Rethinking Removal Attack and Fingerprinting Defense for Model Intellectual Property Protection: A Frequency Perspective

Cheng Zhang[1,2], **Yang Xu**[1,2*], **Tingqiao Huang**[1,2], **Zixing Zhang**[1]
[1]College of Computer Science and Electronic Engineering, Hunan University
[2]College of Cyber Science and Technology, Hunan University
{zhangchengcs, xuyangcs, huangtingqiao, zixingzhang}@hnu.edu.cn

Abstract

Training deep neural networks is resource-intensive, making it crucial to protect their intellectual property from infringement. However, current model ownership resolution (MOR) methods predominantly address general removal attacks that involve weight modifications, with limited research considering alternative attack perspectives. In this work, we propose a frequency-based model ownership removal attack, grounded in a key observation: modifying a model's high-frequency coefficients does not significantly impact its performance but does alter its weights and decision boundary. This change invalidates the existing MOR methods. We further propose a frequency-based fingerprinting technique as a defense mechanism. By extracting frequency-domain characteristics instead of decision boundary or model weights, our fingerprinting defense effectively against the proposed frequency-based removal attack and demonstrates robustness against existing general removal attacks. The experimental results show that the frequency-based removal attack can easily defeat state-of-the-art white-box watermarking and fingerprinting schemes while preserving model performance, and the proposed defense method is also effective. Our code is released at: https://github.com/huangtingqiao/RRA-IJCAI25.

1 Introduction

Deep Neural Networks (DNNs) have achieved great success in many real-world applications [Kälble *et al.*, 2024; Chen *et al.*, 2024; Kim *et al.*, 2024]. However, training a high-performance DNN model is very costly, typically requiring heavy investment in high-quality data, expert knowledge, computing power, and so on. As a result, these valuable models are increasingly becoming targets for thieves. Previous studies [Tang *et al.*, 2024; Chen *et al.*, 2023] have revealed that attackers can acquire an exact replica of the original model via model-sharing platforms, and even steal models through model extraction attacks, such as APIs querying [Carlini *et al.*, 2024] and side-channel exploitation [Gongye *et al.*, 2020].

Model ownership resolution (MOR) [Liu *et al.*, 2024] is an effective method for protecting intellectual property of DNN models, which mainly includes model watermarking [Fan *et al.*, 2019; Zhu *et al.*, 2024] and fingerprinting [Peng *et al.*, 2022; Pan *et al.*, 2022] techniques. MOR enables model owners to verify and assert their ownership if the model is misappropriated, thus providing a means to resolve intellectual property disputes.

DNN watermarking methods embed an ownership-representing watermark into the DNN model during training and later verify ownership by extracting the watermark from suspected models. Specifically, feature-based white-box watermarking methods [Uchida *et al.*, 2017; Li *et al.*, 2022; Fan *et al.*, 2019] embed ownership identifiers within model weights by modifying the loss function during training. In contrast, backdoor-based black-box watermarking methods [Adi *et al.*, 2018; Leroux *et al.*, 2024; Li *et al.*, 2023] insert specific triggers during training, allowing the owner to verify a pirated model based on distinct model outputs. However, the model retraining process for the model watermarking is costly, and the embedded watermark may lead to a loss in model accuracy. Furthermore, while some watermarking methods have demonstrated robustness against various removal attacks, such as pruning or fine-tuning, recent studies indicate that model watermarking still faces threats from tampering and watermark overwriting [Zong *et al.*, 2024; Wang *et al.*, 2023].

By contrast, DNN fingerprinting methods, which have gained increasing attention for their non-intrusive nature, demonstrate model ownership by extracting unique characteristics from the existing model without modifying it. Researchers have found that the decision boundary can reflect the distinct characteristics of a model and have proposed various methods to extract fingerprints from the decision boundary, including utilizing feature points close to the decision boundary [Cao *et al.*, 2021], or the adversarial samples [Wang and Chang, 2021; Peng *et al.*, 2022]. They have also evaluated the robustness of fingerprinting against existing ownership removal attacks. These attacks primarily modify model weights but minimally alter the decision boundary, so it is difficult to destroy the existing model fingerprint. However, the evaluated attacks thus far were not specifically designed for DNN fingerprinting methods, and the robustness of fingerprinting methods has yet to be thoroughly assessed.

In this work, we propose a frequency-based model ownership removal attack, grounded in a key observation: after applying a Discrete Cosine Transform (DCT) to shift model weights into the frequency domain, modifying a model's high-frequency coefficients does not significantly impact its performance but does alter its decision boundary and weights. This boundary shift renders fingerprinting methods ineffective, and the weight changes prevent watermark recognition embedded within the model. We further propose a frequency-based fingerprinting technique as a defense mechanism. By extracting frequency-domain characteristics instead of decision boundary characteristics, our method effectively withstands the proposed frequency-based removal attack and demonstrates robustness against existing general removal attacks. Both the proposed attack and defense approaches are efficient, requiring only frequency transformation of the model without necessitating retraining or adversarial training.

Overall, our contributions are summarized as follows: (1) We reveal a noval attack surface for model ownership removal based on frequency analysis. The proposed attack can defeat MOR methods based on decision boundary or model weights. (2) We design a frequency-based fingerprinting defense against the proposed attack as well as general removal attacks. (3) Extensive experiments show the effectiveness of both our attack and defense.

2 Preliminaries and Related Work

Model Watermarking relies on embedding specific ownership indicators, such as strings or backdoor triggers, within the model. For instance, Uchida et al. [Uchida *et al.*, 2017] pioneered a feature-based white-box watermarking algorithm that embeds ownership information in model weights by adding a regularization term to the loss function. Another category of backdoor-based black-box watermarking [Adi *et al.*, 2018] introduces a trigger into the model, causing the pirated model to produce specific outputs for certain inputs, serving as a unique identifier. Li et al. [Li *et al.*, 2022] used black-box and white-box watermarking to jointly detect suspicious models and enable forensics. Recent studies [Xu *et al.*, 2024] have also explored model watermarking in complex settings, such as personalized federated learning [Zhang *et al.*, 2025]. However, the embedding process of the watermark alters the original model weights, potentially affecting the accuracy. Some studies [Lukas *et al.*, 2021] also show that model watermarking is vulnerable to removal attacks.

Model Fingerprinting leverages the intrinsic characteristics of a model as proof of ownership. Existing studies primarily treat the decision boundary as a unique and representative feature of models, constructing various types of adversarial examples to capture this boundary. IPGuard [Cao *et al.*, 2021] is an efficient adversarial example method that generates data points near the decision boundary to serve as a model fingerprint. Similarly, Wang et al. [Wang and Chang, 2021] utilize the geometry characteristics inherited in the DeepFool attack to extract these data points. Peng et al. [Peng *et al.*, 2022] discovered that the model's decision boundary can be uniquely represented by universal adversarial perturbations (UAP). Besides, Zheng et al. [Zheng *et al.*, 2022] introduced a weight-based fingerprinting approach rooted in the stability of the model's early layers, employing random projection to bind these weights to the owner's identity and enhance the fingerprint's non-repudiation. In this work, we demonstrate that their fingerprints cannot withstand frequency-based ownership removal attack and propose a frequency-based fingerprinting defense with greater robustness.

Discrete Cosine Transform is a reversible linear transformation that maps an input sequence of N real numbers into frequency components represented by orthogonal cosine bases. This transformation expresses the original data as a sum of cosine waves with varying frequencies and amplitudes, effectively converting it to a frequency-domain representation. The resulting coefficients are arranged by increasing frequency, where lower frequencies typically represent the primary features of the data, while higher frequencies capture details or noise. As the dimensionality of the input data increases, DCT can capture a more comprehensive set of features; however, excessive dimensionality may amplify noise, causing feature extraction bias.

Recent studies have explored applying DCT to transform DNN model weights into the frequency domain, enabling more effective poison attack detection [Wang *et al.*, 2018; Fereidooni *et al.*, 2024]. We take convolutional layers (e.g., a tensor containing 64×3 independent 3×3 kernels) as an example to illustrate the method of transforming model weights into the frequency domain using DCT. We apply the DCT to each kernel individually, and then combine all transformed kernels to form a frequency coefficient matrix for the layer, which retains the same dimensions as the original.

3 Frequency-based Removal Attack

3.1 Threat Model

We consider two entities: the model owner and the attacker. The model owner is responsible for training a DNN model, which they may distribute by sharing model parameters or by providing APIs. The owner's objective is to protect the intellectual property of the model through MOR techniques. Specifically, the owner can embed a watermark or extract a fingerprint in the model, allowing them to verify ownership by extracting these identifiers from suspect models if unauthorized use is detected.

The attacker's objective is to acquire a well-performing model at a relatively low cost while avoiding accountability. We assume the attacker has white-box access to the victim model and focus on the way to make watermarks or fingerprints unrecognizable. The attacker may employ various modifications to the model, such as fine-tuning and pruning, but will not generate data to train a new model from scratch due to the high computational cost involved.

3.2 Observation Explanation

OBSERVATION: *After transforming model weights into a frequency domain coefficient matrix using DCT, modifying the high-frequency coefficients leads to minimal performance loss while rendering the MOR mechanisms ineffective. In*

Dimension	Band	Modified Value (V)				
		V=-1	V=-0.5	V=0	V=0.5	V=1
1D DCT	Low	11.92	12.85	37.97	9.88	9.33
	Mid	17.07	25.90	51.68	23.36	14.94
	High	37.51	66.47	73.70	66.29	39.71
2D DCT	Low	12.35	13.47	35.57	10.49	10.05
	Mid	21.57	57.34	72.39	58.50	23.91
	High	66.46	77.80	80.08	78.79	67.51
3D DCT	Low	12.34	13.37	19.31	11.56	10.43
	Mid	76.43	81.73	83.43	82.75	78.02
	High	87.44	87.91	88.01	88.05	87.61

Table 1: Impact of modifying frequency domain coefficients at different locations under various DCT dimensions on the model accuracy (%) of AlexNet (Original accuracy: 88.33%).

Modified Scale	Modified Value (V)				
	V=-1	V=-0.5	V=0	V=0.5	V=1
1/3	87.44	87.91	88.01	88.05	87.61
1/5	88.38	88.30	88.32	88.35	88.30
1/9	88.39	88.38	88.33	88.32	88.31

Table 2: Impact of different modification scales of high-frequency coefficients on the model accuracy (%) of AlexNet.

contrast, altering the low-frequency coefficients significantly degrades the model's usability.

We first demonstrate the impact of modifying frequency domain coefficients of model weights on model ownership recognition and performance. In this experiment, we trained a baseline AlexNet model on the CIFAR-10 dataset, achieving an accuracy of 88.33%. Next, we transformed the convolutional layer weights of the model to the frequency domain using 1D, 2D, and 3D DCT, respectively, and divided the frequency domain coefficients sequentially into low, medium, and high-frequencies. Finally, we modified the frequency domain coefficients and tested the accuracy of the model after applying the Inverse Discrete Cosine Transform (IDCT). The experimental results are shown in Table 1 and Table 2.

As the DCT dimension increases from 1D to 3D, the impact on model performance decreases, suggesting that a higher-dimensional DCT more effectively concentrates the primary feature information in the low-frequency components, thus minimizing disruption to overall model performance when modifying high-frequency coefficients. Additionally, the experiments show a notable difference in the effect of modification values and frequency band positions on model performance: modifications to frequency domain coefficients closer to zero have minimal impact on model accuracy, and alterations in the high-frequency range result in relatively lower performance loss. These findings further confirm the structural characteristics of the DCT frequency domain, where low-frequency components typically carry the main feature information, while high-frequency components reflect finer details and noise, making them more suitable for non-destructive modification.

Method	ACC (%)		ORR (%)	
	Before	After	Before	After
Uchida [Uchida *et al.*, 2017]	83.09	82.21	100.00	69.14
IPGuard [Cao *et al.*, 2021]	88.33	87.61	100.00	39.10
DeepFool [Wang and Chang, 2021]	88.33	87.61	100.00	51.52
FUAP [Peng *et al.*, 2022]	88.33	87.61	100.00	44.64
Zheng et al. [Zheng *et al.*, 2022]	88.33	87.61	100.00	46.87
Ours	88.33	87.61	100.00	100.00

Table 3: Model accuracy (ACC) and ownership recognition rate (ORR) of AlexNet protected by different MOR methods against frequency-based removal attack.

Next, we present the impact of modifying frequency domain coefficients on MOR. We examined various types of MOR methods, including feature-based watermarking, decision boundary-based fingerprinting (represented by adversarial samples, geometric characteristics, and UAP), and weight-based fingerprinting. As shown in Table 3, these methods exhibit a significant decrease in ownership recognition rates after modifying high-frequency coefficients of the model. This decline is due to two primary reasons: first, the high-frequency components capture the finer details of the model, which are critical to defining its decision boundaries. Altering these high-frequency components shifts the decision boundaries, rendering decision boundary-based fingerprints ineffective. Second, changes in the high-frequency coefficients introduce detailed adjustments that, upon inverse DCT transformation, are distributed across all weights, causing global alterations in model weights. This global shift in weights invalidates weight-based ownership resolution methods. In contrast, existing methods are typically effective only against local changes, such as pruning or fine-tuning.

3.3 Removal Attack Design Details

Based on the above observations, we propose a model ownership removal attack by modifying the high-frequency domain coefficients. At the beginning, the attacker obtains the original model M and sets the model's high-frequency modification ratio $ratio$ and the high-frequency coefficient modification value V. The attack is described in Algorithm 1, using convolutional layers as an illustrative example. It mainly consists of the following steps:

Calculate the DCT coefficient of the model weight. Extract the 3D array corresponding to each convolutional kernel from the convolutional layer $f(x, y, z)$. These 3D arrays are then transformed using 3D DCT, resulting in frequency domain coefficients $F(u, v, w)$.

Modify the high-frequency coefficient. In the DCT, the low-frequency components are usually located at smaller index positions, while the high-frequency components correspond to larger index positions. Therefore, for each frequency domain coefficient $F(u, v, w)$, it can be considered as a high-frequency component when its index sum $S_i = u_i + v_i + w_i$ exceeds the preset threshold $ratio \times (n_u + n_v + n_w)$, where n_u , n_v and n_w denote the number of input channels, height and width of the filter. Subsequently, all high-frequency components $F(u, v, w)$ are set to the modified value V to obtain the new frequency domain coefficients $F'(u, v, w)$.

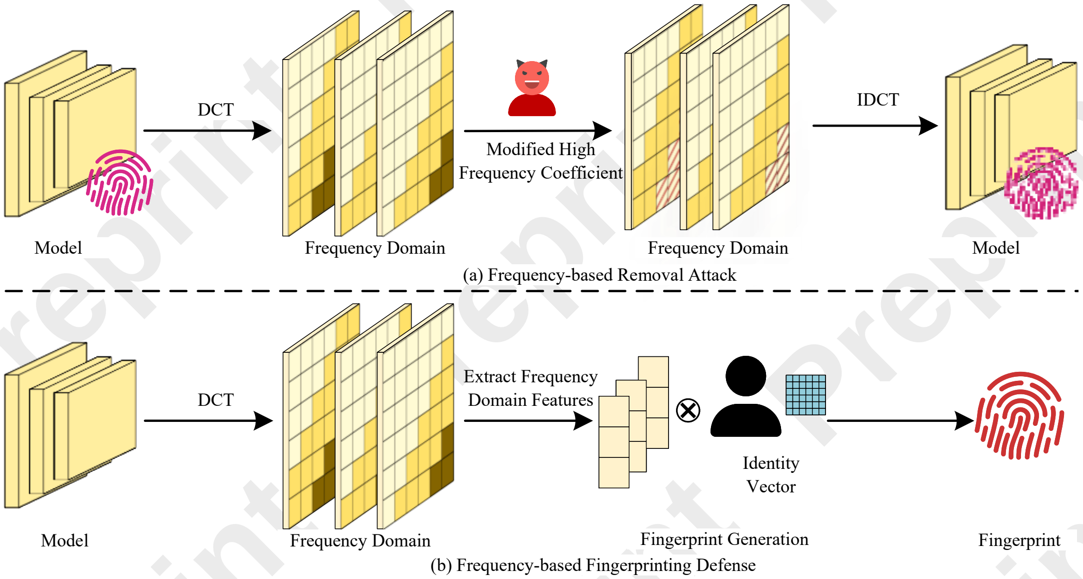

DCT
Modified High Frequency Coefficient
IDCT
Model
Frequency Domain
Frequency Domain
Model
(a) Frequency-based Removal Attack
DCT
Extract Frequency Domain Features
⊗
Identity Vector
Model
Frequency Domain
Fingerprint Generation
Fingerprint
(b) Frequency-based Fingerprinting Defense

Figure 1: Frequency-based removal attack and fingerprinting defense.

Algorithm 1 Frequency-based Removal Attack

```
Input: Original model: $M$, Ratio of the modified high-frequency coefficients: $ratio$, High-frequency coefficient values after the attack: $V$
Output: Removal model: $M'$
Extract the convolutional kernel $f(x, y, z)$ from the front convolutional layer of the original model $M$
for each kernel $f_i(x, y, z) \in f(x, y, z)$ do
    $F_i(u, v, w) \leftarrow \text{DCT}(f_i(x, y, z))$
    $F'_i(u, v, w) \leftarrow \emptyset$
    for $i \in [0, n_u \times n_v \times n_w - 1]$ do
        $d, h, w \leftarrow \left\lfloor \frac{i}{n_v \times n_w} \right\rfloor, \left\lfloor \frac{i}{n_w} \right\rfloor \bmod n_v, i \bmod n_w$
        if $d + h + w \geq ratio \times (n_u + n_v + n_w)$ then
            $F'_i(d, h, w) \leftarrow V$
        else
            $F'_i(d, h, w) \leftarrow F_i(d, h, w)$
        end if
    end for
    $f'_i(x, y, z) \leftarrow \text{IDCT}(F'_i(u, v, w))$
end for
Replace the original kernel with $f'(x, y, z)$ in the convolutional layer to form the new model $M'$
return $M'$
```

Calculate model weights via inverse DCT. The modified frequency domain coefficients F' are converted back to their original spatial domain form through an inverse 3D IDCT transform, resulting in new convolutional kernel parameters $f'(x, y, z)$.

4 Frequency-based Fingerprinting Defense

Observations in Section 3.2 indicate that the low-frequency components of model weights capture the model's primary features. Therefore, we propose extracting the model fingerprint from the low-frequency coefficients of the model weights. After training the model, the owner extracts the fingerprint and registers it with a trusted third party (TTP). In the event of an ownership dispute, the owner can extract the fingerprint from the suspect model and verify it by comparing the cosine similarity between the fingerprints of the suspect and the victim models. A high cosine similarity would suggest that the suspect model is a post-processed version of the victim model. The owner could send the above proof to the TTP to resolve ownership disputes. We assume that the model owner has white-box access to both their own model and the suspect model, or can obtain effective weight information from the suspect model through existing side-channel techniques. The proposed method comprises two main components: fingerprint extraction and ownership verification.

Fingerprint Extraction: The model owner extracts the fingerprint from their model by selecting a specific layer and applying the DCT to convert it into the frequency domain. The sum of the low and mid frequency components of these frequency-domain coefficients is then taken as the model's frequency-domain feature. The owner then binds this feature to their identity using random projection as an intermediary, forming the model fingerprint.

Ownership Verification: The TTP verifies the ownership of the suspect model. In the event of an ownership dispute, the model owner submits the frequency-domain features of the original model along with the suspect model to the TTP. The TTP first checks whether the submitted frequency-domain features match the owne's registered fingerprint. It then evaluates the similarity between the fingerprints of the suspect and original models to determine the model ownership.

4.1 Fingerprint Extraction

Before fingerprinting, the model owner trains a well-performing model M and presets an ownership threshold T.

T is the limit for judging the similarity between the pirated model and the original model. The model owner generates the model fingerprint through three steps: frequency domain transformation, frequency domain feature extraction and fingerprint generation.

Frequency Domain Transformation. Each convolutional kernel in the model's front (usually first) convolutional layer $f(x, y, z)$ undergoes a 3D DCT, generating the corresponding frequency domain coefficients $F(u, v, w)$.

Frequency Component Extraction. The low and mid-frequency components can be filtered based on the position of the indices. When the index sum $S_i = u_i + v_i + w_i$ is below the preset threshold $(r \times (n_u + n_v + n_w))$, the corresponding components are extracted as effective components, denoted as F_e. For each convolution kernel, the components F_e are then combined to form a composite feature vector V_e.

Fingerprint Generation. To generate unique random fingerprints, the extracted feature vector V_e is bound to the model owner's identity information. Specifically, a unique random Bernoulli matrix $U \in \mathbb{R}^{t \times t_0}$ is created using a pseudo-random number generator and the model owner's identity information (e.g., ID number or company entity number). Here, t_0 represents the length of the feature vector V_e, and t corresponds to the length of the final generated fingerprint f. Next, the matrix U is normalized to produce a random projection matrix P, ensuring that each column of P is a unit vector. This normalization follows the Johnson-Lindenstrauss (JL) lemma [Dasgupta and Gupta, 2003], which guarantees the preservation of relative distances between data points in the lower-dimensional space. Finally, as described in Equation 1, the DNN model's feature vector V_e is projected onto the random space defined by the matrix P, resulting in the fingerprint f. The model owner then submits the extracted fingerprint f to the TTP, which performs a timestamp verification on the submitted fingerprint to ensure its validity.

$$f = V_e P^T = \frac{V_e U^T}{\sqrt{t}} \tag{1}$$

4.2 Ownership Verification

When it is necessary to verify the ownership of a suspicious model, we can follow the above steps to compute the fingerprint of that model f'. Subsequently, the computed fingerprints are compared with the enrolled fingerprints in the TTP database. In the comparison process, we use cosine similarity as a measure, which is calculated as follows:

$$\text{FS} = \frac{f \cdot f'}{\|f\| \|f'\|} = \frac{\sum_{i=1}^{n} f_i f_i'}{\sqrt{\sum_{i=1}^{n} f_i^2} \sqrt{\sum_{i=1}^{n} (f_i')^2}} \tag{2}$$

In this process, the fingerprint vector f and f' denote the feature representations of the victim model and the suspect model, respectively. A larger value of cosine similarity indicates a higher match between the two fingerprints, thus indicating a stronger similarity between these two models. If the computed cosine similarity exceeds a preset threshold and the timestamps in the TTP database show that the original model was registered earlier than any of the suspicious models claimed to have been registered, it can be confirmed that the suspicious model belongs to the original owner.

5 Evaluation

5.1 Setup

Datasets. We evaluate our approach on four popular image classification datasets: MNIST [LeCun *et al.*, 1998], Fashion MNIST (FMNIST) [Xiao *et al.*, 2017], CIFAR-10 and CIFAR-100 [Krizhevsky *et al.*, 2009].

Models. We evaluate our method in four different model architectures: CNN with two convolutional layers, a max-pooling layer, and two fully connected layers, AlexNet [Krizhevsky *et al.*, 2012], ResNet18 [He *et al.*, 2016], VGG16 [Simonyan and Zisserman, 2014]. We train innocent models using the same dataset or model structure as the original model, or using the same dataset and model but with different initialization conditions, and apply removal attacks to the original model to obtain pirated versions.

Baselines. To verify the effectiveness of fingerprinting defense, we evaluate it with four classes of advanced model fingerprint methods: (a) IPGuard [Cao *et al.*, 2021] finds data points close to the decision boundaries by optimizing the objective function; (b) Deepfool [Wang and Chang, 2021] uses geometric properties to find the minimum perturbation, so that the input point crosses the decision boundary; (c) FUAP [Peng *et al.*, 2022] represents decision boundary by universal adversarial perturbations; (d) Zheng et al's method [Zheng *et al.*, 2022] uses the weights of the model's front convolutional layers as a fingerprint.

Removal Attacks. We consider following model ownership removal attacks: (a) Weight Pruning (WP): Prune the least important p percent of weights in the target model, beginning at 0.4 and increasing in increments of 0.1; (b) Filter Pruning (FP): Prune the least significant q percent of weights in the target model, starting at 1/16 and increasing in increments of 1/16; (c) Fine-tuning-retrain (Retrain): All the parameters of the target model are updated using the training data with a smaller learning rate and the model is trained for 20 epochs; (d) Fine-tuning-transfer (Transfer): All the parameters of the target model are updated using a different dataset and the model is trained for 20 epochs; (e) Frequency-based Removal: the 1/3 highest frequency coefficients of the model weights modified to V using the proposed attack.

5.2 Result

Robustness. We evaluated the robustness of our scheme against various attacks under different settings and compared it with baseline approaches. As shown in Table 4, our scheme maintains a model accuracy between 85.6% and 89.34% when facing different types of removal attacks, and the fingerprint recognition rates almost 100% are still maintained. Additionally, both our fingerprinting method and baselines effectively resist existing removal attacks. However, when encountering the frequency-based removal attack, the fingerprint recognition rates of IPGuard, DeepFool, and UAPS drop significantly. Due to the use of low-frequency features as fingerprints in our scheme, modifications to high-frequency coefficients do not impact fingerprint effectiveness. In contrast, we observed that the fingerprint recognition rate of the baselines decreases as the degree of high-frequency coefficient modifications increases. This occurs because most high-

Method	Results	Original	WP			FP		Fine-tuning		Frequency-based Removal			
			p=0.4	p=0.5	p=0.6	q=1/16	q=2/16	Retrain	Transfer	V=0.3	V=0.5	V=0.7	V=1
Init	ACC (%)	88.33	88.41	88.41	88.35	87.31	85.60	89.34	-	88.02	88.05	87.87	87.61
IPGuard [Cao *et al.*, 2021]	FS (%)	100.00	97.40	96.80	95.30	86.10	78.00	79.50	-	72.00	63.20	54.10	39.10
DeepFool [Wang and Chang, 2021]	FS (%)	100.00	98.59	97.11	95.67	89.60	82.44	81.16	-	77.80	72.70	63.23	51.52
FUAP [Peng *et al.*, 2022]	FS (%)	100.00	87.39	85.75	85.18	76.67	74.37	82.87	82.72	46.46	39.64	40.24	44.64
Zheng et al. [Zheng *et al.*, 2022]	FS (%)	100.00	99.99	99.99	99.99	99.08	97.84	99.99	99.95	83.33	70.91	59.65	46.87
Ours	FS (%)	100.00	99.99	99.99	99.99	99.05	97.84	99.99	99.95	100.00	100.00	100.00	100.00

Table 4: Model accuracy and fingerprint similarity under different removal attacks compared with different model fingerprints.

Model	Results	Original	WP			FP		Fine-tuning		Frequency-based Removal			
			p=0.4	p=0.5	p=0.6	q=1/16	q=2/16	Retrain	Transfer	V=0.3	V=0.5	V=0.7	V=1
CNN	ACC (%)	80.33	80.26	79.71	73.81	76.82	72.82	82.31	-	80.25	79.83	78.55	76.33
	FS (%)	100.00	99.97	99.92	99.61	98.89	97.72	99.93	97.26	100.00	100.00	100.00	100.00
AlexNet	ACC (%)	88.33	88.41	88.41	88.35	87.31	85.60	89.34	-	88.02	88.05	87.87	87.61
	FS (%)	100.00	99.99	99.99	99.99	99.05	97.84	99.99	99.95	100.00	100.00	100.00	100.00
ResNet18	ACC (%)	85.94	83.25	78.67	71.09	24.81	10.23	88.54	-	84.15	84.38	84.86	85.29
	FS (%)	100.00	99.99	99.97	99.95	99.41	98.67	99.98	99.98	100.00	100.00	100.00	100.00
VGG16	ACC (%)	90.20	90.21	90.19	90.26	89.46	86.99	92.43	-	89.80	89.61	89.61	89.60
	FS (%)	100.00	99.99	99.97	99.99	99.62	99.45	98.98	99.94	100.00	100.00	100.00	100.00

Table 5: Model accuracy and fingerprint similarity of different models on CIFAR-10 under various attacks.

Dataset	Results	Original	WP			FP		Fine-tuning		Frequency-based Removal			
			p=0.4	p=0.5	p=0.6	q=1/16	q=2/16	Retrain	Transfer	V=0.3	V=0.5	V=0.7	V=1
CIFAR-10	ACC (%)	88.33	88.41	88.41	88.35	87.31	85.60	89.34	-	88.02	88.05	87.87	87.61
	FS (%)	100.00	99.99	99.99	99.99	99.05	97.84	99.99	99.95	100.00	100.00	100.00	100.00
CIFAR-100	ACC (%)	60.65	60.74	60.83	60.56	58.82	54.41	64.67	-	60.39	60.30	60.19	59.84
	FS (%)	100.00	99.99	99.99	99.99	99.33	98.51	99.98	99.97	100.00	100.00	100.00	100.00
MMIST	ACC (%)	99.46	99.44	99.43	99.42	99.43	99.45	99.55	-	99.45	99.45	99.42	99.34
	FS (%)	100.00	99.99	99.99	99.99	99.60	98.49	99.98	99.46	100.00	100.00	100.00	100.00
FMNIST	ACC (%)	92.22	92.23	92.21	92.23	92.16	91.65	93.24	-	91.56	91.48	91.32	91.23
	FS (%)	100.00	99.99	99.99	99.99	99.47	98.51	99.98	99.74	100.00	100.00	100.00	100.00

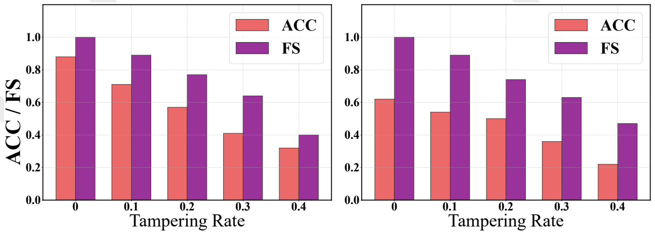
Table 6: Model accuracy and fingerprint similarity of AlexNet on different datasets under various attacks.

Selection Scale	Results	Original	WP			FP		Fine-tuning		Frequency-based Removal			
			p=0.4	p=0.5	p=0.6	q=1/16	q=2/16	Retrain	Transfer	V=0.3	V=0.5	V=0.7	V=1
Init	ACC (%)	88.33	88.41	88.41	88.35	87.31	85.60	89.34	-	88.02	88.05	87.87	87.61
1/3	FS (%)	100.00	99.99	99.99	99.99	94.96	91.62	99.92	99.57	100.00	100.00	100.00	100.00
1/2	FS (%)	100.00	99.99	99.99	99.99	98.85	97.38	99.98	99.93	100.00	100.00	100.00	100.00
2/3	FS (%)	100.00	99.99	99.99	99.99	99.05	97.84	99.99	99.95	100.00	100.00	100.00	100.00
4/5	FS (%)	100.00	99.99	99.99	99.99	99.10	97.91	99.99	99.96	87.80	78.50	68.92	56.50
1	FS (%)	100.00	99.99	99.99	99.99	99.09	97.86	99.99	99.96	84.43	72.70	61.72	48.94

Table 7: Impact of frequency domain feature selection scale r on frequency-based model fingerprinting.

frequency coefficients are initially zero, and as the modification values deviate further from zero, the model weights and decision boundaries shift more, leading to a marked decline in the baselines' fingerprint recognition rates.

Table 5 and Table 6 present the accuracy and fingerprint recognition rates of our method across different models and datasets. The results show that under all attack scenarios, our method consistently achieves a similarity of over 97%. Specifically, in the FP attack scenario, even with the removal of some convolutional kernels leading to partial loss of fingerprint information, our method still maintains a fingerprint recognition rate between 97.72% and 99.62%. This indicates that FP operations on model weights do not invalidate our frequency-based fingerprints, even with some kernel information lost, the remaining fingerprint provides sufficient recognition accuracy. Additionally, we observe that as model complexity increases, the variations in model performance and fingerprint similarity under the same attack conditions de-



(a) AlexNet on CIFAR-10 (b) VGG16 on CIFAR-100

Figure 2: The effect of tampering with fingerprint frequency domain coefficients on model performance (ACC) and fingerprint similarity (FS).

crease. This can be attributed to differences in model structural complexity. Larger models typically exhibit higher parameter redundancy, meaning that the percentage of effective parameters is relatively low. Due to the presence of redundant parameters, the impact of attacks on model performance becomes more distributed, allowing key model features and fingerprint similarity to remain more stable under attack.

Table 7 illustrates the relationship between the range of frequency-domain features extracted for fingerprints and the fingerprint recognition rate. In our experiments, we selected frequency-domain coefficients from low to high, ranging from one-third up to all available coefficients, to generate the model fingerprint. The results show that as the proportion of selected frequency-domain coefficients (r) increases, the overall similarity of the fingerprint also rises when facing pruning and fine-tuning attacks. This indicates that including more frequency-domain coefficients enhances fingerprint stability and improves resistance to such attacks. However, when encountering frequency-domain attacks targeting high-frequency coefficients, the fingerprint similarity drops significantly. Therefore, to achieve an optimal trade-off under various removal attacks, we recommend selecting up to two-thirds of the frequency-domain coefficients, focusing on low and mid-frequency coefficients, for fingerprint extraction.

Adaptive attack. We simulated an adaptive attack scenario in which an attacker, aware of our fingerprinting method, attempts to make our frequency-based fingerprints unrecognizable by altering the model's frequency-domain coefficients. Figure 2 presents the effects on model accuracy and fingerprint recognition rate when the attacker randomly selects modification locations and alters the frequency-domain coefficients to random values at various modification rates. The results indicate that this adaptive attack leads to a significant decline in model accuracy, rendering the model unusable. For example, to reduce the fingerprint recognition rate below 50%, the attacker would need to modify 40% of the frequency-domain coefficients, resulting in a model accuracy drop to 31.69%.

Uniqueness. Even models trained with the same dataset and architecture exhibit low similarity due to the non-convex nature of the neural network loss function, where different initializations lead to distinct local minima. Thus, models trained independently on the same data should be considered distinct. In Table 8, we compare the fingerprint similarity (FS) of models trained independently on the same dataset or with identical architectures. The results show generally low similarity, well below the threshold for detecting pirated models, indicating a very low false-positive rate.

Original models		**Innocent models**		**FS (%)**
Dataset	Model	Dataset	Model	
CIFAR-10	CNN	CIFAR-10	CNN	6.27
CIFAR-10	AlexNet	CIFAR-10	AlexNet	2.51

Table 8: Fingerprint similarity between independently models.

Method	Gen. Time(s)	Ver. Time(s)
IPGuard [Cao *et al.*, 2021]	293.723	0.656
DeepFool [Wang and Chang, 2021]	270.537	9.175
Zheng et al. [Zheng *et al.*, 2022]	0.001	0.001
Ours	0.044	0.024

Table 9: Average time for generating and verifying fingerprints on the CIFAR-10 dataset using four models.

Efficiency. Table 9 presents the time usage for model fingerprint extraction and verification. Compared to other decision-boundary-based fingerprinting methods, our approach and that of Zheng et al. [Zheng *et al.*, 2022] are highly efficient, requiring only a few seconds of computation. This efficiency comes from the fact that decision boundary-based fingerprinting methods require adversarial sample training, which incurs a substantial computational cost. Zheng et al.'s method requires only model weight extraction, resulting in minimal computation overhead. However, it demonstrates weaker robustness against frequency-domain removal attacks. In contrast, our method involves converting the model to the frequency-domain for extraction, introducing additional DCT computation costs. Additionally, Table 9 demonstrates the efficiency of the proposed attack methods, indicating that the computational costs of our removal attacks and fingerprinting defenses are essentially negligible.

6 Conclusion

In this work, we propose a frequency-based model ownership removal attack, which changes the decision boundary and model weight by modifying the high-frequency coefficients, making traditional watermarking and fingerprinting methods ineffective with minimal impact on model accuracy. Additionally, we propose a robust defense mechanism based on frequency-domain fingerprinting, which withstands the proposed frequency-based attack and shows resilience against other general removal attacks. Experimental results validate the efficacy of our attack in erasing ownership indicators while maintaining model performance and demonstrate the robustness of our fingerprinting defense. Our method relies on frequency domain conversion, applicable to models with structured weights like CNN, ResNet, and LSTM. Its applicability to architectures with less spatial or sequential structure is limited. Extending our approach to support these models will be future work.

Acknowledgements

We would like to heartfeltly thank the anonymous reviewers for their constructive comments. This work is supported in part by the National Natural Science Foundation of China (No. 62272154, 62472163), the Hunan Provincial Natural Science Foundation of China (No. 2024JJ5096), the science and technology innovation Program of Hunan Province (No. 2023RC3125), the Science & Technology talents lifting project of Hunan Province (No. 2023TJ–N23), the Training Program for Excellent Young Innovators of Changsha (No. kq2209008), the Hunan Provincial Department of Education Excellent Youth Project (No. 24B0039), and the Hunan Provincial Innovation Foundation for Postgraduate (No. CX20230389). Yang Xu is the corresponding author.

References

[Adi *et al.*, 2018] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *Proceedings of the USENIX security symposium*, pages 1615–1631, 2018.

[Cao *et al.*, 2021] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Ipguard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. In *Proceedings of the ACM Asia Conference on Computer and Communications Security (ASIACCS)*, pages 14–25, 2021.

[Carlini *et al.*, 2024] Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, Eric Wallace, David Rolnick, and Florian Tramèr. Stealing part of a production language model. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[Chen *et al.*, 2023] Yiming Chen, Jinyu Tian, Xiangyu Chen, and Jiantao Zhou. Effective ambiguity attack against passport-based dnn intellectual property protection schemes through fully connected layer substitution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8123–8132, 2023.

[Chen *et al.*, 2024] Jiayuan Chen, Changchang Yin, Yuanlong Wang, and Ping Zhang. Predictive modeling with temporal graphical representation on electronic health records. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5763–5771, 2024.

[Dasgupta and Gupta, 2003] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[Fan *et al.*, 2019] Lixin Fan, Kam Woh Ng, and Chee Seng Chan. Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 4716–4725, 2019.

[Fereidooni *et al.*, 2024] Hossein Fereidooni, Alessandro Pegoraro, Phillip Rieger, Alexandra Dmitrienko, and Ahmad-Reza Sadeghi. Freqfed: A frequency analysis-based approach for mitigating poisoning attacks in federated learning. In *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS)*, 2024.

[Gongye *et al.*, 2020] Cheng Gongye, Yunsi Fei, and Thomas Wahl. Reverse-engineering deep neural networks using floating-point timing side-channels. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2020.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

[Kim *et al.*, 2024] Changhoon Kim, Kyle Min, Maitreya Patel, Sheng Cheng, and Yezhou Yang. Wouaf: Weight modulation for user attribution and fingerprinting in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8974–8983, 2024.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 2009.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 1106–1114, 2012.

[Kälble *et al.*, 2024] Jonas Kälble, Sascha Wirges, Maxim Tatarchenko, and Eddy Ilg. Accurate training data for occupancy map prediction in automated driving using evidence theory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5281–5290, 2024.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Leroux *et al.*, 2024] Sam Leroux, Stijn Vanassche, and Pieter Simoens. Multi-bit black-box watermarking of deep neural networks in embedded applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2121–2130, 2024.

[Li *et al.*, 2022] Bowen Li, Lixin Fan, Hanlin Gu, Jie Li, and Qiang Yang. Fedipr: Ownership verification for federated deep neural network models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(4):4521–4536, 2022.

[Li *et al.*, 2023] Peixuan Li, Pengzhou Cheng, Fangqi Li, Wei Du, Haodong Zhao, and Gongshen Liu. Plmmark: a secure and robust black-box watermarking framework for pre-trained language models. In *Proceedings of the AAAI*

Conference on Artificial Intelligence, pages 14991–14999, 2023.

[Liu *et al.*, 2024] Jian Liu, Rui Zhang, Sebastian Szyller, Kui Ren, and N. Asokan. False claims against model ownership resolution. In *Proceedings of the USENIX Security Symposium*, pages 6885–6902, August 2024.

[Lukas *et al.*, 2021] Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. Deep neural network fingerprinting by conferrable adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[Pan *et al.*, 2022] Xudong Pan, Yifan Yan, Mi Zhang, and Min Yang. Metav: A meta-verifier approach to task-agnostic model fingerprinting. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1327–1336, 2022.

[Peng *et al.*, 2022] Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. Fingerprinting deep neural networks globally via universal adversarial perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13430–13439, 2022.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Tang *et al.*, 2024] Minxue Tang, Anna Dai, Louis DiValentin, Aolin Ding, Amin Hass, Neil Zhenqiang Gong, Yiran Chen, and Hai ”Helen” Li. ModelGuard: Information-Theoretic defense against model extraction attacks. In *Proceedings of the USENIX Security Symposium*, pages 5305–5322, August 2024.

[Uchida *et al.*, 2017] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin’ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the ACM on International Conference on Multimedia Retrieval (ICMR)*, pages 269–277, 2017.

[Wang and Chang, 2021] Si Wang and Chip-Hong Chang. Fingerprinting deep neural networks-a deepfool approach. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2021.

[Wang *et al.*, 2018] Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. Packing convolutional neural networks in the frequency domain. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2495–2510, 2018.

[Wang *et al.*, 2023] Meiqi Wang, Han Qiu, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Mitigating query-based neural network fingerprinting via data augmentation. *ACM Transactions on Sensor Networks*, 2023.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[Xu *et al.*, 2024] Yang Xu, Yunlin Tan, Cheng Zhang, Kai Chi, Peng Sun, Wenyuan Yang, Ju Ren, Hongbo Jiang, and Yaoxue Zhang. Robwe: Robust watermark embedding for personalized federated learning model ownership protection. *arXiv preprint arXiv:2402.19054*, 2024.

[Zhang *et al.*, 2025] Cheng Zhang, Yang Xu, Jianghao Tan, Jiajie An, and Wenqiang Jin. Mingledpie: A cluster mingling approach for mitigating preference profiling in cfl. In *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS)*, 2025.

[Zheng *et al.*, 2022] Yue Zheng, Si Wang, and Chip-Hong Chang. A dnn fingerprint for non-repudiable model ownership identification and piracy detection. *IEEE Transactions on Information Forensics and Security*, 17:2977–2989, 2022.

[Zhu *et al.*, 2024] Peifei Zhu, Tsubasa Takahashi, and Hirokatsu Kataoka. Watermark-embedded adversarial examples for copyright protection against diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24420–24430, 2024.

[Zong *et al.*, 2024] Wei Zong, Yang-Wai Chow, Willy Susilo, Joonsang Baek, Jongkil Kim, and Seyit Camtepe. Ipremover: A generative model inversion attack against deep neural network fingerprinting and watermarking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 7, pages 7837–7845, 2024.