Boosting the Uniqueness of Neural Networks Fingerprints with Informative Triggers

Zhuomeng Zhang
Shanghai Jiao Tong University
zzmsmm@sjtu.edu.cn

Fangqi Li
Shanghai Jiao Tong University
solour_lfq@sjtu.edu.cn

Hanyi Wang
Shanghai Jiao Tong University
why_820@sjtu.edu.cn

Shi-Lin Wang*
Shanghai Jiao Tong University
wsl@sjtu.edu.cn

Abstract

One prerequisite for secure and reliable artificial intelligence services is tracing the copyright of backend deep neural networks. In the black-box scenario, the copyright of deep neural networks can be traced by their fingerprints, i.e., their outputs on a series of fingerprinting triggers. The performance of deep neural network fingerprints is usually evaluated in robustness, leaving the accuracy of copyright tracing among a large number of models with a limited number of triggers intractable. This fact challenges the application of deep neural network fingerprints as the cost of queries is becoming a bottleneck. This paper studies the performance of deep neural network fingerprints from an information theoretical perspective. With this new perspective, we demonstrate that copyright tracing can be more accurate and efficient by using triggers with the largest marginal mutual information. Extensive experiments demonstrate that our method can be seamlessly incorporated into any existing fingerprinting scheme to facilitate the copyright tracing of deep neural networks.

1 Introduction

Recent progress in deep neural network (DNN) models is raising privacy and ethics concerns since they might facilitate the propagation of fake information with negative social impacts [1]. To ensure that AI serves people properly, it is necessary to trace the copyright of DNN models and attribute the misuse of models to specific users. The majority of existing studies concentrate on copyright tracing in the black-box setting where the copyright verifier interferes with the suspicious DNN model as a black box. Two mainstream methods are DNN watermarking [2–8] and DNN fingerprinting [9–16].

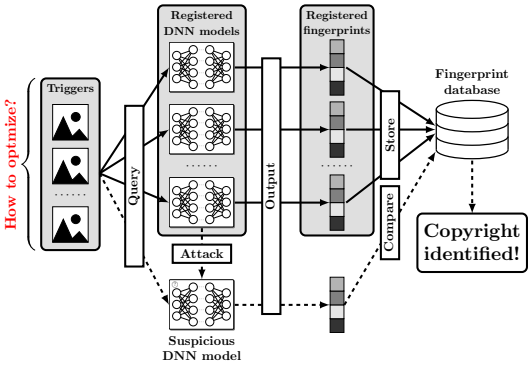


Figure 1: The framework of a fingerprint-based DNN copyright tracing system.

DNN watermarking schemes use a series of inputs as watermark triggers and tune DNN models so that their outputs on triggers are differentiated as pre-defined. Therefore, they appeal only to copyright

*Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

verifiers who can tune the DNN models to be protected or owners who distribute a model to multiple clients. Watermarking schemes inevitably introduce deline in the performance of watermarked DNN models, which is unacceptable in industrial fields.

In contrast, DNN fingerprinting schemes produce a series of fingerprinting triggers. The outputs of a suspicious DNN model on the triggers constitute its fingerprint, with which its identity is recognized. An illustration is given in Fig. 1.

The reliability of DNN fingerprinting schemes has usually been interpreted as their robustness, i.e., the fingerprint of a DNN model should remain invariable under adversarial modifications. Despite established results on robustness, the uniqueness of DNN fingerprinting is still under-explored. This aspect is crucial for industrial copyright tracing especially when the number of model to be traced is large while the expense of retrieving a fingerprint grows in the number of triggers and could become a bottleneck. It is hard to evaluate how every trigger contributes differently to copyright tracing, and under which circumstances does each DNN model have a unique fingerprint. Consequently, it is difficult to optimize the collection of triggers with a fixed cardinality. To address the above challenges, this paper proposes a general method to improve the efficiency of fingerprint-based DNN copyright tracing. The contributions are concluded as follows:

- We adopt an information theoretical perspective to measure the contribution to copyright tracing of each fingerprinting trigger by its conditional mutual information.
- We boost the copyright tracing performance by greedily optimizing the collection of triggers and validate this method through extensive experiments.
- We derive the first necessary condition for the number of fingerprinting triggers to ensure copyright tracing.

2 Preliminaries

2.1 DNN Fingerprint

Notations used in this paper are listed in Table 1. We focus on classifiers that map the input space $\mathcal{X}$ to $\mathcal{Y} = \{1, 2, \cdots, C\}$. Altogether P classifiers $\mathbf{F} = \{f_p\}_{p=1}^{P}$ require copyright tracing. A fingerprinting scheme draws N independent triggers $\mathbf{T} = \{\mathbf{t}_n\}_{n=1}^{N}$ from a distribution $\mathcal{T}$. Choices of $\mathcal{T}$ include random noises [9], outliers [3, 10], normal samples that are close to the centers of each class [15, 16], adversarial samples that are close to the decision boundaries [11, 13], etc.

Table 1: Frequently used notations in this paper.

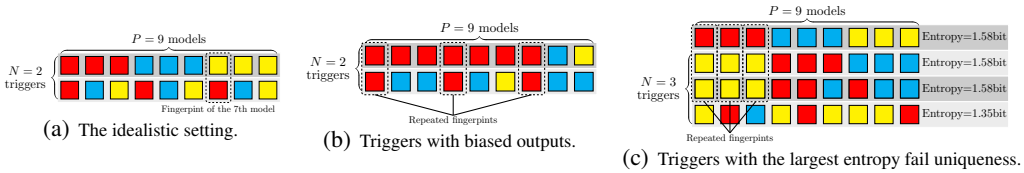
Symbol	Meaning
$\mathcal{X}$	Input space of DNN models.
C	Number of classes.
$\mathcal{Y}$	Output space of DNN models.
P	Number of registered DNN models.
$\mathbf{F}$	Registered DNN models, $\mathbf{F} = \{f_p\}_{p=1}^{P}$.
N	Number of fingerprinting triggers.
$\hat{N}$	Number of greedily selected triggers.
$\mathcal{T}$	Distribution of triggers.
$\mathbf{T}$	Fingerprinting triggers, $\mathbf{T} = \{\mathbf{t}_n\}_{n=1}^{N}$.
ϵ	Verifier's tolerance on the adversary's attack.
$\mathbf{A}_\epsilon$	Verifier's threat model.
$\mathcal{F}$	A randomly selected model from $\mathbf{F}$.
$\mathcal{A}$	A randomly selected attack from $\mathbf{A}_\epsilon$.
ϕ_n	The randomly selected model's prediction on $\mathbf{t}_n$.
u	Uniqueness rate
$I_\epsilon\left(\mathbf{t}_n \middle\| \mathbf{t}_{1:(n-1)}\right)$	Conditional mutual information of $\mathbf{t}_n$.

The fingerprint of the p-th model is its outputs on triggers: $(f_p(\mathbf{t}_1), \cdots, f_p(\mathbf{t}_N))$. The fingerprints of all models are computed and registered in a database. Upon locating a suspicious model, the verifier computes its fingerprint and compares it with records in the database. If there exists a registered model whose fingerprint is close to the suspicious model's, then a copyright issue is reported. Otherwise, the suspicious model is registered as a new model. The distance between two fingerprints is measured by the number of triggers where two models return different outputs.

2.2 Robustness & Uniqueness

The performance of a DNN fingerprinting scheme is inclusively reflected in its *robustness* and *uniqueness*.

The robustness of a DNN fingerprinting scheme is characterized by the difference between registered models' fingerprints before and after adversarial modifications. Formally, the verifier assumes that the adversary has sacrificed the victim model's performance for a probability at most ϵ (it is impossible to



(a) The idealistic setting. (b) Triggers with biased outputs. (c) Triggers with the largest entropy fail uniqueness.

Figure 2: An illustration of the difficulties in deciding the optimal fingerprinting triggers $P = 9$ and $C = 3$. Each column denotes a model. Each row denotes a fingerprinting trigger. Each color denotes a classification label. (a) The idealistic case where $\log_C P = 2$ triggers produce unique fingerprints. (b) When the predictions on triggers are biased, two triggers are insufficient. (c) Choosing the first three triggers with the largest independent entropy fails the uniqueness property. The correct choice is the last three triggers.

trace the copyright of models whose functionality has been arbitrarily modified). The threat model is:

$$\mathbf{A}_\epsilon = \left\{ A : \forall f \in \mathbf{F},\ \Pr_{x \leftarrow \mathcal{X}}(f(x) \neq f^A(x)) \leq \epsilon \right\}. \tag{1}$$

where f^A denotes a model f undertaken an attack A. In general, the adversary's attack always has a measurable influence on the model's performance so $\mathbf{A}_0 = \emptyset$. The verifier should assume a non-trivial adversary and focus on the maximal damage that the adversary might cause [17], so the robustness of a fingerprinting trigger distribution $\mathcal{T}$ is quantified by:

$$\delta_{\mathcal{T}}(\epsilon) = \max_{f \in \mathbf{F}, A \in \mathbf{A}_\epsilon} \left\{ \Pr_{\mathbf{t} \leftarrow \mathcal{T}} \left(f(\mathbf{t}) \neq f^A(\mathbf{t}) \right) \right\}. \tag{2}$$

A fingerprinting trigger distribution with a large $\delta_{\mathcal{T}}(\epsilon)$ yields weak fingerprints, since the adversary can obfuscate the fingerprints with little decline in functionality.

Empirically, the robustness of a fingerprinting trigger distribution is estimated with a finite collection of adversarial modifications including fine-tuning [18, 19], neuron pruning [20], distillation [21], etc. Although it is hard to directly foster the robustness by modifying the trigger distribution, some distributions are reported to be more robust under certain attacks [12, 15, 16].

On the other hand, the fingerprint of each model should be unique so the false positive rate is negligible. The low transferability of Characteristic Examples [22] primarily addresses the fingerprint of individual models, whereas our study focuses on the uniqueness of fingerprints for distinguishing between different models in large-scale deployment scenarios. This property is reflected in the percentage of models whose fingerprints remain unique under any attack. Let $\mathbf{F}_\epsilon(\mathbf{T}) = \{ f \in \mathbf{F} : \exists A \in \mathbf{A}_\epsilon, \exists f' \in \mathbf{F} \setminus \{f\}, \forall \mathbf{t} \in \mathbf{T}, f(\mathbf{t}) = f'(\mathbf{t}) \}$, the uniqueness rate u can be defined as:

$$u = 1 - |\mathbf{F}_\epsilon(\mathbf{T})| / |\mathbf{F}|. \tag{3}$$

Unfortunately, this metric u can hardly be optimized as a function in $\mathbf{T}$. In existing literature, it is generally assumed that the models' outputs on a fingerprinting trigger are randomly distributed as shown in Fig. 2(a), so $\log_C P$ triggers are sufficient and the probability that two arbitrary models have the same fingerprint declines exponentially in N [9]. So any fingerprinting scheme is expected to have $u \to 1$ when N is large.

2.3 Challenges

The assumptions behind the uniqueness of DNN fingerprinting schemes are challenged by the following facts.

(I): The number of triggers is not arbitrarily large. The number of triggers determines the cost of copyright tracing and could become a bottleneck in large-scale or expensive service or when querying the API of suspicious DNN models is expensive. It is necessary to evaluate the performance of DNN fingerprints when the number of triggers is limited.

(II): The value of each trigger has been overestimated. DNN models' outputs on a trigger might not be uniformly distributed, as shown in Fig. 2(b). Moreover, the triggers are not independent of each other. Using triggers with independently the largest entropy might turn out to be misleading as

shown in Fig. 2(c). The value of each trigger is determined by the conditional entropy it contains w.r.t. queried triggers.

(III): The influence of adversarial modifications is unclear. When the adversary modifies the victim model's fingerprint, the information value of each trigger might change. The optimal number of triggers varies with the assumptions of the adversary. So far, the relationship between uniqueness and robustness has not been established.

As a result, the volume of information that a limited number of triggers can provide in the adversarial environment is an intractable bottleneck of the copyright tracing system, leaving the uniqueness property as a risk.

3 Method

3.1 Information in DNN Copyright Tracing

We consider the copyright tracing of DNN models as a communication channel. From the verifier's view, the information source is the identity of the suspicious model, which is represented by a random variable $\mathcal{F}$ whose domain is $\mathbf{F}$. Without loss of generality, we assume that the suspicious model $\mathcal{F}$ is equally likely to be any registered model, so $\mathcal{F}$ contains $\log_2 P$ bits of information.

The adversarial modifications, represented by a random variable $\mathcal{A}$, are noises in this channel. Without prior knowledge, the attack is randomly chosen from $\mathbf{A}_\epsilon$, where ϵ reflects the verifier's tolerance. We further assume that the attack is independent of the triggers and the victim model.

The output from the suspicious model $\mathcal{F}$ on the n-th trigger, denoted by ϕ_n, is another random variable, so is the fingerprint of the suspicious model $\Phi = (\phi_1, \phi_2, \cdots, \phi_N)$. We are interested in how much information the fingerprint reveals about the suspicious model's identity, which is inclusively quantified by the mutual information $I(\Phi; \mathcal{F})$. In fact, the volume of information to secure a uniqueness rate u is at least $-u\log_2 u - (1-u)\log_2(1-u) + \log_2 uP$, so an upper bound of u is

Algorithm 1 Computing $I_0\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right)$.

Input: Registered models $\mathbf{F}$, triggers $\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_n$
Output: $I_0\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right)$

$\mathcal{M} = \emptyset$, $h = 0$
for $i = 1$ to P **do**
  flag = False
  for $\mathbf{M}$ in $\mathcal{M}$ **do**
    if $\exists f \in \mathbf{F}\backslash\{f_i\}, \forall j = 1, \cdots, n-1, f(\mathbf{t}_j) = f_i(\mathbf{t}_j)$ **then**
      $\mathbf{M} = \mathbf{M} \cup \{f_i\}$; flag=True; break
    end if
  end for
  if flag=False **then**
    $\mathcal{M} = \mathcal{M} \cup \{f_i\}$
  end if
end for
for $\mathbf{M}$ in $\mathcal{M}$ **do**
  for $c = 1$ to C **do**
    $u_c = 0$
    for f in $\mathbf{M}$ **do**
      if $f(\mathbf{t}_n) = c$ **then**
        $u_c = u_c + 1$
      end if
    end for
    $u_c = u_c/|\mathbf{M}|$; $h = h - \frac{|\mathbf{M}|}{P} \times u_c \log_2 u_c$
  end for
end for
Return h

$$u = \frac{2^{\log_2 uP}}{P} \leq \frac{2^{-u\log_2 u-(1-u)\log_2(1-u)+\log_2 uP}}{P} \leq \frac{2^{I(\Phi;\mathcal{F})}}{P}.$$

Therefore, a necessary condition for better uniqueness is using informative triggers. We begin with the conditional mutual information of the n-th trigger under threat model $\mathbf{A}_\epsilon$.

Definition 1. *Let the threat model be $\mathbf{A}_\epsilon$, denote the mutual information of $\mathbf{t}_n$ conditioned on queried triggers $\mathbf{t}_1, \cdots, \mathbf{t}_{n-1}$ by:*

$$I_\epsilon\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right) = H(\phi_n|\phi_1, \cdots, \phi_{n-1}) - H(\phi_n|\phi_1, \cdots, \phi_{n-1}, \mathcal{F}). \tag{4}$$

The mutual information of all triggers is decomposed as:

$$I(\Phi; \mathcal{F}) = H(\Phi) - H(\Phi|\mathcal{F}) = \sum_{n=1}^{N} I_\epsilon\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right). \tag{5}$$

In the vanilla setting, the verifier considers $\epsilon = 0$ and the modified model is always recognized as irrelevant from the original version. So $H(\phi_n|\phi_1, \cdots, \phi_{n-1}, \mathcal{F}) = 0$ and:

$$I_0\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right) = H(\phi_n|\phi_1, \cdots, \phi_{n-1}). \tag{6}$$

The complexity in computing the r.h.s. of Eq. (6) by iterating over all possible values of $(\phi_1, \cdots, \phi_{n-1}, \phi_n)$ is of order $\mathcal{O}(C^n)$. Instead, we resort to Algo. 1 where all registered models are segmented into disjoint sets according to their partial fingerprints on the first $(n-1)$ triggers. Fingerprints that never appear are ignored. The complexity is reduced to $\mathcal{O}(\max\left\{P^2 n, PC\right\})$ and is acceptable since $P \ll C^n$ usually holds in practice, especially when n is large.

In non-trivial cases where the tolerance $\epsilon > 0$, the verifier attributes modified models as copies of their original version and $H(\phi_n|\phi_1, \cdots, \phi_{n-1}, \mathcal{F})$ no longer always equals zero. A lower bound of the conditional mutual information of each trigger is given by the following theorem.

Theorem 1. *Let the threat model be $\mathbf{A}_\epsilon$ in Eq.* (1)*, then:*

$$I_\epsilon\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right) \geq I_0\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right) - h(\epsilon). \tag{7}$$

where $h(\epsilon) = -\delta_{\mathcal{T}}(\epsilon)\log_2\delta_{\mathcal{T}}(\epsilon) - (1-\delta_{\mathcal{T}}(\epsilon))\log_2(1-\delta_{\mathcal{T}}(\epsilon)) + \delta_{\mathcal{T}}(\epsilon)\log_2(C-1)$.

Conceptually, Theorem 1 is a variant of Fano's inequality. The complete proof is given in Appendix A.

3.2 Greedy Optimization of Triggers

Being equipped with the conditional mutual information of each trigger, we proceed to optimize the collection of triggers. Given the budget $\hat{N} \leq N$, the optimized collection of triggers $\hat{\mathbf{T}}$ with size $\hat{N}$ is selected by Algo. 2 where the trigger with the largest conditional entropy is iteratively included. Remarkably, $\hat{\mathbf{T}}$ is a permutation of a subset of $\mathbf{T}$.

Theorem 1 indicates that even for $\epsilon > 0$, the mutual information provided by the n-th trigger is lower bounded by I_0 minus a constant. So the cumulative mutual information provided by triggers selected according to Algo. 2 is lower bounded under arbitrary threat models.

Selecting a fixed number of triggers that provides the largest mutual information is essentially a combinatorial optimization problem that is NP-hard. We prove that our strategy selects triggers whose mutual information is lower bounded compared with the optimal triggers in theory. Let $\underline{I}_\epsilon\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right) = \max\left\{0, I_0\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right) - h(\epsilon)\right\}$.

Algorithm 2 Greedily selecting $\hat{N}$ informative triggers.

Input: Budget $\hat{N}$, triggers $\mathbf{T}$, registered models $\mathbf{F}$
Output: A collection of triggers $\hat{\mathbf{T}}$, $|\hat{\mathbf{T}}| = \hat{N}$.

$\hat{\mathbf{T}} = \emptyset$
for $n = 1$ to $\hat{N}$ **do**
  $m = 0$, $\mathbf{r} \in \mathbf{T} \setminus \hat{\mathbf{T}}$
  for $\mathbf{t} \in \mathbf{T} \setminus \hat{\mathbf{T}}$ **do**
    if $I_0(\mathbf{t}|\hat{\mathbf{T}}) > m$ **then**
      $m = I_0(\mathbf{t}|\hat{\mathbf{T}})$, $\mathbf{r} = \mathbf{t}$
    end if
  end for
  $\hat{\mathbf{T}} = \hat{\mathbf{T}} \cup \left\{\hat{\mathbf{t}}_n = \mathbf{r}\right\}$
end for
Return $\hat{\mathbf{T}}$

The mutual information provided by triggers $\mathbf{T}$ is no less than $g(\mathbf{T}) = \sum_{n=1}^{N} \underline{I}_\epsilon\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right)$, which turns out to be a non-negative, monotonically increasing, and submodular function in $\mathbf{T}$ [23–25]. Because for $\mathbf{V} \subset \mathbf{U} \subset \mathbf{T}$ and $\mathbf{t} \in \mathbf{T} \setminus \mathbf{U}$:

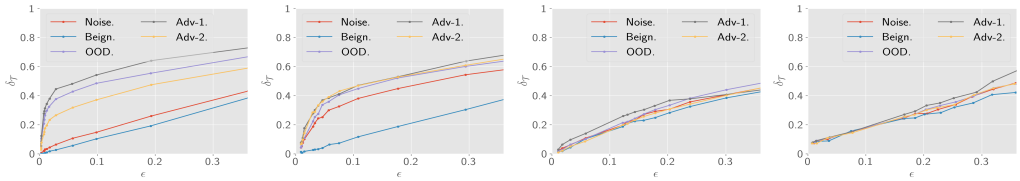
$$\begin{aligned} &g(\mathbf{U} \cup \{\mathbf{t}\}) - g(\mathbf{U}) = \max\left\{0, H\left(\mathbf{t}|\mathbf{U}\right) - h(\epsilon)\right\} \\ &\leq \max\left\{0, H\left(\mathbf{t}|\mathbf{V}\right) - h(\epsilon)\right\} = g(\mathbf{V} \cup \{\mathbf{t}\}) - g(\mathbf{V}). \end{aligned}$$

The submodularity guarantees that the lower bound of mutual information given by $\hat{N}$ greedily selected triggers is no less than $\left(1 - \frac{1}{e}\right)$ of that of $\hat{N}$ optimal triggers in theory, i.e.,

$$\sum_{n=1}^{\hat{N}} \underline{I}_\epsilon\left(\hat{\mathbf{t}}_n|\hat{\mathbf{t}}_{1:(n-1)}\right) \geq \left(1 - \frac{1}{e}\right) \sum_{n=1}^{\hat{N}} \underline{I}_\epsilon\left(\tilde{\mathbf{t}}_n|\tilde{\mathbf{t}}_{1:(n-1)}\right). \tag{8}$$

where $\hat{\mathbf{t}}_n$/$\tilde{\mathbf{t}}_n$ is the n-th trigger in the greedily selected/optimal collection with size $\hat{N}$.

In summary, When the verifier can only query $\hat{N}$ instead of all N triggers, we recommend retrieving the fingerprint with $\hat{\mathbf{T}}$ instead of $\hat{N}$ random triggers in $\mathbf{T}$. This discussion further implies a necessary condition on the number of triggers.



(a) MNIST. (b) FashionMNIST. (c) CIFAR-10. (d) ImageNet.

Figure 3: The robustness of studied DNN fingerprints measured in $\delta_{\mathcal{T}}(\epsilon)$, averaged on 600 models under the fine-pruning attack.

Theorem 2. *Let the threat model be $\mathbf{A}_\epsilon$, a necessary condition for the number of triggers is:*

$$N \geq \min \left\{ \hat{N} : \sum_{n=1}^{\hat{N}} \underline{I}_\epsilon \left(\hat{\mathbf{t}}_n | \hat{\mathbf{t}}_{1:(n-1)} \right) \geq \left(1 - \frac{1}{e} \right) \log_2 P \right\}. \tag{9}$$

If $\epsilon = 0$ and N fails to meet Eq. (9) then any collection of N triggers cannot trace the copyright of all registered models. If $\epsilon > 0$ and N fails to meet Eq. (9) then any collection of N triggers has a risk of failing to trace the copyright of all registered models.

Proof. If N fails to satisfy Eq. (9) then the optimal N triggers might provide less information than $\log_2 P$ bits due to Eq. (8):

$$\sum_{n=1}^{N} \underline{I}_\epsilon \left(\tilde{\mathbf{t}}_n | \tilde{\mathbf{t}}_{1:(n-1)} \right) \leq \frac{1}{1 - \frac{1}{e}} \sum_{n=1}^{N} \underline{I}_\epsilon \left(\hat{\mathbf{t}}_n | \hat{\mathbf{t}}_{1:(n-1)} \right) < \log_2 P.$$

When $\epsilon = 0$ we have $I_0 = \underline{I}_0$, so failing to satisfy Eq. (9) implies a deterministic failure in copyright tracing. Otherwise, we can only assert that copyright tracing has a chance to fail since the lower bound of mutual information provided by any combination of triggers is insufficient to identify all registered models. □

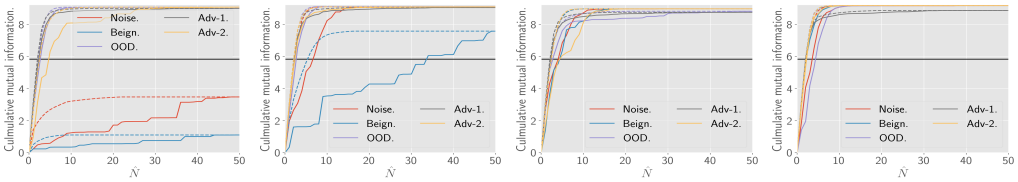
3.3 Remarks

We make three remarks regarding the implications and applications of our analyses.

Remark 1: Optimizing the distribution of triggers. Theorem 2 gives two directions to reduce the number of triggers for copyright tracing. The first is to reduce $\delta_{\mathcal{T}}(\epsilon)$, i.e., to increase the robustness, since $h(\epsilon)$ monotonically increases in $\delta_{\mathcal{T}}(\epsilon)$ when $\delta_{\mathcal{T}}(\epsilon)$ is small. The second is to increase $I_0 \left(\mathbf{t}_n | \mathbf{t}_{1:(n-1)} \right)$. Unfortunately, neither direction can be directly transformed into changes to $\mathcal{T}$. Moreover, there might be a trade-off between robustness and mutual information or uniqueness according to our empirical studies presented in the next section.

Remark 2: Fingerprint vs. watermark. In contrast to fingerprinting triggers, watermarking triggers have labels that are deliberately assigned. It can always be expected that $I_0 \left(\mathbf{t}_n | \mathbf{t}_{1:(n-1)} \right) = \log_2 C$ for watermarking triggers (unless n is so large that queried triggers have already provided enough information for copyright tracing). Despite this advantage, fingerprinting schemes are applicable in more settings so they remain a competitive option.

Remark 3: Generalization to non-classifiers. Our method can be generalized to black-box copyright tracing systems for non-classifiers such as multimedia content generators [26]. The bridge is considering the basic copyright interpreter as a classifier [27, 28]. An example on a copyright tracing system for generative language models is given in Appendix B.



(a) MNIST. (b) FashionMNIST. (c) CIFAR-10. (d) ImageNet.

Figure 4: The cumulative mutual information (in bit) provided by triggers in the original order (the solid curves) and triggers selected by the greedy algorithm (the dashed curves) when $\epsilon = 0$. The black line marks $\left(1 - \frac{1}{e}\right) \log_2 600$ bits.

4 Experiments and Results

4.1 Settings

Following the settings of existing studies [29], we conducted experiments on four classification tasks: MNIST [30], FashionMNIST [31], CIFAR-10 [32], and ImageNet [33]. The number of classes was $C = 10$ (randomly drawn from ImageNet). A series of DNN models were trained as registered models. The sources of heterogeneity were (I) Four network architectures including LeNet-5 [34], VGG-16 [35], ResNet-18, and ResNet-34 [36]. (II) Five learning rates ranging from 0.02 to 0.1. (III) Two learning schedules with step lengths 5 and 10. (IV) Five training epochs ranging from 10 to 60. (V) Three random downsampling of training data. So, there were $P = 600$ models for each task. We used four GeForce RTX 2080 Ti GPUs for acceleration. All experiments were implemented using the PyTorch framework. Link to the code repo is given in the Appendix C.

We considered five representative DNN fingerprinting schemes. For each task, 50 triggers were produced by each scheme. The robustness was measured under the fine-pruning attack [20], results are shown in Fig. 3).

(I) Noise [9]. The pixel of each trigger was randomly generated from a normal distribution whose mean and variance were identical to samples from the training dataset. **(II) Benign** [4, 37, 38]. Each trigger was randomly drawn from the training dataset. **(III) OOD** [3, 10]. Triggers are randomly drawn across training datasets. For classifiers on MNIST and FasionMNIST, each trigger was randomly drawn from the training dataset of CIFAR-10. For classifiers on CIFAR-10 and ImageNet, triggers were drawn from MNIST. **(IV) Adv-1** [11, 13]. The first category of adversarial samples was produced from an ordinary SGD-based adversarial attack [39]. The victim model was a randomly chosen classifier. **(V) Adv-2**. The second category of adversarial samples was produced from the same SGD-based adversarial attack, but initial images were noises.

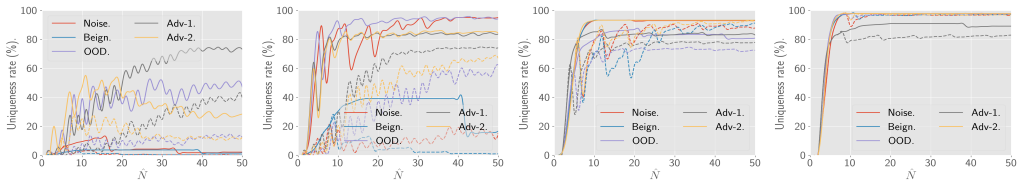
4.2 Results of Using Informative Triggers

4.2.1 Baseline Setting: $\epsilon = 0$

The cumulative mutual information provided by triggers when $\epsilon = 0$ is visualized in Fig. 4. The necessary condition for the number of triggers given by Theorem 2 was uniformly no less than 4 in all combinations of task and fingerprinting scheme, i.e., it is impossible to trace all models with $\lceil \log_{10} 600 \rceil = 3$ triggers. Greedily selected triggers were always more informative than the original triggers. In all cases, the first 15 triggers selected by the greedy algorithm have provided the same amount of information as all 50 triggers, so extra querying is redundant.

4.2.2 Adversarial Setting: $\epsilon > 0$

To delve into the influence of greedy selection on the uniqueness rate u in adversarial environments, we computed u defined by Eq. (3). Results are listed in Table 2. When the number of triggers is limited, using greedily selected triggers always yielded larger mutual information and secured the uniqueness of more models. This phenomenon appeared in all cases regardless of the task, the choice of fingerprinting scheme, the number of triggers, and the threat model with no exception. Notably,



(a) MNIST. (b) FashionMNIST. (c) CIFAR-10. (d) ImageNet.

Figure 5: Uniqueness rate (%) provided by greedily selected informative triggers when $\epsilon = 2.5\%$ (the solid curves) and $\epsilon = 10.0\%$ (the dashed curves).

Table 2: Uniqueness rate of registered models (%). For A/B in each entry, A is the uniqueness rate provided by the original order of triggers, and B is the uniqueness rate provided by greedily selected triggers. The dataset is MNIST, FashionMNIST, CIFAR-10, and ImageNet from top to bottom. The better scheme in each setting is highlighted in bold.

ϵ	Noise			Benign			OOD			Adv-1			Adv-2		
	$\hat{N}=5$	$\hat{N}=10$	$\hat{N}=15$	$\hat{N}=5$	$\hat{N}=10$	$\hat{N}=15$	$\hat{N}=5$	$\hat{N}=10$	$\hat{N}=15$	$\hat{N}=5$	$\hat{N}=10$	$\hat{N}=15$	$\hat{N}=5$	$\hat{N}=10$	$\hat{N}=15$
0.0	1.7/**5.0**	3.0/**9.5**	3.3/**12.2**	0.5/**1.8**	0.7/**3.7**	2.0/**3.7**	63.0/**78.2**	93.7/**96.5**	96.5/**96.5**	69.5/**81.8**	89.5/**91.5**	90.8/**93.5**	23.2/**58.7**	62.5/**92.2**	64.8/**95.3**
5.0	1.7/**5.0**	3.0/**9.5**	3.3/**3.7**	0.5/**1.8**	0.7/**3.7**	2.0/**3.7**	0.5/**0.7**	3.0/**7.2**	7.0/**15.2**	0.7/**0.8**	8.2/**17.7**	16.3/**19.6**	3.8/**12.0**	6.7/**25.2**	7.8/**33.7**
10.0	1.7/**5.0**	0.8/**2.8**	0.2/**3.7**	0.5/**1.8**	0.2/**1.2**	0.5/**1.2**	0.5/**0.7**	3.0/**8.2**	1.7/**4.7**	0.7/**0.8**	1.5/**2.8**	3.7/**4.9**	3.8/**12.0**	6.7/**25.2**	4.7/**20.5**
0.0	14.3/**71.7**	64.0/**96.0**	90.0/**96.0**	0.8/**8.5**	2.5/**30.5**	3.7/**38.2**	54.0/**84.5**	**96.0/96.0**	**96.0/96.0**	72.8/**81.5**	87.8/**92.3**	90.8/**94.2**	71.5/**80.0**	88.8/**95.5**	83.7/**91.0**
5.0	1.5/**12.0**	7.5/**52.8**	7.7/**15.8**	0.8/**8.5**	2.5/**30.5**	3.7/**38.2**	10.2/**21.3**	29.5/**43.2**	**54.7/54.7**	28.5/**30.8**	57.5/**63.7**	68.5/**71.5**	22.2/**26.0**	43.0/**54.2**	59.5/**71.5**
10.0	1.5/**12.0**	1.2/**17.8**	2.3/**5.7**	0.8/**8.5**	0.7/**5.8**	0.7/**10.7**	0.2/**0.8**	4.7/**9.8**	**26.7/26.7**	0.5/**1.2**	25.0/**30.0**	55.7/**60.0**	**0.5/0.5**	16.3/**20.5**	38.7/**47.5**
0.0	22.0/**59.8**	76.8/**93.0**	91.3/**93.3**	17.0/**62.0**	56.3/**93.3**	89.8/**93.3**	47.3/**56.8**	77.3/**80.0**	78.8/**85.8**	65.2/**74.5**	82.0/**85.3**	84.2/**87.7**	17.7/**61.2**	53.2/**93.0**	83.8/**93.3**
5.0	22.0/**59.8**	76.8/**93.0**	79.2/**91.5**	17.0/**62.0**	56.3/**93.3**	69.5/**84.7**	47.3/**56.8**	77.3/**80.0**	74.2/**76.2**	65.2/**74.5**	77.7/**78.3**	80.8/**81.7**	17.7/**61.2**	53.2/**93.0**	53.0/**92.0**
10.0	22.0/**59.8**	36.2/**79.7**	48.3/**80.0**	17.0/**62.0**	22.2/**71.3**	42.8/**60.3**	47.3/**56.8**	69.3/**69.8**	72.0/**72.5**	18.0/**29.2**	67.7/**69.0**	74.8/**75.4**	17.7/**61.2**	18.7/**77.8**	28.5/**80.5**
0.0	40.8/**76.0**	91.3/**97.7**	97.5/**98.0**	66.2/**83.7**	97.7/**98.0**	97.8/**98.0**	29.5/**86.0**	91.5/**97.7**	97.3/**98.0**	74.7/**78.3**	84.8/**87.0**	87.0/**90.0**	64.8/**84.0**	97.5/**98.0**	**98.0/98.0**
5.0	40.8/**76.0**	91.3/**97.7**	97.5/**98.0**	66.2/**83.7**	97.7/**98.0**	97.8/**98.0**	29.5/**86.0**	91.5/**97.7**	97.3/**98.0**	74.7/**78.3**	84.8/**87.0**	87.0/**90.0**	64.8/**84.0**	97.5/**98.0**	**98.0/98.0**
0.0	40.8/**76.0**	74.3/**89.2**	96.2/**97.0**	66.2/**83.7**	97.7/**98.0**	91.7/**96.7**	29.5/**86.0**	69.2/**94.2**	95.0/**96.8**	74.7/**78.3**	78.8/**79.2**	82.7/**83.0**	64.8/**84.0**	89.5/**93.5**	96.5/**97.8**

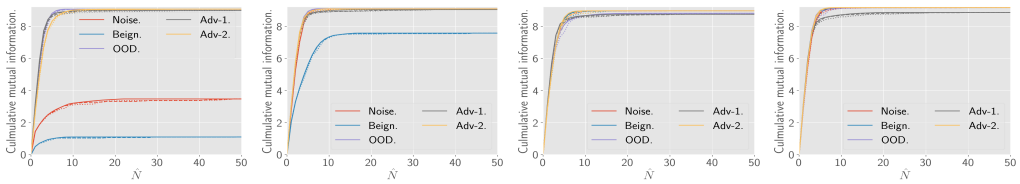
although the necessary condition in Theorem 2 is not a guarantee of perfect uniqueness, satisfying Eq. (9) implies a minimal uniqueness rate of at least 56.8%.

Table 2 also indicates that fewer models' fingerprints could remain unique when the threat model became stronger, i.e., when ϵ increased. One interesting finding is that some fingerprints might not have a higher uniqueness rate when the number of triggers increased. We suspect that when the threat model becomes very strong, the incremental mutual information for a large $\hat{N}$ can be completely nullified, as what is implied by the form of the lower bound given in Theorem 1. To examine this phenomenon, we recorded the uniqueness rate w.r.t. $\hat{N}$ under different threat models. Results shown in Fig. 5 demonstrate that the uniqueness rate was not always increased in $\hat{N}$. The fluctuations are due to rounding issues in computing the uniqueness rate, where only two fingerprints with at most $\lceil \delta_{\mathcal{T}}(\epsilon)\hat{N} \rceil$ differences were recognized as the same.

4.2.3 Re-evaluation of DNN Fingerprinting Schemes

In addition to robustness, statistics in Table 2 and Fig. 5 reveal a different ranking among examined DNN fingerprinting schemes. For example, despite the robustness, both **Noise** and **Benign** failed in MNIST by providing the least information, so did **Benign** in FashionMNIST. The reason is that triggers similar to normal samples are robust due to their entanglement with the classifier's functionality. However, this similarity means models trained on similar data perform alike, offering less model-specific information, especially for simple, high-accuracy tasks. **Adv-2** appeared to be the optimal choices in CIFAR-10 when $\hat{N} \leq 15$ regarding the uniqueness rate. This result is non-trivial since **Adv-2** was not judged as the most informative scheme before applying the greedy algorithm as shown in Fig. 4(c). For ImageNet, **Adv-1** outperformed **Noise** and **OOD** in their original setting when N is small. However, both **Noise** and **OOD** provided perfect uniqueness after the greedy selection process, while **Adv-1** failed. We emphasize that existing robustness-oriented evaluation of DNN fingerprinting schemes tends to overlook these relationships between robustness and uniqueness and overfits oversimplified settings where P is very small and/or $N/\hat{N}$ is very large.

The difficulty in tracing the copyright of DNN models also varies with the task. The more difficult the task is, the easier it is to distinguish models, since models trained on complex datasets tend to be more diversified instead of overfitting a local optimum.



(a) MNIST. (b) FashionMNIST. (c) CIFAR-10. (d) ImageNet.

Figure 6: The culmulative mutual information (in bit) provided by greedily selected triggers with 100 models (the dotted curves), 200 models (the dashed curves), and 600 models (the solid curves)..

Table 3: Uniqueness rate within all 600 models (%). The dataset is MNIST, FashionMNIST, CIFAR-10, and ImageNet from top to bottom. R denotes randomly drawn triggers, 100/200/600 denotes the number of models used to greedily select triggers (i.e., the size of $\mathbf{F}$ in Algo. 1). Settings that satisfy the order $R \leq 100 \leq 200 \leq 600$ are highlighted in green.

ϵ	OOD												Adv-2											
	$\hat{N}=5$				$\hat{N}=10$				$\hat{N}=15$				$\hat{N}=5$				$\hat{N}=10$				$\hat{N}=15$			
	R	100	200	600	R	100	200	600	R	100	200	600	R	100	200	600	R	100	200	600	R	100	200	600
0.0	63.0	71.8	73.2	78.2	93.7	96.2	96.2	96.5	96.5	96.5	96.5	96.5	23.2	50.2	59.2	58.7	62.5	83.3	91.5	92.2	64.8	92.0	93.8	95.3
5.0	0.5	0.5	1.0	0.7	3.0	8.0	8.0	8.2	7.0	10.7	14.2	15.2	3.8	8.5	11.2	12.0	6.7	7.3	20.2	25.2	7.8	24.7	28.3	33.7
10.0	0.5	0.5	1.0	0.7	3.0	8.0	8.0	8.2	1.7	4.2	5.2	4.7	3.8	8.5	11.2	12.0	6.7	7.3	20.2	25.2	4.7	13.0	16.7	20.5
0.0	54.0	76.2	79.3	84.5	96.0	96.0	96.0	96.0	96.0	96.0	96.0	96.0	71.5	77.2	74.5	80.0	88.8	91.0	94.2	95.5	83.7	94.2	94.8	91.0
5.0	10.2	16.3	20.8	21.3	29.5	34.5	37.0	43.2	54.7	54.7	54.7	54.7	22.2	24.2	24.8	26.0	43.0	49.5	52.2	54.2	59.5	59.7	60.3	71.5
10.0	0.2	0.5	1.0	0.8	4.7	5.0	8.2	9.8	26.7	26.7	26.7	26.7	0.5	0.5	0.5	0.5	16.3	18.3	18.7	20.5	38.7	39.7	40.0	47.5
0.0	47.3	55.0	55.8	56.8	77.3	77.8	79.8	80.0	78.8	83.7	85.2	85.8	17.7	47.0	55.8	61.2	53.2	82.7	91.7	93.0	83.8	93.0	93.0	93.3
5.0	47.3	55.0	55.8	56.8	77.3	77.8	79.8	80.0	74.2	75.8	76.0	76.2	17.7	47.0	55.8	61.2	53.2	82.7	91.7	92.0	53.0	84.2	86.2	92.0
10.0	47.3	55.0	55.8	56.8	69.3	69.7	69.7	69.8	72.0	72.0	72.3	72.5	17.7	47.0	55.8	61.2	18.7	44.3	69.3	77.8	28.5	51.8	55.7	80.5
0.0	29.5	65.2	82.3	86.0	91.5	94.8	95.8	97.7	97.3	97.5	97.5	98.0	64.8	79.2	83.3	84.0	97.5	97.7	97.9	98.0	98.0	98.0	98.0	98.0
5.0	29.5	65.2	82.3	86.0	91.5	94.8	95.8	97.7	97.3	97.5	97.5	98.0	64.8	79.2	83.3	84.0	97.5	97.7	97.9	98.0	98.0	98.0	98.0	98.0
10.0	29.5	65.2	82.3	86.0	69.2	73.7	85.0	94.2	95.0	96.5	96.5	96.8	64.8	79.2	83.3	84.0	89.5	93.8	94.2	93.5	96.5	97.2	97.3	97.8

4.3 Scalability to Online Copyright Tracing

An final concern is whether greedily selected triggers remain informative when more models are registered online. Specifically, we assume that the verifier only obtains a small number of DNN models at the beginning, with which he/she greedily selects a series of triggers. We are interested in whether these triggers still outperform randomly drawn triggers on unseen models or not.

The greedily selected triggers according to either 100 or 200 models performed almost identically to those selected according to all 600 models regarding the culmulative information, as shown in Fig. 6. The failure to refer to all registered DNN models had a small impact on the uniqueness rate as shown in Table 3. In almost all cases, the greedily selected triggers outperformed randomly selected triggers, even if only 100 models were considered. In general, the more models verifier can observe during greedy selection process, the larger the uniqueness rate is. This fact is justified by the dominance of green entries in Table 3. Therefore, we recommend that the copyright verifier collects as many models as possible to select informative triggers, or choose to update the collection of triggers when the number of registered DNN models increases. Even if the number of observed models is small, the greedily selected triggers are more informative than randomly drawn triggers and result in a larger uniqueness rate.

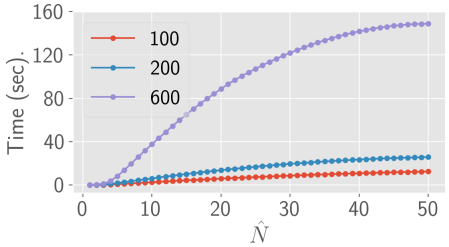


Figure 7: The time consumption of greedy trigger selection.

4.4 Overhead

We also evaluated the overhead of our method. The time consumption is no more than 160 seconds even when all 600 models were used to optimize 50 triggers (it has been shown that 15-30 triggers

were sufficient for copyright tracing), as shown in Fig. 7. This overhead is acceptable and is independent from the dataset.

5 Discussions

The fingerprint optimization algorithm proposed in this paper adopts informative triggers during the fingerprint selection phase, aiming to achieve enhanced uniqueness. It does not involve the design or implementation of the fingerprint scheme itself, and can be seamlessly integrated with trigger set-based fingerprint schemes to improve their effectiveness in practical application scenarios.

Limitations. More complex task scenarios beyond text generation, such as image and video generation, have not yet been discussed, which will also be the focus of our future work.

6 Conclusions

This paper explores uniqueness, a less frequently studied yet important dimension in evaluating DNN fingerprinting schemes for copyright tracing. After highlighting the significance and challenges regarding this property, we adopt an information theoretical perspective to quantify the contribution of each fingerprinting trigger. We design an algorithm to efficiently estimate the conditional mutual information of each trigger and propose a greedy algorithm that facilitates the efficiency of copyright tracing. Extensive experiments show that our method can be easily combined with arbitrary DNN fingerprinting schemes to improve the performance regarding uniqueness, even in the online setting. Our studies reveal several new insights in evaluating and comparing DNN fingerprinting schemes and suggest more attentions on uniqueness in addition to robustness.

Acknowledgments and Disclosure of Funding

The work described in this paper was supported in part by the National Natural Science Foundation of China (62271307).

The AI-driven experiments, simulations and model training were performed on the robotic AI-Scientist platform of Chinese Academy of Sciences.

References

[1] D. Guo, H. Chen, R. Wu, and Y. Wang, "Aigc challenges and opportunities related to public safety: a case study of chatgpt," *Journal of Safety Science and Resilience*, vol. 4, no. 4, pp. 329–339, 2023.

[2] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *27th USENIX security symposium (USENIX Security 18)*, 2018, pp. 1615–1631.

[3] J. Zhang, Z. Gu, J. Jang, H. Wu, and M. P. Stoecklin et al., "Protecting intellectual property of deep neural networks with watermarking," in *Proceedings of the 2018 on Asia conference on computer and communications security*, 2018, pp. 159–172.

[4] R. Namba and J. Sakuma, "Robust watermarking of neural network with exponential weighting," in *Proceedings of the 2019 ACM ASIACCS*, 2019, pp. 228–240.

[5] J. Zhang, D. Chen, J. Liao, H. Fang, W. Zhang, W. Zhou, H. Cui, and N. Yu, "Model watermarking for image processing networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 805–12 812.

[6] J. Zhang, D. Chen, J. Liao, W. Zhang, G. Hua, and N. Yu, "Passport-aware normalization for deep model protection," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 619–22 628, 2020.

[7] J. Zhang, D. Chen, J. Liao, W. Zhang, H. Feng, G. Hua, and N. Yu, "Deep model intellectual property protection via deep watermarking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 4005–4020, 2021.

[8] J. Zhang, D. Chen, J. Liao, Z. Ma, H. Fang, W. Zhang, H. Feng, G. Hua, and N. Yu, "Robust model watermarking for image processing networks via structure consistency," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[9] T. Dong, S. Li, G. Chen, M. Xue, H. Zhu, and Z. Liu, "Rai 2: Responsible identity audit governing the artificial intelligence," in *2022 Network and Distributed System Security Symposium*, 2022, pp. 1–18.

[10] A. Bansal, P.-y. Chiang, M. J. Curry, R. Jain, C. Wigington, and V. Manjunatha et al., "Certified neural network watermarks with randomized smoothing," in *International Conference on Machine Learning*. PMLR, 2022, pp. 1450–1465.

[11] J. Zhao, Q. Hu, G. Liu, X. Ma, F. Chen, and M. M. Hassan, "Afa: Adversarial fingerprinting authentication for deep neural networks," *Computer Communications*, vol. 150, pp. 488–497, 2020.

[12] N. Lukas, Y. Zhang, and F. Kerschbaum, "Deep neural network fingerprinting by conferrable adversarial examples," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=VqzVhqxkjH1

[13] X. Cao, J. Jia, and N. Z. Gong, "Ipguard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary," in *Proceedings of the 2021 ACM ASIACCS*, 2021, pp. 14–25.

[14] Z. Peng, S. Li, G. Chen, C. Zhang, H. Zhu, and M. Xue, "Fingerprinting deep neural networks globally via universal adversarial perturbations," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 13 420–13 429. [Online]. Available: https://doi.org/10.1109/CVPR52688.2022.01307

[15] K. Yang, R. Wang, and L. Wang, "Metafinger: Fingerprinting the deep neural networks with meta-training." in *IJCAI*, 2022, pp. 776–782.

[16] S. Wang, P. Zhao, X. Wang, S. Chin, T. Wahl, and Y. Fei et al., "Intrinsic examples: Robust fingerprinting of deep neural networks," in *British Machine Vision Conference (BMVC)*, 2021.

[17] F. Li, H. Zhao, W. Du, and S. Wang, "Revisiting the information capacity of neural network watermarks: Upper bound estimation and beyond," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 19, 2024, pp. 21 331–21 339.

[18] W. Aiken, H. Kim, S. Woo, and J. Ryoo, "Neural network laundering: Removing black-box backdoor watermarks from deep neural networks," *Computers & Security*, vol. 106, p. 102277, 2021.

[19] X. Chen, W. Wang, C. Bender, Y. Ding, R. Jia, B. Li, and D. Song, "Refit: a unified watermark removal framework for deep learning systems with limited data," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2021, pp. 321–335.

[20] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *International symposium on research in attacks, intrusions, and defenses*. Springer, 2018, pp. 273–294.

[21] N. Chattopadhyay, C. S. Y. Viroy, and A. Chattopadhyay, "Re-markable: Stealing watermarked neural networks through synthesis," in *Security, Privacy, and Applied Cryptography Engineering: 10th International Conference, SPACE 2020, Kolkata, India, December 17–21, 2020, Proceedings 10*. Springer, 2020, pp. 46–65.

[22] S. Wang, X. Wang, P.-Y. Chen, P. Zhao, and X. Lin, "Characteristic examples: High-robustness, low-transferability fingerprinting of neural networks," in *International joint conferences on artificial intelligence organization (IJCAI)*, 2021.

[23] A. Karanam, K. Killamsetty, H. Kokel, and R. Iyer, "Orient: Submodular mutual information measures for data subset selection under distribution shift," *Advances in neural information processing systems*, vol. 35, pp. 31 796–31 808, 2022.

[24] C. Li, S. Kothawade, F. Chen, and R. Iyer, "Platinum: Semi-supervised model agnostic meta-learning using submodular mutual information," in *International Conference on Machine Learning*. PMLR, 2022, pp. 12 826–12 842.

[25] C. Di, F. Li, P. Xu, Y. Guo, C. Chen, and M. Shu, "Learning automata-accelerated greedy algorithms for stochastic submodular maximization," *Knowledge-Based Systems*, vol. 282, p. 111118, 2023.

[26] Z. Xi, W. Huang, K. Wei, W. Luo, and P. Zheng, "Ai-generated image detection using a cross-attention enhanced dual-stream network," in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2023, pp. 1463–1470.

[27] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, "Detectgpt: Zero-shot machine-generated text detection using probability curvature," in *International Conference on Machine Learning*. PMLR, 2023, pp. 24 950–24 962.

[28] L. Dugan, D. Ippolito, A. Kirubarajan, S. Shi, and C. Callison-Burch, "Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 11, 2023, pp. 12 763–12 771.

[29] C. Xiong, G. Feng, X. Li, X. Zhang, and C. Qin, "Neural network model protection with piracy identification and tampering localization capability," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 2881–2889.

[30] M. B. Abdulrazzaq and J. N. Saeed, "A comparison of three classification algorithms for handwritten digit recognition," in *2019 International Conference on Advanced Science and Engineering (ICOASE)*. IEEE, 2019, pp. 58–63.

[31] M. Kayed, A. Anter, and H. Mohamed, "Classification of garments from fashion mnist dataset using cnn lenet-5 architecture," in *2020 international conference on innovative trends in communication and computer engineering (ITCE)*. IEEE, 2020, pp. 238–243.

[32] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," *Procedia computer science*, vol. 132, pp. 377–384, 2018.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 248–255.

[34] M. R. Islam and A. Matin, "Detection of covid 19 from ct image by the novel lenet-5 cnn architecture," in *2020 23rd International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2020, pp. 1–5.

[35] A. Vedaldi and A. Zisserman, "Vgg convolutional neural networks practical," *Department of Engineering Science, University of Oxford*, vol. 66, 2016.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] H. Wu, G. Liu, Y. Yao, and X. Zhang, "Watermarking neural networks with watermarked images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 7, pp. 2591–2601, 2021. [Online]. Available: https://doi.org/10.1109/TCSVT.2020.3030671

[38] T. Maho, T. Furon, and E. Le Merrer, "Fingerprinting classifiers with benign inputs," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5459–5472, 2023.

[39] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE transactions on neural networks and learning systems*, pp. 2805–2824, 2019.

[40] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, J. Dean, and S. Ghemawat, "Language models are unsupervised multitask learners," in *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, pp. 137–150.

[41] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima *et al.*, "The pile: An 800gb dataset of diverse text for language modeling," *arXiv preprint arXiv:2101.00027*, 2020.

[42] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.

[43] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff *et al.*, "Pythia: A suite for analyzing large language models across training and scaling," in *International Conference on Machine Learning*. PMLR, 2023, pp. 2397–2430.

[44] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[45] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, "Scaling instruction-finetuned language models," *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.

[46] B. Workshop, T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon *et al.*, "Bloom: A 176b-parameter open-access multilingual language model," *arXiv preprint arXiv:2211.05100*, 2022.

[47] C. Tan, V. Niculae, C. Danescu-Niculescu-Mizil, and L. Lee, "Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions," in *Proceedings of the 25th international conference on world wide web*, 2016, pp. 613–624.

[48] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, 2015.

[49] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization," *arXiv preprint arXiv:1808.08745*, 2018.

[50] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, "Eli5: Long form question answering," *arXiv preprint arXiv:1907.09190*, 2019.

[51] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," *arXiv preprint arXiv:1805.04833*, 2018.

[52] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen, "A corpus and cloze evaluation for deeper understanding of commonsense stories," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 839–849.

[53] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "Hellaswag: Can a machine really finish your sentence?" *arXiv preprint arXiv:1905.07830*, 2019.

[54] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[55] N. S. Moosavi, A. Rücklé, D. Roth, and I. Gurevych, "Learning to reason for text generation from scientific tables," *arXiv preprint arXiv:2104.08296*, 2021.

[56] Y. Chen, H. Kang, V. Zhai, L. Li, R. Singh, and B. Raj, "Token prediction as implicit classification to identify llm-generated text," in *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[57] K. Wu, L. Pang, H. Shen, X. Cheng, and T.-S. Chua, "Llmdet: A third party large language models generated text detection tool," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 2113–2133.

[58] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause, "Lazier than lazy greedy," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract has stated the claims made, including the contributions made in the paper and important assumptions and limitations.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations are discussed in conclusions.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Self-contained proofs have been given for theoretical results. Additional assumptions and theories have been given sufficient references.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We disclose as many details as we could, and provide the codes. We used public datasets and toolkits.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The anonymous repo has been given in the appendix D.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Most settings showned in section 4.1, and more details can be found and checked in the anonymous code repo.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We recorded and visualized each experiment in Figures, including noises.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Section 4.1.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This paper is with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: This paper address a crucial aspect of AI security

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [Yes]

    Justification: This paper aims to analyze a critical threat to the misuse of AI models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: They have been explicitly mentioned and properly respected.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: This paper does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Proof of Theorem 1

Proof. Consider an attack that changes each position of the fingerprint with a probability δ. This attack is featured by a C-ary code of length N, among which $(1-\delta)N$ positions equal zero to represent triggers whose predictions remain invariable. The other δN positions are distributed in $\{1, 2, \cdots, C-1\}$ to represent that the corresponding predictions have been shifted to the next $1, 2, \cdots, (C-1)$-th class (modulo C).

We use ϕ_n^0 to denote the output of the suspicious model on $\mathbf{t}_n$ before being attacked. For the first term in Eq. (4), :

$$\begin{aligned} H(\phi_n|\phi_1, \cdots, \phi_{n-1}) &\overset{(a)}{\geq} H(\phi_n|\phi_1, \cdots, \phi_{n-1}, \mathcal{A}) \\ \overset{(b)}{=} H(\phi_n^0|\phi_1, \cdots, \phi_{n-1}, \mathcal{A}) &\overset{(c)}{=} I_0\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right). \end{aligned} \tag{10}$$

in which *(a)* follows the basic properties of entropy, *(b)* and *(c)* hold since once the attack is known, the entire case can be reduced to the vanilla setting as if no attack has been applied.

The second term in Eq. (4) equals:

$$\begin{aligned} & H(\phi_n, \mathcal{A}_{1:n}|\phi_1, \cdots, \phi_{n-1}, \mathcal{F}) - H(\mathcal{A}_{1:n}|\phi_1, \cdots, \phi_n, \mathcal{F}) \\ \overset{(a)}{=} & H(\phi_n, \mathcal{A}_{1:n}|\phi_1, \cdots, \phi_{n-1}, \mathcal{F}) \\ \overset{(b)}{=} & H(\mathcal{A}_{1:n}|\phi_1, \cdots, \phi_{n-1}, \mathcal{F}) \overset{(c)}{=} H(\mathcal{A}_n) \\ \overset{(d)}{\leq} & -\delta\log_2\delta - (1-\delta)\log_2(1-\delta) + \delta\log_2(C-1), \end{aligned} \tag{11}$$

where $\mathcal{A}_{1:n}$ and $\mathcal{A}_n$ denote the attack on corresponding triggers. All *(a)-(d)* use the attack's representation, *(b)* also relies on the chain rule of entropy. Combining Eq. (10) and Eq. (11) yields:

$$\begin{aligned} I_\epsilon\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right) \geq & I_0\left(\mathbf{t}_n|\mathbf{t}_{1:(n-1)}\right) + \delta\log_2\delta \\ & + (1-\delta)\log_2(1-\delta) - \delta\log_2(C-1). \end{aligned} \tag{12}$$

The r.h.s. of Eq. (12) monotonically decreases in δ (when $\delta \leq 0.5$). Combining this observation with Eq. (2) yields Eq. (7). □

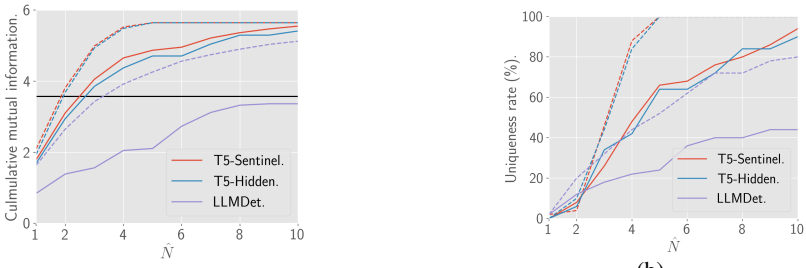
B Applications on Copyright Tracing of Generative Language Models

We demonstrate the application of the proposed method on the copyright tracing of generative language models.

We built a collection of $P = 50$ generative language models including `GPT-2` [40] with 10 fine-tuned versions, `GPT-Neo-125M` [41] with 10 fine-tuned versions, `OPT-125M` [42] with 10 fine-tuned versions, `OPT-350M` with 10 fine-tuned versions, `Pythia-70M` [43], `Pythia-160M`, `Pythia-160M-deduped`, `T5-Small` [44], `Flan-T5-Small` [45], and `BLOOM-560M` [46]. Each fine-tuned version used one corpus from CMV [47], Yelp [48], TLDR [49], XSum [49], ELI5 [50], WP [51], ROC [52], HellaSwag [53], SQuAD [54], and SciGen [55].

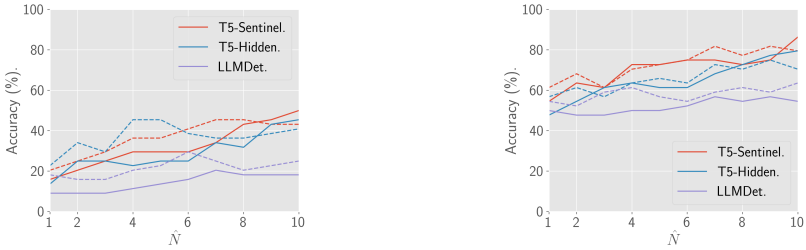
Candidate basic copyright tracing algorithms were T5-Sentinel [56] ($C = 5$), T5-Hidden ($C = 5$), and LLMDet [57] ($C = 9$). Given a series of prompts, the suspicious model generates a series of texts, which are fed into the basic copyright tracing algorithm. The fingerprint of the suspicious model is the list of outputs from the copyright tracing algorithm. For example, LLMDet has captured texts generated from {0:`Human`, 1:`GPT-2`, 2:`OPT`, 3:`UniLM`, 4:`LLaMA`, 5:`BART`, 6:`T5`, 7:`BLOOM`, 8:`GPT-Neo`}. Given a list of seven prompts, `T5-Small` returns six sentences, which might be classified by LLMDet into [`Human`,`GPT-2`,`T5`,`LLaMA`,`Human`,`T5`,`GPT-2`], so the fingerprint of `T5-Small` under LLMDet can be encoded as [0,1,6,4,0,6,1], which can be interpreted as the fingerprint of a nine-class classifier.

Initially, $N = 200$ prompts were random drawn from the union of 10 corpura for fine-tuning. The culmulative mutual information provided by fingerprints from three algorithms is visualuzed in Fig. 8. It turns out that either T5-Sentinel or T5-Hidden was capable of distinguishing all 50 models with five prompts, although they were trained on only five models (`Human`, `GPT3.5`, `PaLM`, `LLaMA`, and `GPT2-XL`). Meanwhile, LLMDet has learned data generated from nine different models, but it failed



(a) (b)

Figure 8: (a) The culmulative mutual information (in bit) and (b) the uniqueness rate provided by prompts in the original order (the solid curves) and prompts seleted by the greedy algorithm (the dashed curves). The black line marks $\left(1-\frac{1}{e}\right)\log_2 50$ bits.



(a) Top-5. (b) Top-10.

Figure 9: The copyright tracing accuracy provided by prompts in the original order (the solid curves) and prompts seleted by the greedy algorithm (the dashed curves).

to differentiate all models even with ten prompts. After all, all three schemes' performance was boosted under the greedy selection framework.

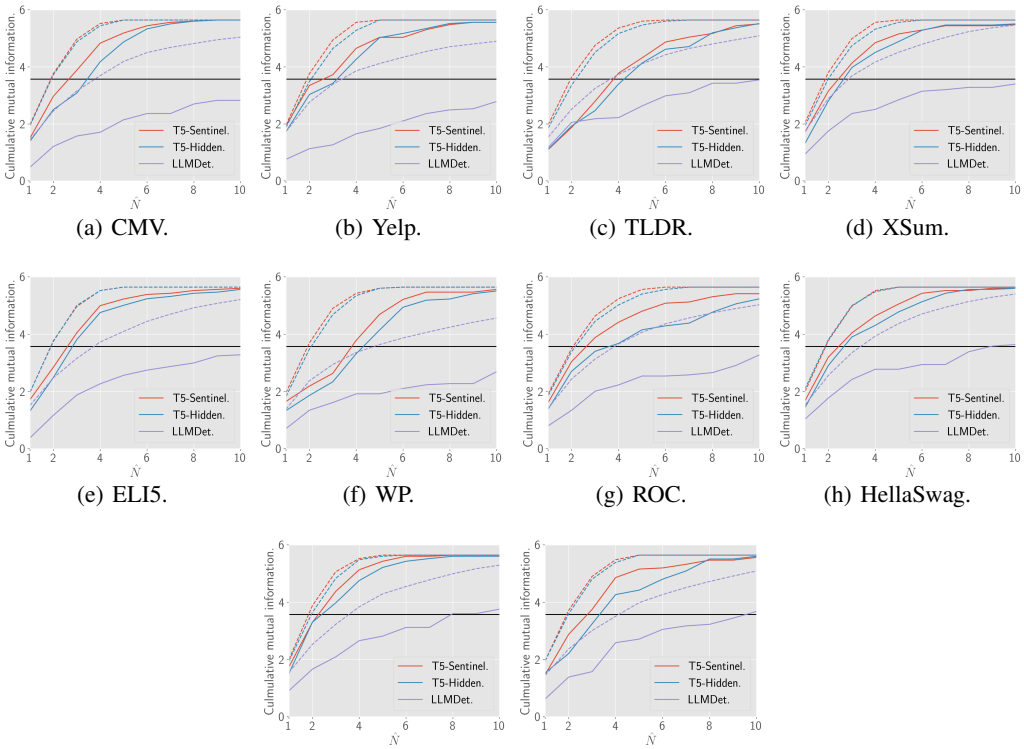
The uniqueness declined sharply in the adversarial setting where we implemented fine-tuning with Wikimedia corpus [2] as adversarial modifications. Greedily selected triggers outperformed the baseline random setting as well. This is reflected in the copyright tracing accuracy in Fig. 9.

We remark that for generative language models, the conditional mutual information of a prompt depends on both the prompt's source corpus and the classifier algorithm (there is no extra copyright tracing classifier for DNN classifiers to be protected), our experiments suggested that the first factor also had a small influence as shown in Fig. 10. Although T5-Sentinal and T5-Hidden performed differently across corpura (using random prompts from a corpus), their performed almost identically after incorporating the greedy selection scheme. Meanwhile, the culmulative mutual information of LLMDet remained the lowest in all cases, yet our greedy selection scheme uniformly boosted its performance.

To simplify the evaluation, the random seeds within language models were manually fixed. In practice, the verifier is encouraged to feed a prompt to a language model for multiple times, record the predictions returned from the basic copyright tracing algorithm, and conduct a voting. It can be proven that when the error of this estimation for each prompt is bounded by ϵ (i.e., the probability that the prediction for this prompt differs from the statistical mode is no larger than ϵ) then the bound in Eq. (8) should be relaxed into:

$$\begin{aligned}&\sum_{n=1}^{\hat{N}} \underline{I}_\epsilon\left(\hat{\mathbf{t}}_n|\hat{\mathbf{t}}_{1:(n-1)}\right) \geq \\ &\left(1-\frac{1}{e}\right)\left(\sum_{n=1}^{\hat{N}} \underline{I}_\epsilon\left(\tilde{\mathbf{t}}_n|\tilde{\mathbf{t}}_{1:(n-1)}\right) - \epsilon\hat{N}^2\left(1-\frac{1}{\hat{N}}\right)\log_2 C\right) - \epsilon\hat{N}^2\log_2 C.\end{aligned} \tag{13}$$

[2] https://dumps .wikimedia.org.



(a) CMV. (b) Yelp. (c) TLDR. (d) XSum.

(e) ELI5. (f) WP. (g) ROC. (h) HellaSwag.

(i) SQuAD. (j) Sci-Gen.

Figure 10: The culmulative mutual information (in bit) provided by prompts in the original order (the solid curves) and prompts seleted by the greedy algorithm (the dashed curves). The black line marks $\left(1-\frac{1}{e}\right)\log_2 50$ bits.

The proof is similar to the induction in Lemma 2 in [58].

In conclusion, our scheme can be generalized to other non-classifiers and boost the performance of copyright tracing by increasing the uniqueness rate. Additionally, it can be used in the open setting where the models to be traced have not been included into the training set of basic copyright tracing algorithms, so it is necessary to use multiple prompts to extract their fingerprints.

C Code Repo Link

All codes for reproducibility in `https://github.com/zzmsmm/Informative_Triggers`.