Not Just Change the Labels, Learn the Features: Watermarking Deep Neural Networks with Multi-View Data

Yuxuan Li[1], Sarthak Kumar Maharana[2], and Yunhui Guo[2]

[1] Harbin Institute of Technology, China
[2] The University of Texas at Dallas, Richardson, USA
lyxzcx@outlook.com, {skm200005, yunhui.guo}@utdallas.edu

Abstract. With the increasing prevalence of Machine Learning as a Service (MLaaS) platforms, there is a growing focus on deep neural network (DNN) watermarking techniques. These methods are used to facilitate the verification of ownership for a target DNN model to protect intellectual property. One of the most widely employed watermarking techniques involves embedding a trigger set into the source model. Unfortunately, existing methodologies based on trigger sets are still susceptible to functionality-stealing attacks, potentially enabling adversaries to steal the functionality of the source model without a reliable means of verifying ownership. In this paper, we first introduce a novel perspective on trigger set-based watermarking methods from a feature learning perspective. Specifically, we demonstrate that by selecting data exhibiting multiple features, also referred to as *multi-view data*, it becomes feasible to effectively defend functionality stealing attacks. Based on this perspective, we introduce a novel watermarking technique based on Multi-view dATa, called MAT, for efficiently embedding watermarks within DNNs. This approach involves constructing a trigger set with multi-view data and incorporating a simple feature-based regularization method for training the source model. We validate our method across various benchmarks and demonstrate its efficacy in defending against model extraction attacks, surpassing relevant baselines by a significant margin. The code is available at https://github.com/liyuxuan-github/MAT.

1 Introduction

Deep neural networks (DNNs) have demonstrated superior performance across various tasks, including computer vision [16,17,24]and natural language processing [10, 20, 31], and speech recognition [9, 14, 51]. Their remarkable performance has led to widespread adoption in Machine Learning as a Service (MLaaS) platforms, allowing users to submit input data and retrieve model-generated outputs from the cloud [23, 42, 48].

MLaaS empowers users to leverage the strong capabilities of DNNs but also exposes MLaaS providers to potential risks. Specifically, as these providers invest significant resources in developing the source model, protecting their intellectual

property rights becomes a critical concern. Although the attackers do not have access to the source model nor the source training data, the attackers can still extract the DNN models' functionality using black-box functionality stealing attacks [34,39,40,46]. For example, a model extraction attack leverages a surrogate model for imitating the output of the source model on a surrogate dataset for functionality stealing [39].

To enable the ownership verification of a stolen model, one of the most effective approaches is to use a trigger set for model watermarking [1, 2, 19, 29, 36, 37, 47, 50]. Specifically, during the training of the source model, the model owner utilizes a pre-selected trigger set, where the labels of the samples are known only to the owner. To verify ownership, the model owner assesses the output of the suspicious model. If the output aligns with the intended labels, the model owner can confidently assert ownership of the suspicious model.

Using a trigger set for model watermarking offers several advantages over alternative methods. Firstly, it removes the necessity for the model owner to have white-box access to the suspicious model, making it applicable in scenarios with restricted access. Secondly, it avoids the need to modify the architecture of the source model, simplifying the deployment process. However, it is crucial to note that trigger set-based watermarking methods are still susceptible to model extraction attacks [32]. After extraction, the stolen model may fail to produce the intended labels on the trigger set, effectively removing the watermark.

Several recent efforts have sought to enhance the efficacy of trigger set-based watermarking methods [22, 27, 36]. For instance, a margin-based watermarking technique was introduced recently in [22] to optimize the margin of samples in the trigger set through projected gradient ascent. The main idea is that by emulating the decision boundary of the source model, the surrogate model can replicate predictions on the trigger set from the source model. However, this approach encounters two challenges. Firstly, the use of projected gradient ascent significantly slows the training speed of the source model. Secondly, in a more realistic attack scenario where the attacker does not leverage the source data [2], it is difficult for the margin-based approach to effectively watermark the model.

In this paper, we first introduce a novel perspective on trigger set-based watermarking methods based on feature learning. Specifically, we demonstrate the efficacy of employing multi-view data [3] as the trigger set. Multi-view data, defined as data exhibiting diverse features, is common in practice. For instance, a given horse image may possess the shape of a horse but mirror the color of a dog, exhibiting two distinct features. In model extraction attacks, we demonstrate that when the source model utilizes color as a key feature for classifying a horse as a dog, the output of the source model on this multi-view trigger set can be successfully transferred to the surrogate model. This transfer makes the removal of the watermark a challenging task. Building on this perspective, we propose a simple approach to extract multi-view data from the source data for constructing the trigger set based on logit margin. To enhance the efficacy of using the trigger set, we additionally introduce a feature regularization loss, aiming to encourage the model to learn the desired features from this specific trigger set. Aligning

with recent advancements in trigger-set-based watermarking methods [2, 22], we conduct experiments across various widely used benchmarks. Our results demonstrate the superior performance of the proposed method compared to relevant baselines.

The contributions of the paper can be summarized as follows: **1)** We present a novel perspective on trigger set-based watermarking methods, aiming to understand the conditions under which trigger set-based watermarking can effectively defend model extraction attacks. To the best of our knowledge, this perspective is novel in understanding trigger set-based watermarking methods. **2)** We introduce an efficient trigger set-based watermarking approach, called MAT, by constructing a trigger set comprising multi-view data. The proposed method is simple to implement and does not require any changes to model architectures or the training process. **3)** We evaluate MAT against various attack methods and the results demonstrate that MAT enables reliable ownership identification in situations where baseline methods fail.

2 Related Work

Watermarking Deep Neural Networks. Watermarking [15] was originally proposed to prove ownership of multimedia. The first model watermarking method [44] is proposed to protect intellectual property by adding watermarks to model parameters with a parameter regularizer during training. Based on this, more methods [4, 45] have been proposed to make model watermarks more difficult to detect and remove. However, these white-box model watermarking methods usually require access to the specific parameters of the suspect model to verify whether it contains a watermark or not. This is usually difficult to do in practical scenarios. Inspired by the backdoor attack technique [12], trigger set-based model watermarking methods [1, 50] are proposed to verify model ownership in black-box scenarios by simply querying the accuracy of a suspicious model on a specific trigger set. To make the watermark more robust, CEM [33] uses the trigger set constructed by conferrable adversarial examples when training the model, while margin-based method [22] uses adversarial training to maximize the margins of samples from the trigger set. The use of randomized smooth [5] during the training process has also been proven [2] to increase the non-removable watermark of the model.

Model Stealing. Although model watermarking is very effective in verifying ownership, it becomes vulnerable when adversaries employ attacks to attempt to remove the watermark. In the scenario of a white-box attack, the adversary can obtain the parameter weights of the model, allowing classic defense mechanisms against backdoor attacks, such as fine-tuning and pruning [30], to easily eliminate the model watermark. In a more realistic case, black-box attacks such as distillation [18] and model extraction [39] can also steal clean models without watermarks through API queries of the source model. In particular, model extraction [39] is a functionality stealing method and is regarded as the most powerful black box attack currently [22,32]. Some recent work [2,22] claims to be

resistant to functionality stealing, however under some restrictions. For example, random smoothing [2] requires limiting the size of perturbation for functionality stealing, while margin-based [22] requires the attacker to use the same training data set as the source model. Our approach effectively mitigates functionality theft in the most challenging and realistic scenarios. This includes situations where the attacker can employ an entirely different dataset, either mirroring the distribution of the source model training set or utilizing a completely unrelated dataset for model extraction.

3 Background

Threat model. Similar to existing trigger set-based watermarking methods [2,22], we assume that adversaries aim to emulate the functionality of the source model using surrogate data, possibly from a distribution distinct from the source dataset used to train the model. They are aware of watermarking techniques but lack knowledge of the trigger set. Meanwhile, our defense operates in a black-box scenario, where defenders seek to establish ownership of the suspected model through a trigger set. Defenders can query predicted probabilities from the suspected model but may not have access to its parameters.

Trigger set-based approach for watermarking. Given a source dataset $S = (x_i, y_i)_{i=1}^n$ with K classes, with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, where both $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are sampled from a joint distribution $\mathbb{P}_{x,y}$, a specific source model $M_\theta : \mathcal{X} \to \mathbb{R}^K$ is trained by minimizing the loss function ℓ on the source dataset,

$$\ell(M_\theta; S) = \frac{1}{|S|} \sum_{(x,y)\in S} \ell(M_\theta(x), y) \tag{1}$$

Specifically, the output of the source model consists of logits represented as $M_\theta(x) = (M_\theta^1(x), M_\theta^2(x), ..., M_\theta^K(x))$, where $M_\theta^k(x)$ signifies the score assigned to class k by the model.

In functionality stealing attacks, the attacker aims to train a surrogate model $\hat{M}_{\hat{\theta}}$ to imitate the functionality of the source model [19]. In particular, the attackers leverage a surrogate data $\hat{S} = \{(x_j, y_j)\}_{j=1}^m$ to extract the output of the source model and leverage model extraction attack for functionality stealing,

$$\min_{\hat{\theta}} \ell_{\text{extraction}}(\hat{\theta}; \theta, \hat{S}) = \frac{1}{|\hat{S}|} \sum_{(x,y)\in \hat{S}} D_{KL}(\hat{M}_{\hat{\theta}}(x), M_\theta(x)) \tag{2}$$

where D_{KL} is the Kullback-Leibler divergence [6] between two probability distributions. Note that the logits need to be normalized via a softmax function. Although the attacker does not have access to the source dataset or source model parameters, the functionality of the source model can still be stolen based on the output of the source model on the surrogate data [22, 39].

Ownership Identification with a Trigger Set. To protect intellectual property, the owner of the source model must assess whether a particular surrogate

model reproduces the functionality of the source model, a process commonly referred to as ownership identification [4, 7, 41, 47]. One of the most commonly used ownership identification methods is based on watermarking using a trigger set [1, 2, 19, 22, 36]. In particular, for training the source model, the model owner randomly samples a subset of samples $\{(x_k, y_k)\}_{k=1}^{q}$ from the source dataset S without replacement and replaces the true label y_k with $\hat{y}_k \neq y_k$ to construct a trigger set $D_t = \{(x_k, \hat{y}_k)\}_{k=1}^{q}$. Meanwhile, the trigger set is excluded from the original source dataset S and the remaining clean training set is denoted as $D_c = \{(x_i, y_i)\}_i^m$, where $q<m$. With the additional trigger set, the model owner trains the source model by minimizing the following loss,

$$\min_{\theta} \ell(M_\theta; D_c) + \ell(M_\theta; D_t) \tag{3}$$

Although the labels of the trigger set are not the true labels, by minimizing the above loss the source model can still achieve 100% accuracy on the trigger set due to the high capacity of the DNNs [49]. As the labels of the trigger set are randomly assigned, a conventionally trained model is anticipated to attain 0% accuracy on the trigger set when assessed using the modified labels. Should a surrogate model be trained to mimic the output of the source model, and subsequently classify the image x_k in the trigger set as $\hat{y}_k$, the owner could claim that the source model has been stolen by the attacker.

Ideally, the model owner expects the surrogate model to produce the desired label for the trigger set. However, as outlined in Eq. 2, the attack is based solely on surrogate data for functionality stealing. Consequently, there is no guarantee that the surrogate model will accurately extrapolate its predictions to the trigger set, especially considering the potential lack of correlation between the assigned labels and the images within it.

4 A New Perspective for Ownership Identification

Conceptually, because the attacker does not depend on the source data for extraction, transferring the predictions of the trigger set from the source model to the surrogate model can only rely on the output of the source model on the *surrogate data*—a task that seems challenging. Here, we present a simple thought experiment to demonstrate the potential feasibility of such a transfer. Suppose we have an image of a horse that shares certain attributes, such as color, with those of a dog. If we select this image as the trigger set and assign the label "dog" to it, we create a scenario where the attributes of a horse are mislabeled as those of a dog. During the training of the source model using Eq. 1, the model is likely to rely on color cues in its classification process to identify the image as a dog. In this specific scenario, color serves as the exclusive distinguishing feature for classifying the image into the dog category. Now, in the context of a model extraction attack, let's assume there exists a dog image within the surrogate dataset that shares the same color as the horse image in the trigger set. The source model, when presented with this dog image, would predict it as a dog,

and this prediction would transfer to the surrogate model based on Eq. 2. As a result, the surrogate model, having adopted this prediction, would subsequently categorize the horse image within the trigger set as a dog based on the color cue. To make the above idea concrete, we will leverage the following definition of multi-view data,

Multi-view data. The multi-view hypothesis [3] states that a given sample may exhibit multiple distinct features which can be used for classification. In the example above, the horse image exhibits both the distinctive shape characteristics of a horse and the color attributes of a dog. Suppose we have two classes: dog and horse, for simplicity. We assume each class c has one feature $v_c \in \mathbb{R}^{p\times 1}$ which represents the mean feature of this class. We further assume the feature vectors of different classes are orthogonal, *i.e.*, $v_0^T v_1 = 0$. For the sample x_i^c belonging to class c, if the sample exhibits features from both classes, then we can represent the feature of this sample as $f_i^c = w_i^0 v_0 + w_i^1 v_1$, where $w_i^0 \in \mathbb{R}$ and $w_i^1 \in \mathbb{R}$ are some constant weights. For example a given horse image may resemble a dog such that the feature of the horse can be written as $f_{horse} \approx 0.2v_0 + 0.8v_1$. As shown in [3], multi-view data commonly exists in real-world datasets, as natural images often possess diverse features that can be exploited for classification.

Watermarking using multi-view data. Here we show that multi-view data can be naturally used for constructing the trigger set. Assume that we assign a dog class to the horse image. Consider a simple binary classifier with weights $y = Wx + b$, where $W \in \mathbb{R}^{2\times p}$ and $b \in \mathbb{R}^2$. We use W_i to denote the i-th row of W. Since the trigger set is trained alongside the clean data and is relatively small in comparison, it is reasonable to assume that the classifier can perfectly classify the two classes and the weights are perfectly aligned with the corresponding class, that is, $\cos(W_i, v_i) = 1$. Given the sample x_i^1 with feature $f_i^1 = w_i^0 v_0 + w_i^1 v_1$, the logit can be computed as,

$$\begin{aligned} z &= W f_i^1 + b = w_i^0 * W v_0 + w_i^1 * W v_1 + b \\ &= \begin{bmatrix} w_i^0 W_0 v_0 + w_i^1 W_0 v_1 + b_1 \\ w_i^0 W_1 v_0 + w_i^1 W_1 v_1 + b_2 \end{bmatrix} \end{aligned} \tag{4}$$

The logit will be converted into probabilities using a softmax function,

$$\text{softmax}(z)_k = \frac{e^{z_k}}{\sum_{j=1}^{2} e^{z_j}}, \quad k = 0, 1 \tag{5}$$

If we assign a class 0 to this image, the binary cross-entropy loss can be simplified to $\ell = -\log(\text{softmax}(z)_0)$. Based on our assumptions, $\text{softmax}(z)_0 = \frac{e^{w_i^0 W_0 v_0 + b_1}}{\sum_{j=1}^{2} e^{z_j}}$ as $W_0 v_1 = 0$. Thus the source model will leverage the feature v_0 for classifying the images. In a model extraction attack, when a sample with feature v_0 is provided, the source model predicts class 0. This prediction is then leveraged to train a surrogate model, resulting in the surrogate model predicting the same label for the sample within the trigger set.

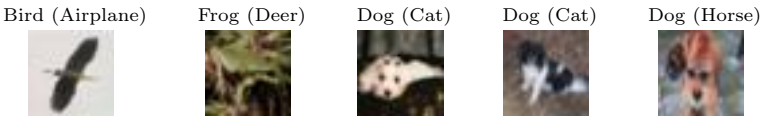


Fig. 1: Some sample images in the trigger set selected by MAT on CIFAR10. The images exhibit features from different classes as expected. The true classes, along with the classes having the second-highest scores, are displayed in parentheses.

5 MAT

Based on the above analysis, we introduce MAT, a novel watermarking method using trigger sets to defend against model extraction attacks. MAT incorporates a novel trigger sample selection method and a new approach to source model training. In the Supplementary, we illustrate our entire training pipeline.

Margin-based Trigger Set Selection. While our analysis in Section 4 demonstrates the effectiveness of transferring the source model's predictions on the trigger set to the surrogate model through the utilization of data with features from different classes, it remains unclear how to identify samples exhibiting multi-view features within a given source dataset. To tackle this challenge, we introduce a simple and effective method for extracting multi-view data from the source dataset and subsequently employ them to construct the trigger set.

Given a source model M_θ and the logits $M_\theta(x)$ for a sample x with label y, if the sample exhibits features from different classes, then the model will assign high scores to the respective classes. This intuition can be captured by the logit margin loss which is defined as, $LM = \max_{j \neq y} M_\theta^j(x) - M_\theta^y(x)$. If the model makes the correct prediction, a large logit margin loss indicates that the features of sample x comprise characteristics of both the true classes and the class with the second-highest score. Thus, we can select a trigger set consisting of samples with large logit margin loss. In practice, we first train the model on the source dataset S using Eq. 1 and then select the top q samples with the largest logit margin loss as the trigger set. The label of each selected sample is *modified* to the class $\hat{y} = \arg\max_{j \neq y} M_\theta^j(x) - M_\theta^y(x)$. Fig. 1 shows the selected images using the proposed margin-based trigger set selection on CIFAR10. It can be observed that the images are difficult to classify and exhibit features from different classes.

In Fig. 2, we show a simple example with two classes: circle and rectangle. With the logit margin loss, we select samples close to the decision boundary that exhibit features of both the circle and rectangle classes. Fig. 2 a) shows the decision boundary of the source model on the source data. Fig. 2 b) shows the decision boundary of the source model after training with the clean data and the chosen trigger set. It is noteworthy that this selection not only enhances watermarking performance but also impose minimal affect on clean accuracy, given that all other samples from the rectangle class are considerably far from the chosen trigger set. This intuition is validated by the experimental results.

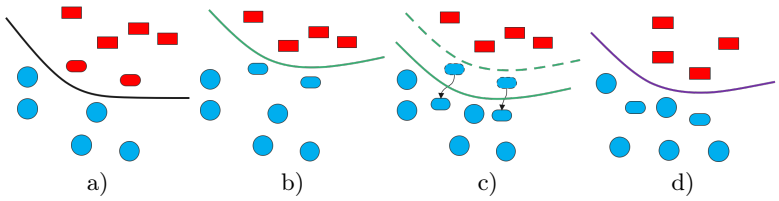


Fig. 2: a) Source training (Eq. 1), b) Trigger set training (Eq. 3), c) Feature regularization (Eq. 6) d) Surrogate model (Eq. 2). The proposed MAT identifies samples close to the decision boundary as the trigger set and adjusts the features of this set to align closely with the class center of the modified label.

Feature Regularization. While the chosen trigger set possesses multi-view features, it is still important to encourage the source model to learn the desired features effectively. In Fig. 2 b), ideally, the source should employ the circle feature to classify samples in the trigger set. For instance, in a model extraction attack involving a circular image, the source model should be capable of transferring predictions from the trigger set to the surrogate model. Thus, we propose to enhance the model's feature learning by incorporating a feature regularization loss. In particular, for a given sample $(x_k, \hat{y}_k)$ in the trigger set, we aims to push the feature of the sample to be close to the mean feature of class $\hat{y}_k$. This can be achieved by minimizing the following loss,

$$\min_{\theta} \ell(\theta; D_c) + \ell(\theta; D_t) + \alpha \frac{1}{|D_t|} \sum_{(x_k, \hat{y}_k) \in D_t} \|f(x_k) - f_{\hat{y}_k}\|_2 \tag{6}$$

where α is a balance parameter. As the mean feature of the class $\hat{y}_k$ is not directly available, we employ an approximation by computing them through the average features of samples belonging to the class $\hat{y}_k$. Specifically, we calculate the average features for each class during the previous epoch of model training and utilize these averages to regularize the feature learning of the trigger set in the current epoch. Fig. 2 c) demonstrates that additional feature regularization pushes trigger set features towards the circle class center, shifting the source model's decision boundary accordingly. In Fig. 2 d), the surrogate model's decision boundary after a model extraction attack aligns with the source model's predictions on the trigger set, as it aims to replicate them on the surrogate dataset. We include the discussions and results of an alternative regularization method in the Supplementary.

Crucially, MAT does not require the surrogate data to exhibit the features of the source dataset to successfully identify ownership. This is because the output of the source model for any given input can be used to approximate the decision boundary of the source model. By shaping the decision boundary of the source model using the chosen trigger set, the prediction of the source model on the trigger set can be transferred to the the surrogate model. We experimentally validate this in Sec. 6.4.

Ownership Verification with Hypothesis Test. Our pipeline for ownership verification is similar to existing watermarking-based methods [13,22,26,28]. The defender can assert ownership of suspicious models by demonstrating statistically significant deviations in their behavior compared to a benign model. Specifically, we train a model M_θ^c directly on the clean dataset D_c, which serves as our benign model. We then evaluate the predictions of both the benign model M_θ^c and the suspicious model $\hat{M}_{\hat{\theta}}$ on the trigger set D_t, denoting their predictions as P and $\hat{P}$, respectively. These predictions can be represented as a binomial distribution, which indicates whether watermarked data is classified as the target class [19]. Validation is conducted through a two-sample t-test, comparing the models' predictions on the trigger set. If the resulting p-value is less than the significance level, defenders can claim ownership of the model.

6 Experiments

6.1 Experimental Settings

Datasets. Building upon prior studies [2, 22], we utilize the CIFAR10 [25] and CIFAR100 [25] datasets. Additionally, we extend our evaluation to include ImageNet [8], a large-scale image dataset that challenges existing watermarking techniques. Following [2], we use half of the training dataset as the source data, with the remaining half serving as the surrogate dataset.

Baselines. We compare MAT with the following baseline approaches:

- **Base** [50]. Randomly selects a trigger set and trains the model using Eq. 3.
- **Randomized Smoothing (RS)** [2]. It employs randomized smoothing to achieve certified robustness when the modifications to the source model parameters θ are limited in size.
- **Margin-based method** [22]. It maximizes the margin of the samples in the trigger set using the projected gradient descent.
- **Datasets Inference (DI)** [36]. It verifies ownership by determining whether the suspected model contains private knowledge of the source model's training dataset.
- **Embedded External Features (EEF)** [27]. It transfers the data style of the trigger set by embedding external features.

Metrics. We evaluate the accuracy of the surrogate model $\hat{M}_{\hat{\theta}}$ on trigger set D_t, and report the p-value using a two-sample t-test, where a small p-value indicates distinguishability between the stolen and benign models. Additionally, we report the clean accuracy of both the source and surrogate models.

Attack methods. We consider several commonly used attack methods:

- **Soft-label model extraction attack**. The attacker leverages the predictions of the source model on the surrogate model as in Eq. 2.
- **Hard-label model extraction**. Instead of depending on the soft labels $M_\theta(x) = (M_\theta^1(x), M_\theta^2(x), ..., M_\theta^K(x))$, the attacker exploits the hard label $y_{hard} = \arg\max_k M_\theta^c(x)_{k=1}^K$. The hard label will be converted into a one-hot representation, which then replaces the soft label in Eq. 2 for model extraction attack.

Table 1: Our proposed MAT achieves the best watermarking performance on CIFAR10 compared with the baselines. Even after hard-label model extraction attack, MAT still achieves 56% accuracy on the trigger set.

	Source Acc. (%)	Soft-Label			Hard-Label		
		Surro. Acc. (%)	Trig. Acc. (%)	p-Value	Surro. Acc. (%)	Trig. Acc. (%)	p-Value
Base [50]	90.40	89.07	0	10^{-1}	85.22	4	10^{-1}
RS [2]	91.10	89.74	2	10^{-1}	85.75	2	10^{-1}
Margin-based [22]	83.80	86.51	46	10^{-14}	84.45	10	10^{-3}
DI [36]	-	-	-	10^{-6}	-	-	10^{-3}
EEF [27]	-	-	-	10^{-7}	-	-	10^{-4}
MAT (no reg.)	91.70	89.44	52	10^{-4}	86.05	37	10^{-1}
MAT	87.90	88.50	**74**	10^{-11}	85.40	**56**	10^{-4}

- **Fine-tuning** [30]. Given the access to the source model, the attacker fine-tunes the source model on a clean data set under the same data distribution.
- **Fine-pruning** [30]. Given access to the source model, the attacker prunes the neurons in the last convolutional layer based on their activation levels using a small batch of clean data, and then proceeds with fine-tuning.

Implementation details. We utilize ResNet18 [17] as the source and surrogate models for CIFAR10 and CIFAR100, and ViT-base-patch16-384 [11] for ImageNet. Additional results with alternative architectures are provided in the Supplementary. We assume that all surrogate models are randomly initialized, as attackers do not have access to the parameters of the source model. In all the experiments, the size of the trigger set is 100. We first train a clean model on the source data for 200 epochs for extracting multi-view data for constructing the trigger set. Then, a source model is trained on the clean data and the trigger set data as in Equation 6. We follow the same training strategies as in [22]. The source model is trained for 200 epochs and optimized using SGD [21] with an initial learning rate of 0.1. A linear decay is used to decay the learning rate every 50 epochs. For the details of various attack methods, please see the Supplementary.

6.2 Results on Functionality Stealing

Tables 1, 2 and 3 present the source accuracy (Source Acc.), surrogate accuracy (Surro. Acc.), trigger set accuracy (Trig. Acc) and p-value for all the methods on CIFAR10, CIFAR100 and ImageNet, respectively. *For the statistical testing based methods - DI [36] and EEF [27], we report only the p-values.* It is evident that following a model extraction attack, all the baselines exhibit a substantial number of errors on the trigger set, posing a significant challenge to ownership verification. Specifically, in the case of a hard-label model extraction attack, none of the baselines can transfer the prediction of the source model on the trigger set to the surrogate model, rendering ownership verification impossible.

In contrast, the proposed MAT effectively retains the majority of predictions on the trigger set. Specifically, when compared to the margin-based method,

Table 2: Our proposed MAT achieves the best watermarking performance on CIFAR100 compared with the baselines after soft-label model extraction attack. With the hard-label model extraction attack, the attack itself becomes more challenging, yet MAT still achieves higher trigger accuracy compared to other methods.

		Soft-Label			Hard-Label		
	Source Acc. (%)	Surro. Acc. (%)	Trig. Acc. (%)	p-Value	Surro. Acc. (%)	Trig. Acc. (%)	p-Value
Base [50]	64.30	63.99	2	10^{-1}	15.01	1	10^{-1}
RS [2]	66.80	64.85	0	10^{-1}	15.61	0	10^{-1}
Margin-based [22]	60.95	61.33	11	10^{-3}	15.74	1	10^{-1}
DI [36]	-	-	-	10^{-2}	-	-	10^{-1}
MAT (no reg.)	66.02	65.65	62	10^{-11}	15.44	**9**	10^{-1}
MAT	61.20	59.73	**77**	10^{-20}	15.19	**9**	10^{-1}

Table 3: Our proposed MAT achieves the best watermarking performance on ImageNet compared with the baselines.

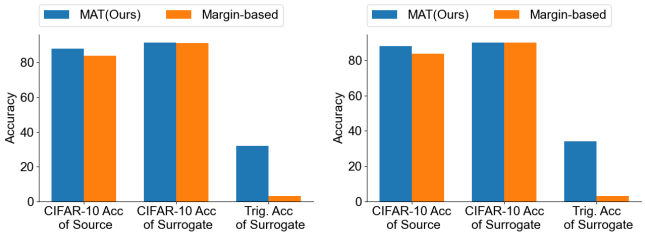
		Soft-Label			Hard-Label		
	Source Acc. (%)	Surro. Acc. (%)	Trig. Acc. (%)	p-Value	Surro. Acc. (%)	Trig. Acc. (%)	p-Value
Base [50]	75.30	74.44	1	10^{-1}	5.84	4	10^{-1}
Margin-based [22]	70.06	71.92	9	10^{-2}	5.68	1	10^{-1}
MAT (no reg.)	74.94	72.76	**82**	10^{-7}	5.36	**10**	10^{-1}
MAT	76.25	74.16	**82**	10^{-7}	5.84	**6**	10^{-1}

the trigger set accuracy experiences a substantial improvement, rising from 46% to 74%. Even when subjected to a hard-label model extraction attack, MAT still maintains a trigger set accuracy of 56% on CIFAR10. On CIFAR100, all baseline methods fail to maintain trigger set accuracy following a soft-label model extraction attack, whereas our proposed MAT demonstrates an accuracy of 77%. Similarly, MAT achieves much higher trigger set accuracy and a lower p-value on ImageNet, demonstrating its scalability to large image datasets.

Consistent with the findings in [2], we observe that all methods struggle to preserve trigger set accuracy following a hard-label model extraction attack. This phenomenon may be attributed to the challenges of the CIFAR100 dataset and ImageNet, coupled with the low accuracy of the source model on the surrogate dataset. We also present results for MAT (no reg.), which does not use feature regularization during training. MAT (no reg.) achieves higher clean accuracy but lower trigger set accuracy compared to MAT. Thus, MAT (no reg.) can be employed when prioritizing high clean accuracy over ownership verification.

6.3 Results with White-box Attacks

Model extraction attack is a black-box attack which assumes that the attacker cannot have access to the parameters of the source model. In some cases, the attacker may even know the parameters of the source model in which case the attacker can perform white-box attack. In this section, we examine two white-



a) Fine-tuning b) Fine-pruning

Fig. 3: MAT significantly outperforms the margin-based approach in watermarking effectiveness when facing white-box attacks.

Table 4: MAT still outperforms the margin-based approach with heterogeneous surrogate data and architecture.

Method	Surrogate Clean Acc. (%)	Trigger Set Acc. (%)
Surrogate Dataset with SVHN		
Margin-based [22]	82.53	39
MAT	82.43	**67**
Surrogate Model with VGG11		
Margin-based [22]	85.69	43
MAT	85.77	**62**

box attacks, namely fine-tuning and fine-pruning. As illustrated in Fig. 3, our proposed MAT demonstrates the ability to sustain a reasonably high trigger set accuracy, even when subjected to white-box attacks. This represents a significant strength of MAT in contrast to margin-based approaches, which lack the capability to effectively defend against such attacks.

6.4 Ablation Studies

Heterogeneous Surrogate Dataset & Model. MAT does not need to assume that the surrogate data possess the features of the source dataset. In this section, we address a more realistic scenario where the attacker lacks knowledge of the distribution of the source data or the architecture of the source model. Instead of the CIFAR10 dataset, we employ the SVHN dataset [38] as the surrogate dataset, maintaining ResNet18 as the architecture for the surrogate model. Additionally, we explore the use of the CIFAR10 dataset as the surrogate dataset while employing VGG11 [43] as the architecture for the surrogate model. Table 4 reports that in both the cases, the proposed MAT consistently outperforms the margin-based approach in terms of trigger set accuracy. Visualization results using t-SNE [35] can be found in the Supplementary.

Different trigger set selection strategies. In this section, we further demonstrate the effectiveness of the proposed trigger set selection strategy in comparison to alternative approaches. Specifically, we examine the following strategies:

- Random selection (Random): This strategy randomly selects a subset of samples from the source dataset to form the trigger set.
- Highest confidence (Highest conf.): This strategy selects samples with the smallest logit margin loss over those with the largest.

For both of these strategies, we maintain consistency with the labeling strategy employed by MAT. The results presented in Table 5 and 6 show the performance of these strategies alongside MAT. With the adoption of the proposed trigger set selection strategy, MAT significantly outperforms the other two approaches, emphasizing the effectiveness of choosing multi-view data as the trigger set.

Table 5: Results of trigger set selection strategies on CIFAR10.

	Clean Acc. Source	Clean Acc. Surrogate	Trig. Acc. (%)
Random	87.73	87.74	12
Highest conf.	82.68	83.80	1
MAT	87.90	88.50	**74**

Table 6: Results of trigger set selection strategies on CIFAR100

	Clean Acc. Source	Clean Acc. Surrogate	Trig. Acc. (%)
Random	59.14	58.97	56
Highest conf.	58.75	59.23	9
MAT	61.20	59.73	**77**

Different trigger set labelling strategies. In addition to choosing the trigger set, how to modify the label of the trigger set is also critical for transferring the prediction of the source model on the trigger set to the surrogate model. To further show the effectiveness of our labelling strategy, we examine the following two alternatives:

- Random labelling (Random): Randomly assign one of the labels, excluding the true label, to the selected sample.
- Minimize confidence (Min. conf.): Identify the label that minimizes the logit margin loss. The label can be calculated as $\tilde{y} = \arg\min_{j \neq y} \left(M_\theta^j(x) - M_\theta^y(x) \right)$.

For both of these strategies, we maintain consistency with the trigger set selection strategy employed by MAT. The results presented in Table 7 and 8 showcase the performance of these strategies alongside MAT. With the adoption of the proposed trigger set labelling strategy, MAT significantly outperforms the other two approaches. In particular, MAT assigns the label to the sample in the trigger that the model finds most confusing, making it easier for the model to learn the distinctive features of the assigned classes. Consequently, it becomes feasible to transfer the model's predictions on the trigger set to the surrogate model.

Table 7: Results of trigger set labelling strategies on CIFAR10.

	Clean Acc. (%) Source	Clean Acc. (%) Surrogate	Trig. Acc. (%)
Random	85.66	85.82	32
Min conf.	86.92	86.91	2
MAT	87.90	88.50	**74**

Table 8: Results of trigger set labelling strategies on CIFAR100.

	Clean Acc. (%) Source	Clean Acc. (%) Surrogate	Trig. Acc. (%)
Random	57.43	58.45	15
Min conf.	55.94	56.61	9
MAT	61.20	59.73	**77**

Effect of feature regularization. While the inclusion of the feature regularization loss enhances feature learning on the trigger set, it is possible that it may compromise clean accuracy. In Fig. 4, we explore the impact of varying the balance parameter α within the range of $\{0.0, 0.01, 0.05, 0.1\}$. The findings indicate that an increase in α does indeed negatively affect clean accuracy. This can be attributed to the fact that features of different classes are not strictly orthogonal, leading to a decrease in the clean accuracy. Despite this, an increased value of α significantly enhances the effectiveness of the trigger set for watermarking. In practice, to achieve a balance between clean accuracy and watermarking effectiveness, a smaller value of α is preferred to guide the feature learning process. It also worth noting that even without the feature regularization loss, MAT still outperforms the margin-based watermarking method which further demonstrates the effectiveness of the trigger set selection strategy.

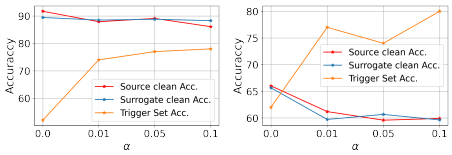


Fig. 4: A large α enhances feature regularization, thereby resulting in improved watermarking performance.

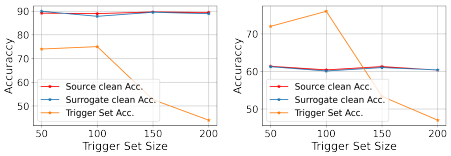


Fig. 5: MAT can achieve strong watermarking performance with a small trigger set consisting of multi-view data.

Size of the trigger set. In this section, we vary the trigger set size, choosing from $\{50, 100, 150, 200\}$, to assess its impact. Fig. 5 shows that changing the trigger set size minimally affects the clean accuracy of both the source and surrogate models. However, the accuracy on the trigger set drops significantly as its size increases. This can be understood from a feature learning perspective: with limited samples demonstrating multiple features, expanding the trigger set size adds only marginal multi-view data. For example, in CIFAR10, a trigger set size of 200 yields 88 correctly classified samples, while a size of 100 yields 75. Notably, the absolute number of correctly classified samples remains similar.

7 Conclusion

In this paper, we propose a novel trigger set-based watermarking method, called MAT, using multi-view data. The proposed MAT is easy to interpret and highly effective for watermarking deep neural networks (DNNs). Extensive experiments have been conducted to demonstrate the efficacy of MAT. The findings indicate that MAT not only effectively safeguards against model extraction attacks but also exhibits robustness in the face of whitebox attacks, including fine-tuning and fine-pruning. In summary, MAT stands out as a robust watermarking method, offering enhanced protection for the intellectual property of DNNs.

Acknowledgements. We would like to thank the anonymous reviewers for their helpful comments. This project was supported by a grant from the University of Texas at Dallas.

References

1. Adi, Y., Baum, C., Cisse, M., Pinkas, B., Keshet, J.: Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In: 27th USENIX Security Symposium (USENIX Security 18). pp. 1615–1631 (2018)
2. Bansal et al., A.: Certified neural network watermarks with randomized smoothing. In: ICML (2022)
3. Allen-Zhu, Z., Li, Y.: Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. arXiv preprint arXiv:2012.09816 (2020)
4. Chen, H., Rohani, B.D., Koushanfar, F.: Deepmarks: A digital fingerprinting framework for deep neural networks. arXiv preprint arXiv:1804.03648 (2018)
5. Chiang, P.y., Ni, R., Abdelkader, A., Zhu, C., Studer, C., Goldstein, T.: Certified defenses for adversarial patches. arXiv preprint arXiv:2003.06693 (2020)
6. Cover, T.M.: Elements of information theory. John Wiley & Sons (1999)
7. Darvish Rouhani, B., Chen, H., Koushanfar, F.: Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. pp. 485–497 (2019)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. Deng, L., Platt, J.: Ensemble deep learning for speech recognition. In: Proc. interspeech (2014)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
12. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)
13. Guo, J., Li, Y., Wang, L., Xia, S.T., Huang, H., Liu, C., Li, B.: Domain watermark: Effective and harmless dataset copyright protection is closed at hand. Advances in Neural Information Processing Systems **36** (2024)
14. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al.: Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567 (2014)
15. Hartung, F., Kutter, M.: Multimedia watermarking techniques. Proceedings of the IEEE **87**(7), 1079–1107 (1999)
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

18. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
19. Jia, H., Choquette-Choo, C.A., Chandrasekaran, V., Papernot, N.: Entangled watermarks as a defense against model extraction. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 1937–1954 (2021)
20. Kamath, U., Liu, J., Whitaker, J.: Deep learning for NLP and speech recognition, vol. 84. Springer (2019)
21. Kiefer, J., Wolfowitz, J.: Stochastic estimation of the maximum of a regression function. The Annals of Mathematical Statistics pp. 462–466 (1952)
22. Kim, B., Lee, S., Lee, S., Son, S., Hwang, S.J.: Margin-based neural network watermarking. International Conference on Machine Learning (2023)
23. Kim, H., Kim, M., Seo, D., Kim, J., Park, H., Park, S., Jo, H., Kim, K., Yang, Y., Kim, Y., et al.: Nsml: Meet the mlaas platform with a real-world case study. arXiv preprint arXiv:1810.09957 (2018)
24. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
25. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
26. Li, Y., Bai, Y., Jiang, Y., Yang, Y., Xia, S.T., Li, B.: Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. Advances in Neural Information Processing Systems **35**, 13238–13250 (2022)
27. Li, Y., Zhu, L., Jia, X., Jiang, Y., Xia, S.T., Cao, X.: Defending against model stealing via verifying embedded external features. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 1464–1472 (2022)
28. Li, Y., Zhu, M., Yang, X., Jiang, Y., Wei, T., Xia, S.T.: Black-box dataset ownership verification via backdoor watermarking. IEEE Transactions on Information Forensics and Security **18**, 2318–2332 (2023)
29. Li, Z., Hu, C., Zhang, Y., Guo, S.: How to prove your model belongs to you: A blind-watermark based framework to protect intellectual property of dnn. In: Proceedings of the 35th Annual Computer Security Applications Conference. pp. 126–137 (2019)
30. Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. In: International symposium on research in attacks, intrusions, and defenses. pp. 273–294. Springer (2018)
31. Lopez, M.M., Kalita, J.: Deep learning applied to nlp. arXiv preprint arXiv:1703.03091 (2017)
32. Lukas, N., Jiang, E., Li, X., Kerschbaum, F.: Sok: How robust is image classification deep neural network watermarking? In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 787–804. IEEE (2022)
33. Lukas, N., Zhang, Y., Kerschbaum, F.: Deep neural network fingerprinting by conferrable adversarial examples. arXiv preprint arXiv:1912.00888 (2019)
34. Ma, C., Chen, L., Yong, J.H.: Simulating unknown target models for query-efficient black-box attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11835–11844 (2021)
35. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
36. Maini, P., Yaghini, M., Papernot, N.: Dataset inference: Ownership resolution in machine learning. arXiv preprint arXiv:2104.10706 (2021)

37. Namba, R., Sakuma, J.: Robust watermarking of neural network with exponential weighting. In: Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security. pp. 228–240 (2019)
38. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
39. Orekondy, T., Schiele, B., Fritz, M.: Knockoff nets: Stealing functionality of black-box models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4954–4963 (2019)
40. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security. pp. 506–519 (2017)
41. Quan, Y., Teng, H., Chen, Y., Ji, H.: Watermarking deep neural networks in image processing. IEEE transactions on neural networks and learning systems **32**(5), 1852–1865 (2020)
42. Ribeiro, M., Grolinger, K., Capretz, M.A.: Mlaas: Machine learning as a service. In: 2015 IEEE 14th international conference on machine learning and applications (ICMLA). pp. 896–902. IEEE (2015)
43. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
44. Uchida, Y., Nagai, Y., Sakazawa, S., Satoh, S.: Embedding watermarks into deep neural networks. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval. pp. 269–277 (2017)
45. Wang, T., Kerschbaum, F.: Robust and undetectable white-box watermarks for deep neural networks. arXiv preprint arXiv:1910.14268 **1**(2) (2019)
46. Wang, Y., Li, J., Liu, H., Wang, Y., Wu, Y., Huang, F., Ji, R.: Black-box dissector: Towards erasing-based hard-label model stealing attack. In: European Conference on Computer Vision. pp. 192–208. Springer (2022)
47. Yang, P., Lao, Y., Li, P.: Robust watermarking for deep neural networks via bi-level optimization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14841–14850 (2021)
48. Yao, Y., Xiao, Z., Wang, B., Viswanath, B., Zheng, H., Zhao, B.Y.: Complexity vs. performance: empirical analysis of machine learning as a service. In: Proceedings of the 2017 Internet Measurement Conference. pp. 384–397 (2017)
49. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM **64**(3), 107–115 (2021)
50. Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M.P., Huang, H., Molloy, I.: Protecting intellectual property of deep neural networks with watermarking. In: Proceedings of the 2018 on Asia conference on computer and communications security. pp. 159–172 (2018)
51. Zhang, Z., Geiger, J., Pohjalainen, J., Mousa, A.E.D., Jin, W., Schuller, B.: Deep learning for environmentally robust speech recognition: An overview of recent developments. ACM Transactions on Intelligent Systems and Technology (TIST) **9**(5), 1–28 (2018)