Where Am I From? Identifying Origin of LLM-generated Content

Anonymous ACL submission

Abstract

Generative models, particularly Large Language Models, have demonstrated remarkable proficiency in producing natural and high-quality content. However, the widespread use of such models raises significant concerns related to copyright, privacy, and security vulnerabilities associated with AI-generated material. In response to these concerns, our objective is to develop digital forensics methods tailored for large language model to trace the generator given a AI-generated content. Our methodology begins with the incorporation of a secret watermark into the generated output, facilitating traceability without necessitating model retraining. To enhance effectiveness, especially in scenarios involving short outputs, we introduce a depth watermark. This framework ensures the traceability of content back to its original source, achieving both accurate tracing and the generation of high-quality output. Extensive experiments have been conducted across diverse settings and datasets to validate the effectiveness and robustness of our proposed framework.

1 Introduction

Generative models, including GPT (OpenAI, 2023a), Gemini (Google, 2023), and LLama (Meta, 2023), have shown exceptional ability to produce natural and high-quality outputs. However, they also pose potential risks for malicious applications, such as the generation of fake news and unfounded rumors. Consequently, the imperative arises to construct a robust security auditing system capable of tracing the origin of outputs. This tracing ability should not only ensures that model utilization aligns with regulatory requirements, compliance, and ethical standards but also elevates the overall integrity of the system. Furthermore, considering generative models' proficiency in content creation and manipulation, safeguarding users' copyrights and intellectual property becomes paramount. By discerning the source of each user's generated output, we can protect legitimate contributions, preventing unauthorized replication or use of their content.

In this paper, our objective is to pioneer digital forensics methods tailored for deep learning systems to audit AI-generated content. Our innovative approach involves embedding a secret watermark into the generated output, ensuring source traceability. Importantly, we strive to make watermarking implicit, balancing the challenge of being hard to remove and undetectable without compromising the quality of the generated output. During inference and without necessitating model fine-tuning or retraining, our method dynamically adjusts the probabilities of specific tokens based on a unique code associated with the user, effectively imprinting a watermark into the text. Recognizing the need for enhanced performance in scenarios with short outputs, we introduce a depth watermark as an additional mechanism to embed the entire binary user code. This involves dividing the vocabulary into sub-lists and utilizing the added watermark to adjust the probability of the output token falling into these groups. This approach ensures accurate identification of the content generator from the user pool, relying on unique probability shifts.

Comprehensive experiments were conducted across various settings and datasets to validate our method's efficacy. We assessed accuracy on five datasets for tracing the generator from a single output sentence, and even with only 25 tokens, our proposed framework achieved over 90% top-10 accuracy across all datasets. The quality of watermarked outputs was evaluated through perplexity measures, and the robustness of our method was confirmed against text alteration attacks, maintaining a 90% accuracy for identifying the top generator in outputs of 200 tokens, even when up to 30% of the text was changed.

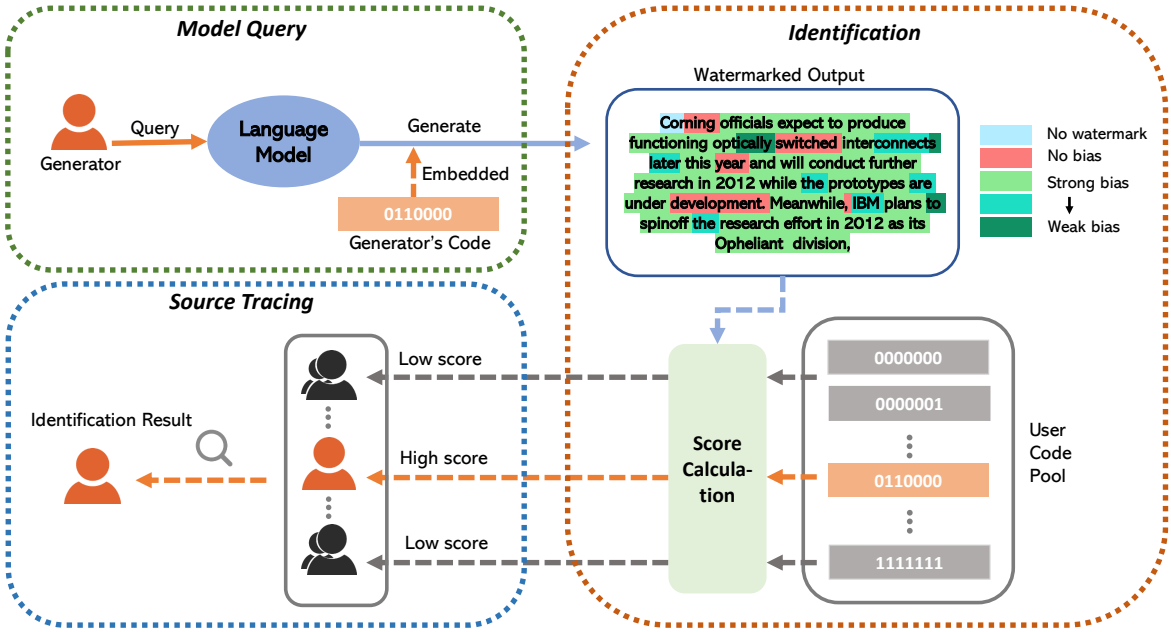

Figure 1: Illustration of the proposed watermark generation and identification flow.

2 Related Work

Watermarking. Watermarking serves as a crucial technique for embedding unique identifiers or signatures into digital signals or content to protect intellectual property or verify authenticity. In the scene of language models (LMs), watermarking plays a vital role in safeguarding the creators' intellectual property or ensuring the authenticity of generated text. Early rule-based approaches work in text watermarking involved embedding watermarks within a sentence's syntactic structure, as proposed in the post-hoc strategy by Atallah et al. (2001). Subsequently, synonym substitution techniques (Topkara et al., 2006) and semantic combination patterns for watermark construction (Xiang et al., 2021) were introduced for natural language text. Venugopal et al. (2011), were the first to propose using bit information for watermarking, while He et al. (2022) introduced the innovative concept of context-conditional textual watermarking. However, applying post-hoc watermarks to LMs faces challenges due to the diverse and versatile nature of output generation. To address this, Fang et al. (2017) proposed restricting generative models to produce tokens only from an 'allowed' set. Unfortunately, as demonstrated by Kirchenbauer et al. (2023), this constraint often leads to low-quality output, especially in low-entropy cases.

Detection of AI-generated Content. With the proliferation of large language models capable of generating high-quality and natural text, detecting the usage of machine-generated text has become increasingly important. Early methods employed natural heuristics and statistical analyses, while entropy scoring (Lavergne et al., 2008) utilized local syntactic and semantic consistency of short phrases to identify AI-generated sentences. GLTR (Gehrmann et al., 2019) assessed the expected probability of individual tokens, setting a threshold for detection. DetectGPT (Mitchell et al., 2023) observed that paragraphs created by AI often reside in the negative curvature of the log probability of texts. Some detectors, like a tool released by OpenAI (OpenAI, 2023b), incur additional training costs by using a fine-tuned GPT model to discern whether a sentence is human-written or AI-generated. Kirchenbauer et al. (2023) introduced a novel approach by creating a 'green list' randomly sampled from the entire vocabulary for every output token generation. They added a preference bias with 'green' tokens during sampling, enabling the change in output distribution to be used for watermark detection. In our work, we address a more challenging problem by not only detecting AI-generated content but also identifying the content generator for auditing purposes.

3 Problem Setting

Language Model. Let V denote the vocabulary set of the Language Model (LM), which contains all possible tokens that the LM can gener-

ate. The size of V is typically $50,000$ tokens or more (Radford et al., 2019). A language model (LM) can then be defined as a function, which accepts a sequence with arbitrary length of tokens $s_{(-N)}, s_{(-N+1)}, .., s_{(-1)}$, referred to as the prompt, and generates an output $s_{(0)}, s_{(1)}, ..., s_{(M)}$. For simplicity, output $s_{(n)}, ..., s_{(m)}$ will be denoted as $s_{(n:m)}$. Since the output $s_{(n:m)}$ is generated recursively, let $f : s_{(-N:t-1)} \rightarrow Z_t$ be the generation function at step t, where Z_t represents a score vector with logit values for each token in V. At the initial step, $t = 0$, the LM accepts the prompt $s_{(-N:-1)}$ and computes a corresponding logit which is later processed by a softmax operator to obtain a probability distribution over the vocabulary, denoted as p_0. The token at position t is sampled from this probability distribution, denoted as $s_{(0)}$. In the subsequent step, at $t = 1$, the function f takes $s_{(-N:0)}$ as input. During t-th step, f generates Z_t, and p_t, $s_{(t)}$ are computed sequentially.

Threat Model. The problem of tracing the output generated by a Large Language Model (LLM) is formalized as follows. The language model server supports text generation tasks for n users. When a user employs the language model to generate an output sentence s, it becomes imperative for the server to identify the specific generator responsible for producing that output without external information. This traceability is essential for secure audits and intellectual property protection. To protect intellectual property and ensure maintenance, the model architecture and weight details are encrypted and concealed from customers. As a result, customers only have access to input and output sentences. Conversely, the server possesses access to some user information such as a unique ID that serves as a 'fingerprint' for each client.

4 A Simple Watermark

Inspired by Kirchenbauer et al. (2023), our methodology begins with a straightforward yet highly effective watermarking method, ensuring that the generated output retains user-specific information without compromising its quality.

The process begins by assigning a unique ID to each client, which is embedded within the generated output to establish a clear link between the client and their output. This ID is then transformed into a unique binary user code denoted by $E \in \{0,1\}^k$, where k represents the length of this binary code. During Language Model (LM) generation, at each t-th step, our method initiates the process by generating a pseudo-random preferred list L_t. To construct L_t, we define $I(s_{(t)})$ as the fixed index for token $s_{(t)}$ within the vocabulary list. The context c_t for step t is defined as $c_t = I(s_{(t-1)}) * I(s_{(t-2)})$. Utilizing this c_t to seed a random number generator that splits the entire vocabulary list into two equally sized, non-overlapping lists. These lists, $V0_t$ and $V1_t$, then serves as two candidates for the preferred list L_t. Notably, the selection of the previous two tokens as the seed, instead of just the preceding one, enhances the differentiation in the splitting process.

Following this, an $indicator \in \{0,1\}$ is introduced to select the preferred list L_t from the two candidate lists, $V0_t$ and $V1_t$. To infuse user ID intuitively into the generation process, the $indicator$ is determined by a specific bit of the generator's binary user code E_i, calculated as follows:

$$indicator = \text{The } n\text{-th bit of } E_i$$

where $n = c_t \mod k$ and k is the length of E_i. Finally, the preferred list L_t is decided by the $indicator$: if $indicator$ is 0, then L_t is set to $V0_t$, otherwise L_t is set to $V1_t$.

Subsequently, the logit vector Z_t at step t, produced by the final layer of the language model, undergoes modification by adding a bias parameter δ to the logit corresponding to the token in L_t. This adjusted logit vector is then fed into a softmax operator, resulting in a biased probability distribution denoted as $\tilde{p}_t$. The calculation is shown in Equation 1, where $\tilde{p}_t^x$ signifies the biased probability of sampling the x-th token during the t-th step.

$$\tilde{p}_t^x = \begin{cases} \frac{\exp(Z_t^x+\delta)}{\sum_{u\in L_t}\exp(Z_t^u+\delta)+\sum_{v\notin L_t}\exp(Z_t^v)}, x \in L_t \\ \frac{\exp(Z_t^x)}{\sum_{u\in L_t}\exp(Z_t^u+\delta)+\sum_{v\notin L_t}\exp(Z_t^v)}, x \notin L_t \end{cases} \quad (1)$$

This process ensures a seamless integration of user-specific information into the watermarking technique, preserving both the quality of the generated output and the traceability of user-specific content.

Identification of the Simple Watermark. Given an output sentence $s_{(0:m)}$ produced by a language model and the set of all binary user E, our goal is to determine the most likely generator among all users. For each user i, we compute a confidence score indicating the likelihood of user i being the

generator. The confidence score for each user i is calculated by examining the frequency of every output token appeared in the user's exclusive preferred list. Ultimately, we select the top k users with the highest confidence scores as the top k suspects.

Limitations of the Simple Watermark. However, we find that the identification performance can be poor in cases of inadequate output length, especially when this length is shorter than that of the binary user code. This issue arises due to the resemblance in binary user codes, leading to close proximity in confidence scores. In a scenario where the k-bit binary user codes of two users differ by only one bit, the probability of choosing the same preferred lists per step is $\frac{k-1}{k}$. With insufficient output length, these two users may produce identical results with high probability, resulting in similar confidence scores. Consequently, short output length can lead to numerous users' confidence scores remarkably resembling that of the generator, thereby deteriorating the identification performance. Addressing these challenges becomes crucial, particularly in scenarios where output length constraints may impact the watermark's efficacy.

5 Depth Watermarking

To improve tracing performance and address the previously mentioned limitation, especially for short outputs, we introduce depth watermarking as an additional mechanism to embed the entire binary user code within a single generation step. Therefore, we design a more sophisticated token probability distribution, aiming to maximize the distribution difference by introducing varying biases to different sub-lists within the preferred list.

As shown in Algorithm 1, in each generation step, we construct the preferred list L_t using the same algorithm as in the simple watermark. We utilize the hash value of the generator's binary user code E_i as a seed in the random number generator to partition the preferred list L_t into several equally sized sub-lists without overlap, denoted as $L_t^0, L_t^1, ..., L_t^n$, where $L_t^0 \cup ... \cup L_t^n = L_t$. For each sub-list L_t^j within $L_t^0, \ldots, L_t^n$, we incorporate a scaled bias $(\frac{1}{2})^j \delta$ to the logit associated with L_t^j. The adjusted logits are then subjected to the softmax operator, producing a biased probability distribution $\tilde{p}_t$:

$$\tilde{p}_t^x = \begin{cases} \frac{\exp(Z_t^x + \frac{\delta}{2^j})}{\sum_{j=0}^{n} \sum_{u \in L_t^j} \exp(Z_t^u + \frac{\delta}{2^j}) + \sum_{v \notin L_t} \exp(Z_t^v)}, x \in L_t^j \\ \frac{\exp(Z_t^x)}{\sum_{j=0}^{n} \sum_{u \in L_t^j} \exp(Z_t^u + \frac{\delta}{2^j}) + \sum_{v \notin L_t} \exp(Z_t^v)}, x \notin L_t \end{cases} \tag{2}$$

Algorithm 1 Depth Watermark Generation

Input: Prompt $s_{(-N:-1)}$, binary code of generator E_i, length of binary code k, language model f, bias intensity δ

Output: Watermarked output

1: **for** $t = 0, 1, ...$ **do**
2: Apply language model on prior tokens $s_{(-N:t-1)}$ to get a logit vector Z_t.
3: Compute the product of vocabulary index of $s_{(t-1)}$ and $s_{(t-2)}$ as c_t.
4: Utilize c_t to seed a random number generator R_t, and use R_t to partition vocabulary V into $V0_t$ and $V1_t$, where $V0_t \cup V1_t = V$.
5: Calculate the result of c_t with modulus k and use the result n to determine an $indicator \in \{0, 1\}$, as n-th bit of E_i. If the identifier equals to 0, set the preferred list L_t as $V0_t$, otherwise $V1_t$.
6: Utilize the hash of E_i to seed a random generator to partition L_t into sub lists $L_t^0, ..L_t^n$, where $L_t^0 \cup ... \cup L_t^n = L_t$.
7: **for** $j = 0, ..., n$ **do**
8: For each token $\in L_t^j$, add a scaled bias $0.5^j \delta$ to its corresponding logit.
9: **end for**
10: Apply softmax operator to the new logit and get biased probability distribution $\tilde{p}_t$ over the vocabulary.
11: **end for**

Identification of the Depth Watermark. By introducing exponentially decreasing biases to the logits corresponding to each sub-list, we enhance the probability of sampling tokens from these sub-lists to varying extents. Subsequently, the 'depth score' is employed to measure the similarity between the actual and ideal probability distributions of tokens within each sub-list. Although precisely calculating the ideal probability distribution for each individual token is not feasible, we can estimate the overall trend P_{known} for these n categories in the probability distribution. This entails experimentally obtaining a probability distribution of n classes, where each probability indicates the chance of a token from that class being selected.

To estimate P_{known}, we randomly generate a fixed preferred list and its sub-lists, calculating the corresponding bias values for each sub-list. We then construct a metric dataset by sampling data from different domains. For each prompt in the metric dataset, we query the model to obtain clean logits. The bias is then added to the set of clean logits according to the sub-lists, and then processed with the softmax operator to generate the probability distribution. We calculate the probabilities of output tokens falling into each of the n sub-lists or outside the preferred list P_{known} by sampling the output from the resulting probability distribution.

Having obtained P_{known}, given a sentence $s_{(0:m)}$ generated by the LM and the set of all binary user codes E, we proceed to calculate the actual probability distribution $P_{practical}$. We examine the frequency of each token in the sentence falling into each sub-list and simultaneously track the count of tokens that do not belong to the preferred list. The resulting values, representing the count of tokens falling into the n sub-lists and the non-preferred list, are then normalized by dividing them by the total number of checked tokens, yielding $P_{practical}$.

Finally, for each user i, their depth score is defined as the negative cross-entropy of $P_{practical}$ and P_{known}. The higher the score, the more likely it is that user i is the generator.

Theoretical Analysis. Our task involves identifying the most probable user from a model's output, framed as a multi-class classification problem. By employing strategies such as one-vs-one or one-vs-all, we transform this into a series of binary classification tasks, similar to the approach proposed by Kirchenbauer et al. (2023) where they formulated a binary task to detect if a text sequence is language model-generated by checking the probability of output tokens falling within a predetermined "green list".

For each binary task, we adopt their theoretical derivation, providing an error bound. By combining these bounds, we deduce the highest error rate for our multi-class system. Specifically, in our simple watermark scheme, each generation step's preferred list L_t is determined by the user code's bit. For each bit prediction, the expected value and variance of preferred list token counts are reduced to the bounds established by Kirchenbauer et al. (2023). Given a threshold, the probability of binary misclassification ϵ can be derived. With an n-bit code, the worst-case probability of wrong prediction is $1-(1-\epsilon)^k$, where 2^k equals the number of users in a full capacity assignment. As ϵ is actually quite small(on the order of 10^{-5}), our proposed scheme would achieve good detection performance. Our average-case bound is even better, as some bit misclassifications do not lead to incorrect tracing. Sparse assignment (where not all n-bit codes are assigned) further improves this bound.

Our depth watermark approach partitions the preferred list into n sub-lists with an exponentially decreasing bias. Compared to the simple watermark's fixed bias on half the vocabulary, the depth scheme affects fewer words, increasing the post-softmax probability of sampled tokens landing in the selected sub-list. This reduces the binary misclassification probability ϵ, leading to a lower rate of wrong predictions in our multi-class system.

6 Experiments

6.1 Experimental Setup

Implementation Details. We assess the performance of the proposed depth method under various configurations using OPT-1.3B (Zhang et al., 2022) as the language model with 1.3 billion parameters. Five diverse datasets spanning various everyday domains are employed: News-like text from the C4 dataset (Raffel et al., 2020) (odc-by license), news articles from the XSum dataset (Narayan et al., 2018) (cc-by-sa-4.0 license), prompted stories from the Reddit WritingPrompts dataset (Fan et al., 2018) (MIT license), Wikipedia paragraphs from SQuAD contexts (Rajpurkar et al., 2016) (cc-by-4.0 license), and long-form answers written by human experts in the PubMedQA dataset (Jin et al., 2019) (MIT license). We use the Pytorch interface of Huggingface library (Wolf et al., 2020) to implement the proposed watermarking method. Every experiment is conducted with 200 distinct fixed-length prompt sentences. For each prompt, five users were randomly selected from the user pool with 1024 users to act as generator. The mean of multiple experiment results is reported. Identification accuracy is examined across five datasets using the fixed P_{known} derived from the C4 dataset. All experiments are configured with the number of sub-lists n set to 3, bias intensity δ ranging from 3 to 5, the number of tokens per output fixed at 25, and the binary user code length set to 10, corresponding to a user pool size of $2^{10} = 1024$. All experiments are conducted on NVIDIA RTX 3090, and identification for a sentence takes around 0.45

minutes.

Evaluation Metrics. To evaluate the efficacy of our proposed method, we use the top-1, top-3, and top-10 identification accuracy to reflect the probability of the generator belongs to the top 1, 3, or 10 suspects, respectively. The quality of the watermarked output was assessed using perplexity (Beresneva, 2016) with an oracle model of OPT-1.3B, which quantifies the divergence between the original distribution of output and that of our watermarked model.

6.2 Main Results

Table 1 illustrates a notable enhancement in the accuracy of text generator identification across a user pool of 1024 clients with varying δ values. Particularly, when $\delta = 5$, the top-10 accuracy metric of our methodology consistently surpasses 90% for each dataset. For top-1 accuracy, the majority of outcomes exhibit satisfactory performance, with values approximating or exceeding 90%. Notably, in the absence of our proposed method, the probability of correct prediction through random guessing would be a mere 0.1%, 0.3%, and 1% for top-1, top-3, and top-10 identification accuracy, respectively, in the given context.

This assessment of identification accuracy across datasets emphasizes the adaptability and robustness of our watermarking approach, with the depth watermarking method yielding consistently high accuracy. However, it is noteworthy that the performance on the WritingPrompts dataset is comparatively less favorable than on other datasets. This outcome might be influenced by the unique characteristics of the WritingPrompts dataset, which encompasses a diverse range of creative and narrative texts. Addressing this challenge could be the focus of future works, aiming to formulate an improved watermarking scheme.

6.3 Ablation Studies

We exam the effect of hyperparameters on the quality of output and the identification performance. For each experiment, a corpus of 200 distinct, fixed-length text was sourced from the C4 dataset as prompts for the experiments. Correspondingly, a generator is randomly selected from the user pool for each prompt. The default settings include bias intensity$\delta = 5$, number of sub-lists=3, output length=200, and user pool size=1024.

Bias Intensity δ. By selecting a large bias intensity δ, the output generated by different users can be more distinct, thereby enhancing the watermark efficiency. However, creating a strong watermark may perturb the output distribution of the language model, potentially leading to a deterioration in output quality. Table 2 shows the correlation between the efficiency of watermarking (as measured by top-1 identification accuracy) and the quality of output (as quantified by perplexity, smaller is better) across various bias intensities. Additionally, we have included illustrative instances of human written prompts, unwatermarked and watermarked outputs, as well as watermarked perplexity within Table 3, thereby offering a qualitative perspective on the performance of the quantitative measurement.

The Length of Output Sequence. The length of the output sequence is considered to be proportional to the efficacy of identification. Figure 2 presents the correlation between identification performance and output lengths, across a spectrum of bias intensities δ. We also include the simple watermark in Section 4 with $\delta = 5$ as a reference. From Figure 2, we can see depth watermarking consistently outperforms simple watermarking across nearly all tested δ, with a particularly significant advantage for shorter output lengths. For the depth method, it is evident that a delta value of 5 yields extremely strong watermarking performances. As output sequence length exceeds 50, the top-1 identification accuracy is nearly 100%, indicating the exceptional performance of our proposed watermark.

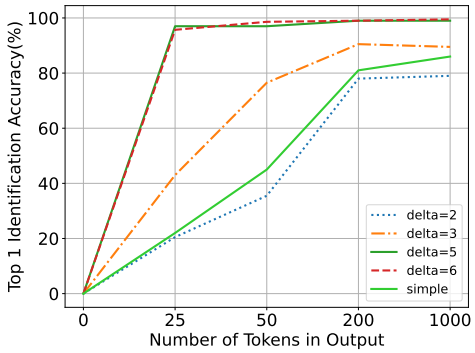

Figure 2: The top-1 identification accuracy against the length of the generated text.

Number of Sub-lists. Recall that the preferred list is partitioned into several sub-lists in our depth watermarking. The division inherently impacts

Dataset	$\delta = 4$			$\delta = 5$			$\delta = 6$		
	Top1 Acc	**Top3 Acc**	**Top10 Acc**	**Top1 Acc**	**Top3 Acc**	**Top10 Acc**	**Top1 Acc**	**Top3 Acc**	**Top10 Acc**
C4	77.3	87	92.6	89.1	92.5	95.3	95.7	97.9	98.4
XSum	78.5	86.9	93.1	90.3	94.3	95.9	95.0	97.2	98.0
WritingPrompts	57.3	67	76.4	77	85.4	90.6	87.5	91.3	94.2
SQuAD	76.2	86.5	92.6	93.7	96.6	98.1	97.9	99.6	99.9
PubMedQA	83.7	90.4	95.1	94.1	97.5	98.5	96.9	98.7	99.2

Table 1: Identification accuracy (%) on different datasets with varying δ.

δ	**Top1 Acc** %	**Perplexity**
4	78	11.17
5	89	17.56
6	96	23.75
7	97	30.44

Table 2: Trade-off between watermark strength and output quality.

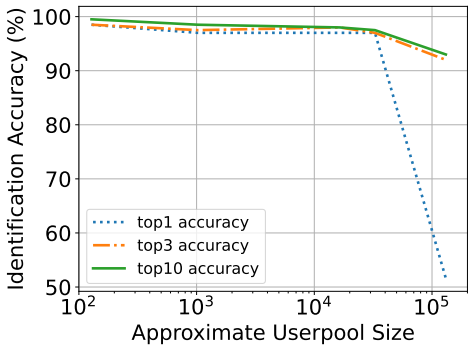

Figure 3: Identification performance against different size of userpool.

the variation in the bias vector to be added to the logit, consequently weaken the output quality, as measured by perplexity. Table 4 demonstrates that perplexity worsens as the number of sub-lists increases, conforming with the hypothesis we made above. While a configuration employing three sub-lists offers the most effective identification performance. This accuracy can be influenced by the distribution variance within the sub-lists. An increase in the number of sub-lists can enhance this variance, yet may also result in negligible differences. (Increment of the number of sub-lists results in a more uniform distribution.) Evidently, the configuration with 3 sub-lists strikes an optimal balance.

Size of User Pool. An intriguing question is about the boundaries of the effectiveness of the depth watermark scheme. Specifically, the question arises as to whether the proposed method retains its accuracy when applied to significantly large user pools. This experiment was conducted to evaluate the performance of our method across user pools of varying sizes. It involved testing binary user code lengths of 7, 10, 14, 15, and 17, corresponding to user pool sizes of 128, 1,024, 16,384, 32,768, and 131,072, respectively. As illustrated in Figure 3, the top-3 identification accuracy remarkably maintains a level above 90%, even with a user pool expanding to 131,072 members. In such a context, a reduced top-1 accuracy is deemed acceptable, considering that identifying a set of top-3 suspects remains sufficiently precise for a user pool of this magnitude.

6.4 Estimation of Ideal Distribution

In Section 5, the calculation of the depth score relies on the ideal distribution P_{known}, derived from the C4 dataset. It is imperative to ensure the proximity of P_{known} to estimations obtained from other datasets. To achieve this, we have undertaken the generation of outputs using 5000 different prompts from every datasets where each output consists of 25 tokens. The ideal probability P_{known} is computed for each of the five datasets, employing a bias intensity of $\delta = 5$ and a number of sub-lists $n = 3$, as detailed in Table 5. It can be observed that the P_{known} generated with different datasets are largely consistent so that the ideal distribution could be used across different datasets.

7 Robustness Analysis

The robustness of the proposed watermarking scheme, particularly in maintaining the stability of identification results, presents an essential inquiry. In this section, we further discuss the performance of our scheme under adversarial attacks.

We investigate a black-box text substitution attack designed to mimic the perturbations one might encounter in the real-world application. An adversary alters the original output text by one token each step using a substitution model, and iteratively conduct the attack until a alteration budget ϵT is reached, where T denotes the output length. This is

Prompt (Written by Human)	Watermarked Output	Unwatermarked Output	Perplexity
...The experts concluded that each 50 gram portion of processed meat eaten daily increases the risk of	stomach cancer significantly, similar to the findings from analyses on blue, red, salted and unsalted meat, poultry and pork[...continues]	stomach cancer by 16 per cent." In February, the Brazilian Centre for Public Health Research (CBP) said[...continues]	15.95
...One thing we have to do is measure the contribution of arts to the big picture. Take for instance the yearly jazz fes	itval, for one, and then look at the economic, social development and political effects. It will show you how much[...continues]	itval held in Johannesburg. I have been on the committee that has been looking at this for five years. We have[...continues]	17.11
...I am really sad they decided to cancel it and I have no idea why. I hope the cast move on to other things that I will enjoy	watching and I hope to see the last 7 seasons on tv, and maybe an online movie, there are some great actors that[...continues]	. I wouldnt say the show was bad, but it wasnt as good as it once was, but I still[...continues]	19.62

Table 3: Prompt and output w/o the proposed watermark generated by prompts in the C4 dataset.

# of sub-lists	Top1 Acc	Perplexity
2	68.5	14.35
3	89	17.56
4	91.5	20.93
5	94.5	21.40

Table 4: Top-1 identification accuracy(%) and output quality against number of sub-lists.

Dataset	Ideal Distribution P_{known}
C4	[*0.5304, 0.1397, 0.1060*, 0.2239]
XSum	[*0.4828, 0.2038, 0.1365*, 0.1769]
WritingPrompts	[*0.6494, 0.1040, 0.1100*, 0.1366]
SQuAD	[*0.5508, 0.1474, 0.0709*, 0.2309]
PubMedQA	[*0.5892, 0.1031, 0.1177*, 0.1900]

Table 5: Probability Distribution P_{known} of tokens in sub-lists and outside the preferred list for various datasets (First 3 coordinates: probability in sub-list; Last element: probability outside preferred list)

accomplished without the adversary's knowledge of the output's distribution or the prior knowledge P_{kown}. It is imperative to highlight that the budget ϵ serves as a regulatory constraint, ensuring that the resultant text maintains a degree of similarity to the original. Specifically, we employ the OPT-1.3b - the watermarking model with bias $\delta = 0$ as the adversary's substitution model. The process begin with tokenizing the watermarked text S of length T, and randomly select a token $s_{(t)} \in S$ that never been modified for attack. We extract the preceding 30 tokens of $s_{(t)}$ in the original watermarked text to form a prompt for the substitution model. We attack the original watermark text with the output length $T = 200$. As depicted in Figure 4, we assess the effectiveness of our scheme through the top k identification accuracy in the face of outputs generated with $\delta = 5$ subjected to attacks with varying attack budgets $\epsilon = 0.05, 0.1, 0.3, 0.5, 0.7$. It is encouraging to note that, when the attack budget is limited to 0.3 or less, ensuring that at least 70% of the original content remains unaltered, our scheme's top-1 accuracy showcases exceptional resilience, maintaining an identification accuracy exceeding 90%. This underscores the robustness of our methodology in achieving high accuracy levels, even when confronted with adversarial modifications.

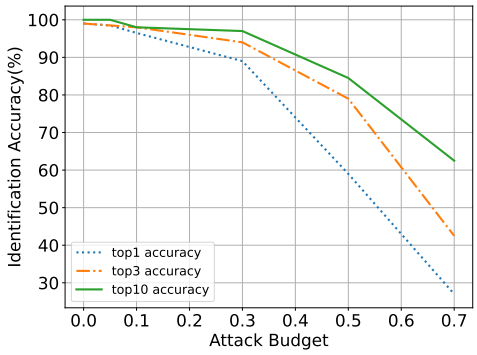

Figure 4: Identification accuracy under attack with various attack budget ϵ.

8 Conclusion

In this study, we introduced a comprehensive traceability framework designed to identify the origin of specific sentences generated by language models. Our novel watermarking technique, which uniquely embeds a user code within each user's output, enables precise source identification through a robust scoring mechanism. Empirical evaluation of our framework yielded promising results. Notably, for outputs exceeding 50 tokens, we achieved a precision of nearly 100% across a cohort of 1024 users, underscoring the effectiveness of our methodology. Furthermore, rigorous experimentation confirmed the robustness of our framework, demonstrating its resilience in diverse scenarios and challenges.

Limitations

A significant limitation is the scalability in terms of user capacity. Our proposed scheme might face challenges when deployed in super-heavyweight application like ChatGPT with around 180.5 million users. We have demonstrated in Section 6.3 that the largest user pool size of our method can be about 10^5 (unless using excessively long output). We expect that a more carefully designed watermarking scheme could support even larger user pool size, which we leave as future works. Moreover, the robustness of the scheme against sophisticated adversarial attack, including but not limited to the generative attack that could alter the watermarked content remains a great concern. In Section 7, we only discussed the robustness under 'text alteration' attack. The robustness against other attacks remains to be tested.

References

Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. 2001. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding: 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings 4*, pages 185–200. Springer.

Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *Natural Language Processing and Information Systems: 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK, June 22-24, 2016, Proceedings 21*, pages 421–426. Springer.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Tina Fang, Martin Jaggi, and Katerina Argyraki. 2017. Generating steganographic text with LSTMs. In *Proceedings of ACL 2017, Student Research Workshop*, pages 100–106, Vancouver, Canada. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Google. 2023. Introducing gemini, our largest and most capable ai model. https://llama.meta.com.

Xuanli He, Qiongkai Xu, Yi Zeng, Lingjuan Lyu, Fangzhao Wu, Jiwei Li, and Ruoxi Jia. 2022. Cater: Intellectual property protection on text generation apis via conditional watermarks. *Advances in Neural Information Processing Systems*, 35:5431–5445.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

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

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. *Pan*, 8(27-31):4.

Meta. 2023. Discover the power of llama,. https://blog.google/technology/ai/google-gemini-ai/#sundar-note.

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

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

OpenAI. 2023a. Chatgpt is a free-to-use ai system. https://chat.openai.com/.

OpenAI. 2023b. New ai classifier for indicating ai-written text. https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Umut Topkara, Mercan Topkara, and Mikhail J Atallah. 2006. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In *Proceedings of the 8th workshop on Multimedia and security*, pages 164–174.

Ashish Venugopal, Jakob Uszkoreit, David Talbot, Franz Josef Och, and Juri Ganitkevitch. 2011. Watermarking the outputs of structured prediction with an application in statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1363–1372.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Tao Xiang, Chunlong Xie, Shangwei Guo, Jiwei Li, and Tianwei Zhang. 2021. Protecting your nlg models with semantic and robust watermarks. *arXiv preprint arXiv:2112.05428*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A More Results of Bias intensity.

We report the additional results for top-1, top-3, top-10 accuracy, along with the perplexity measurement for δ ranging from 0 to 7 in Table 6. The experimental setting aligns with which in Section 6.3, wherein each result represents the average across 200 runs. For each experiment, a corpus comprising 200 unique text excerpts, each fixed at a length of 25 tokens, was extracted from the C4 dataset as prompts, and the number of sub-lists is fixed to 3.

δ	Top1 Acc %	Top3 Acc %	Top10 Acc %	Perplexity↓
0	0	0	0	5.3692
1	1.5	5.5	11	5.3705
2	20.5	35.5	52	6.8072
3	43	58.5	71.5	8.549
4	78	87	92.5	11.1726
5	89	97.5	98.5	17.5585
6	96	98	98.5	23.7451
7	97	98	98.5	30.4353

Table 6: Top1,3,10 identification accuracy(%) and output quality against bias intensity δ.

B Additional Results of User Pool Size

Accuracy Result. We report the additional experimental result for size of user pool in this section. The experimental setting is also aligned with section 6.3, with a configuration of 3 sub-lists with a bias intensity $\delta = 5$. The experiment involved testing binary user code lengths of 7, 10, 14, 15, and 17, corresponding to user pool sizes of 128, 1,024, 16,384, 32,768, and 131,072, respectively. As demonstrated in Figure 5, there is a notable decline in the performance of the watermarking scheme for output length equal to 25 when applied to a large user pool. This phenomenon is reasonable considering the constraint of having too short outputs, wherein the variance in output sentences among different users is inherently limited.

Running Time. As the size of the user pool increases, the complexity of the identification task expands correspondingly, resulting in extended processing times. We documented the time consumption to trace a single sentence generated by an individual user within a single execution, across various magnitudes of user pool sizes. Each experiment was conducted on a single RTX 3090 GPU, with the number of output sentence fixed to 25, sub-lists number fixed to 3 and bias intensity $\delta = 5$.

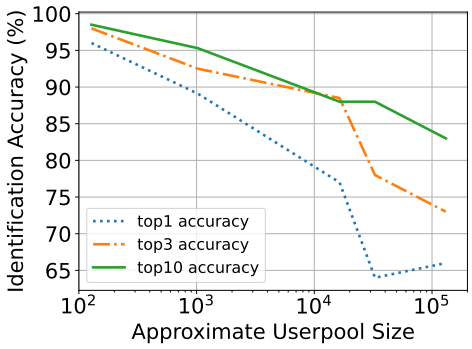

Figure 5: Identification performance against different size of userpool. (output length = 25)

User Code Length	User Pool Size	Time Used per Single Run
7	128	0.25
10	1024	0.45
14	16384	11.5
15	32768	17.2
17	131072	41
18	262114	228
19	524288	663

Table 7: Running time (minutes per single run) at different user pool size.

C More Result of Watermarked and Unwatermarked Samples.

Here we demonstrates how our depth watermark maintains the output quality while achieving high trace accuracy. Table 8 presents a range of examples comparing human written prompt, unwatermarked output, and watermarked output. The watermark's bias intensity δ is set to 5.

Prompt (Written by Human)	Unwatermarked Output	Watermarked Output
Plump up a cushion, have a listen, and post your feedback on the blog below.	And remember, you can subscribe and have your say on our app for iOS and Android, by searching FFT. WEEKEND FOOTBALL April 25, 2017 AC Jimbo is back with Football Weekly	And, remember - if you're liking our content on our website, please leave a review in the comments, so we'll get to know you better! And, don't forget that you can also follow your favourites - and get updates on their stories
irits and our emotions. We cannot continue that way, because we are missing that sense of belonging.	What are your views on the topic of white on black crime? The issue is not about white on black crime. There is a difference between white on black and black on white. It is about the dignity of every South African.	That sense is what gives away the real deal that South Africans have. It's important for us to remember the spirit, which has to be brought alive through things, music or paintings or bigger expressions that involve people of culture – not about collecting
over and over are over. She urged student to take advantage of the Open University because struct	uring of the classes gives them the freedom to learn on their own time. LAWEH University offers a wide variety of programmes and degrees including the BSc in Chemistry, BSc in Biochemistry, BSc in Environmental Science, BSc in	uring of programmes, online and face-to-face courses, combined with assessment methods, have been streamlined to make things easier and make things happen better for the learner. "Ours is one open access model so you will get the
mber of new flights lead this week's air travel news agents should keep on their radar. Take a look.	Virtuoso's Luxe Report surveyed 1,071 of its advisors and found travelers are increasingly looking to personalize their trips with staged photo shoots, private helicopter transfers and more. The hotel offers 276 rooms, five distinct	Airport infrastructure upgrades have boosted Denver as the fourth-hottest travel destination. The company has invested around $500 million and plans for more expansion in 2017. Coast to coast to airport: Delta Airlines has revealed plans for its first
Each one is made artisanally in Queens and Brooklyn, at what we call chop shops," Dubrovsky says. "E	ighty-five percent of the materials and labor are on site, and the rest are shipped in." The MRV100 has a big, bright green paint job, and large, black wheels that are easy to steer. The cart is	ighty per cent are made by the vendors from scraps." To give some context: It can take two people to fill and operate a food cart. "It costs a pretty big sum, and these vendors are running on their last cents
usually provide the theme several days in advance, and encouraged anyone interested to get involved.	"It's a lot of fun. It's a lot of fun for us. It's a lot of fun for the community. It's a lot of fun for the tourists, too," she said. "There are a lot of	The final competition is set for 4:30 p.m. Saturday at the Colorado River Mall. The winner(s) will win money toward travel to participate in a two-day contest of ice carving in Idaho at Vashon Island
archants Plants, also uses it in his grassled garden and he describes it as "remarkable and lofty".	Diascias are a riot of colour on a sunny window sill, and they are a good substitute for potted dahlias, which are often seen as dahlias. As mentioned above, diascias will flower for	Next time I'm away I'm taking some 'Appleby apricot' D. barberae with the pale blue and pink roses from Appleby's. I'm hoping to add to this collection and, if the weather holds out,
video of our conversation. Let me know what you thought of the show. See you next Sunday at 11 a.m.!	Copyright 2021 NPR. To see more, visit https://www.npr.org. DAVID GREENE, HOST: We want to turn now to the latest big story about Hillary Clinton and her memoir, "Hard Choices	Copyright 2021 NPR. To see more, visit https://www.npr.org. TERRY GROSS, HOST: This is FRESH AIR. And welcome back. And welcome back to FRESH AIR as well. Today
or Indiancitizenship is under the consideration of Joint parliamentary committee of both the Houses.	Congress has been opposing the bill and has also threatened to form an alliance with another political party if the bill is passed. The bill has triggered series of protest in Assam, All Assam Students Union and several other organisations have announced fresh agitation	Recently the Congress had also slammed and asked why the Bill is going to become another issue after 15 January 1946 in the state. Congress said that in February this year a protest will start with burning effigies and in March, students will start anti
I am really sad they decided to cancel it and I have no idea why. I hope the cast move on to other	things and I would love to see them return to college campus, I miss them all. I hope they come back to the Greek culture, the show was about Greek life and I think it would be great if they come back with a new Greek	projects somewhere. I will miss it so much. So good. I will watch all of it!!!!!!!!! I really thought they could of at least ended with this season if they were going to go through with this season so long

Table 8: A intuitive comparison of the watermarked outputs and unwatermarked outputs.