32	32	31	32	31	31
4	left	RP2	60	56	55	54	56	57	58	53	57	55	59
4	left	RP3	83	78	78	93	83	85	80	80	77	77	80
4	left	RP4	107	101	104	117	111	107	102	104	98	96	96
4	left	RP5	133	125	129	140	136	129	125	129	119	116	107
4	left	SICO	1477	1079	1715	1799	3153	2734	3130	3753	2258	2910	3521
4	left	RB	4	7	11	14	18	21	25	28	31	35	-
4	left	SCTS	2148	4421	6240	8416	10571	12828	13600	13619	13619	13619	-
8	left	RP1	35	33	37	34	31	34	34	33	33	33	34
8	left	RP2	56	62	71	64	61	62	63	66	60	65	67
8	left	RP3	85	91	104	96	93	93	93	95	93	93	94
8	left	RP4	117	121	135	123	122	125	130	124	122	120	116
8	left	RP5	188	159	165	154	154	157	158	151	148	144	131
8	left	SICO	1484	1104	1696	1814	3155	2733	3130	3774	2263	2899	3501
8	left	RB	4	8	13	17	21	25	29	33	37	41	-
8	left	SCTS	2713	6378	9215	12278	14906	17733	18460	18460	18460	18460	-
1	self	RP1	42	39	40	38	39	41	39	36	40	38	40
1	self	RP2	78	66	72	68	72	70	73	68	70	73	78
1	self	RP3	104	99	102	99	104	104	108	101	102	106	109
1	self	RP4	135	136	134	130	137	138	142	134	132	136	135
1	self	RP5	171	168	164	161	166	176	169	167	164	170	154
1	self	SICO	1481	1084	1707	1797	3152	2744	3092	3747	2266	2908	3505
1	self	RB	6	12	17	23	28	34	40	45	50	56	-
1	self	SCTS	1636	3250	4693	6228	7984	9762	11412	13181	14853	16772	-
2	self	RP1	42	40	38	39	36	39	35	38	39	36	38
2	self	RP2	76	75	69	70	71	70	65	70	68	71	73
2	self	RP3	109	108	102	101	102	100	97	102	94	104	99
2	self	RP4	144	142	134	134	135	130	121	129	122	130	119
2	self	RP5	176	175	169	160	165	159	148	158	147	154	133
2	self	SICO	1477	1091	1699	1778	3150	2748	3086	3745	2271	2898	3508
2	self	RB	6	11	16	21	27	32	37	42	47	51	-
2	self	SCTS	2008	4025	6334	8484	11055	13072	15032	15111	15111	15111	-
4	self	RP1	39	36	36	35	35	37	34	37	36	35	36
4	self	RP2	68	70	64	64	62	64	64	65	66	66	73
4	self	RP3	98	98	96	90	95	94	89	93	95	90	101
4	self	RP4	129	127	125	117	121	123	116	120	120	114	131
4	self	RP5	162	159	157	144	151	150	143	147	144	135	151
4	self	SICO	1474	1103	1701	1781	3131	2735	3130	3754	2267	2911	3512
4	self	RB	4	8	13	17	21	25	29	33	37	42	-
4	self	SCTS	1973	4191	6599	9006	11201	13340	14866	14993	15000	15000	-
8	self	RP1	42	36	37	37	38	38	39	38	36	37	38
8	self	RP2	75	62	71	68	69	70	69	70	65	67	73
8	self	RP3	116	92	100	98	97	98	99	104	99	91	103
8	self	RP4	144	122	133	125	126	127	127	129	123	112	127
8	self	RP5	176	148	159	156	157	154	152	155	148	133	145
8	self	SICO	1465	1080	1706	1765	3129	2737	3139	3753	2278	2901	3506
8	self	RB	4	9	13	18	22	27	31	35	40	44	-
8	self	SCTS	2709	6373	9398	12717	15919	19723	20976	20976	20976	20976	-

Table 15: # LLM calls for `Llama-2-7b-chat-hf`, 10 samples, UMD watermarking. The trend is consistent compared with `vicuna-7b-v1.5-16k` over different c and hashing methods.

c	hashing	method	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
1	left	RP1	10	10	10	10	10	10	10	10	10	10	10
1	left	RP2	20	20	20	20	20	20	20	20	20	20	20
1	left	RP3	30	30	30	30	30	30	30	30	30	30	30
1	left	RP4	40	40	40	40	40	40	40	40	40	40	40
1	left	RP5	50	50	50	50	50	50	50	50	50	50	50
1	left	SICO	235	223	251	259	474	388	422	504	297	398	498
1	left	RB	0	0	0	0	0	0	0	0	0	0	-
1	left	SCTS	317	642	947	1280	1602	1901	2172	2234	2234	2234	-
2	left	RP1	10	10	10	10	10	10	10	10	10	10	10
2	left	RP2	20	20	20	20	20	20	20	20	20	20	20
2	left	RP3	30	30	30	30	30	30	30	30	30	30	30
2	left	RP4	40	40	40	40	40	40	40	40	40	40	40
2	left	RP5	50	50	50	50	50	50	50	50	50	50	50
2	left	SICO	235	223	251	259	474	388	422	504	297	398	498
2	left	RB	0	0	0	0	0	0	0	0	0	0	-
2	left	SCTS	338	664	1051	1496	1901	2270	2481	2516	2516	2516	-
4	left	RP1	10	10	10	10	10	10	10	10	10	10	10
4	left	RP2	20	20	20	20	20	20	20	20	20	20	20
4	left	RP3	30	30	30	30	30	30	30	30	30	30	30
4	left	RP4	40	40	40	40	40	40	40	40	40	40	40
4	left	RP5	50	50	50	50	50	50	50	50	50	50	50
4	left	SICO	235	223	251	259	474	388	422	504	297	398	498
4	left	RB	0	0	0	0	0	0	0	0	0	0	-
4	left	SCTS	344	706	991	1343	1690	2049	2177	2180	2180	2180	-
8	left	RP1	10	10	10	10	10	10	10	10	10	10	10
8	left	RP2	20	20	20	20	20	20	20	20	20	20	20
8	left	RP3	30	30	30	30	30	30	30	30	30	30	30
8	left	RP4	40	40	40	40	40	40	40	40	40	40	40
8	left	RP5	50	50	50	50	50	50	50	50	50	50	50
8	left	SICO	235	223	251	259	474	388	422	504	297	398	498
8	left	RB	0	0	0	0	0	0	0	0	0	0	-
8	left	SCTS	427	1005	1444	1920	2328	2772	2885	2885	2885	2885	-
1	self	RP1	10	10	10	10	10	10	10	10	10	10	10
1	self	RP2	20	20	20	20	20	20	20	20	20	20	20
1	self	RP3	30	30	30	30	30	30	30	30	30	30	30
1	self	RP4	40	40	40	40	40	40	40	40	40	40	40
1	self	RP5	50	50	50	50	50	50	50	50	50	50	50
1	self	SICO	235	223	251	259	474	388	422	504	297	398	498
1	self	RB	0	0	0	0	0	0	0	0	0	0	-
1	self	SCTS	241	481	701	938	1209	1473	1713	1971	2235	2531	-
2	self	RP1	10	10	10	10	10	10	10	10	10	10	10
2	self	RP2	20	20	20	20	20	20	20	20	20	20	20
2	self	RP3	30	30	30	30	30	30	30	30	30	30	30
2	self	RP4	40	40	40	40	40	40	40	40	40	40	40
2	self	RP5	50	50	50	50	50	50	50	50	50	50	50
2	self	SICO	235	223	251	259	474	388	422	504	297	398	498
2	self	RB	0	0	0	0	0	0	0	0	0	0	-
2	self	SCTS	316	635	1005	1345	1762	2096	2404	2418	2418	2418	-
4	self	RP1	10	10	10	10	10	10	10	10	10	10	10
4	self	RP2	20	20	20	20	20	20	20	20	20	20	20
4	self	RP3	30	30	30	30	30	30	30	30	30	30	30
4	self	RP4	40	40	40	40	40	40	40	40	40	40	40
4	self	RP5	50	50	50	50	50	50	50	50	50	50	50
4	self	SICO	235	223	251	259	474	388	422	504	297	398	498
4	self	RB	0	0	0	0	0	0	0	0	0	0	-
4	self	SCTS	319	663	1055	1436	1796	2136	2379	2398	2399	2399	-
8	self	RP1	10	10	10	10	10	10	10	10	10	10	10
8	self	RP2	20	20	20	20	20	20	20	20	20	20	20
8	self	RP3	30	30	30	30	30	30	30	30	30	30	30
8	self	RP4	40	40	40	40	40	40	40	40	40	40	40
8	self	RP5	50	50	50	50	50	50	50	50	50	50	50
8	self	SICO	235	223	251	259	474	388	422	504	297	398	498
8	self	RB	0	0	0	0	0	0	0	0	0	0	-
8	self	SCTS	414	959	1415	1913	2389	2966	3153	3153	3153	3153	-

Table 16: Semantic similarity for `Llama-2-7b-chat-hf`, 10 samples, UMD watermarking. The trend is consistent compared with `vicuna-7b-v1.5-16k` over different c and hashing methods.

c	hashing	method	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	1 (Unconstrained)
1	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9925	1.0000	0.8462
1	left	RP2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9916	0.9887	0.9959	0.9896	0.7681
1	left	RP3	1.0000	1.0000	1.0000	0.9954	1.0000	1.0000	0.9916	0.9790	0.9919	0.9789	0.6907
1	left	RP4	1.0000	1.0000	1.0000	0.9954	0.9945	0.9983	0.9916	0.9671	0.9813	0.9523	0.6563
1	left	RP5	1.0000	1.0000	1.0000	0.9954	0.9945	0.9983	0.9916	0.9671	0.9738	0.9303	0.4560
1	left	SICO	0.9988	0.9934	0.9960	0.9850	1.0000	1.0000	0.9932	1.0000	0.9696	0.9682	0.7528
1	left	RB	0.9894	0.9792	0.9660	0.9511	0.9415	0.9315	0.9148	0.8932	0.8994	0.8809	-
1	left	SCTS	0.9903	0.9727	0.9558	0.9353	0.9136	0.8999	0.8916	0.8916	0.8916	0.8916	-
2	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9943	0.9930	0.9069
2	left	RP2	1.0000	1.0000	1.0000	1.0000	1.0000	0.9798	0.9982	0.9900	0.9741	0.9669	0.8994
2	left	RP3	1.0000	1.0000	1.0000	0.9963	0.9933	0.9792	0.9902	0.9739	0.9562	0.9492	0.8323
2	left	RP4	1.0000	1.0000	1.0000	0.9959	0.9941	0.9848	0.9881	0.9742	0.9592	0.9352	0.8028
2	left	RP5	1.0000	1.0000	1.0000	0.9959	0.9941	0.9773	0.9828	0.9681	0.9498	0.9014	0.6752
2	left	SICO	0.9981	0.9974	0.9929	0.9695	0.9879	0.9972	1.0000	0.9877	0.9733	0.4321	0.8733
2	left	RB	0.9896	0.9759	0.9577	0.9443	0.9332	0.9229	0.9088	0.8966	0.8829	0.8632	-
2	left	SCTS	0.9881	0.9740	0.9558	0.9427	0.9310	0.9204	0.9173	0.9173	0.9173	0.9173	-
4	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	0.9967	0.9943	0.9900	0.9814	0.9666	0.8900
4	left	RP2	1.0000	1.0000	1.0000	1.0000	1.0000	0.9870	0.9842	0.9900	0.9712	0.9441	0.8323
4	left	RP3	1.0000	1.0000	1.0000	1.0000	0.9951	0.9715	0.9842	0.9900	0.9697	0.9397	0.6899
4	left	RP4	1.0000	1.0000	1.0000	1.0000	0.9931	0.9715	0.9790	0.9882	0.9681	0.9161	0.6418
4	left	RP5	1.0000	1.0000	1.0000	1.0000	0.9916	0.9715	0.9772	0.9912	0.9663	0.8877	0.5192
4	left	SICO	0.9997	1.0000	1.0000	0.9835	0.9547	0.9900	0.9863	0.9774	0.9885	0.9657	0.6946
4	left	RB	0.9843	0.9703	0.9579	0.9425	0.9285	0.9071	0.9013	0.8830	0.8630	0.8630	-
4	left	SCTS	0.9847	0.9725	0.9551	0.9430	0.9222	0.8984	0.8726	0.8726	0.8726	0.8726	-
8	left	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9946	0.9856	0.9893	0.9243
8	left	RP2	1.0000	1.0000	0.9960	1.0000	0.9982	0.9958	0.9896	0.9716	0.9773	0.9676	0.8236
8	left	RP3	1.0000	1.0000	0.9960	0.9955	0.9984	0.9879	0.9845	0.9710	0.9697	0.9591	0.7269
8	left	RP4	1.0000	0.9995	0.9960	0.9918	0.9893	0.9876	0.9854	0.9602	0.9576	0.9214	0.7133
8	left	RP5	1.0000	0.9995	0.9960	0.9918	0.9875	0.9876	0.9854	0.9602	0.9511	0.9232	0.6308
8	left	SICO	0.9932	0.9993	0.9967	0.9686	0.9964	0.9959	0.9995	0.9952	0.9732	0.9892	0.7148
8	left	RB	0.9837	0.9724	0.9613	0.9473	0.9332	0.9172	0.8987	0.8815	0.8705	0.8705	-
8	left	SCTS	0.9890	0.9699	0.9556	0.9368	0.9108	0.8912	0.8912	0.8912	0.8912	0.8912	-
1	self	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9936	0.8943
1	self	RP2	1.0000	1.0000	1.0000	0.9989	0.9874	0.9996	0.9958	0.9988	0.9873	0.9869	0.8921
1	self	RP3	1.0000	0.9989	0.9997	0.9988	0.9869	0.9925	0.9828	0.9854	0.9866	0.9741	0.8119
1	self	RP4	1.0000	0.9989	0.9998	0.9989	0.9887	0.9930	0.9607	0.9750	0.9820	0.9753	0.7244
1	self	RP5	1.0000	0.9989	0.9998	0.9985	0.9881	0.9913	0.9601	0.9712	0.9555	0.9660	0.6341
1	self	SICO	0.9943	1.0000	0.9752	0.9943	0.9917	0.9892	1.0000	0.9912	0.9380	0.9796	0.8004
1	self	RB	0.9902	0.9745	0.9631	0.9500	0.9390	0.9252	0.9082	0.8912	0.8817	0.8605	-
1	self	SCTS	0.9868	0.9735	0.9589	0.9433	0.9304	0.9126	0.8907	0.8738	0.8529	0.8383	-
2	self	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9930	0.9873	0.9100
2	self	RP2	1.0000	0.9967	1.0000	1.0000	1.0000	0.9866	0.9860	0.9930	0.9699	0.9787	0.9045
2	self	RP3	1.0000	0.9967	1.0000	0.9951	1.0000	0.9776	0.9610	0.9780	0.9578	0.9214	0.8391
2	self	RP4	1.0000	0.9967	1.0000	0.9951	1.0000	0.9663	0.9611	0.9622	0.9510	0.9207	0.7726
2	self	RP5	1.0000	0.9967	0.9966	0.9951	0.9908	0.9669	0.9632	0.9498	0.9431	0.8934	0.7188
2	self	SICO	0.9992	1.0000	0.9908	0.9828	1.0000	0.9878	0.9979	0.9997	0.9798	0.9876	0.8152
2	self	RB	0.9867	0.9719	0.9545	0.9404	0.9250	0.9117	0.9004	0.8936	0.8802	0.8554	-
2	self	SCTS	0.9821	0.9606	0.9446	0.9270	0.9040	0.8851	0.8807	0.8807	0.8807	0.8807	-
4	self	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9933	0.9977	0.9918	0.8954
4	self	RP2	1.0000	0.9979	1.0000	1.0000	0.9972	1.0000	0.9894	0.9922	0.9762	0.9521	0.8938
4	self	RP3	1.0000	0.9979	1.0000	1.0000	0.9953	1.0000	0.9894	0.9741	0.9653	0.9389	0.7950
4	self	RP4	1.0000	0.9979	1.0000	1.0000	0.9953	0.9977	0.9894	0.9667	0.9245	0.9313	0.7871
4	self	RP5	1.0000	0.9979	1.0000	1.0000	0.9953	0.9977	0.9894	0.9651	0.8967	0.9282	0.7494
4	self	SICO	0.9968	0.9969	0.9942	0.9826	0.9722	0.9952	0.9913	0.9983	0.9888	0.9648	0.8156
4	self	RB	0.9901	0.9754	0.9654	0.9497	0.9353	0.9102	0.8965	0.8728	0.8636	0.8636	-
4	self	SCTS	0.9879	0.9763	0.9590	0.9330	0.9147	0.8999	0.8975	0.8932	0.8932	0.8932	-
8	self	RP1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9973	1.0000	0.9812	0.9279
8	self	RP2	1.0000	0.9957	0.9965	0.9960	0.9955	0.9874	0.9955	0.9822	0.9760	0.9455	0.9102
8	self	RP3	1.0000	0.9957	0.9965	0.9946	0.9862	0.9883	0.9902	0.9719	0.9428	0.9318	0.8043
8	self	RP4	1.0000	0.9952	0.9965	0.9923	0.9862	0.9874	0.9788	0.9444	0.9397	0.9201	0.6657
8	self	RP5	1.0000	0.9952	0.9965	0.9886	0.9862	0.9874	0.9788	0.9478	0.9220	0.9171	0.6358
8	self	SICO	0.9997	0.9973	0.9929	0.9837	0.9781	0.9939	0.9530	1.0000	0.9778	0.9931	0.8032
8	self	RB	0.9910	0.9667	0.9500	0.9296	0.9153	0.9056	0.8976	0.8744	0.8577	0.8577	-
8	self	SCTS	0.9809	0.9609	0.9292	0.9160	0.8993	0.8864	0.8864	0.8864	0.8864	0.8864	-