Intellectual Property Protection for Deep Models: Pioneering Cross-Domain Fingerprinting Solutions

Tianhua Xu, Sheng-hua Zhong, Zhi Zhang, and Yan Liu

***Abstract*—The high cost of developing high-performance deep models highlights their value as intellectual property for creators. However, it is important to consider the potential risks of theft. Although various techniques have been developed to protect the intellectual property of deep models, there is still room for improvement in terms of efficiency, comprehensiveness, and generalization. Compared with the intrusiveness of watermarking methods, fingerprinting methods do not affect the training process of the source model. Consequently, this paper proposes a fingerprinting method to address the paucity of attempts in fingerprinting methods for model protection. Our method consists of two efficient algorithms for generating fingerprinting samples, where the first one possesses the advantage of efficiency, while the second one is better in terms of robustness. The first algorithm takes a comprehensive approach to modeling the fingerprint of the deep model. The generated samples are distributed within the stable region and near the decision boundary of the model, taking into account both the duality and the conviction factors. Then, a heuristic sample perturbation algorithm is introduced, which generates a fingerprint with solid stability and generalization across multiple domains. The two algorithms proposed in this paper have been shown to be capable of withstanding attacks on intellectual property removal, detection, and evasion. They also show some advantages in terms of efficiency. In addition, the proposed method is the first to apply fingerprinting techniques in a cross-domain context.**

***Index Terms*—Model protection, intellectual property, model fingerprint.**

I. Introduction

DEEP learning models have become central to various multimedia and multi-modal content understanding tasks, delivering significant advancements in performance across many domains. Recently, these models have grown in scale, with some reaching trillions of parameters, leading to impressive capabilities. However, increasing the model size has also led to a corresponding increase in training costs, often running into tens of millions of dollars [1]. As a result, these models represent substantial Intellectual Property (IP) assets for their creators. Unfortunately, this immense value also makes them prime targets for malicious actors who seek to steal, misuse, or redistribute the models, leading to serious IP infringements. Recent studies have shown that attackers can replicate a model's functionality simply by exploiting its Application Programming Interfaces (APIs) [2]. This highlights a critical need for robust protection mechanisms to protect the IP of deep models.

One of the most promising research areas is the development of IP protection techniques for deep models. Traditional multimedia watermarking methods have been adapted to embed unique markers within these models [3], [4]. In contrast, model fingerprinting has emerged as a method to extract unique identifiers from deep models [5], [6]. Both watermarking and fingerprinting are still in the early stages of development, with fingerprinting receiving comparatively less attention than watermarking. While watermarking is generally reliable, it often alters the internal structure of the model, potentially leading to unpredictable performance effects. In contrast, fingerprinting focuses on extracting inherent model characteristics—such as decision boundaries, adversarial robustness, and other intrinsic properties—without modifying the model's architecture or parameters. This non-intrusive approach is essential to preserve the model's integrity and minimize the risk of performance degradation.

Both watermarking and fingerprinting have primarily been applied in traditional domains such as Computer Vision (CV) and Natural Language Processing (NLP). However, one area that remains underexplored regarding IP protection is Brain-Computer Interfaces (BCI), particularly Electroencephalogram (EEG)-based models. These models, which analyze sensitive physiological data to predict users' movements, moods, and other states, are becoming increasingly complex and require effective IP protection. To date, only a few studies [7], [8] have explored protection mechanisms for EEG-based models, highlighting a significant gap in research.

Despite progress in watermarking and fingerprinting, protecting deep models across diverse domains remains an open challenge. The unique characteristics of each domain, such as the type of data processed, require tailored protection strategies. Specifically, fingerprinting holds significant promise

Received 4 August 2024; revised 1 January 2025 and 16 February 2025; accepted 10 March 2025. Date of publication 21 March 2025; date of current version 2 April 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62472291, in part by the Natural Science Foundation of Guangdong Province under Grant 2025A1515012154 and Grant 2023A1515012685, and in part by the Open Fund of National Engineering Laboratory for Big Data System Computing Technology under Grant SZU-BDSC-OF2024-14. The associate editor coordinating the review of this article and approving it for publication was Prof. Guowen Xu. *(Corresponding author: Sheng-hua Zhong.)*

Tianhua Xu and Sheng-hua Zhong are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: xutianhua2021@email.szu.edu.cn; csshzhong@szu.edu.cn).

Zhi Zhang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: 22038275r@connect.polyu.hk).

Yan Liu is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: yan.liu@polyu.edu.hk).

Digital Object Identifier 10.1109/TIFS.2025.3552175

for domains like BCI, where model integrity is critical, as even small alterations can introduce bias or lead to substantial performance degradation. In response to this challenge, we propose a heuristic fingerprint identification strategy that minimizes the impact on model performance while overcoming the limitations of existing methods. Our approach aims to broaden the scope of IP protection across diverse domains, offering a more robust and adaptable solution to safeguard deep models in the future.

In this paper, we present a novel approach to IP protection for deep models, designed to address key challenges in safeguarding IP across multiple domains. Our efforts are threefold:

(I) Efficient sample generation: We propose a creative strategy for generating samples based on duality and conviction factors, which enables efficient and comprehensive extraction of model identifiers. This method ensures that the generated identifiers are distinctive and resilient, effectively identifying deep models while minimizing resource consumption during the fingerprinting process.

(II) Heuristic algorithm for sample perturbation: We introduce a heuristic algorithm for perturbing samples to enhance the stability of the resulting fingerprints. Our approach ensures that the generated fingerprints are robust and reliable even when applied to cross-domain tasks.

(III) Cross-domain application: To our knowledge, we are the first to apply fingerprinting techniques in a cross-domain context, including CV, NLP, and EEG-based BCI models, for IP protection. Our approach is the first to effectively extend fingerprinting to secure IP across such diverse and complex fields.

In summary, our contributions focus on developing a robust, efficient, and cross-domain applicable strategy for deep model fingerprinting, which significantly advances IP protection for deep models. Our method enables effective protection without compromising the performance of the model, and we make our code publicly available for future research and implementation.[1]

II. Related Work

In this section, we briefly review the existing literature on model IP protection methods, which can be broadly categorized into model watermarking and model fingerprinting.

A. Model Watermarking

Model watermarking involves embedding knowledge into various components of a target model, such as weights, layers, and outputs. As demonstrated by [9], early approach focused on embedding watermarks directly into the model weights. Subsequent research explored backdoor watermarks, which utilized trigger sets to create a hidden watermark within the model [8]. Fan et al. introduced a passport-based method for deep model ownership verification, which ensures robustness against removal and ambiguity attacks by adjusting the model's inference performance based on valid passports [10]. Szyller et al. developed Dynamic Adversarial Watermarking of Neural Networks (DAWN), a dynamic watermarking method for deep models at their prediction APIs to deter IP theft through model extraction. DAWN has been shown to be robust against adversarial manipulations and model extraction attacks, with minimal impact on the utility of the model [11]. Most recently, He et al. presented a novel method for safeguarding IP in language generation APIs by embedding lexical watermarks in generated text, which remain effective for identifying IP infringements [4]. It is evident that model watermarking methods share certain similarities with data watermarking methods, which has led to the dominance of model watermarking in the field of model protection. However, it is important to recognize that model watermarking can influence the model's learning process, making it challenging to avoid potential impacts on model accuracy.

B. Model Fingerprinting

Model fingerprinting is a critical technique for uniquely identifying a model through its inherent properties. Cao et al. introduced IPGuard, a method that effectively protected the IP of deep model classifiers by uniquely fingerprinting their classification boundaries without compromising the accuracy of the model [12]. However, the effectiveness of these methods can be compromised by so-called model extraction attacks that modify the boundary. Zheng et al. proposed a method that took advantage of the distinctiveness of the lower-layer weights and random projection to embed an undeniable and irrevocable proof within a fingerprint, significantly advancing IP protection for deep models [13]. Given that generative methods show good performance on many tasks [14], the generated samples can also be used for fingerprinting. Concurrently, multiple studies have focused on identifying unique samples that reflect behaviors exclusive to the source model and are not replicated in irrelevant models. The generation of such distinctive samples is based on a compilation of both pirated and irrelevant models [6], [15], [16]. For example, in the study by Lukas et al. [16], the authors proposed a method for extracting adversarial samples that are transferable exclusively to stolen models and not to independently trained models that are irrelevant to the original model. This approach effectively enhanced the robustness of models against extraction attacks, but it required a significant amount of additional training. Quan further advanced the field by introducing a groundbreaking fingerprinting framework specifically designed for deep models in image restoration tasks, using critical images to extract fingerprints [17]. From these existing work we can get the conclusion that the primary focus of watermarking and fingerprinting is on single-domain IP protection. However, the expansion of these techniques' practical applicability to various domains has the potential to enhance their overall effectiveness and utility. We are particularly drawn to the non-invasive nature of model fingerprinting. Consequently, we design a novel model fingerprinting strategy that is both stable and efficient across various domains.

[1]Our code is available here: https://github.com/munchidelufu/primary evolvedfinger

III. Method

This section comprises four components: Sample-Driven Model Identification, Model Fingerprint Construction, Model Fingerprint Matching, and Theoretical Insights into Fingerprints. In Model Fingerprint Construction, we utilize two distinct algorithms: the Primary Fingerprint and Evolved Fingerprint algorithms. The Primary Fingerprint algorithm stands out for its simplicity and efficiency, making it a fast and effective solution. In contrast, though more complex, the Evolved Fingerprint algorithm provides greater stability and reliability in the fingerprinting process.

A. Sample-Driven Model Identification

A well-trained model effectively encapsulates the complexity of the decision space, skillfully segregating the dataset according to the patterns and features learned. This segregation is not merely a division but a nuanced classification of samples into distinct categories or decision regions. To quantitatively assess the efficacy of this classification, cross-entropy loss is employed as a pivotal metric and is encapsulated as follows:

$$\ell_i = -\sum_{k=1}^{K} q(k \mid x_i) \log p(k \mid x_i) \tag{1}$$

Here, $q(k \mid x_i)$ denotes the true probability distribution of sample x_i belonging to class k, $p(k \mid x_i)$ represents the predicted probability distribution by the model, and ℓ_i represents the cross-entropy loss for a sample x_i. We believe that the scale of ℓ_i is influenced by two key factors:

- Duality: It assesses the accuracy of the model's decision on x_i as either *correct* or *wrong*.
- Conviction: It reflects the confidence of the model in the decision about x_i, *confident* or *uncertain*.

Duality is a binary factor where the model knows whether the samples are classified correctly or wrongly. Conviction, on the other hand, is a continuous factor. It distinguishes between confident and uncertain samples based on the similarity between $q(k \mid x_i)$ and $p(k \mid x_i)$. By combining duality and conviction, we categorize samples into four types: *correct-confident* (*cc*), *correct-uncertain* (*cu*), *wrong-confident* (*wc*), and *wrong-uncertain* (*wu*). Previous fingerprinting methods relied on samples near the decision boundary to determine the model's fingerprint. Based on duality and conviction, these methods mainly focus on *cu* and *wu* samples. However, we argue that incorporating samples *cc* and *wc* can enhance the discriminative power of the model fingerprints. Specifically, we calculate ℓ_i of x_i and divide the samples according to duality and conviction to form the model's fingerprint.

B. Model Fingerprint Construction

Within context, we designate the model intended for protection as the source model M_s, the models derived from unauthorized copying as the pirated model M_p, and models without association to plagiarism as the irrelevant model M_i.

Primary Fingerprint. Since a model tends to exhibit higher confidence in its train set compared to test set [18], we propose using the model-inferred ℓ_i of x_i in the train set to collect the four types of samples mentioned above. Departing from duality, the train set includes the correct and wrong samples predicted by M_s. Given the above facts, we rank the correctly and wrongly classified samples in ascending order based on ℓ_i. Finally, as shown in Eq. (2) and Eq. (3), we get the sequential samples expressed as SC and SW. Within the context, x_1 denotes the first sample in the forward sequence, whereas x_{-1} corresponds to the first sample in the reverse sequence.

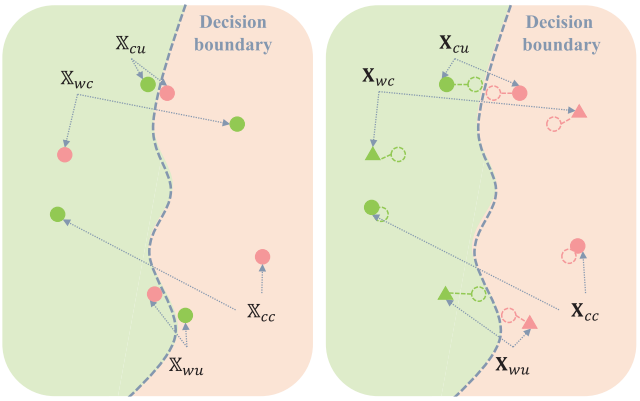

Fig. 1. The intuition of our proposed Primary Fingerprint algorithm (left) and Evolved Fingerprint algorithm (right). Samples near the decision boundary are uncertain, while those farther away are confident. On the right side of the figure, the labels of evolved samples are set by source model, while the triangular samples represent the label changes from the primary samples.

$$SC = \{x_1 \ldots x_i \ldots x_{-1}\},\ \ell_1 < \ell_i < \ell_{-1} \text{ and } x_i \text{ is correct} \tag{2}$$

$$SW = \{x_1 \ldots x_i \ldots x_{-1}\},\ \ell_1 < \ell_i < \ell_{-1} \text{ and } x_i \text{ is wrong} \tag{3}$$

Considering the conviction, the confident and uncertain samples correspond to the lower and higher ℓ_i, respectively. Hence, as illustrated in Eq. (4) and Eq. (5), we integrate duality and conviction to derive four types of samples, where n indicates the number of samples selected.

$$\begin{cases} \mathbb{X}_{cc} = \{x_1 \ldots x_i \ldots x_n\},\ x_i \in SC \\ \mathbb{X}_{cu} = \{x_{-n} \ldots x_i \ldots x_{-1}\},\ x_i \in SC \end{cases} \tag{4}$$

$$\begin{cases} \mathbb{X}_{wc} = \{x_1 \ldots x_i \ldots x_n\},\ x_i \in SW \\ \mathbb{X}_{wu} = \{x_{-n} \ldots x_i \ldots x_{-1}\},\ x_i \in SW \end{cases} \tag{5}$$

Regarding the location of these samples in the decision space of M_s, the left side of Fig. 1 shows our intuition. The samples of $\mathbb{X}_{cu}$ and $\mathbb{X}_{wu}$ are close to the decision boundary of M_s, and the samples of $\mathbb{X}_{cc}$ and $\mathbb{X}_{wc}$ are far from the decision boundary of M_s. The accuracy of samples in $\mathbb{X}_{cc}$ calculated by M_s is recorded as $\mathbb{A}_{cc}$, and the accuracy of samples in $\mathbb{X}_{wc}$ calculated by M_s is recorded as $\mathbb{A}_{wc}$. Additionally, the accuracy of samples in $\mathbb{X}_{cu}$ calculated by M_s is recorded as $\mathbb{A}_{cu}$, and the accuracy of samples in $\mathbb{X}_{wu}$ calculated by M_s is recorded as $\mathbb{A}_{wu}$. These accuracies are encapsulated as the Primary Fingerprint of M_s, which is articulated as $\mathbb{F}_s = (\mathbb{A}_{cc}, \mathbb{A}_{cu}, \mathbb{A}_{wc}, \mathbb{A}_{wu})$.

Considering the duality of the above samples, the decision of M_s is correct for both and wrong for the remaining. Thus, $\mathbb{F}_s$ of M_s is equal to $(1, 1, 0, 0)$. Recognizing that the decision space of M_p exhibits a stronger similarity to that of M_s compared to M_i, the calculated fingerprint $\mathbb{F}_p$ of the pirated

model M_p shows a higher level of alignment with $\mathbb{F}_s$. In contrast, the calculated fingerprint $\mathbb{F}_i$ of the irrelevant model M_i shows a lower alignment level than $\mathbb{F}_s$.

Algorithm 1 Evolved Fingerprint Algorithm (EFA)

Require: $M_s, \mathbb{X}_{\_}, \lambda, \tau, \mu, \omega$
The source model M_s
Samples of any of the four primary type $\mathbb{X}_{\_}$
Learning rate λ
Maximum number of iterations τ
Probability boost factor μ
Distance scaling factor ω
Ensure: *Samples of any of the four evolved type* $\mathbf{X}_{\_}$
for x_i**in**$\mathbb{X}_{\_}$ **do**
  $\mathbf{p} \equiv \sigma(M_s(x_i))$
  $a = argmax(\mathbf{p}), \{1 \ldots a \ldots b \ldots K\}$
  $b = argmax(\mathbf{p}), \{1 \ldots a \ldots b \ldots K\} \setminus \{a\}$
  if $x_i \in \mathbb{X}_{cu} \cup \mathbb{X}_{wu}$ **then**
    $d = \mathbf{p}_a - \frac{1}{K}$
    $\mathbf{p}_a = \mathbf{clip}\,(\mathbf{p}_a, \omega, d)$
    $\ell_{efa} = -\log(\mathbf{p}_a)$
  else if $x_i \in \mathbb{X}_{cc} \cup \mathbb{X}_{wc}$ **then**
    $\ell_{efa} = -\left(\frac{\mathbf{p}_a - \mathbf{p}_b}{1 - \mathbf{p}_a + \mu}\right)$
  end if
  repeat
    $\widetilde{\mathbf{p}} = \sigma(M_s(\widetilde{x_i}))$
    $\nabla \ell_{efa} = \frac{\partial \ell_{efa}}{\partial \widetilde{x_i}}$
    $\widetilde{x_i} = \widetilde{x_i} - \lambda \nabla \ell_{efa}$
    $\tau - -$
  until $\tau = 0$
  $\mathbf{X}_{\_} \cup \widetilde{x_i}$
end for
return: $\mathbf{X}_{\_}$

Evolved Fingerprint. Aggressive model attacks may significantly change the decision space of M_s, thereby challenging the Primary Fingerprint. Specifically, $uncertain(cu, wu)$ samples near the decision boundary of M_s can change from correct to wrong in the decision space of M_p and vice versa. The duality of the samples *cu* and *wu* has changed. These can be seen as easily learned samples for $confident(cc, wc)$ of M_s, making them easy for M_p to master. Although the duality of *cc* and *wc* samples is difficult to fluctuate, their confidence levels may change. In order to eliminate the above-hidden dangers of Primary Fingerprint, we propose another novel algorithm: Evolved Fingerprint. Adding perturbations for the *cu* and *wu* samples makes them slightly away from the decision boundary while adding perturbations for the *cc* and *wc* samples increases their confidence.

Our proposed Evolved Fingerprint Algorithm (EFA) is detailed in Algorithm 1. Specifically, the categorical probability distribution $\mathbf{p}$ is defined as $\mathbf{p} = \sigma(M_s(x_i))$, where M_s represents the source model, and σ denotes the softmax activation function. The categories in the dataset are indexed as a consecutive sequence $1, \ldots, a, \ldots, b, \ldots, K$, where K is the total number of categories. For a given input $x_i \in \mathbb{X}_{cc} \cup \mathbb{X}_{cu} \cup \mathbb{X}_{wc} \cup \mathbb{X}_{wu}$, the softmax function σ produces the $\mathbf{p}$ over these categories. Within this distribution, a and b refer to the classes with the highest and second-highest probabilities, respectively.

For samples $x_i \in \mathbb{X}_{cu} \cup \mathbb{X}_{wu}$, the probability distance d is calculated as $d = \mathbf{p}_a - \frac{1}{K}$, where $\mathbf{p}_a$ represents the probability assigned to the most likely class a, and $\frac{1}{K}$ corresponds to the mean probability across all categories, serving as the decision boundary of the source model. The value of d quantifies the distance of x_i from this decision boundary. To prevent $\mathbf{p}_a$ from becoming excessively large and deviating significantly from the decision boundary, it is truncated according to Eq. (6).

$$\mathbf{clip}\,(\mathbf{p}_a, \omega, d) = \begin{cases} \mathbf{p}_a, & \mathbf{p}_a < \mathbf{p}_a + \omega d \\ \mathbf{p}_a + \omega d, & \mathbf{p}_a \geq \mathbf{p}_a + \omega d \end{cases} \tag{6}$$

In this way, our objective is to constrain the difference between the optimized $\mathbf{p}_a$ of the sample x_i and $\frac{1}{K}$ within the range between d and ωd. Here, ω serves as a distance scaling factor, with larger values recommended when d is small to ensure an appropriate adjustment range.

$$\ell_{efa} = -\log(\mathbf{clip}\,(\mathbf{p}_a, \omega, d)) \tag{7}$$

In Eq. (7), we take the logarithm of the result from Eq. (6) and negate it, defining the outcome as ℓ_{efa}. By minimizing ℓ_{efa} through optimization, $\mathbf{p}_a$ becomes slightly larger, indicating that the sample $x_i \in \mathbb{X}_{cu} \cup \mathbb{X}_{wu}$ moves marginally further away from the decision boundary.

$$\ell_{efa} = -\left(\frac{\mathbf{p}_a - \mathbf{p}_b}{1 - \mathbf{p}_a + \mu}\right) \tag{8}$$

When the sample $x_i \in \mathbb{X}_{cc} \cup \mathbb{X}_{wc}$, the objective function is defined in Eq. (8). Due to the application of a negation operation to the loss function, the optimization goal can be reformulated as maximizing the numerator $\mathbf{p}_a - \mathbf{p}_b$ while simultaneously minimizing the denominator $1 - \mathbf{p}_a + u$, where u is a fixed constant. As a result, the optimization process ultimately drives the value of $\mathbf{p}_a$ to increase further. Following this procedure, every sample $x_i \in \mathbb{X}_{cc} \cup \mathbb{X}_{cu} \cup \mathbb{X}_{wc} \cup \mathbb{X}_{wu}$ undergoes perturbation after τ iterations with a learning rate of λ.

The right side of Fig. 1 illustrates our intuition regarding positioning these optimized samples within the decision space of M_s. Similarly to the Primary Fingerprint, we focus on black-box forensic scenarios in this context. Specifically, we evaluate the performance of M_s in four evolved sample types, encapsulating these metrics into the Evolved Fingerprint, denoted as $\mathbf{F}_s = (\mathbf{A}_{cc}, \mathbf{A}_{wc}, \mathbf{A}_{cu}, \mathbf{A}_{wu})$. Unlike the Primary Fingerprint, where the original samples are used, the current approach involves perturbed samples, with the predictions of M_s serving as their labels. Consequently, these perturbed samples of the four types are guaranteed to be correctly classified by M_s. This ensures that the Evolved Fingerprint $\mathbf{F}_s$ of M_s simplifies to $(1, 1, 1, 1)$, reflecting perfect classification for all types in this framework.

C. Model Fingerprint Matching

The source model M_s and the pirated model M_p share similar decision spaces, resulting in high accuracy across all four types. However, the irrelevant model M_i does not

exhibit such parallels. Consequently, we expect that most samples of the four types, whether they are $\mathbb{X}_{-}$ or $\mathbf{X}_{-}$, will demonstrate behavior more consistent with M_s when tested on M_p compared to M_i.

$$\|\mathbf{F}_s - \mathbf{F}_p\|_1 \ll \|\mathbf{F}_s - \mathbf{F}_i\|_1 \tag{9}$$

As shown in Eq. (9), the similarity between different models can be assessed by calculating the Manhattan norm between their fingerprints. Ideally, the distance between $\mathbf{F}_p$ and $\mathbf{F}_s$ is significantly less than the distance between $\mathbf{F}_i$ and $\mathbf{F}_s$. Here, the calculation of the Evolved Fingerprint is used as an example, while the Primary Fingerprint is also calculated similarly.

D. Theoretical Insights Into Fingerprints

To further analyze the theoretical advantages of the Evolved Fingerprint over the Primary Fingerprint, we consider a dataset containing K categories, where it is guaranteed that $\mathbf{p}_a > \mathbf{p}_b$ and $\mathbf{p}_a > \frac{1}{K}$. The loss defined in Eq. (7) and Eq. (8) are designed to perturb the samples in the Primary Fingerprint by minimizing the corresponding loss values. The optimization process increases the value of $\mathbf{p}_a$. To analyze the impact of this increase, we first assume the case where $K = 2$, such that the probabilities for the two categories are $\mathbf{p}_a$ and $1 - \mathbf{p}_a$, respectively. The information entropy of this binary classification scenario is expressed as Eq. (10):

$$H(\mathbf{p}_a) = -\mathbf{p}_a \log \mathbf{p}_a - (1 - \mathbf{p}_a) \log(1 - \mathbf{p}_a) \tag{10}$$

and its derivative is:

$$H'(\mathbf{p}_a) = -log\mathbf{p}_a + log(1 - \mathbf{p}_a) = log\left(\frac{1 - \mathbf{p}_a}{\mathbf{p}_a}\right) \tag{11}$$

Based on the previously posited condition that $\frac{1-\mathbf{p}_a}{\mathbf{p}_a} < 1$, which further leads to $H'(\mathbf{p}_a) < 0$. Consequently, the information entropy $H(\mathbf{p}_a)$ decreases monotonically as $\mathbf{p}_a$ increases. This result demonstrates that, after perturbation, the model's information entropy concerning the probability distribution of the perturbed samples is reduced.

The same conclusion can be drawn by extending this analysis to the case where $K > 2$. Specifically, the monotonicity of the entropy function ensures that an increase in $\mathbf{p}_a$ reduces the overall uncertainty of the model's output distribution. This reduction in entropy further highlights the stability and robustness of perturbed fingerprints, reinforcing their effectiveness in preserving discriminative model characteristics in various scenarios.

IV. Threats to IP Protection

IP protection in deep models faces various threats from adversaries with varying capabilities. To comprehensively assess these risks, we establish a structured threat landscape: first, by profiling attacker capabilities based on their access to data and knowledge of the model; second, by evaluating six IP removal attacks that exploit these capabilities; and third, after successfully removing IP through these attacks, deploying advanced adversarial strategies to obfuscate the provenance of the pirated model and evade forensic detection mechanisms. The following subsections elaborate on this framework, starting with classifying attackers based on their awareness of data distribution and model transparency. We then benchmark IP removal attacks about these differing capability levels. Finally, the framework addresses sophisticated adversaries who employ detection and evasion tactics across domains such as CV, NLP, and BCI. This hierarchical analysis connects threat modeling with practical countermeasures.

A. Capabilities of Attackers

We profile attackers' capabilities from two angles: data and model. We classify the attackers' data into three levels according to the quality of their data.

- Labeled in-distribution (LID): Attackers who have access to a sufficient amount of labeled in-distribution data, meaning the labeled data is drawn from the same distribution as the source model's training data.
- Unlabeled in-distribution (UID): Attackers with access to a sufficient amount of unlabeled in-distribution data, where the data is from the same distribution but lacks labels.
- Labeled out- of-distribution (LOD): Attackers who possess labeled data from a distribution different from the source model. While they have access to labels, this data may not directly reflect the characteristics of the original training data.

As for the model aspect, we classify attackers into two levels based on their knowledge of the source model.

- White-box (WB): Attackers with comprehensive access to the source model, including its architecture, parameters, and training data. With this knowledge, attackers can fully understand the model's inner workings.
- Black-box (BB): Attackers with limited access, typically only to the source model's API. They can query the model and receive predictions but do not have direct knowledge of its internal structure.

B. IP Removal Attacks

Our evaluation of the proposed method's efficacy involves six distinct IP removal attacks: Fine Tune (FT), Fine Prune (FP), Model-extraction Labels (ML), Model-extraction Probabilities (MP), Model-extraction Adversarial (MA), and Transfer-Learning (TL).

- FT attack involves updating the weights of the model's final layer or all layers. Attackers with LID and WB can exploit this method.
- FP is a fine pruning strategy introduced in [19], which selectively removes certain model parts to reduce complexity while maintaining functionality. Attackers with LID and WB capabilities can use this approach.
- ML assumes attackers have access only to the predicted labels of the source model without any knowledge of the internal model parameters or structure. This is a common threat scenario for attackers with UID and BB.
- MP assumes attackers can access the model's predicted probability distributions rather than just the final labels.

This provides richer information for model extraction and is particularly effective for attackers with UID and BB.
- MA is performed by an attacker with UID and BB, using adversarial samples generated by techniques such as Projected Gradient Descent (PGD) to destroy the decision boundary of the model.
- TL involves using knowledge from one domain to accelerate learning in another, typically by fine-tuning the source model for a target domain. Attackers with LOD and BB may utilize this technique.

C. IP Detection and Evasion Attacks

A savvy attacker may adopt a multi-step strategy to eliminate IP from the source model. Initially, the attacker applies one of the six IP removal attacks to obtain the pirated model. Then, the attacker deploys it and exposes its API for service access, and an intermediate link is established between the API and the pirated model. This scenario is called an IP detection and evasion attack; the attacker seeks to obscure the origin of the pirated model and utilizes advanced adversarial strategies to obfuscate its provenance and evade forensic detection mechanisms.

- Query Attack is designed for CV tasks. Inspired by [20], we train an auto-encoder to pre-process all samples, whether they are normal samples or samples with model IP forensics capabilities. This step brings uncertainty to the forensics of the model IP.
- Synonym Attack is designed for NLP tasks. Inspired by [4], the input text undergoes preprocessing, where adjectives are replaced by their synonyms. This modification can impair the effectiveness of model IP forensics.
- Input Smooth attack is designed for BCI tasks and is inspired by [21]. This approach involves applying smoothing and noise reduction techniques to process EEG signals, which can degrade the quality of the original data while maintaining essential signal features.

V. Experimental Setup

This section presents the experimental objectives, data preparation, model architectures, comparison methods, evaluation metrics, hyper-parameters of EFA, and implementation details.

A. Experimental Objectives

In the abstract, we emphasize the advantages of the proposed methodology. Here, the experimental objectives are provided from efficiency, comprehensiveness, and generalization perspectives.

- Efficiency: We dedicate Section VI-C to experimentally validate the efficiency of our proposed methods. Integrating IP protection into the model, i.e., obtaining the fingerprint, introduces a computational overhead, which we define as generation time. Similarly, extracting IP evidence from a model incurs additional computational costs, referred to as inference time. We introduce the Temporal Fusion Metric (TFM) to balance these two factors. The experimental results across all four datasets consistently show that our proposed Primary and Evolved Fingerprint achieve lower TFM values, outperforming other methods. This demonstrates the superior efficiency of our approach.
- Comprehensiveness: Previous fingerprinting methods have mainly focused on modeling decision boundaries. In contrast, our approach introduces fingerprint samples that incorporate duality and confidence to enhance comprehensiveness. Furthermore, while prior IP protection strategies have concentrated on evaluating robustness against IP removal attacks, limited efforts have been made to assess their effectiveness against IP detection and evasion attacks. IP detection involves identifying the presence of embedded IP within a deep model. IP evasion refers to adversarial strategies that bypass IP verification by perturbing or corrupting input samples. In our study, we test the effectiveness of the proposed method and other methods in a scenario where an attacker steals the source model using IP removal attacks. The results demonstrate that IP detection and evasion attacks are both highly disruptive and easy to implement, underscoring the necessity of evaluating the stability of IP protection methods against such threats. Our methods also show the advantages of comprehensiveness against IP removal attacks, IP detection, and evasion attacks.
- Generalization: Another crucial aspect we address is the generalization capability of model IP protection. Existing methods focus primarily on CV tasks and are tightly coupled to image data, limiting their applicability to other domains and potentially resulting in suboptimal performance. To overcome this limitation, we propose the Primary and Evolved Fingerprint, which rely on the decision-making information of samples concerning the converged source model. This information, defined as duality and conviction in our paper, is independent of the modality of the data. Our experiments provide strong evidence of the cross-domain effectiveness of our methods, demonstrating their robustness and applicability across datasets with varying modalities, including CV, NLP, and BCI. These findings highlight the versatility and generalization of our approach in protecting model IP across various domains.

B. Data Preparation

We evaluate our proposed approach in three domains: CV, NLP, and BCI, involving tasks such as image classification, text classification, and sentiment analysis. For the CV task, we use CIFAR10, CIFAR100 [22], and CIFAR10C [23]. CIFAR10 consists of 10 classes with 50,000 train images and 10,000 test images, while CIFAR100 comprises 100 classes with 600 color images each, all with dimensions of 32×32. CIFAR10C is an extended version of CIFAR10 that introduces disturbances such as blur, noise, and brightness variations. In the

NLP task, we employ the THUCNews [24] and Onlineshop.[2] THUCNews is a Chinese news dataset covering 65,000 news articles across 10 categories (sports, finance, real estate, home, education, technology, fashion, current affairs, games and entertainment). Onlineshop contains over 60,000 Chinese reviews spanning 10 categories (books, tablets, mobile phones, fruits, shampoos, water heaters, milk, clothes, computers, hotels). We use DEAP [25] and MAHNOB-HCI [26] datasets for the BCI task, including EEG and other physiological signals recorded during emotional stimulation experiments. DEAP focuses on rating the emotional aspects of 40 one-minute music videos, while MAHNOB-HCI captures EEG and physiological signals from participants rating the mood of 20 music videos. In our BCI analysis, we select only the EEG signal and its valence label from DEAP and MAHNOB-HCI for classification validation.

C. Model Architectures

In our cross-domain IP protection framework, we establish task-specific experimental settings across three modalities:

- In the CV image classification task, the source model is a VGG16 [27] pre-trained on CIFAR10, which is adapted for target datasets such as CIFAR10, CIFAR100, or CIFAR10C by replacing the final fully connected layer with a modified version. Pirated models include architectures like VGG13, ResNet18 [28], DenseNet121 [29], and MobileNetV2 [30], whose weights are derived from IP removal attacks applied to the source model, as detailed in Section IV-B. Irrelevant models refer to structurally identical counterparts trained independently from scratch, with no involvement of IP attacks.
- In the NLP text classification task, the source model is a bert-base-chinese [31] transformer pre-trained on THUCNews and fine-tuned with task-specific classification layers. The pirated and irrelevant models consist of four architectures (TextCNN [32], TextRNN [33], DPCNN [34], and TextRCNN [35]), distinguished by the origin of their weights. The pirated versions inherit transformed weights through IP removal attacks described in Section IV-B, while the irrelevant versions are trained conventionally from scratch without any IP involvement.
- In the BCI emotion classification task, the source model is EEG-Conformer [36], a transformer-based architecture pre-trained on the DEAP dataset for EEG signal processing. The pirated and irrelevant models include ShallowNet and DeepNet [37], with key differences in the origins of their weights. The pirated models obtain their weights through IP removal attacks on the source model, while the irrelevant models are independently trained from scratch, with no dependency on the source model.

All architectural implementations, including layer configurations and hyper-parameters, are detailed in our publicly available code. This systematic design enables controlled evaluation of IP persistence across heterogeneous model architectures and data modalities.

[2]Onlineshop is available here: https://github.com/SophonPlus/ChineseNlpCorpus/blob/master/datasets/online_shopping_10_cats/online_shopping_10_cats.zip

D. Comparing Methods

Our analysis covers several IP protection methods: IPGuard [12], which uses decision boundary samples for fingerprints; CAE [16], which employs transferable adversarial examples; and both Trigger [38] and Trigger-EEG [8], which are retrained using randomly labeled samples. SAC [15] focuses on pairwise sample relationships and offers two sub-methods, SAC-w and SAC-m, which enhance protection strategy diversity. These methodologies are amongst the most effective in the domain of model protection.

E. Evaluation Metrics

We assess the local effectiveness of our IP protection methods by comparing the fingerprints of the pirated and irrelevant models to the source model using the Area Under the Curve (AUC). Additionally, we introduce the Area of Polygon Boundary (APB) as a comprehensive metric to gauge the overall efficacy against IP infringements. The APB is derived from AUC values when each IP protection method encounters multiple IP removal attacks, providing a quantifiable performance measure.

F. Hyper-Parameters of Efa

We propose two innovative algorithms, namely Primary Fingerprint and Evolved Fingerprint. Primary Fingerprint is an efficient algorithm that does not require any training process. On the other hand, Evolved Fingerprint refines perturbations based on the samples obtained from the Primary Fingerprint to enhance the stability of the model fingerprint. Our Evolved Fingerprint Algorithm (EFA) includes the following hyper-parameters: n is the number of samples from each fingerprint component, the learning rate λ, the number of iterations τ, μ is used to assist in perturbing confident samples, and the distance scaling factor ω is used to control uncertain samples away from the decision boundary. For CV task on the CIFAR10, we set the hyper-parameters as follows: $n = 80$, $\lambda = 0.001$, $\tau = 10$, $\mu = 1\times10^{-5}$, and $\omega = 2$. On the CIFAR100, the corresponding hyper-parameters are: $n = 20$, $\lambda = 0.01$, $\tau = 20$, $\mu = 1 \times 10^{-5}$, and $\omega = 2$. Since in the NLP task, we do not use optimization and derivation methods (IPGuard, Our Evolved Fingerprint), only $n = 500$ involved in the Primary Fingerprint is given here. In the BCI task, the hyper-parameters are set as follows: $n = 100$, $\lambda = 0.05$, $\tau = 50$, $\mu = 1 \times 10^{-5}$, and $\omega = 10000$. A larger ω is set in the BCI task due to the binary nature of our task, where the class probabilities for uncertain samples are too close, requiring a larger ω to move these samples slightly away from the decision boundary. A growing number of methods attempt to improve the model's performance with a limited number of samples [39]. Therefore, we also set the sample size to a relatively small value.

TABLE I

THE MODELS REQUIRED FOR EACH EXPERIMENT IN THE FOUR DATASETS AND THEIR QUANTITIES ARE LISTED. MODELS MARKED WITH AN ASTERISK (*) DENOTE PRUNED VERSIONS. THE CONTENT IN PARENTHESES ON THE FIRST LINE OF THE RECORD INDICATES THE SOURCE MODEL. THE TRANSFER LEARNING PARADIGM IS REPRESENTED AS (A>B)

Removal attacks	CIFAR10 (VGG16)	CIFAR100 (VGG16)	DEAP (Conformer)	THUCNews (BERT)
FT	20 VGG16	20 VGG16	20 Conformer	20 BERT
FP	10 VGG16*	10 VGG16*	10 Conformer*	10 BERT*
ML	5 VGG13 5 ResNet18 5 DenseNet121 5 MobileNetV2	5 VGG13 5 ResNet18 5 DenseNet121 5 MobileNetV2	10 ShallowNet 10 DeepNet	5 DPCNN 5 TextCNN 5 TextRCNN 5 TextRNN
MP	5 VGG13 5 ResNet18 5 DenseNet121 5 MobileNetV2	5 VGG13 5 ResNet18 5 DenseNet121 5 MobileNetV2	10 ShallowNet 10 DeepNet	5 DPCNN 5 TextCNN 5 TextRCNN 5 TextRNN
MA	5 VGG13 5 ResNet18 5 DenseNet121 5 MobileNetV2	5 VGG13 5 ResNet18 5 DenseNet121 5 MobileNetV2	10 ShallowNet 10 DeepNet	5 DPCNN 5 TextCNN 5 TextRCNN 5 TextRNN
TL	5 ResNet18 5 VGG16 (CIFAR10>CIFAR10C)	5 ResNet18 5 VGG16 (CIFAR100>CIFAR10)	10 Conformer (DEAP>MAHNOB-HCI)	10 BERT (THUCNews>Onlineshop)

G. Implementation Details

In this work, we implement the proposed method using PyTorch and train our models on NVIDIA Tesla V100 GPUs. We adopt the same experimental setup as [15] for the CV task. For the NLP task, we split the THUCNews dataset into two parts: one for training the source model and the other for simulating IP attacks. The Onlineshop dataset implements transfer learning attacks on the source model trained with THUCNews. Following the experimental setup of SAC, each IP attack in every task simulates multiple attacks, resulting in multiple piracy models to mitigate experimental result contingencies. For the BCI task, we divided the DEAP dataset into two parts: one for training and tuning the source model and the other for simulating IP attacks. Transfer learning attacks are performed using the MAHNOB-HCI dataset on the source model trained with DEAP. As shown in Table I, we have enumerated the models required for each attack on the four datasets and their corresponding quantities. A model with * denotes the pruned version, while those as (A>B) indicate the transfer learning paradigm.

VI. EXPERIMENTAL RESULTS

In the experimental section, the effectiveness of the proposed method is measured in four parts. First, the methods' performance is verified on IP removal attacks. Second, the methods' performance on IP detection and evasion attacks is evaluated. Third, comparative methods are evaluated in terms of efficiency. Finally, the results of the ablation experiments on fingerprint components are given.

A. Identifiable Traits Post IP Removal

To validate the effectiveness of Primary Fingerprint and Evolved Fingerprint, we compare them with existing methods, including IPGuard, CAE, Trigger, SAC-w, and SAC-m.

TABLE II

THE AUC FOR SEVEN IP PROTECTION METHODS AGAINST SIX IP REMOVAL ATTACKS ON CIFAR10. THE VALUE WITH BOLD AND '__' INDICATE OPTIMAL AND SUBOPTIMAL PERFORMANCE RESPECTIVELY

M\R	FT	FP	ML	MP	MA	TL	Average
IPGuard	1.00	0.88	0.56	0.65	0.37	0.38	0.69
CAE	1.00	0.97	0.90	0.96	0.91	0.83	0.94
Trigger	1.00	1.00	0.66	0.82	0.73	1.00	0.88
SAC-w	1.00	1.00	1.00	0.95	1.00	1.00	<u>0.99</u>
SAC-m	1.00	1.00	1.00	0.95	0.91	1.00	0.98
Primary(Ours)	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**
Evolved(Ours)	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**

Table II shows the AUC of the seven IP protection methods against six IP removal attacks in CIFAR10. The bold values indicate optimal performance, while values with '__' indicate suboptimal performance. This convention is also followed in the subsequent tables as well.

IPGuard, which focuses on samples near the decision boundary, proves vulnerable to IP removal attacks that significantly alter the boundary. Trigger's distributed samples, confidently recognized by the source model, highlight the consideration of uncertain and confident samples. CAE's fingerprinting samples, derived from supervised learning involving pirated and irrelevant models, display commendable robustness. SAC-w and SAC-m, which emphasize pairwise relationships in source model outputs, are robust against IP removal attacks. However, SAC-w relies on irrelevant models, and SAC-m's fingerprints, through image stitching, are more prone to detection. Our Primary and Evolved Fingerprint methods exhibit robustness against almost all IP removal attacks, eliminating the need for specific processes. To further evaluate the proposed fingerprinting methods, we conduct a parallel experiment on CIFAR100, adjusting the

TABLE III

THE AUC FOR SEVEN IP PROTECTION METHODS AGAINST SIX IP REMOVAL ATTACKS ON CIFAR100. THE VALUE WITH BOLD AND '_' INDICATE OPTIMAL AND SUBOPTIMAL PERFORMANCE RESPECTIVELY

M\R	FT	FP	ML	MP	MA	TL	Average
IPGuard	0.81	0.62	0.45	0.42	0.11	0.00	0.40
CAE	1.00	0.79	0.69	0.65	0.63	0.34	0.68
Trigger	0.81	0.92	0.63	0.61	0.77	0.49	0.71
SAC-w	0.83	0.83	0.66	0.66	0.65	0.00	0.61
SAC-m	0.80	0.63	0.68	0.70	0.82	0.00	0.61
Primary(Ours)	1.00	0.87	1.00	1.00	1.00	0.50	<u>0.90</u>
Evolved(Ours)	1.00	0.89	1.00	1.00	1.00	0.62	**0.92**

TABLE IV

THE AUC FOR FOUR IP PROTECTION METHODS AGAINST FIVE IP REMOVAL ATTACKS ON THUCNEWS. THE VALUE WITH BOLD AND '_' INDICATE OPTIMAL AND SUBOPTIMAL PERFORMANCE RESPECTIVELY. IPGUARD AND EVOLVED FINGERPRINT ARE EXCLUDED HERE

Method\Removal	FT	FP	ML	MP	TL	Average
Trigger-Text	1.00	0.69	0.42	0.43	0.00	<u>0.51</u>
SAC-w	0.00	0.00	0.47	0.45	0.16	0.21
SAC-m	0.00	0.00	0.36	0.36	0.78	0.30
Primary(Ours)	1.00	1.00	0.76	0.77	0.52	**0.81**

TABLE V

THE AUC FOR SIX IP PROTECTION METHODS AGAINST SIX IP REMOVAL ATTACKS ON DEAP. THE VALUE WITH BOLD AND '_' INDICATE OPTIMAL AND SUBOPTIMAL PERFORMANCE RESPECTIVELY. IPGUARD IS EXCLUEDED HERE

M\R	FT	FP	ML	MP	MA	TL	Average
CAE	0.57	0.73	0.56	0.56	0.55	0.24	0.54
Trigger-EEG	1.00	0.80	0.85	0.84	0.76	1.00	**0.88**
SAC-w	0.67	0.84	0.44	0.69	0.84	0.97	0.74
SAC-m	0.44	0.62	0.83	0.65	0.42	0.59	0.59
Primary(Ours)	0.73	0.90	0.93	0.94	0.93	0.44	<u>0.81</u>
Evolved(Ours)	0.80	0.90	0.93	0.94	0.92	0.78	**0.88**

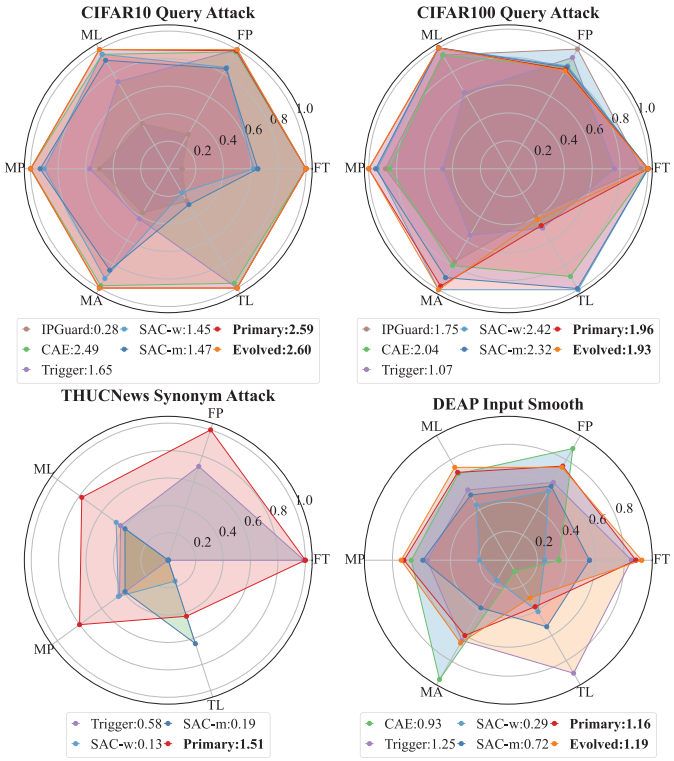

Fig. 2. Various IP protection methods are assessed for domain-adaptive IP detection and evasion attacks across four datasets. A higher value in legend indicates greater IP identification post-attack.

transfer learning from CIFAR100 to CIFAR10. The results in Table III show that both Primary Fingerprint and Evolved Fingerprint consistently outperform other methods in various IP removal attacks. This highlights the robustness and stability of our methods, even when dealing with diverse categories.

We also compare the effectiveness of IP protection methods in NLP tasks. To ensure the integrity and fairness of our experiments, we acknowledge that IPGuard and Evolved Fingerprint may not be suitable for handling discrete token samples due to their derivation process. Therefore, we evaluate the efficacy of four alternative IP protection methods against a range of five IP removal attacks. This approach is applied consistently in subsequent NLP tasks. Trigger-Text is implemented by randomly replacing input tokens and as an adaptation scheme for NLP tasks. The results detailed in Table IV demonstrate that our proposed Primary Fingerprint continues to outperform other methods. The mechanism employed by SAC-w yields only two samples here, which may not fully capture the fingerprints of the source model. In contrast, SAC-m uses the adaptation of sentence fragment splicing. However, experimental results indicate that SAC-m is unsuitable for text samples. Although our method is not optimal regarding TL, the average AUC significantly exceeds other methods.

Furthermore, we conduct comparative experiments for the BCI task, with the results shown in Table V. IPGuard, which relies on the third category of samples, is excluded from the binary classification DEAP task, a convention we also maintain for subsequent BCI tasks. The trigger's stable performance across both tasks can be attributed to two factors: the model was fine-tuned using the trigger set, and these samples are not confined to the decision boundary. Although CAE generates fingerprint samples through supervised training, its efficiency falls short. The lower performance of SAC-w and SAC-m is due to the reduction in categories when transitioning from the CV 10-class classification task to the binary BCI task. This reduction limits the representation of fingerprint samples in the pairwise output, affecting the ability to accurately distinguish pirated and irrelevant models. In contrast, our approach, with the Primary Fingerprint, demonstrates superior versatility, achieving an average AUC that exceeds that of CAE and SAC-m. Although Trigger-EEG's average AUC is comparable to Evolved Fingerprint's, it suffers from invasiveness.

B. Identifiable Traits Post IP Detection and Evasion

We simulate a scenario in which attackers first employ any IP removal attack to steal the source model. Subsequently, domain-specific IP detection and evasion attacks are applied to forensic samples from various IP protection methods. We then evaluate the effectiveness of these samples in identifying

pirated and irrelevant models. Fig. 2 illustrates the AUC of each IP protection method after undergoing multiple-step attacks, represented by solid coordinate points. The enclosed polygonal area denoted as APB, reflects the overall efficacy of each IP protection method against threats, with a higher APB indicating stronger protection capabilities.

In the upper-left of Fig. 2, it is evident that our proposed methods consistently outperform existing methods in countering Query Attack on CIFAR10. Trigger and CAE maintain a stable APB in this scenario, while SAC-w and SAC-m exhibit a lower APB against Query Attack. In the upper-right of Fig. 2, our proposed methods continue to effectively resist Query Attack on CIFAR100 compared to other methods. The superior performance of Primary Fingerprint and Evolved Fingerprint in countering Query Attack can be attributed to the minimal or nonexistent perturbations applied to the samples.

Drawing inspiration from related work [4], we devise an IP detection and evasion attack tailored for NLP tasks. Specifically, we posit that the attacker uses adjective substitution on the query sample to impede the victim's forensic analysis. As shown in the lower-left of Fig. 2, our proposed Primary Fingerprint remains more effective than the other three protection methods, even after the Synonym Attack. In the BCI domain, we experimentally evaluate the impact of the Input Smooth attack on various IP protection methods. As shown in the lower-right of Fig. 2, the Evolved Fingerprint demonstrates better stability than the Primary Fingerprint in APB. However, its performance is weaker relative to transfer learning, which can be attributed to the challenging nature of cross-device transfer learning in BCI [40]. The other methods consistently generate lower APBs than Evolved Fingerprint and display considerable variations, regardless of whether input smooth is applied.

C. Efficiency of IP Protection Methods

Efficiency in real-world scenarios becomes an important metric for evaluating methods [41]. An ideal IP protection method should be both effective and efficient. As shown in Fig. 3, we measure the generation time, inference time, and the TFM of various IP protection methods in multi-domain.

The generation time represents the duration needed to create fingerprinting or watermarking samples for the source model. In contrast, the inference time refers to the time required to distinguish pirated models from irrelevant models using these fingerprints or watermarks. TFM integrates the generation time and the inference time, while AVG (the average AUC) recorded in Tables II, III, IV, V quantifies the method's coordination time, as expressed by Eq. (12). For intuitive understanding, a method that corresponds to the point near the lower left corner is more effective.

$$TFM = \log_{10}\left(\frac{generation\ time + inference\ time}{AVG}\right) \quad (12)$$

The findings reveal minor variations in the inference time yet a pronounced disparity in generation time among IP protection methods. Regarding generation time, IPGuard, CAE, and SAC-w show longer periods—the prolonged generation time of IPGuard results from its random selection of initial samples for optimization. For CAE and SAC-w, the generation

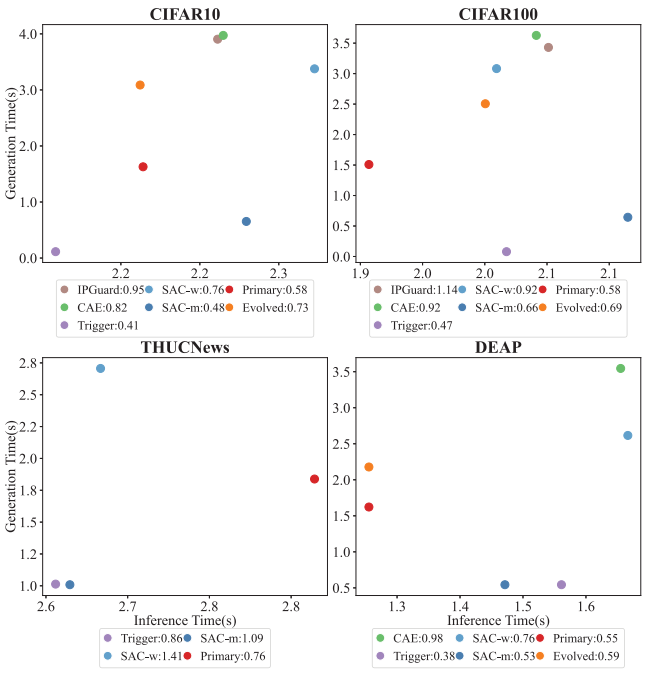

Fig. 3. The results of different IP protection methods regarding generation time and inference time. Values in the legend represent the corresponding TFM, where smaller values indicate higher efficiency.

time is affected by the complexity of the model, which is particularly noticeable when using bert-base-chinese. Conversely, Trigger, SAC-m, Primary Fingerprint, and Evolved Fingerprint have a shorter generation time. The Trigger and SAC-m generate efficiently by adding noise to the original samples. Primary Fingerprint quickly synthesizes fingerprints based on the inference from the train set, while Evolved Fingerprint, a refined version of Primary Fingerprint, introduces minimal performance overhead. It is important to mention that Trigger's watermarking involves retraining the source model, which is time-intensive. Similarly, CAE and SAC-w require training when generating fingerprinting samples from pirated and irrelevant models. These lengthy processes were excluded from our analysis. In essence, the TFM serves as a metric for assessing the efficiency of IP protection methods. As illustrated in Fig. 3, the values depicted in the legend correspond to the TFM for each method. A lower score indicates that the protection method is more efficient. Consequently, our primary and evolved fingerprinting techniques exhibit efficiency across different domains, surpassing the performance of alternative methods.

D. Ablation Study of Fingerprint Components

To further validate the effectiveness of our method, we conduct an ablation experiment on the components of the model fingerprint. The experimental results are presented in Table VI, recording the average AUCs of the two proposed fingerprint algorithms against IP removal attacks in multi-domain tasks. Among them, we denote *both*, *three*, and *four* as the two-components fingerprint (cu, wu), three-components fingerprint (wc, cu, wu), and four-components fingerprint (cc, wc, cu, wu), respectively, where the values in the table indicate the average AUC of the current fingerprint against IP removal attacks.

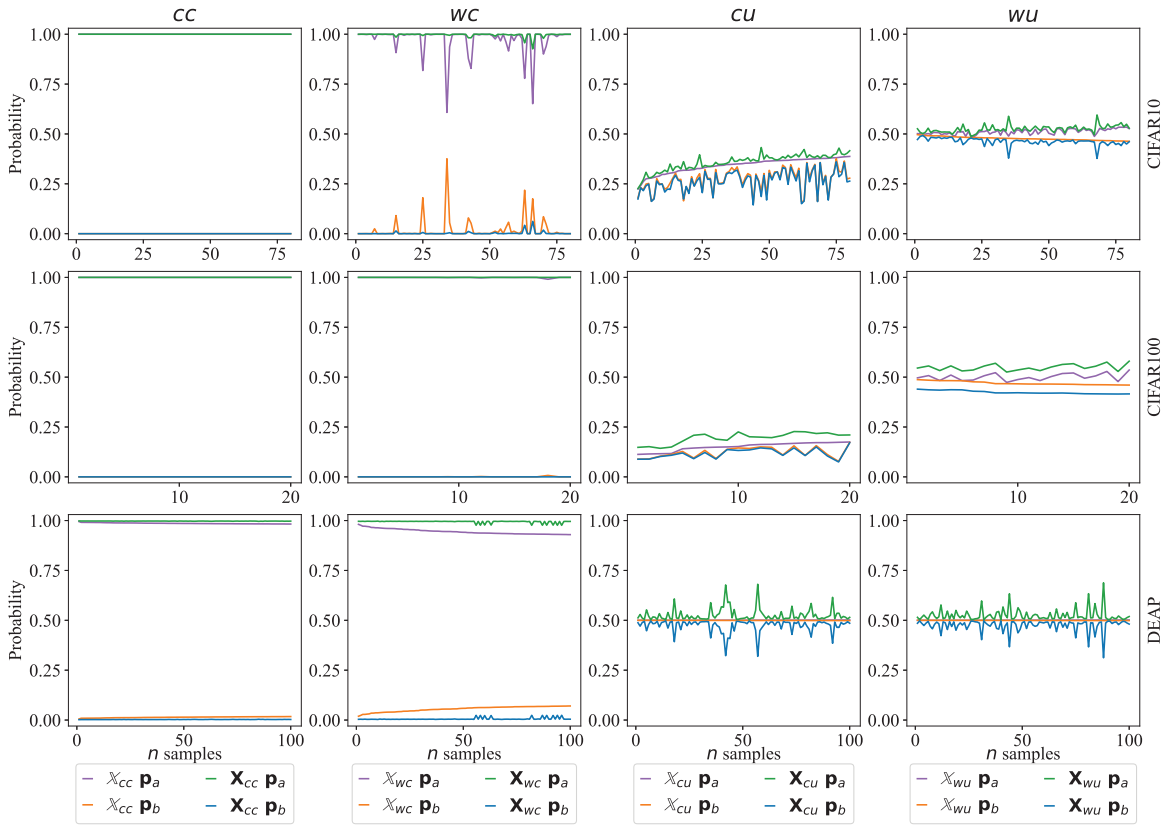

Fig. 4. The line chart displays the $\mathbf{P}_a$ and $\mathbf{P}_b$ for four types of both primary and evolved fingerprints, where $\mathbf{P}_a$ and $\mathbf{P}_b$ represent the activation probabilities of the source model. Due to the absence of Evolved Fingerprint data for THUCNews, results are only shown for CIFAR10, CIFAR100, and DEAP.

TABLE VI

THE AVERAGE AUCS OF PRIMARY FINGERPRINT AND EVOLVED FINGERPRINT WITH DIFFERENT COMPONENTS AGAINST IP REMOVAL ATTACKS IN MULTI-DOMAIN, WITH HIGHER VALUES DENOTING STRONGER RESILIENCE. HERE, THE BOTH, THREE, AND FOUR DENOTE (cu, wu), (wc, cu, wu), AND (cc, wc, cu, wu), RESPECTIVELY

Fingerprint components		*both*	*three*	*four*
CIFAR10	Primary	0.64	1.00	1.00
	Evolved	0.65	1.00	1.00
CIFAR100	Primary	0.83	0.82	0.90
	Evolved	0.90	0.92	0.92
THUCNews	Primary	0.70	0.80	0.81
DEAP	Primary	0.80	0.81	0.81
	Evolved	0.84	0.86	0.88

The results show that in the case of *both*, that only considers samples with uncertain decision boundaries, leading to relatively poor effectiveness in all tasks. However, *three* and *four* incorporate confident samples in addition to uncertain samples, thus improving the robustness of the fingerprints. Additionally, we notice that the Evolved Fingerprint shows a greater performance improvement with increasing fingerprint components compared to the Primary Fingerprint. In conclusion, our methods, which integrate confident and uncertain samples as fingerprints, prove robust and effective against diverse IP attacks in multi-domain tasks.

VII. INTERPRETABILITY ANALYSIS

In this section, an analysis will be performed to determine the efficacy of the proposed method from the standpoint of interpretability. To gain a more intuitive understanding of the differences between Primary Fingerprint and Evolved Fingerprint, we provide the analysis of the activation probability distributions of the fingerprinting samples by the source model. We then examine the distribution of four types of fingerprinting samples within the source model's decision space and present the topological map of EEG fingerprinting samples.

A. Probability Gap of Fingerprinting Samples

The Evolved Fingerprint is obtained by modifying the probability distribution of the Primary Fingerprint concerning model activation through EFA. We examine the change in probabilities for the first and second categories, denoted as $\mathbf{p}_a$ and $\mathbf{p}_b$. Fig. 4 displays these probabilities, with the number of samples n on the horizontal axes and the probability on the vertical axes. In the first and second columns of Fig. 4, we observe that since the source model is confident about the samples *cc* and *wc*, the difference between $\mathbf{p}_a$ and $\mathbf{p}_b$ is relatively large. For the *wc* samples in CIFAR10 and DEAP, the EFA significantly increases this difference between $\mathbf{p}_a$ and $\mathbf{p}_b$. In the third and fourth columns of Fig. 4, where the source model is uncertain about the *cu* and *wu* samples, the difference between $\mathbf{p}_a$ and $\mathbf{p}_b$ is minimal. After applying EFA,

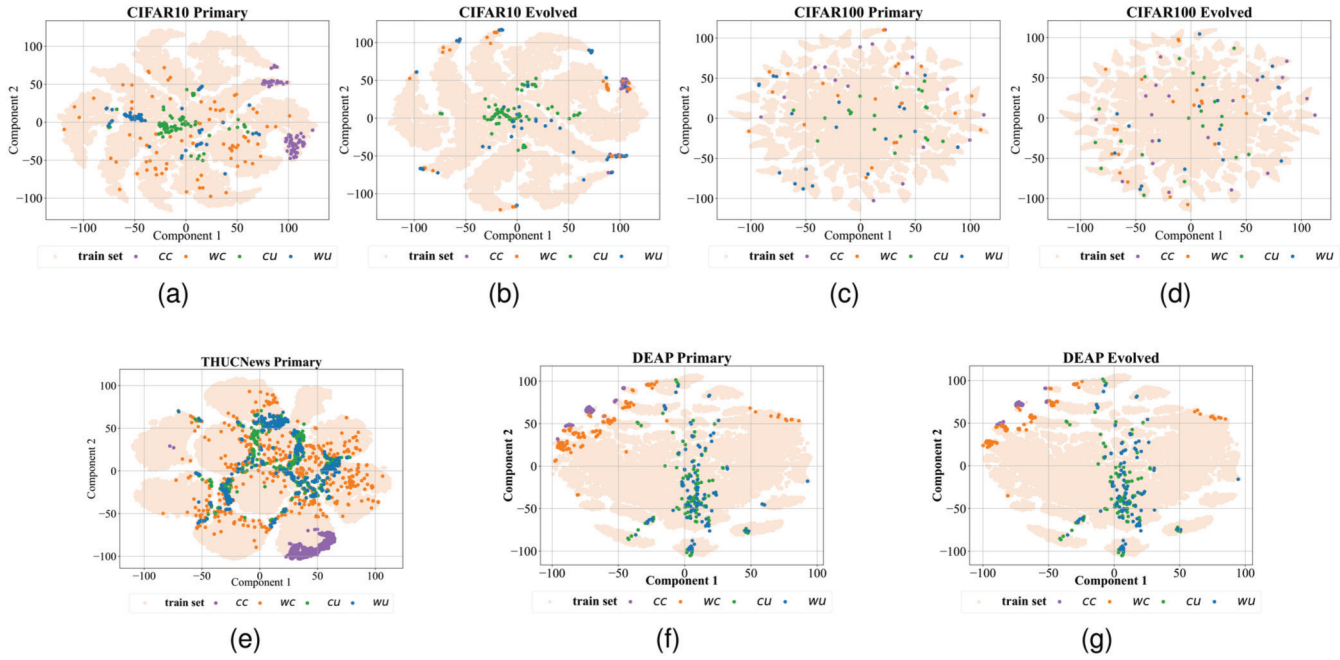

Fig. 5. Visualization of four types of sample' distributions in the decision space of source models across CIFAR10, CIFAR100, DEAP, and THUCNews.

the gap between $\mathbf{p}_a$ and $\mathbf{p}_b$ does not increase substantially. This suggests that these samples remain close to the decision boundary.

B. T-SNE Visualization of Fingerprinting Samples

The Primary Fingerprint consists of four types of sample: theoretically, *cu* and *wu* are close to the decision boundary of the model, while *cc* and *wc* are far from it. The Evolved Fingerprint is created by adding perturbations to the Primary Fingerprint. The perturbations aim to shift the *cu* and *wu* samples slightly away from the decision boundary, thereby increasing the model's confidence in the *cc* and *wc* samples. To validate our intuition about the two algorithms, we use t-SNE [42] to visualize fingerprinting samples in four datasets. Specifically, we concatenate the input and output of the source model's last decision layer to characterize each sample, applying this process to both the samples from the train set and the fingerprinting samples.

Fig. 5(a) displays the visualization of the Primary Fingerprint from CIFAR10. We can observe that the *cu* and *wu* samples are located at the decision boundary, while the *cc* samples are far from it. However, the *wc* samples do not behave like the *cc* samples. Upon further examination of Fig. 5(b), which shows the Evolved Fingerprint, we can see that the *cu* and *wu* samples remain near the decision boundary, while the *cc* and *wc* samples are far from the decision boundary after being perturbed. As shown in Fig. 5(c) and Fig. 5(d), we cannot draw a similarly intuitive conclusion. We attribute this to two factors: first, the dataset contains many categories; second, we used only 20 samples of each type of fingerprint in this dataset. Both of the above points are not conducive to the visualization of results. The results in Fig. 5(e) come from the THUCNews, in which we only apply the Primary Fingerprint. The *cu* and *wu* samples are distributed along the boundaries between different categories. In contrast, the *cc* and *wc* samples are either far from the decision boundary or located in the center of their respective categories. In our task on the DEAP, which involves binary classification of valence, we can see the decision boundary depicted by the *cu* and *wu* samples in Fig. 5(f) and Fig. 5(g). The *cc* and *wc* samples are far from the decision boundary. Upon closer inspection, it is evident that the distance between *cu* and *wu* samples in Fig. 5(g) is noticeably greater in some areas compared to Fig. 5(f). This phenomenon better validates our optimization objective for both fingerprinting algorithms.

In summary, the visual analysis of fingerprinting samples across four datasets in three fields aligns with our algorithm's expectations. We can more accurately model the fingerprint by incorporating both samples near the decision boundary and those far from it.

C. Topological Map of EEG Fingerprinting Samples

To further clarify the distinctions between Primary and Evolved Fingerprints, we provide topographic maps of four types of EEG samples in the BCI task. As shown in Fig. 6, the EEG samples that comprise the Primary Fingerprint and Evolved Fingerprint are presented, respectively. Existing neuroscience research has identified correlations between emotion and the asymmetry of the frontal and temporal lobes [43]. The left frontal and right temporal lobes are commonly activated for positive emotion (high valence), while the left temporal and right frontal lobes are activated for negative emotion (low valence).

In the *cc* case (correct confident predictions by M_s), the primary samples show activation in the left frontal and right temporal lobes. The evolved samples exhibit a noticeable shift in the activation area of the right temporal lobe. In the *cu* case (correct uncertain predictions) with low valence ground

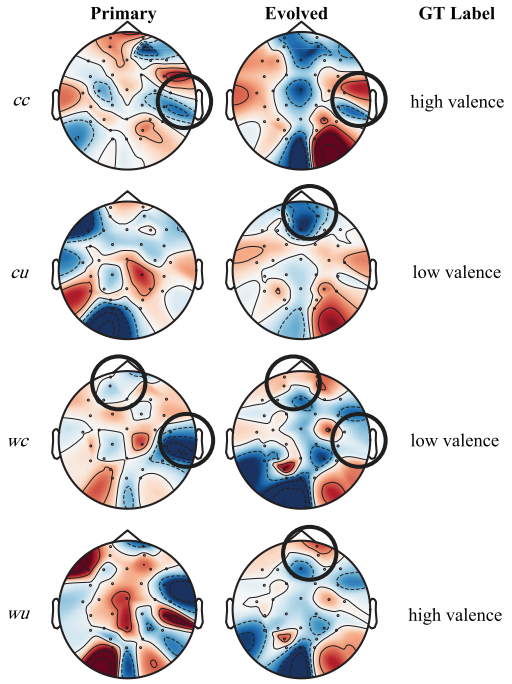

Fig. 6. Topological map of EEG samples. The two middle columns represent primary and evolved samples, respectively. The leftmost column denotes sample type, and the rightmost column indicates actual label.

truth, the primary samples lack apparent asymmetry despite the correct low-confidence predictions. However, evolved samples show increased activation in the right frontal lobe, aligning more with asymmetrical characteristics. In the *wc* case (wrong confident predictions) where the predicted label is high and the correct label is low, the primary samples show activation in the left frontal lobe and right temporal lobe, contributing to the high-confidence wrong prediction. The evolved samples attenuate the activation level in the left frontal lobe. Lastly, in the *wu* case (wrong uncertain predictions) where the predicted label is low, and the correct label is high, the primary samples show contradictory activation areas, causing low-confidence wrong predictions. The evolved samples enhance activation in the right frontal lobe, aligning with a more reasonable low-valence prediction.

VIII. Conclusion and Future Work

This paper concludes that the proposed fingerprinting methods exhibit robust stability against traditional IP removal attacks and more advanced IP detection and erasure techniques. The Primary Fingerprint algorithm is highlighted for its efficiency, as it does not require additional training, while the Evolved Fingerprint algorithm, despite its complexity, provides enhanced stability and reliability. The effectiveness of these methods is demonstrated through extensive experiments and analyses, which show their superiority in mitigating attacks compared to existing approaches.

It is worth noting that current methods focus on IP protection for decision-making models, and future research will aim to address IP protection for generative deep models. This will involve developing new techniques and algorithms specifically designed to meet the unique challenges posed by generative models, thereby ensuring broader applicability and stronger protection mechanisms in the rapidly evolving field.

References

[1] J. Achiam et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.
[2] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High accuracy and high fidelity extraction of neural networks," in *Proc. 29th USENIX Conf. Secur. Symp.*, 2020, pp. 1345–1362.
[3] G. Gan, Y. Li, D. Wu, and S.-T. Xia, "Towards robust model watermark via reducing parametric vulnerability," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 4728–4738.
[4] X. He, Q. Xu, L. Lyu, F. Wu, and C. Wang, "Protecting intellectual property of language generation APIs with lexical watermark," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, Jun. 2022, pp. 10758–10766.
[5] Z. Jiang, M. Fang, and N. Z. Gong, "IPCert: Provably robust intellectual property protection for machine learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2023, pp. 3614–3623.
[6] K. Yang, R. Wang, and L. Wang, "MetaFinger: Fingerprinting the deep neural networks with meta-training," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 776–782.
[7] T. Wang and S.-H. Zhong, "Fingerprinting in EEG model IP protection using diffusion model," in *Proc. Int. Conf. Multimedia Retr.*, May 2024, pp. 120–128.
[8] T. Xu, S.-H. Zhong, and Z. Xiao, "Protecting intellectual property of EEG-based model with watermarking," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2023, pp. 37–42.
[9] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proc. ACM Int. Conf. Multimedia Retrieval*, Bucharest, Romania, 2017, pp. 269–277.
[10] L. Fan, K. W. Ng, C. S. Chan, and Q. Yang, "DeepIPR: Deep neural network ownership verification with passports," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6122–6139, Oct. 2022.
[11] S. Szyller, B. G. Atli, S. Marchal, and N. Asokan, "DAWN: Dynamic adversarial watermarking of neural networks," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 4417–4425.
[12] X. Cao, J. Jia, and N. Z. Gong, "IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, May 2021, pp. 14–25.
[13] Y. Zheng, S. Wang, and C.-H. Chang, "A DNN fingerprint for non-repudiable model ownership identification and piracy detection," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2977–2989, 2022.
[14] R. Zhang, J. Chen, B. Fang, L.-B. Zhang, A. K. Singh, and Z. Lyu, "Artificial intelligence generated data augmentation for abdominal multi-organ segmentation," *IEEE Trans. Consum. Electron.*, vol. 70, no. 3, pp. 6031–6041, Aug. 2024.
[15] J. Guan, J. Liang, and R. He, "Are you stealing my model? Sample correlation for fingerprinting deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 1–14.
[16] N. Lukas, Y. Zhang, and F. Kerschbaum, "Deep neural network fingerprinting by conferrable adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2019, pp. 1–18.
[17] Y. Quan, H. Teng, R. Xu, J. Huang, and H. Ji, "Fingerprinting deep image restoration models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 13239–13249.
[18] Y. Sun et al., "Deep intellectual property protection: A survey," 2023, *arXiv:2304.14613*.
[19] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Proc. Int. Symp. Res. Attacks, Intrusions, Defenses*, 2018, pp. 273–294.
[20] R. Namba and J. Sakuma, "Robust watermarking of neural network with exponential weighting," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Auckland, New Zealand, Jul. 2019, pp. 228–240.
[21] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–15.
[22] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009, pp. 1–58.
[23] D. Hendrycks and T. G. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2019, pp. 1–16.
[24] M. Sun et al., *THUCTC: An Efficient Chinese Text Classifier*. San Francisco, CA, USA: GitHub Repository, 2016.

[25] S. Koelstra et al., "DEAP: A database for emotion analysis; Using physiological signals," *IEEE Trans. Affect. Comput.*, vol. 3, no. 1, pp. 18–31, Jun. 2011.
[26] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic, "A multimodal database for affect recognition and implicit tagging," *IEEE Trans. Affect. Comput.*, vol. 3, no. 1, pp. 42–55, Jan. 2012.
[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.
[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput Vis. Pattern Recognit.*, Jun. 2017, pp. 4700–4708.
[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[32] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*.
[33] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," 2016, *arXiv:1605.05101*.
[34] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 562–570.
[35] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, 2015, vol. 29, no. 1, pp. 2267–2273.
[36] Y. Song, Q. Zheng, B. Liu, and X. Gao, "EEG conformer: Convolutional transformer for EEG decoding and visualization," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 710–719, 2022.
[37] R. T. Schirrmeister et al., "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapp.*, vol. 38, no. 11, pp. 5391–5420, 2017.
[38] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *Proc. 27th USENIX Conf. Security Symp.*, Baltimore, MD, USA, 2018, pp. 1615–1631.
[39] H. Gao, J. Xiao, Y. Yin, T. Liu, and J. Shi, "A mutually supervised graph attention network for few-shot segmentation: The perspective of fully utilizing limited samples," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4826–4838, Apr. 2024.
[40] D. Wu, Y. Xu, and B.-L. Lu, "Transfer learning for EEG-based brain–computer interfaces: A review of progress made since 2016," *IEEE Trans. Cognit. Develop. Syst.*, vol. 14, no. 1, pp. 4–19, Mar. 2022.
[41] H. Gao, B. Qiu, Y. Wang, S. Yu, Y. Xu, and X. Wang, "TBDB: Token bucket-based dynamic batching for resource scheduling supporting neural network inference in intelligent consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1134–1144, Feb. 2024.
[42] L. Van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, Jan. 2008.
[43] E. Harmon-Jones, P. A. Gable, and C. K. Peterson, "The role of asymmetric frontal cortical activity in emotion-related phenomena: A review and update," *Biol. Psychol.*, vol. 84, no. 3, pp. 451–462, 2010.