

Attention Distraction: Watermark Removal Through Continual Learning with Selective Forgetting

Author

Zhong, Q, Zhang, LY, Hu, S, Gao, L, Zhang, J, Xiang, Y

Published

2022

Conference Title

2022 IEEE International Conference on Multimedia and Expo (ICME)

Version

Accepted Manuscript (AM)

DOI

[10.1109/ICME52920.2022.9858930](https://doi.org/10.1109/ICME52920.2022.9858930)

Rights statement

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Downloaded from

<http://hdl.handle.net/10072/422365>

Griffith Research Online

<https://research-repository.griffith.edu.au>

ATTENTION DISTRACTION: WATERMARK REMOVAL THROUGH CONTINUAL LEARNING WITH SELECTIVE FORGETTING

Qi Zhong*, Leo Yu Zhang*^(✉), Shengshan Hu[†], Longxiang Gao^{‡§}, Jun Zhang[#], Yong Xiang*

*Deakin University; [†]Huazhong University of Science and Technology; [‡]Qilu University of Technology; [§]Shandong Computer Science Center; [#]Swinburne University of Technology
email: leo.zhang@deakin.edu.au

ABSTRACT

Fine-tuning attacks are effective in removing the embedded watermarks in deep learning models. However, when the source data is unavailable, it is challenging to just erase the watermark without jeopardizing the model performance. In this context, we introduce Attention Distraction (AD), a novel source data-free watermark removal attack, to make the model selectively forget the embedded watermarks by customizing continual learning. In particular, AD first anchors the model’s attention on the main task using some unlabeled data. Then, through continual learning, a small number of *lures* (randomly selected natural images) that are assigned a new label distract the model’s attention away from the watermarks. Experimental results from different datasets and networks corroborate that AD can thoroughly remove the watermark with a small resource budget without compromising the model’s performance on the main task, which outperforms the state-of-the-art works.

Index Terms— Watermarking, selective forgetting, continual learning, deep learning, intellectual property

1. INTRODUCTION

Deep neural network (DNN) watermarking, deriving wisdom from digital watermarking, protects the intellectual property of DNN models by endowing the model with the ability to trace illegal distributions. Since the first work proposed by Uchida *et al.* [1] in 2017, researchers have given a lot of enthusiasm to this field and a variety of solutions have been put forward [2–5]. The vulnerabilities of watermarks in DNN, meanwhile, have also attracted the attention of researchers.

The watermark removal attack based on fine-tuning technology is one of the most serious threats to watermarked models. It aims to make the victim model forget the embedded watermarks but preserve accuracy on the main task through fine-tuning the model with source training data or/and proxy data (unrelated data acts as a substitution or augmentation of the source training data) [6–9]. However, it is nontrivial to achieve such a problem containing two conflicting goals in

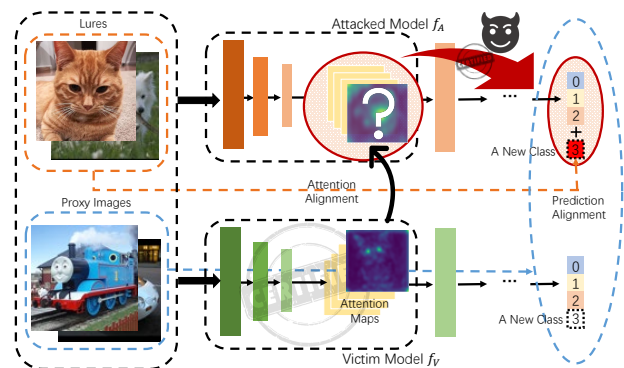


Fig. 1. The framework of AD. It first assigns a new class to the lure samples and feeds them into the victim model, which motivates the model to continue learning the lure task by minimizing the cross-entropy loss. It then fine-tunes unlabeled proxy samples by minimizing attention alignment and prediction alignment losses.

the source data-free regime due to the catastrophic forgetting phenomenon [10] of deep learning systems.

The work in [11] is the first to explore the feasibility of watermark removal in DNN using unlabeled data obtained from open sources. Its improved version, proposed in [12], uses a dataset transformation method called PST (Pattern embedding and Spatial-level Transformation) to preprocess the data before model fine-tuning. However, they either have no effect on particular watermarks or impair the model’s performance after erasing the watermark.

To mitigate this gap and solve the selective forgetting watermarks (SFW) problem in DNNs, we propose a novel continual learning-based [13] DNN watermark removal attack in the source data-free regime: Attention Distraction (AD). As depicted in Fig. 1, AD takes a small part of out-of-distribution (OOD) unlabeled samples as *lures* and assigns them an additional new class to distract the victim’s attention from the watermark task. The lures can be easily collected from open sources. Meanwhile, in the fine-tuning process, AD anchors model attention on the main task by aligning the intermediate-

layer attention maps and the final softmax outputs of proxy data with that of the original victim model. The rationale behind this is simple but effective: through continual learning, distracting the model’s attention from watermarks endows the model with the ability to selectively forget the knowledge about watermarks.

In summary, we make the following contributions:

- We propose AD, a novel continual learning-based DNN watermark removal framework that leverages attention anchoring and attention distraction strategies to achieve SFW, and present intuitive explanations for its efficacy.
- Unlike previous works that require access to source training data, computationally intensive, or only applicable to specific watermark types, AD requires no source training data and is resource-efficient as well as agnostic to watermarking mechanism types.
- Comprehensive empirical experiments on various benchmark datasets and network architectures validate that AD can effectively remove existing watermarks.

2. BACKGROUND AND RELATED WORKS

2.1. DNN Watermarking

This work focuses on the query-response-based watermarking [2–5], which is built on a more realistic assumption that only API access to the to-be-protected model is required. Mathematically, the general process of query-response-based DNN watermarking can be characterized as follows.

Considering a victim \mathcal{V} who trains a classifier $f_{\mathcal{V}}$ on a clean training data $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$ ($y^{(i)} \in \{0, 1, \dots, C - 1\}$). The watermark is embedded into the model by minimizing the following loss function:

$$\mathcal{L}(\theta_{\mathcal{V}}) = \mathbb{E}_{\mathbf{x} \in \mathcal{D} \cup \mathcal{T}} [\mathcal{L}_{CE}(f_{\mathcal{V}}(\mathbf{x}; \theta_{\mathcal{V}}), y)] + \alpha \cdot \mathcal{R}(\theta_{\mathcal{V}}), \quad (1)$$

where $\mathcal{T} = \{\mathbf{x}_t^{(i)}, y_t\}_{i=1}^M$ is a self-designed trigger set, $y_t \in \{0, 1, \dots, C - 1\}$ is the target label selected from existing classes, $\theta_{\mathcal{V}}$ represents all trainable parameters, \mathcal{L}_{CE} denotes the cross-entropy loss, and α is a parameter that controls the strength of the regularization term $\mathcal{R}(\theta_{\mathcal{V}})$ to prevent overfitting.

If \mathcal{V} suspects model theft, to confirm whether the suspected model contains the watermark, she queries this API and checks the accuracy

$$Acc = \text{Ver}_{\mathbf{x} \in \mathcal{T}}(f_{\mathcal{V}}(\mathbf{x}; \theta_{\mathcal{V}}), y_t), \quad (2)$$

where Ver is the verification function returns the probability for $f_{\mathcal{V}}(\mathbf{x}; \theta_{\mathcal{V}}) = y_t$. If $Acc > \tau$ (τ is a threshold close to 1), \mathcal{V} can claim her ownership of the model.

2.2. Watermark Removal Attacks in DNN

Previous researchers explored various fine-tuning-based attacks to remove DNN watermarks, which can be roughly divided into three main types according to the resources budget they can apply: the source data available regime [2, 3], the source data limited regime [6–9], and the source data-free regime [11, 12]. The following briefly introduces the fine-tuning based watermark removal attacks targeted at the source data-free regime.

The authors in [11] proposed a vanilla attack that employs unlabeled data obtained from open source as proxy data and annotates them using the original victim model as a labeling oracle. However, it causes the model’s performance on the main task to degrade since the pseudo labels of the proxy data provided by the model contain noise. To mitigate this drawback, Guo *et al.* in [12] proposed PST-FT attack, which combines an elaborately designed data transformation method called PST with fine-tuning. It is mainly based on the assumption that the fragile mapping between the triggers and the target label learned by the victim model can be easily destroyed after fine-tuning using the PST-transformed dataset. In the ownership verification phase, the clean instances processed by PST can still be recognized by the attacked model, while the triggers after the PST processing become unrecognizable.

3. PROBLEM STATEMENT

3.1. Attack Setting

The attacker \mathcal{A} has obtained the victim model $f_{\mathcal{V}}$, he is aware of the existence of watermarks, but he neither knows the watermarking method nor the knowledge of triggers, *e.g.*, the type of the triggers or the target label. He has a certain power of computing resources, but he cannot access the source training data \mathcal{D} . In other words, the attacker has no knowledge of the watermarks or the source training data, but he can access the victim model and do some manipulations on it.

3.2. Problem Formulation

We consider a victim model $f_{\mathcal{V}}$, which is watermarked by optimizing the object Eq. (1) using the clean dataset \mathcal{D} and the trigger set \mathcal{T} . We assume the victim model $f_{\mathcal{V}}$ has the state-of-the-art performance on both the main task (*i.e.*, classification on the clean dataset \mathcal{D}) and the watermark task (classification on the trigger set \mathcal{T}). The constraints are that only a certain number of auxiliary data \mathcal{D}_A are available in the AD attack, while the triggers or watermarking methods are agnostic.

The attacker aims to eliminate the watermark from the victim model. We view SFW as an instance of multi-task learning with conflicting objectives. Namely, using a continual learning approach with a limited resource budget to find

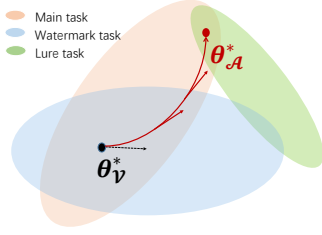


Fig. 2. Conceptual diagram for selective forgetting of the watermark task. Attention distraction happens whilst learning the lure task. The overlapping region of multiple ellipses represents the common solution space for those tasks.

an optimal self-balancing loss function that achieves high accuracy on the main task but low accuracy on the watermark task. In summary, to get the optimum classifier $f_{\mathcal{A}^*}$, the attacker can optimize

$$\min_{\theta_{\mathcal{A}}} \mathcal{L}_{AD}(\theta_{\mathcal{A}}), \text{ for } \mathbf{x} \in \mathcal{D}_A, \quad (3)$$

such that the following equations are hold:

$$|\text{Prob}_{\mathbf{x} \in \mathcal{T}}[f_{\mathcal{A}^*}(\mathbf{x}) = y_i] - \frac{1}{C}| \leq \epsilon, \quad (4)$$

$$|\text{Prob}_{\mathbf{x} \in \mathcal{D}}[f_{\mathcal{A}^*}(\mathbf{x}) = y] - \text{Prob}_{\mathbf{x} \in \mathcal{D}}[f_{\mathcal{V}}(\mathbf{x}) = y]| \leq \text{negl}, \quad (5)$$

where \mathcal{L}_{AD} is the desired objective loss function for the attacker, negl is a negligible value and ϵ determines to what extent the fine-tuned model forgets the original watermark.

4. ATTENTION DISTRACTION ATTACK

4.1. The Design Rationale

Recent studies on DNN visualization [14, 15] find that activation maps can be used to build heatmaps (*i.e.*, attention maps) that imply models' attention on images. And different models often share similar attention when making the right decisions for the same inputs [16]. Intuitively, in the fine-tuning/continual learning process, if we can maintain the attention maps for the main task while disrupting the attention maps for the watermark task, then we can achieve SFW.

As shown in Fig. 2, initially, the optimum $\theta_{\mathcal{V}}^*$ comes with acceptable error allowed by both the main task and original watermark task. The red arrows direct the path to update weights whilst learning the lure task. The continual learning of the lure task prompts the model weights to update in the direction that is conducive for the classification of the lures. The design of the lure task should ensure that it does not overlap with the watermark task, so it is more likely for the model to find a path that is only conducive for the classification of the main task and the new task, *i.e.*, the red arrows' direction. At the end, it can reach an optimum $\theta_{\mathcal{A}}^*$ with acceptable error fits for both the main task and lure task but not the original watermark task.

4.2. Auxiliary Dataset Crafting

As we have mentioned above, for scarce of source training data, to selectively forget watermarks, we need assistance from a certain number of auxiliary data, *i.e.*, proxy samples and lure samples.

In our method, the unlabeled proxy samples $\mathcal{D}_P = \{\mathbf{x}_p^{(i)}, \perp\}_{i=1}^{N_P}$ can be collected from the Internet, drawn from the in-distribution (ID) or OOD area of the source training data. The lure dataset \mathcal{D}_L can be drawn from the OOD area of the source training data, *e.g.*, randomly selected from natural images. To prevent feature collision between proxy data and lure data, there should be no intersection between them (*i.e.*, $\mathcal{D}_L \cap \mathcal{D}_P = \emptyset$) and their distributions are different. Then a new label Δ ($\Delta = C$) is assigned to the lure samples, and the whole lure set is $\mathcal{D}_L = \{\mathbf{x}_l^{(i)}, \Delta\}_{i=1}^{N_L}$. And the proxy samples and lure samples consist of the required auxiliary data, *i.e.*, $\mathcal{D}_A = \mathcal{D}_P \cup \mathcal{D}_L$.

4.3. Watermark Removal

Since the lure data \mathcal{D}_L is a new task for the victim model, manipulations of the model are necessary before the continual learning process. We add a new class to the output layer of $f_{\mathcal{V}}$ and make it fully connected to the layer beneath. The ground-truth label of the lures is $\Delta = C$, we formulate the prediction loss function for the lures as

$$\mathcal{L}_{\delta} = \mathbb{E}_{\mathbf{x} \in \mathcal{D}_L} [\mathcal{L}_{CE}(f_{\mathcal{A}}(\mathbf{x}; \theta_{\mathcal{A}}), C)]. \quad (6)$$

The authors in [2] have demonstrated that fine-tuning or retraining the deep layers (*i.e.*, the fully-connected layers) only has little effect to remove the watermark. Based on this consideration, we fine-tune the whole network. We rewrite the victim model into a composite function

$$f_{\mathcal{V}}(\mathbf{x}; \theta_{\mathcal{V}}) = F^{(2)}(F^{(1)}(\mathbf{x}; \theta_{\mathcal{V}}^{(1)}); \theta_{\mathcal{V}}^{(2)}), \quad (7)$$

in which $F^{(1)} : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H_l \times W_l \times C_l}$ denotes the attention mapping function that propagates an input \mathbf{x} through the network to the l -th ($l \in [1, 2, \dots, L]$) activation layer, $F^{(2)} : \mathbb{R}^{H_l \times W_l \times C_l} \rightarrow \mathbb{R}^{C+1}$ propagates the attention maps from $F^{(1)}$ to the softmax output layer. In our approach, we select a relatively deep layer, *e.g.*, the penultimate activation layer, to construct the sub-net $F^{(1)}$.

Since the source victim model $f_{\mathcal{V}}$ contains watermarks, their attention maps on the proxy data will inevitably, more or less, carry the watermark information. Accordingly, we penalize the attention maps of the model on proxy data and formulate our attention anchoring loss function as

$$\mathcal{L}_{AA} = \lambda_1 \cdot \mathcal{L}_v + \lambda_2 \cdot \mathcal{L}_a + \lambda_3 \cdot \mathcal{R}(\theta_{\mathcal{A}}), \quad (8)$$

where λ_i is the coefficient that regulates the importance of the item, \mathcal{L}_v is the vanilla prediction loss term defined as

$$\mathcal{L}_v = \mathbb{E}_{\mathbf{x} \in \mathcal{D}_P} [\mathcal{L}_{KL}(f_{\mathcal{A}}(\mathbf{x}; \theta_{\mathcal{A}}), f_{\mathcal{V}}(\mathbf{x}; \theta_{\mathcal{V}}))], \quad (9)$$

\mathcal{L}_{KL} is the Kullback-Leibler divergence, \mathcal{L}_a is the attention alignment loss term

$$\mathcal{L}_a = \mathbb{E}_{\mathbf{x} \in \mathcal{D}_P} \left[\|F^{(1)}(\mathbf{x}; \theta_A) - F^{(1)}(\mathbf{x}; \theta_V)\|_2 \right], \quad (10)$$

and $\mathcal{R}(\theta_A)$ is the attention penalty term for the proxy data

$$\mathcal{R}(\theta_A) = \mathbb{E}_{\mathbf{x} \in \mathcal{D}_P} \left[\|F^{(j)}(\mathbf{x}; \theta_A)\|_2 \right], \quad (11)$$

for $j = 1, 2$. To summarize, the total loss for solving the SFW problem with AD is

$$\mathcal{L}_{AD} = \mathcal{L}_\delta + \mathcal{L}_{AA}. \quad (12)$$

5. EXPERIMENTS

5.1. Experiment Setting

All the experiments are performed on a PC equipped with a NVIDIA Geforce RTX 2070 GPU and Ubuntu 20.04 OS, and Keras is the underlying framework.

In our experiments, the watermarked models are mainly targeted at classifying CIFAR-10 [17] and GTSRB [18]. The implemented network architectures of victim models include VGG16 [19], ResNet18 [20], and WRN-16-4 (Wide Residual Network) [21]. We evaluate the attack effectiveness of AD on three state-of-the-art watermarks introduced in [3], including the content watermark (as shown in Fig. 3(b)), the noise watermark, and the unrelated watermark. We set the number of triggers used to embed watermarks into victim models as 1000, and they are assigned to class 0 (the target label).

We consider two attack scenarios for obtaining the proxy datasets: using OOD data (*i.e.*, CIFAR-100 [17]) and ID data (half of the original testing data). For each case of the obtained proxy datasets, 1000 images are randomly sampled and used in the fine-tuning. In addition, only 10 randomly select abstract images (introduced in [2]) are used as the lure data (the implementation details can be found in the appendix).

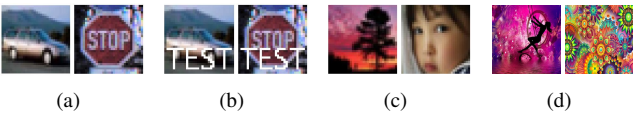


Fig. 3. Examples of datasets: (a) source training samples from CIFAR-10 and GTSRB, (b) trigger samples using ‘TEST’ as perturbation, (c) proxy instances drawn from the CIFAR-100 dataset, and (d) lure images drawn from abstract images.

Evaluation metrics. We mainly consider the main task accuracy (**MTA**) and watermark accuracy (**WMA**) metrics for evaluation. It is performed for the top-1 predicted categories. For CIFAR-10, we set ϵ in Eq. (4) to 0.1 since $C = 10$.

5.2. Results Analyses

Table 1 presents the performance of victim models before and after being attacked by comparing AD with PST-FT on the CIFAR-10 dataset (the results of GTSRB can be found in the appendix). We can observe that almost all three types of watermarks are vulnerable to the PST method. In addition, using PST-FT can generally improve the MTA or decrease the WMA of victim models. However, this is not always the case, especially when the triggers’ perturbation is noise. This is because the trigger noise could be mixed with the distribution mismatch noise.

For AD, it is clear that all the WMAs of the attacked victim are below 20%, achieving the goal of watermark removal. For all the network architectures, AD retains the MTAs of the victim models. Moreover, though AD generally performs better with ID proxy data, the differences between MTAs for OOD and ID proxy data are almost always within 2%. This fact validates that AD can avoid the noise caused by the data distribution difference between the proxy and original training data while removing the planted watermark.

5.3. Ablation Study

In this section, we perform an ablation study to investigate the role of each component of AD on the CIFAR-10 task and the content-based watermark task (more analysis can be found in the appendix).

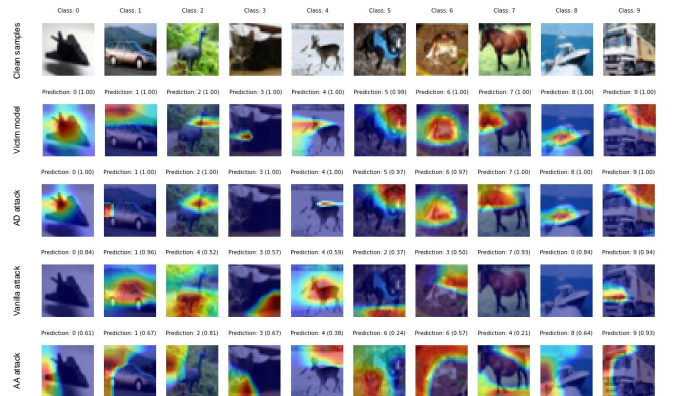


Fig. 4. The attention heatmaps of ResNet18 on CIFAR-10 samples. The first row depicts the clean images from different classes. The following rows respectively depict the heatmaps of the victim model before and after the AD attack, vanilla attack, and AA attack with OOD proxy data.

Role of the lure task. To study the role of the lure task on SFW, we design two experiments that only use proxy data, including OOD and ID, as comparisons. That said, one method uses the vanilla loss (the vanilla attack proposed in [11]) \mathcal{L}_v (defined in Eq. (9)) as the objective, and the other uses attention anchoring (called AA attack) loss \mathcal{L}_{AA} as the objective (defined in Eq. (8)). And the results are listed in Table 2.

Table 1. Performance of watermark removal attacks on the CIFAR-10, where x/y denotes MTA/WMA.

Watermark task	Networks	Victim models	PST	PST-FT		AD	
				OOD	ID	OOD	ID
Content	VGG16	93.82 / 99.96	88.20 / 14.84	84.70 / 19.14	82.80 / 17.80	93.58 / 7.70	93.38 / 18.28
	ResNet18	93.74 / 99.78	84.60 / 12.44	88.66 / 17.98	88.46 / 15.14	92.76 / 13.94	92.34 / 6.78
	WRN-16-4	93.42 / 99.94	88.62 / 26.82	85.24 / 18.70	88.56 / 26.68	92.82 / 13.50	91.16 / 1.98
Unrelated	VGG16	93.68 / 100	88.50 / 4.76	86.13 / 16.56	88.26 / 9.34	89.56 / 14.78	92.40 / 0.22
	ResNet18	92.88 / 100	84.34 / 12.74	88.00 / 1.64	87.90 / 2.12	88.88 / 1.16	90.02 / 0.30
	WRN-16-4	93.80 / 100	88.50 / 6.30	89.60 / 1.76	89.18 / 1.90	90.86 / 3.98	91.74 / 2.38
Noise	VGG16	93.58 / 100	88.36 / 99.86	72.16 / 2.22	73.82 / 0.34	93.78 / 0.78	93.94 / 1.32
	ResNet18	93.72 / 100	84.84 / 1.88	86.38 / 13.76	85.52 / 18.46	93.22 / 5.32	93.64 / 0.96
	WRN-16-4	93.32 / 100	88.58 / 11.92	88.22 / 18.32	90.00 / 2.84	93.28 / 14.30	93.88 / 3.48

Table 2. Performance of watermark removal attacks of AD without using lure data on CIFAR-10.

Networks	Vanilla		AA	
	OOD	ID	OOD	ID
VGG16	59.96 / 6.32	73.18 / 19.60	61.52 / 19.94	76.56 / 19.70
ResNet18	66.78 / 19.52	76.04 / 18.94	79.10 / 10.90	85.94 / 18.26
WRN-16-4	75.54 / 19.48	81.40 / 18.50	50.50 / 18.96	85.18 / 13.22

From Table 2, we observe that, no matter using OOD or ID proxy data, neither the vanilla attack nor the AA attack is as good as AD in solving the SFW. This confirms that the lure task is important for solving SFW: the lure task will compensate for the negative effect brought by the distribution difference between the proxy data and the original training data.

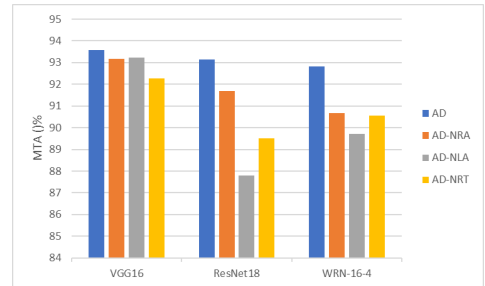
To further investigate the role of the lure task, Fig. 4 visualizes the attention heatmaps (produced by Grad-CAM [15]) of ResNet18, before and after different attacks using OOD proxy data, on CIFAR-10 images. From the second and third rows of this figure, it is easy to see that the attention maps of the model after the AD attack are highly similar to those of the victim before attack. For both the vanilla and the AA attacks, the attention maps of the model after attack deviate from those of the victim before attack. That said, without the participation of lure, using OOD proxy data alone cannot effectively anchor the main task attention maps while forgetting the target watermark task.

Table 3. Influence of the lure data label on CIFAR-10.

Networks	$\Delta = 0$	$\Delta = 1$	$\Delta = 5$
VGG16	78.70 / 99.34	93.58 / 18.78	93.54 / 6.10
ResNet18	74.88 / 55.90	92.20 / 18.14	92.26 / 11.20
WRN-16-4	60.70 / 100	93.18 / 0.00	93.76 / 0.00

Influence of the label of the lure task. To investigate the influence of the label of the lure task, we conduct experiments using different predefined lure labels, and the results are listed in Table 3. From this table, we can observe that when the lure label Δ is not the same as the target label 0, *i.e.*, $\Delta = 1$ or 5, the results of the AD attack are comparable to those obtained by using a new class C as the label for lure data. However, when $\Delta = 0$, the AD attack fails for all victim models.

As discussed previously and visualized in Fig. 2, when the lure label does not overlap with the target label, it encourages the continual learning process to find a path that is suitable for the main task and the lure task, while forgetting the target watermark task. For the consideration of watermark agnostic, the most convenient way to prevent the label collision is to set $\Delta = C$.

**Fig. 5.** The MTA of AD with different regularization on CIFAR-10. All WMAs are lower than the predefined threshold. The blue bars represent the complete version of AD, the orange bars, gray bars, and yellow bars represent different versions of AD, respectively.

Role of the regularization terms. We investigate the role of the regularization terms in Eq. (12) by designing several different versions of AD attack and analyzing their attack effectiveness with OOD proxy data. In what follows, AD-NRA refers to the AD attack without using $\mathcal{R}(\theta_A)$ defined in Eq. (11), AD-NLA refers to the AD attack without using \mathcal{L}_a defined in Eq. (10), and AD-NRT refers to the AD attack without using any regularization terms, *i.e.*, directly use $\mathcal{L}_\delta + \lambda_1 \cdot \mathcal{L}_v$ (defined in Eq. (6) and Eq. (9) respectively) as

the objective function.

Figure 5 depicts the model performance on the main task after erasing the watermarks by different versions of AD attacks. From this figure, it is clear that the blue bars, which represent the complete AD, achieve the highest MTA for all considered network architectures.

6. CONCLUSION

This paper has proposed a novel DNN watermark removal attack, AD, at the source data-free regime through continual learning with selective forgetting. We introduced two strategies to achieve the attack goal: attention distraction and attention anchoring. The key to the success of AD is to ensure that the new lure task does not collide with the to-be-forgotten watermark task, and this can be easily achieved by assigning a new label to the lure data. Though conceptually simple, extensive experimental results validated that AD outperforms other state-of-the-art works in various benchmark datasets and network architectures.

7. REFERENCES

- [1] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh, "Embedding watermarks into deep neural networks," in *ICMR*, 2017, pp. 269–277.
- [2] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *USENIX Security 18*, 2018, pp. 1615–1631.
- [3] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy, "Protecting intellectual property of deep neural networks with watermarking," in *ASIACCS*, 2018, pp. 159–172.
- [4] Qi Zhong, Leo Yu Zhang, Jun Zhang, Longxiang Gao, and Yong Xiang, "Protecting IP of deep neural networks with watermarking: A new label helps," in *PAKDD*, 2020, vol. 12085, pp. 462–474.
- [5] Meng Li, Qi Zhong, Leo Yu Zhang, Yajuan Du, Jun Zhang, and Yong Xiang, "Protecting the intellectual property of deep neural networks with watermarking: The frequency domain approach," in *TrustCom. IEEE*, 2020, pp. 402–409.
- [6] Xinyun Chen, Wenxiao Wang, Chris Bender, Yiming Ding, Ruoxi Jia, Bo Li, and Dawn Song, "REFIT: A unified watermark removal framework for deep learning systems with limited data," in *ASIACCS*, 2021, pp. 321–335.
- [7] Xuankai Liu, Fengting Li, Bihan Wen, and Qi Li, "Removing backdoor-based watermarks in neural networks with limited data," *arXiv preprint arXiv:2008.00407*, 2020.
- [8] William Aiken, Hyoungshick Kim, Simon Woo, and Jungwoo Ryoo, "Neural network laundering: Removing black-box backdoor watermarks from deep neural networks," *Computers & Security*, vol. 106, pp. 102277, 2021.
- [9] Haoqi Wang, Mingfu Xue, Shichang Sun, Yushu Zhang, Jian Wang, and Weiqiang Liu, "Detect and remove watermark in deep neural networks via generative adversarial networks," *arXiv preprint arXiv:2106.08104*, 2021.
- [10] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan, "Measuring catastrophic forgetting in neural networks," in *AAAI*, 2018, vol. 32.
- [11] Masoumeh Shafieinejad, Jiaqi Wang, Nils Lukas, Xinda Li, and Florian Kerschbaum, "On the robustness of the backdoor-based watermarking in deep neural networks," in *IH & MMSEC*, 2021, pp. 177–188.
- [12] Shangwei Guo, Tianwei Zhang, Han Qiu, Yi Zeng, Tao Xiang, and Yang Liu, "Fine-tuning is not enough: A simple yet effective watermark removal attack for DNN models," in *IJCAI*, 2021, pp. 3635–3641.
- [13] Zhizhong Li and Derek Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, 2017.
- [14] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *ECCV*. Springer, 2014, pp. 818–833.
- [15] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017, pp. 618–626.
- [16] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R Lyu, and Yu-Wing Tai, "Boosting the transferability of adversarial samples via attention," in *CVPR*, 2020, pp. 1161–1170.
- [17] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," Technical Report, 2009.
- [18] Johannes Stalldkamp, Marc Schlipfing, Jan Salmen, and Christian Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, vol. 32, pp. 323–332, 2012.
- [19] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
- [21] Sergey Zagoruyko and Nikos Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.
- [22] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao, "Neural cleanse: Identifying and mitigating backdoor

attacks in neural networks,” in *SP. IEEE*, 2019, pp. 707–723.

8. APPENDIX

Implementation details. We evaluate the attack effectiveness of AD on three state-of-the-art watermarks introduced in [3], including (1) the content watermark (stamp with a self-designed ‘TEST’ pattern on clean samples), (2) the noise watermark (add Gaussian noise on the clean samples), and (3) the unrelated watermark (use the images in MNIST with label ‘1’ as the triggers).

We set the number of triggers used to embed watermarks into victim models as 1000, and they are assigned to class 0, *i.e.*, label ‘Airplane’ in CIFAR-10 and label ‘Speed Limit 20’ in GTSRB. The content watermark triggers and the noise watermark triggers are generated by randomly selecting certain images from each class of the clean training data and stamping the corresponding perturbations on them with the following equation [22]:

$$\mathbf{x}_t^{(i)} = (1 - m) \cdot \mathbf{x}^{(i)} + m \cdot \boldsymbol{\epsilon}^{(i)}, \quad (13)$$

where $\boldsymbol{\epsilon}^{(i)}$ is the trigger pattern determined by \mathcal{V} , m is a *mask* defines both the position and intensity of the trigger pattern overwritten into the clean data.

The victim watermarked models with VGG16, ResNet18 and WRN-16-4 are all trained using the Stochastic Gradient Descent (SGD) optimizer with an initial learning rate 0.1, momentum of 0.9 and batch size of 128. And we employed data augmentation techniques, including image rotation, and horizontal and vertical image shift during the training of the victims. As for fine-tuning with AD and PST-FT [12], we use batch size 100 and no data augmentation strategy are used. The hyper parameters in AD, *e.g.*, λ_i ($i = [1, 3]$), are determined by binary search.

Results on GTSRB tasks. Table 4 presents the performance of victim models before and after being attacked by comparing AD with PST-FT on the GTSRB dataset. As we can see from this table, unlike the results on the CIFAR-10 task, the PST method failed to remove watermarks in most of the victim models on the GTSRB task, *i.e.*, the WMAs in most of the victim models using PST are still higher than the threshold (since we define the $\epsilon = 0.1$, so for the GTSRB task, the ideal threshold is 12%). This is mostly because the main task (classification of the GTSRB dataset) and the watermark task of the victim models are easier than those on the CIFAR-10 task, so the victim model is overfitted to the watermark triggers. For the VGG16 victim with unrelated watermarks, the WMAs almost show no decrease while the MTAs drop about 3% with the PST method.

Using PST-FT can effectively improve the watermark removal effects. However, most of the victim models using OOD proxy data still cannot make the WMAs below the predefined threshold. Using ID proxy data can generally improve

the MTAs or reduce the WMAs, and the MTA gaps between the results of using OOD proxy data and ID proxy data are obvious. Similar to what we have analyzed for the experimental results of the CIFAR-10 dataset, it is because of the inherent noise caused by the proxy data distribution mismatch to that of the source domain.

As for AD, it is obvious that all of the three types of watermarks can be removed while the MTAs are close to the victim models before attacks. In addition, just like in the CIFAR-10 task, AD attack with ID proxy data performs better than that using OOD proxy data, but the MTA gaps between the results of using OOD proxy data and ID proxy data are small (within 3%).

Role of the lure task. Table 5 list the results using vanilla attack and AA attack on various victim models with content-based watermark. We can observe from the table that, when using OOD proxy data, neither the vanilla attack nor the AA attack is as good as AD in solving the SFW. Specifically, for VGG16, although the WMA of victim models drops to below 12%, their MTAs also drop drastically after attacks (as low as 58.07% for vanilla attack and 57.48% for AA attack). And in both attacks, the gap in WMA for OOD and ID proxy data is huge (with the largest gap being about 38%), while this value is only 3% for AD as shown in Table 4. This, once again, corroborates the lure task plays a key role in solving SFW: the lure task will compensate for the negative impact caused by the distribution difference between the proxy data and the source training data.

The heatmaps of the victim models before and after attacks on clean instances can further explain the role of the lure task in a more vivid form. As presented in Fig. 6, which visualizes the attention heatmaps of ResNet18, before and after different attacks using OOD proxy data, on GTSRB images. From the second and third rows of this figure, it is easy to see that the attention maps of the model after the AD attack are highly similar to those of the victim before the attacks. For the vanilla and AA attacks, the attention maps of the model after attack deviate from those of the victim before attack. That said, without the participation of lure, using OOD proxy data alone cannot effectively anchor the main task attention maps while forgetting the target watermark task.

Influence of the label of the lure task. The experiment results using different predefined lure labels are listed in Table 6. We can observe from this table that when the lure label Δ is not overlap with the target label 0, *i.e.*, $\Delta = 10$ or 20, the results of AD attack is comparable to that using a new class C as the label for lure data. Specifically, when Δ is 10 or 20, the WMA of the AD-attacked models is below 12% and the loss in MTA is small. However, when $\Delta = 0$, *i.e.*, the label of the lure task overlaps with the target label, the AD attack fails for all victim models. This phenomenon once again verifies the necessity of using an additional defined new label for the lure task when the watermark is agnostic, *i.e.*, preventing label collision.

Table 4. Performance of watermark removal attacks on the GTSRB task, where x/y stands for MTA/WMA.

Watermark task	Networks	Victim models	PST	PST-FT		AD	
				OOD	ID	OOD	ID
Content	VGG16	96.11 / 100	93.58 / 50.28	82.90 / 9.14	91.87 / 0.60	95.47 / 0.00	96.12 / 0.40
	ResNet18	96.30 / 99.48	91.55 / 25.08	78.49 / 11.92	92.46 / 15.14	92.10 / 0.04	95.36 / 8.86
	WRN-16-4	95.66 / 100	92.98 / 51.14	81.15 / 11.56	92.56 / 26.68	94.97 / 5.08	96.00 / 2.84
Unrelated	VGG16	96.37 / 100	93.41 / 99.66	62.44 / 99.68	92.14 / 0.44	94.46 / 0.10	96.19 / 0.00
	ResNet18	96.61 / 100	92.29 / 99.98	92.00 / 1.64	92.90 / 2.12	94.98 / 0.00	96.17 / 0.00
	WRN-16-4	95.28 / 100	91.28 / 99.80	80.18 / 99.94	87.73 / 94.02	94.84 / 0.34	94.74 / 2.38
Noise	VGG16	96.34 / 100	93.72 / 10.06	93.31 / 3.80	93.00 / 3.68	95.69 / 0.30	96.30 / 0.00
	ResNet18	95.44 / 99.98	92.90 / 0.72	92.38 / 0.26	92.52 / 0.00	94.10 / 0.00	95.14 / 0.00
	WRN-16-4	95.79 / 100	92.11 / 47.34	82.31 / 11.40	82.60 / 2.30	93.00 / 0.00	95.88 / 3.48

Table 5. Performance of watermark removal attacks of AD without using lure data on GTSRB.

Networks	Vanilla		AA	
	OOD	ID	OOD	ID
VGG16	58.07 / 19.78	94.39 / 12.04	57.48 / 1.72	95.67 / 3.80
ResNet18	66.29 / 50.48	95.19 / 0.16	89.23 / 21.68	95.41 / 0.20
WRN-16-4	85.37 / 99.98	95.37 / 8.20	79.13 / 99.90	95.31 / 1.90

Table 6. Influence of the lure data label on GTSRB.

Network	$\Delta = 0$	$\Delta = 10$	$\Delta = 20$
VGG16	69.00 / 99.96	95.64 / 5.12	95.47 / 0.64
ResNet18	79.88 / 96.50	92.24 / 7.98	92.04 / 0.88
WRN-16-4	61.18 / 100	93.76 / 11.18	94.31 / 3.10

Influence of lure data budget. We also investigate the attack effects of different lure data budgets (using abstract images as the lure data), and the results are depicted in Fig. 8. We can see that there is an improvement in the MTA of most victim models as the amount of lure data increases, even though the improvement is limited (within 1%). This indicates that the proposed AD is a serious threat to existing DNN watermarking methods.

Influence of proxy data budget. We evaluate the performance of AD and PST-FT with different proxy data budgets on content-based watermarks, as shown in Fig. 9. We observe that for both the AD attack and the PST-FT attack, the MTAs of the victim models are preserved with the increase of the volume of the proxy data. And for some networks, their MTAs have slightly increased when using more proxy data in fine-tuning. However, the MTA gaps between PST-FT and AD are still obvious even using more proxy data.

Role of the regularization terms. Figure 7 depicts the model performance on the main task (GTSRB) after different versions of AD attacks. From the above figure, we can see that almost all the blue bars, which represent the AD, reach the highest MTA. For the VGG16, the attack effects of AD-NPA and AD-NAA are almost the same, reaching the complete version of AD. By comparing the orange bars, grey bars, and yel-

low bars, for most of the networks and watermarks, AD-NPA and AD-NAA are better than AD-NRT. For the ResNet18 and WRN-16-4, the difference in attack effects between different versions of AD is more obvious. Specifically, for the ResNet18 with the unrelated-based watermark, the attack effects of AD-NRT are obviously less than those of the other versions of AD. By comparing the blue and yellow bars, we find that the attention loss term and the attention penalty term help to maintain the main task performance. Employing the prediction alignment is not sufficient to achieve good performance in MTA.

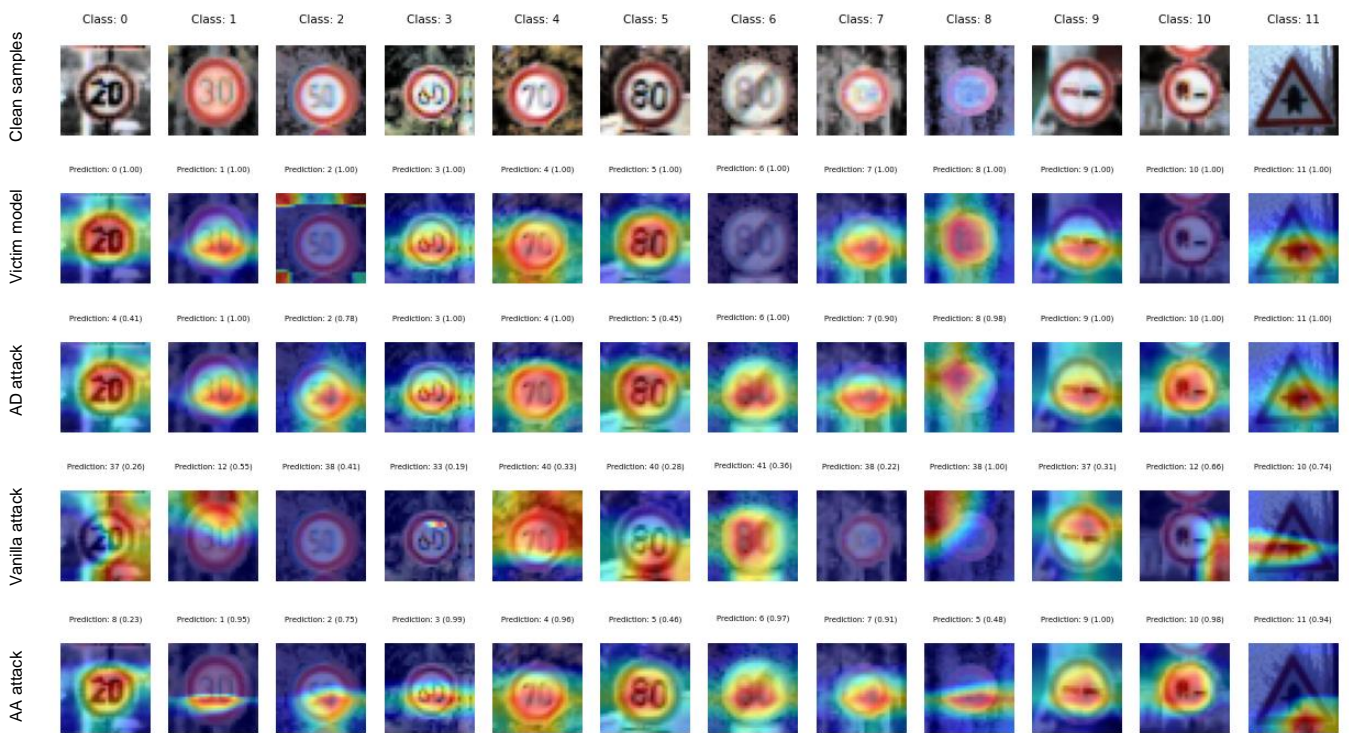


Fig. 6. The attention heatmaps of ResNet18 on part of GTSRB clean instances. The first row depicts the clean images from different classes. The second row depicts the heatmaps of the victim model on the clean images. The following rows respectively depict the heatmaps of the model after the AD attack, vanilla attack, and AA attack with OOD proxy data.

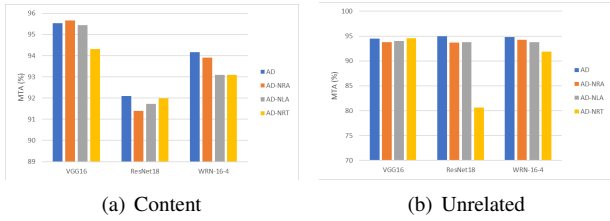


Fig. 7. The MTAs of regularization terms on the GTSRB task, where (a) refers to the content-based watermark, (b) refers to the unrelated-based watermark. Their corresponding WMA is lower than the predefined threshold. The blue bars represent the complete version of AD, the orange bars, grey bars, and yellow bars represent different versions of AD, respectively.

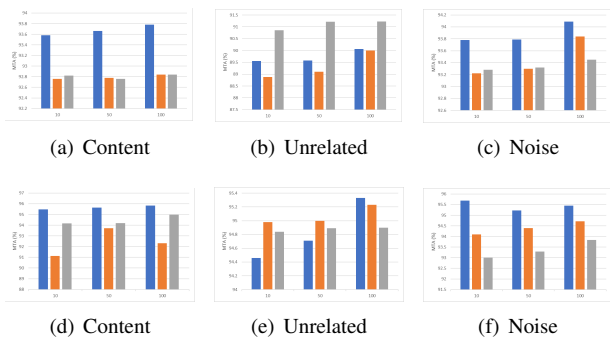


Fig. 8. Sensitivity to the lure data budget on different victim models, where the lure data used is abstract images, (a)-(c) refer to the models on CIFAR-10 task, (d)-(f) refer to the models on GTSRB task. The blue bars, orange bars, and grey bars represent the VGG16, ResNet18, and WRN-16-4 networks, respectively. Their corresponding WMAs are below the threshold.

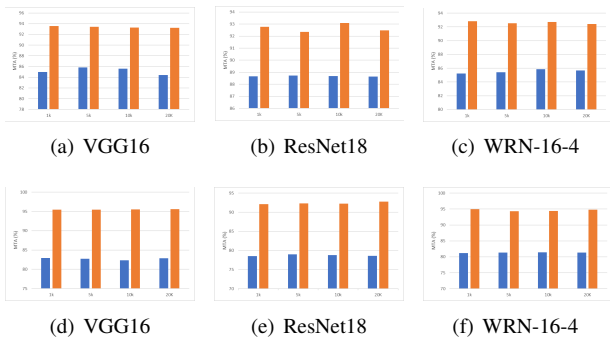


Fig. 9. Performance of AD attack and PST-FT attack with different amount of OOD proxy data, where (a)-(c) refer to the models on CIFAR-10 task, (d)-(f) refer to the models on GTSRB task. The blue bars represent the PST-FT attack, and the orange bars represent the AD attack. Their corresponding WMAs are below the corresponding thresholds.