

# CoTSRF: Utilize Chain of Thought as Stealthy and Robust Fingerprint of Large Language Models

Zhenzhen Ren GuoBiao Li Sheng Li  
Zhenxing Qian Xinpeng Zhang

Fudan University

{24110240140, 20210240200, lisheng, zxqian, zhangxinpeng}@fudan.edu.cn

## Abstract

Despite providing superior performance, open-source large language models (LLMs) are vulnerable to abusive usage. To address this issue, recent works propose LLM fingerprinting methods to identify the specific source LLMs behind suspect applications. However, these methods fail to provide stealthy and robust fingerprint verification. In this paper, we propose a novel LLM fingerprinting scheme, namely CoTSRF, which utilizes the Chain of Thought (CoT) as the fingerprint of an LLM. CoTSRF first collects the responses from the source LLM by querying it with crafted CoT queries. Then, it applies contrastive learning to train a CoT extractor that extracts the CoT feature (i.e., fingerprint) from the responses. Finally, CoTSRF conducts fingerprint verification by comparing the Kullback-Leibler divergence between the CoT features of the source and suspect LLMs against an empirical threshold. Various experiments have been conducted to demonstrate the advantage of our proposed CoTSRF for fingerprinting LLMs, particularly in stealthy and robust fingerprint verification.

## 1 Introduction

Recent advanced large language models (LLMs) demonstrate powerful natural language understanding, generation, and reasoning capabilities and have been widely applied in various fields such as healthcare (Wang et al., 2024b), education (Wang et al., 2024a), software development (Xia et al., 2024), and scientific research (Xia et al., 2024). Despite their remarkable success, training a high-performing LLM is not a trivial task, requiring a large scale high-quality data and massive amount of computation resources. Fortunately, in actively embracing the ethos of openness, many leading teams in the AI industry have generously released their trained LLMs on open-source platforms such as GitHub and Hugging Face. Notable examples of these open-source LLMs include LLaMA (Touvron et al., 2023), Guanaco (Dettmers et al., 2024),

and Vicuna (Chiang et al., 2023), which empowers practitioners with limited resources to conduct further experimentation, fine-tuning, and downstream application development.

For both commercial and ethical reasons, LLM providers typically release their trained LLMs with crafted licenses (Creative Commons, 2024; Free Software Foundation, 2024). These licenses restrict the use of the published LLMs, prohibiting their application in commercial or illegal activities. However, tempted by the enormous profits, some downstream developers may bypass these restrictions, building entities based on open-source LLMs to provide services through APIs, even if these services directly compete with the LLM providers. On the other hand, malicious users may intentionally compromise the LLM’s internal alignment mechanisms by fine-tuning, using the LLMs to spread harmful content. Therefore, it becomes a pressing concern for LLM providers to safeguard their released source LLMs against abusive usage (termed as LLM infringement for short) that violates their licenses.

A promising way to address the above issues is model fingerprinting, which non-intrusively extracts the unique features of a model. Generally, the external manifestation of most of the fingerprinting methods is specific input-output pairs. The inputs are carefully designed to trigger the source model to produce a unique answer while making other benign models to generate different responses. As such, by comparing the unique answer with the response of a suspect API when fed the specific inputs, the model providers could determine whether the model used behind the API is their released one. Currently, model fingerprinting has achieved significant success in protecting deep neural networks (DNN) (Lukas et al., 2019; Zheng et al.; Peng et al., 2022; Guan et al., 2022; Quan et al., 2023). However, the research on LLM fingerprinting is still in its infancy, possibly due to computational resource

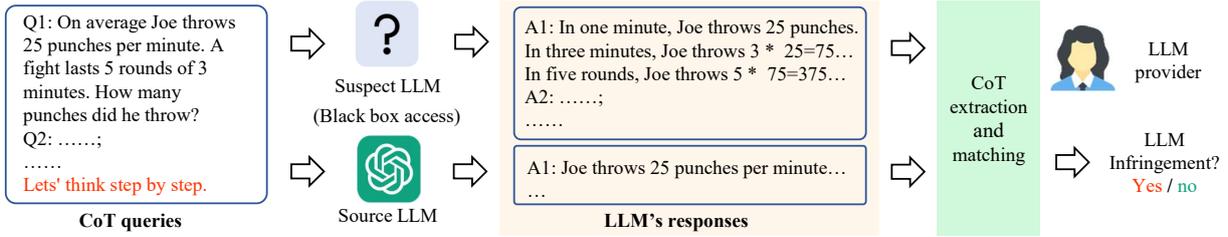


Figure 1: LLM fingerprint verification process of the proposed method in a black-box access setting, where the model provider only has API access to the suspect LLM.

limitations and the discrete nature of text data.

A recent proposed LLM fingerprinting method is TRAP (Gubri et al., 2024), which combines adversarial prefix (Zou et al., 2023) with queries to induce the source LLM to generate a predefined content. It verifies LLM infringement by comparing the content with the response of the suspect LLM when fed the combined prompts. Unfortunately, TRAP is not stealthy, as its added adversarial prefixes is meaningless characters, which disrupt the semantic coherence of prompts. This may alert malicious downstream users and let them to obstruct LLM infringement verification through denial-of-service measures. On the other hand, TRAP is not robust to output perturbation. If the malicious downstream developers modify the LLM’s hyperparameters (e.g., temperature), the infringing LLM (i.e., a copy of the source LLM) may return response that does not match the predefined content.

To bridge this gap, we propose a novel method, namely CoTSRF, which utilizes the Chain of Thought (CoT) as the LLM’s Stealthy and Robust Fingerprint. Our key insight is that the profile of an LLM can be uniquely characterized by its logical reasoning pattern represented by the CoT. Unlike TRAP (Gubri et al., 2024), Our CoTSRF verifies the LLM infringement in a stealthy and robust manner, as shown in Fig. 1. LLM provider first queries the suspect API using a combination of reasoning questions and a standard CoT prompt (e.g., let’s think step by step). The queried LLM then returns responses that implicitly reveal its logical reasoning patterns. Finally, the provider extracts the CoT features from the responses and compares them with those of the source LLM to identify the LLM infringement. CoTSRF uses the widely adopted CoT prompt to query the infringing LLM, ensuring that it does not alert malicious developers. Moreover, it performs fingerprint matching at the feature level, which is more robust against the output perturbation.

In methodology, CoTSRF begins by obtaining the responses of the source LLM by querying it using reasoning questions and the standard CoT prompt. During this, a High-Temperature Data Augmentation (HTDA) strategy is designed and utilized to generate diverse positive responses that vary in word space but follow the same logical reasoning pattern. Additionally, benign LLMs are used to create negative responses with distinct CoT features. These positive and negative responses are then employed in a contrastive learning framework to train a CoT extractor for accurate CoT feature extraction. In the LLM infringement verification, CoTSRF compares the KL divergence between the CoT features of the source and suspect LLMs against an empirical threshold. Various experiments have been conducted to demonstrate the advantages of our proposed CoTSRF for LLM infringement verification. The main contributions are summarized below:

- We present the first attempt to leverage CoT as LLM’s fingerprint for black-box LLM infringement verification.
- We propose a novel LLM fingerprinting method CoTSRF that achieves highly competitive results in terms of the stealthy and robustness.
- We adopt contrastive learning to train a CoT extractor that accurately extracts the LLM’s fingerprint from its responses and propose an HTDA strategy to create diverse responses from the LLM.

## 2 Related Works

### 2.1 Model fingerprinting

Model fingerprinting technology non-intrusively extracts the unique features of a source model and uses these features to identify infringing models from benign ones. It has flourished in protecting

DNNs (Lukas et al., 2019; Zheng et al.; Peng et al., 2022; Guan et al., 2022; Quan et al., 2023). Currently, few attempts have been made in LLM fingerprinting. Zeng *et al.* take the vector direction of the LLM’s parameters as the fingerprint and achieve remarkable performance (Zeng et al., 2023). However, their method requires white-box access to the internal parameters of the suspect LLM during fingerprint verification, making it unsuitable for cases where the malicious developer only provides API access. Gubri *et al.* propose TRAP, which utilizes an adversarial prefix to trigger the source LLM to output a unique answer and takes the mapping of the adversarial prefix and the answer as the fingerprint (Gubri et al., 2024). Despite supporting fingerprint verification in the black-box setting, TRAP is not stealthy enough to avoid prompt filtering and lacks robustness against output perturbation.

## 2.2 LLM Watermarking

LLM watermarking presents a potential way to address the issue we have highlighted. It intrusively embeds watermark information within the weights or outputs of the LLM. Most of the LLM watermarking methods follow a black-box paradigm, where the LLM is fine-tuned to remember distinctive input-output pairs (Li et al., 2023; Peng et al., 2023; Xu et al., 2024). They verify the LLM infringement by comparing the distinctive outputs with the responses of a suspect LLM when fed the specific inputs, which is similar to that of LLM fingerprinting. However, LLM watermarking technology modifies the LLM when embedding the watermark, which inevitably affects the LLM’s performance. On the other hand, it cannot adapt to those LLMs that have been released without being watermarked.

## 2.3 Chain of Thought

Chain of thought (CoT) mirrors LLM’s logical reasoning path when solving systematic and complex problems. Recent researches propose CoT prompting, which significantly enhances the reasoning abilities of LLMs and makes the output logic of LLMs more reasonable and the results more accurate. This technology designs CoT prompts to guide the LLM in deconstructing complex problems into orderly sequences of logical steps (Wei et al., 2022). Currently, CoT prompting methods could be broadly divided into two categories: zero-shot CoT and few-shot CoT. The former enables models to generate reasoning steps and solve tasks

without any prior examples, effectively leveraging their pretrained knowledge (Kojima et al., 2022; Wang et al., 2023; Chen and Liu, 2023; Yuan et al., 2024). The latter involves generating intermediate reasoning steps for a task, leveraging a handful of examples to enhance model performance (Huang et al., 2023; Song et al., 2023; Liu et al., 2024a).

Recent works have demonstrated that the CoT is LLM’s internal attribute, which is highly related to LLM’s architecture, training dataset, and training strategy (Feng et al., 2024; Liu et al., 2024b). Therefore, in this paper, we argue that CoT could uniquely characterize an LLM and serve as its fingerprint. We then make full use of the popular zero-shot CoT prompting methods to extract LLM’s CoT. Through comprehensive experiments, we empirically demonstrate its effectiveness in identifying specific LLMs and its advantages in stealthy and robust LLM infringement verification.

# 3 Problem Formulation

## 3.1 Threat Model

The threat model of the paper involves two parties: the LLM provider and the malicious downstream developer. The LLM provider releases a source LLM under a carefully designed license (e.g., a non-commercial license), while the malicious developer ignores these restrictions and builds entities based on the downloaded source LLM to offer profitable services through APIs.

The provider’s goal is to identify the infringing LLM that is remotely deployed by the malicious developer. The provider has the following capabilities: 1) white-box access to the source LLM, 2) black-box access to the infringing LLM, and 3) a limited set of queries to conduct fingerprint verification. The malicious developer’s goal is to utilize the infringing LLM to provide commercial services without being noticed by the LLM provider. To prevent the fingerprint verification, the malicious developer conducts the following strategies: 1) at the input level, preset a prompt filter module to check and filter out the odd queries; 2) at the LLM level, fine-tuning the infringing LLM to alter its fingerprint; and 3) at the output level, imposing perturbation to disrupt the fingerprint detection. It should be noted that, the intensity of these strategies must be carefully controlled, as excessive manipulation could compromise the performance of the infringing LLM.

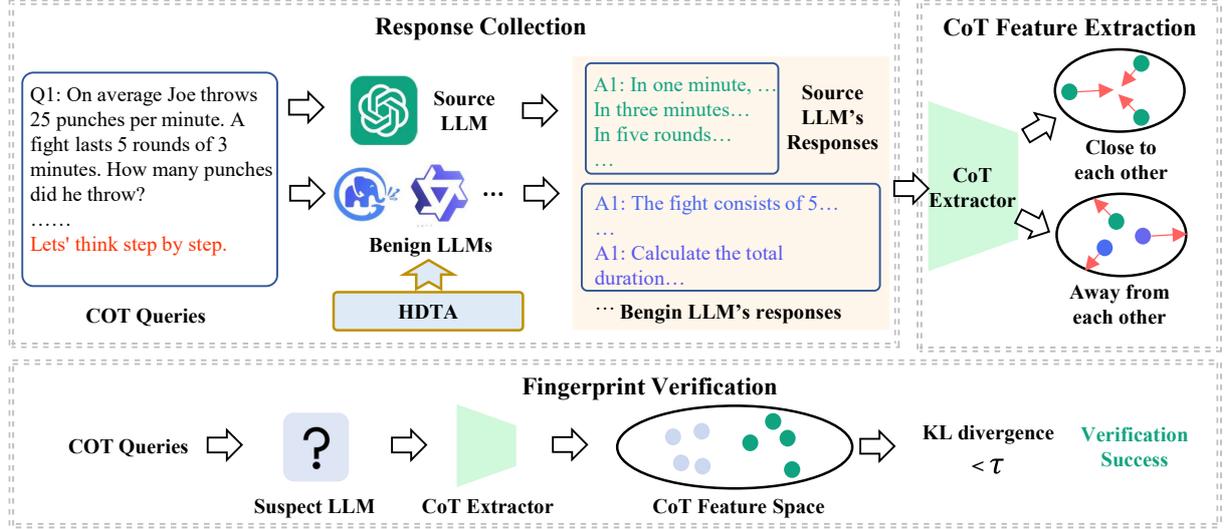


Figure 2: Framework of the proposed CoTSRF.

### 3.2 Designed Goals

The design of the LLM fingerprinting method should satisfy the following properties:

- 1) **Effectiveness**: the fingerprint should accurately identify the infringing LLM;
- 2) **Reliability**: false positives, where the fingerprint misidentifies a benign LLM released by other providers as an infringing LLM, should be minimized;
- 3) **Stealthiness**: the fingerprint queries should be normal prompts with coherent semantics to avoid being filtered out;
- 4) **Robustness**: the fingerprint remains consistent even if the LLM undergoes output perturbation and fine-tuning attacks.

## 4 Methodology

The flowchart of the proposed method is shown in Fig. 2. It consists of 1) a response collection module, which obtains the responses of the LLM by querying it using crafted CoT queries, where a High-Temperature Data Augmentation (HTDA) strategy is designed and used to make LLM generate diverse positive responses that vary in word space but follow the same logical reasoning pattern. It also introduces several benign LLMs to create negative responses with different logical reasoning patterns. 2) a CoT feature extraction module that adopts contrastive learning to train a CoT extractor to extract the logical reasoning path of the responses. 3) An fingerprint verification module,

which verifies the LLM infringement by comparing the KL divergence between the CoT features of the source and suspect LLMs against an empirical threshold. In what follows, we elaborate on each step in detail.

### 4.1 Response Collection

Response collection module involves two types of LLMs: the source LLM  $\mathcal{M}^S$  and a group of benign LLMs  $\{\mathcal{M}_1^B, \mathcal{M}_2^B, \dots, \mathcal{M}_K^B\}$ , where  $K$  is the total number of benign LLMs. This module begins by building a set of CoT queries  $Q = \{q_1, q_2, \dots, q_I\}$ , where  $I$  is the total number of queries.  $q_i$  ( $1 \leq i \leq I$ ) refers to the  $i$ -th CoT query and is composed of a reasoning question and a standard CoT prompt. After that, the response collection module collects diverse responses by feeding  $Q$  within the source and benign LLMs.

For  $\mathcal{M}^S$ , when fed with  $Q$ , it generates  $I$  different responses. Here, we design a High-Temperature Data Augmentation (HTDA) strategy to increase the diversity of the source LLM's outputs. Specifically, we set the temperature  $T$  of the last softmax layer in  $\mathcal{M}^S$  to a high value and let  $\mathcal{M}^S$  generate  $J > 3$  different responses for each  $q_i$  in  $Q$ . These responses differ in word space but follow the same logical reasoning pattern of  $\mathcal{M}^S$ . After querying  $\mathcal{M}^S$ , we obtain a total of  $I \times J$  responses, namely  $R^S = \{r_{1,1}^S, r_{i,j}^S, \dots, r_{I,J}^S\}$ .

By the same token, for the  $k$ -th benign LLM  $\mathcal{M}_k^B$ , we obtain a responses set  $R_k^B = \{r_{1,1,k}^B \dots r_{i,j,k}^B \dots r_{I,J,k}^B\}$ , with  $r_{i,j,k}^B$  being the  $j$ -th times response of  $\mathcal{M}_k^B$  for  $q_i$ . By querying all the benign LLMs, we obtain  $R^B =$

$\{R_1^B, R_2^B, \dots, R_K^B\}$  which consists of  $I \times J \times K$  responses.

## 4.2 CoT Feature Extraction

The goal of the CoT feature extraction module is to train a CoT extractor to accurately extract the CoT features from the responses. For reliable fingerprint verification, the extracted CoT features should be similar for the two responses that are both derived from the source LLM but be different when one of them is generated by the benign LLM. To achieve this, contrastive learning with a triplet loss function is adopted to train the CoT extractor.

For the  $i$ -th query  $q_i$  in  $Q$ , we consider the two of the responses in  $r_i^s$  as positive pairs, and treat the response from  $r_i^s$  and that from  $r_i^b$  as negative pairs. Denote the CoT extractor parameterized by  $\theta$  as  $E_\theta(\cdot)$ , we optimize  $\theta$  by minimize the following triplet margin loss:

$$\mathcal{L} = \sum_{q_i \in Q} \max(0, \|z_i^a - z_i^p\| - \|z_i^a - z_i^n\| + \delta), \quad (1)$$

where  $z_i^a = E_\theta(r_{i,j_1}^s)$  is set as anchor CoT feature, and is extracted from the response by the  $\mathcal{M}^S$  for  $q_i$  in  $j_1$ -th time;  $z_i^p = E_\theta(r_{i,j_2}^s)$  is set as a positive CoT feature, which is extracted from the response by the  $\mathcal{M}^S$  for  $q_i$  in  $j_2$  ( $j_2 \neq j_1$ )-th time;  $z_i^n = E_\theta(r_{i,j,k}^b)$  is set as a negative CoT feature, which is extracted from one of the responses by the benign LLMs for  $q_i$ .  $\|\cdot\|$  denotes the Euclidean distance and  $\delta$  is the margin enforcing a minimum distance between positive and negative pairs.

## 4.3 Fingerprint Verification

Fingerprint verification module uses CoT queries  $Q = \{q_1, q_2, \dots, q_I\}$  and source LLM's responses  $R^S = \{r_{1,1}^s, \dots, r_{i,j}^s, \dots, r_{I,J}^s\}$  to verify the LLM infringement. Let's denote the suspect LLM to be verified as  $\mathcal{M}^V$ . Fed with  $Q$ ,  $\mathcal{M}^V$  return a set of outputs  $R^V = \{r_1^v, \dots, r_2^v, \dots, r_I^v\}$ , with  $r_i^v$  denotes the response of  $\mathcal{M}^V$  for  $q_i$ . During the response collection, for  $q_i$ ,  $\mathcal{M}^S$  have generated  $J$  different responses  $\{r_{i,1}^s, r_{i,2}^s, \dots, r_{i,J}^s\}$ , from which, we select the first three responses for fingerprint verification.

Specifically, using trained  $E_\theta(\cdot)$ , we extract the CoT feature vectors of  $r_{i,1}^s$  and  $r_{i,2}^s$  and measure their distance by

$$d_i^s = \|E_\theta(r_{i,1}^s) - E_\theta(r_{i,2}^s)\|. \quad (2)$$

We then extract the CoT feature vectors of  $r_{i,3}^s$  and  $r_i^v$  and measure their distance by

$$d_i^v = \|E_\theta(r_{i,3}^s) - E_\theta(r_i^v)\|. \quad (3)$$

Using all the CoT queries  $Q$ , we obtain  $D^S = \{d_1^s, d_2^s, \dots, d_I^s\}$  for source LLM  $\mathcal{M}^S$  and  $D^V = \{d_1^v, d_2^v, \dots, d_I^v\}$  and for verified LLM  $\mathcal{M}^V$ . After that, we calculate the distance between the  $D^S$  and  $D^V$  using the KL divergence with kernel density estimation, as follows:

$$\text{KL}(D^S \| D^V) = \sum_{x \in \mathcal{X}} D^S(x) \log \frac{D^S(x)}{D^V(x)}, \quad (4)$$

where  $D^S(x)$  and  $D^V(x)$  are probability densities estimated via Gaussian Kernel Density Estimation (KDE) with Silverman's bandwidth rule. The evaluation grid  $\mathcal{X}$  spans  $[\min(\mathbf{x}_S, \mathbf{x}_T), \max(\mathbf{x}_S, \mathbf{x}_T)]$  using 1,000 equally spaced points, where  $\mathbf{x}_S$  and  $\mathbf{x}_T$  denote observed values from each distribution. Numerical stability is ensured by flooring probabilities at  $\epsilon = 10^{-10}$ .

Finally, we compare  $\text{KL}(D^S \| D^V)$  with an empirical threshold  $\tau$  and identify the verified suspect LLM  $\mathcal{M}^V$  as an infringing LLM if  $\text{KL}(D^S \| D^V) \geq \tau$ , and as a benign LLM otherwise.

## 5 Experiments

### 5.1 Experimental Settings

Three popular open-source LLMs are used in our implementation, including llama-2-7b-chat-hf, vicuna-7b-v1.3, and guanaco-7b-HF. When one of them is used as the source LLM, the remaining two serve as benign LLMs. The reasoning questions are derived from the dataset in Wang et al. (2023). The standard CoT prompt used to build our CoT queries is: "Let's first understand the problem and devise a plan to solve it. Then, let's carry out the plan and solve the problem step by step." (Wang et al., 2023). The number of CoT queries is either 50 or 100. In the HTDA strategy, the temperature  $T$  is set to 1.5, and  $J$  is set to 4. The Longformer encoder from (Beltagy et al., 2020) is adopted as our CoT extractor. For each source LLM, we train a unique CoT extractor 3000 epochs using the Adam (Diederik, 2014) optimizer. The hyperparameter  $\delta$  of the Triplet Margin Loss is set to 5.

For fingerprint verification, the thresholds  $\tau$  for vicuna-7b-v1.3, llama-2-7b-chat-hf,

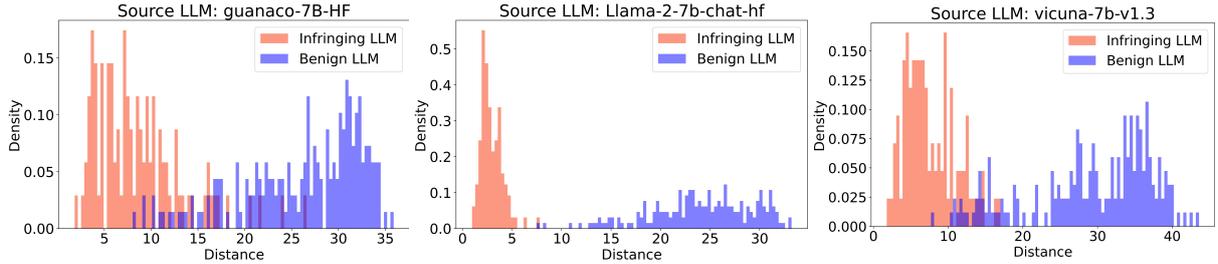


Figure 3: Distribution of the Euclidean distance between the CoT features of the source LLM and those of the infringing/benign LLM.

Table 1: Effectiveness and reliability comparison.

Source LLM	$I$	CoTSRF					TRAP	
		SI-KL	SB-KL	$\tau$	TPR	FPR	TPR	FPR
guanaco-7b-HF	50	1.5	303.6	8.0	100.0%	0.0%	-	-
	100	0.8	298.6		100.0%	0.0%	100.0%	0.0%
Llama-2-7b-chat-hf	50	8.9	400.6	85.0	99.5%	0.0%	-	-
	100	5.9	425.8		100.0%	0.0%	95.2%	0.2%
vicuna-7b-v1.3	50	1.7	151.2	18.0	98.8%	0.0%	-	-
	100	1.1	153.5		100.0%	0.0%	97.0%	0.0%

and guanaco-7B-HF are empirically set to 8.0, 85.0, and 18.0, respectively. Moreover, we introduce internlm2\_5-7b-chat and llama3.1-8b-instruct as unseen benign LLMs to test the reliability of the proposed CoTSRF by verifying its effectiveness in distinguishing those benign LLMs that were not used to train the CoT extractor. TRAP (Gubri et al., 2024) is used as the benchmark method, which, to the best of our knowledge, is the state-of-the-art LLM fingerprinting method that supports fingerprint verification under a black-box setting. For a fair comparison, we run TRAP with its default settings.

## 5.2 Effectiveness

Effectiveness requires that the fingerprinting method accurately identifies the infringing LLM. Table 1 presents the verification results of the proposed CoTSRF and the comparison method. In the table,  $I$  represents the number of queries, SI-KL/SB-KL represent the KL divergences between the CoT features of the source LLM and the infringing/benign LLM, respectively. The True Positive Rate (TPR) measures the accuracy of correctly identifying an infringing LLM, while the False Positive Rate (FPR) indicates the rate of mistakenly identifying a benign LLM as infringing. To obtain the TPR and FPR for Our CoTSRF, we conduct fingerprint verification 100 times for each source LLM. The TRAP’s results are duplicated from its

original paper, with “-” indicating no data.

We can see that the difference in values between SI-KL and SB-KL is significant, indicating that the distance between the CoT features of benign LLMs and the source LLM is much greater than that between infringing LLMs and the source LLM. This serves as the foundation of our method’s effectiveness. For TPR and FPR, our CoTSRF achieves the best results in all cases. Specifically, when  $I$  is 100, we provide a 100.00% TPR across all source LLMs. In contrast, TRAP exhibits inferior performance and only provides 95.2% TPR when taking Llama-2-7b-chat-hf as the source LLM. This highlights the effectiveness of our CoTSRF.

We further visualize the distribution of the Euclidean distance between the CoT features of different types of LLMs using a histogram, as depicted in Fig. 3. We can observe that, in all cases, the distance between the source LLM and the infringing LLM is significantly lower than that between the source LLM and a benign LLM. This demonstrates that our CoT extractor effectively captures the differences in CoT features between different types of LLMs.

## 5.3 Reliability

The FPR results in Table 1 are 0.00% in all cases, indicating that the proposed CoTSRF can effectively identify the benign LLMs used for training the CoT extractor  $E_{\theta}(\cdot)$ . To

Table 2: Reliability of CoTSRF in identifying unseen benign LLMs.

Source LLM	$I$	Unseen benign LLMs	SI-KL	SB-KL	TPR	FPR
guanaco-7b-HF	50	internLM2.5-7b	1.2	19.4	100.0%	3.0%
		llama3-8-instruct	1.3	30.6	100.0%	0.0%
	100	internLM2.5-7b	0.9	30.1	99.5%	0.0%
		llama3-8-instruct	0.9	32.6	100.0%	0.0%
Llama-2-7b-chat-hf	50	internLM2.5-7b	8.7	318.8	100.0%	0.0%
		llama3-8-instruct	8.7	333.9	100.0%	0.0%
	100	internLM2.5-7b	5.7	329.3	100.0%	0.0%
		llama3-8-instruct	5.7	331.9	100.0%	0.0%
vicuna-7b-v1.3	50	internLM2.5-7b	2.1	39.6	100.0%	0.0%
		llama3-8-instruct	2.3	51.2	97.5%	0.0%
	100	internLM2.5-7b	1.2	37.6	100.0%	0.0%
		llama3-8-instruct	1.3	48.2	99.5%	0.0%

Table 3: Perplexity of the fingerprint queries of different methods.

Methods	Avg	Min	Max
Trap	16467.4	1989.5	71995.5
CoTSRF	28.4	10.6	61.3
Normal	75.7	10.2	1204.5

further evaluate the reliability of our method on unseen benign LLMs, we conduct additional tests using internlm2\_5-7b-chat and llama3.1-8b-instruct, which were not included in the training pipeline of  $E_{\theta}(\cdot)$ . The results are presented in Table 2. We can observe that the SB-KL values remain significantly higher than the SI-KL values, demonstrating a substantial difference between the CoT features of unseen benign LLMs and the source LLM. Moreover, when a large number of CoT queries  $I$  is used, the FPR results remain at 0.00%. This indicates that the proposed method can generalize to distinguish unseen benign LLMs from the source LLM, further emphasizing its reliability.

#### 5.4 Stealthiness

In this section, we evaluate the stealthiness of TRAP and the proposed CoTSRF. Specifically, we use perplexity (Gonen et al., 2022) to measure the semantic coherence of the fingerprint queries generated by different methods. A higher perplexity score indicates lower semantic coherence and a higher probability of being detected and filtered out by a malicious developer. In our implementation, the standard GPT-2 language model (Radford et al., 2019) is used to calculate the perplexity score. The number of queries is set to 100.

Table 3 presents the perplexity scores of the fin-

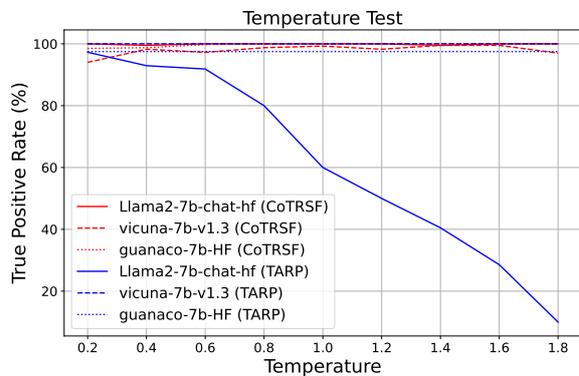


Figure 4: TPR of CoTSRF and TRAP under different temperature settings (from 0.2 to 1.8).

gerprint queries generated by different methods, with the last row showing the perplexity of normal queries (i.e., reasoning questions without a CoT prompt or adversarial prefix). We can observe that the average perplexity of TRAP’s queries reaches 16467.4, which is significantly greater than that of normal queries (i.e., 75.7). In contrast, the perplexity of our CoT-based queries is lower than that of normal queries in terms of the average, maximum, and minimum values. This indicates that the added CoT prompt not only preserves the original logical structure of the queries but also enhances their semantic coherence. These findings demonstrate the stealthiness of our proposed method.

In real applications, a malicious developer could implement a perplexity-based filtering module in front of the infringing model to block fingerprint verification by filtering out queries with high perplexity scores. To avoid negatively impacting the performance of the infringing model, they could set the perplexity threshold to the maximum perplexity observed in normal queries (i.e., 1204.5). Under

Table 4: Robustness of CoTSRF against the output perturbation attack in identifying unseen benign LLMs (TPR/FPR).

Source LLM	Temperature $T$ (0.2 - 1.8)								
	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8
Llama-2-7b-chat-hf	99.5/0.0	99.5/0.0	100.0/0.0	100.0/0.0	100.0/0.0	99.5/0.0	100.0/0.0	100.0/0.0	100.0/0.0
vicuna-7b-v1.3	95.5/9.5	94.0/2.0	91.0/5.0	96.5/1.0	94.0/2.0	97.5/0.0	99.5/0.0	99.0/0.0	98.0/0.0
guanaco-7b-HF	98.0/0.0	100.0/0.0	99.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/1.0	100.0/14.5	99.5/85.5

Table 5: Robustness against finetuning attacks; (CoT Query number  $I = 50$ ).

Step	LoRA Rank: 8		LORA Rank: 16	
	TPR	FPR	TPR	FPR
1600	100.0%	0.0%	100.0%	0.0%
3200	100.0%	0.0%	100.0%	0.0%
4800	100.0%	0.0%	100.0%	0.0%
6400	100.0%	0.0%	100.0%	0.0%

such conditions, the TRAP method would fail because all of its fingerprint queries would be filtered out. In contrast, all our CoT-based queries can successfully bypass the perplexity-based detection and complete the LLM infringement verification.

## 5.5 Robustness

### 5.5.1 Robustness Against Output Perturbation Attack

After downloading the source LLM, the malicious developer may perturb the LLM’s output before returning it to its users to prevent fingerprint verification. Here, we simulate such an attack by adjusting the temperature coefficient  $T$  of the source LLM’s last Softmax layer. The values of  $T$  range from 0.2 to 1.8, where a higher  $T$  results in more diverse and random outputs.

Fig. 4 shows the TPR of CoTSRF and TRAP across three LLMs. We can see that CoTSRF consistently achieves TPRs above 94% across all temperature settings, demonstrating substantial robustness against output perturbation. In contrast, TRAP suffers significant performance degradation at higher temperature settings (i.e.,  $T \geq 1.0$ ), with TPR dropping below 20% when  $T$  is 1.8. This highlights CoTSRF’s ability to effectively counter the output perturbation attacks.

We further test CoTSRF’s robustness against output perturbation attack in identifying unseen benign LLMs, including internlm2\_5-7b-chat and llama3.1-8b-instruct, under varying  $T$  and show the results in Table 4. We can see that CoTSRF maintains strong detection performance across moderate temperatures ( $T \in [0.2, 1.4]$ ), achieving

TPRs above 95% while keeping FPRs below 2%. At  $T = 1.8$ , where the output becomes excessively random, CoTSRF’s performance begins to degrade for some LLMs (e.g., guanaco-7b-HF). We would like to mention that such extreme temperatures are rare in practical applications, as they may significantly affect the LLM’s performance.

### 5.5.2 Robustness against Fine-Tuning Attack

The malicious developer may also fine-tune the infringing LLM to erase its fingerprint. To verify the robustness of the proposed CoTSRF against such an attack, we fine-tune Llama-2-7b-chat-hf on the dataset ‘timdettmers/openassistant-guanaco’ (Köpf et al., 2024) using Low-Rank Adaptation (LoRA) technology (Hu et al., 2021) on the xTuner framework (Contributors, 2023). The rank of LoRA is set to 8 or 16. The learning rate is set to  $1 \times e^{-5}$ . Table 5 gives CoTSRF’s detection performance across different fine-tuning step. As can be seen, even with a high LoRA rank (i.e., 16) and long fine-tuning step (i.e., 4800), our CoTSRF maintains a TPR of 100.00% and an FPR of 0%, indicating that CoTSRF is robust to the fine-tuning attack.

## 6 Conclusion

In this paper, we propose a novel LLM fingerprinting method, namely CoTSRF, to identify LLM infringement in a black-box access setting. We take CoT as LLMs’ fingerprint, which allows stealthy and robust fingerprint verification. Specifically, we first collect the responses of the source LLM by querying it using crafted CoT queries. During this, a High-Temperature Data Augmentation (HTDA) strategy is proposed to boost the diversity of the responses. We then employ a contrastive learning framework with triplet margin loss to train a CoT extractor for accurate CoT extraction. Various experiments demonstrate the advantages of our method for verifying LLM infringement, achieving satisfactory performance in terms of effectiveness, reliability, stealthiness, and robustness.

## 7 Limitation

While our proposed CoTSRF method demonstrates strong performance in fingerprinting and identifying a wide range of LLMs, it is not without limitations. These limitations primarily stem from highly contrived corner cases and the need for broader validation across diverse model scales and architectures. Below, we discuss these challenges in detail.

**1. Challenges in Highly Contrived Corner Cases** The method faces difficulties in scenarios where two LLMs exhibit extreme architectural and training homogeneity. For instance, if two entities independently train models using identical open-source architectures (e.g., LLaMA) with precisely replicated training protocols—same data sources, identical preprocessing—the resulting models’ reasoning pathways could become nearly indistinguishable. This would reduce the effectiveness of CoT-based fingerprinting, leading to a decline in the reliability of our method, as it would struggle to differentiate between the source model and the benign model. However, such scenarios are exceptionally rare in practice due to the low probability of two entities independently replicating the exact same training pipeline.

**2. Need for Broader Validation** While our experiments demonstrate promising results on several models under 10B parameters, the generalizability of CoTSRF requires further validation across a broader range of LLM architectures and scales. The rapid evolution of LLM architectures necessitates testing on larger and more diverse model families. Future work should extend evaluations to larger-scale models (e.g., 70B+ parameters) and emerging paradigms such as mixture-of-experts and multimodal architectures.

## Acknowledgements

## References

- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Qi Chen and Dexi Liu. 2023. Dynamic strategy chain: Dynamic zero-shot cot for long mental health support generation. *arXiv preprint arXiv:2308.10444*.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing

gpt-4 with 90%\* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2(3):6.

- XTuner Contributors. 2023. Xtuner: A toolkit for efficiently fine-tuning llm.
- Creative Commons. 2024. Noncommercial licenses. [https://wiki.creativecommons.org/wiki/NonCommercial\\_interpretation](https://wiki.creativecommons.org/wiki/NonCommercial_interpretation). [Online; accessed 2024-08-22].
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.
- P Kingma Diederik. 2014. Adam: A method for stochastic optimization. (*No Title*).
- Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2024. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36.
- Free Software Foundation. 2024. Gnu general public license. <https://www.gnu.org/licenses/gpl-faq.html>. [Online; accessed 2024-08-22].
- Hila Gonen, Srinu Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2022. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*.
- Jiyang Guan, Jian Liang, and Ran He. 2022. Are you stealing my model? sample correlation for fingerprinting deep neural networks. *Advances in Neural Information Processing Systems*, 35:36571–36584.
- Martin Gubri, Dennis Ulmer, Hwaran Lee, Sangdoon Yun, and Seong Joon Oh. 2024. Trap: Targeted random adversarial prompt honeypot for black-box identification. *arXiv preprint arXiv:2402.12991*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Xijie Huang, Li Lina Zhang, Kwang-Ting Cheng, and Mao Yang. 2023. Boosting llm reasoning: Push the limits of few-shot learning with reinforced in-context pruning. *arXiv preprint arXiv:2312.08901*.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. 2024. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36.

- Peixuan Li, Pengzhou Cheng, Fangqi Li, Wei Du, Haodong Zhao, and Gongshen Liu. 2023. Plmmark: a secure and robust black-box watermarking framework for pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14991–14999.
- Wenhao Liu, Tianxing Bu, Erchen Yu, Dailin Li, Ding Ai, Zhenyi Lu, and Haoran Luo. 2024a. Optimizing few-shot learning: From static to adaptive in qwen2-7b. In *Amazon KDD Cup 2024 Workshop*.
- Xiao Liu, Zirui Wu, Xueqing Wu, Pan Lu, Kai-Wei Chang, and Yansong Feng. 2024b. Are llms capable of data-based statistical and causal reasoning? benchmarking advanced quantitative reasoning with data. *arXiv preprint arXiv:2402.17644*.
- Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. 2019. Deep neural network fingerprinting by conferrable adversarial examples. *arXiv preprint arXiv:1912.00888*.
- Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. 2023. Are you copying my model? protecting the copyright of large language models for eaaS via backdoor watermark. *arXiv preprint arXiv:2305.10036*.
- Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. 2022. Fingerprinting deep neural networks globally via universal adversarial perturbations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13430–13439.
- Yuhui Quan, Huan Teng, Ruotao Xu, Jun Huang, and Hui Ji. 2023. Fingerprinting deep image restoration models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13285–13295.
- 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.
- Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.
- Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024a. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.
- Xiyue Wang, Junhan Zhao, Eliana Marostica, Wei Yuan, Jietian Jin, Jiayu Zhang, Ruijiang Li, Hongping Tang, Kanran Wang, Yu Li, et al. 2024b. A pathology foundation model for cancer diagnosis and prognosis prediction. *Nature*, 634(8035):970–978.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. 2024. Agentless: Demystifying llm-based software engineering agents. *arXiv preprint arXiv:2407.01489*.
- Jiashu Xu, Fei Wang, Mingyu Derek Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. 2024. Instructional fingerprinting of large language models. *arXiv preprint arXiv:2401.12255*.
- Xiaosong Yuan, Chen Shen, Shaotian Yan, Xiaofeng Zhang, Liang Xie, Wenxiao Wang, Renchu Guan, Ying Wang, and Jieping Ye. 2024. Instance-adaptive zero-shot chain-of-thought prompting. *arXiv preprint arXiv:2409.20441*.
- Boyi Zeng, Lizheng Wang, Yuncong Hu, Yi Xu, Chenghu Zhou, Xinbing Wang, Yu Yu, and Zhouhan Lin. 2023. Huref: Human-readable fingerprint for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Yue Zheng, Si Wang, and Chip-Hong Chang. A dnn fingerprint for non-repudiable model ownership identification and piracy detection. *IEEE Transactions on Information Forensics and Security*, 17.
- Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

## A Example Appendix

This is a section in the appendix.