Subnetwork-Lossless Robust Watermarking for Hostile Theft Attacks in Deep Transfer Learning Models

Ju Jia, Yueming Wu, Anran Li, Siqi Ma, and Yang Liu

Abstract—Recently, considerable progress has been made in providing solutions to prevent intellectual property (IP) theft for deep neural networks (DNNs) in ideal classification or recognition scenarios. However, little work has been dedicated to protecting the IP of DNN models in the context of transfer learning. Moreover, knowledge transfer is usually achieved through knowledge distillation or cross-domain distribution adaptation techniques, which will easily lead to the failure of the IP protection due to the high risk of the underlying DNN watermark being corrupted. To address this issue, we propose a subnetwork-lossless robust DNN watermarking (SRDW) framework, which can exploit out-of-distribution (OOD) guidance data augmentation to boost the robustness of watermarking. Specifically, we accurately seek the most rational modification structure (i.e., core subnetwork) using the module risk minimization, and then calculate the contrastive alignment error and the corresponding hash value as the reversible compensation information for the restoration of carrier network. Experimental results show that our scheme has superior robustness against various hostile attacks, such as fine-tuning, pruning, cross-domain matching, and overwriting. In the absence of malicious jamming attacks, the core subnetwork can be recovered without any loss. Besides that, we investigate how embedding watermarks in batch normalization (BN) layers affect the generalization performance of the deep transfer learning models, which reveals that reducing the embedding modifications in BN layers can further promote the robustness to resist hostile attacks.

Index Terms—IP protection, DNN watermarking, deep transfer learning models, reversibility and robustness, OOD guidance.

1 Introduction

The protection of intellectual property (IP) for deep neural networks (DNNs) is attracting more and more attention due to the expensive cost and technical complexity for constructing these models. Following this trend, several DNN watermarking methods have been proposed to respond to the growing IP protection concern recently [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. Conceptually, DNN watermarking is implemented by injecting certain behaviors into the model whose existence can be easily verified when needed, usually by using a series of key samples.

DNN watermarking can be divided into white-box and black-box settings depending on whether demands full access to the model during watermark insertion and authentication. In the white-box setting, the model information is required to be fully available, including the potential feature mappings for extracting the watermarks, and the expected behaviors can be embedded into the inherent structure or potential space of the DNN model [1], [5]. While in the black-box setting, the watermark is designed to correlate the desired predictions to the key samples by using the application programming interface (API) in machine learning as a service (MLaaS) [2], [3], [8], [9], [11]. For example, the attacker can first feed a large amount of input into the model API and obtain the corresponding output. Then, the attacker takes these input-output pairs as training samples and constructs a reliable surrogate model, which is called surrogate model attacks (SMAs) or model extraction attacks (MEAs).

Many scientific studies have revealed the hidden vulnerability and fragile robustness of DNN watermarking [15], [16], [17], [18]. The reasons can be attributed to two aspects. On the one hand, some watermarking schemes are designed based on simple backdoor techniques, which makes them hard to be robust to adversarial samples and OOD samples with complex patterns. On the other hand, it is difficult for DNN watermarking to resist various advanced model transformations only through identifying the verification queries. For example, existing work usually either adds a weight consolidation regularizer to the loss function to guarantee the learned weights have some specific patterns, or uses the verifiable prediction results of a series of special indicator images as watermarks [1], [2], [3], [9], [10]. Although these methods perform well to increase the robustness within a certain range, they only consider ideal classification settings and structure fault attacks like fine-tuning or pruning, which hinders their application in many mission-critical scenarios (e.g., transfer learning [19] or domain adaptation [20]). Deep transfer learning aims to substantially reduce the efforts to obtain high-performance DNN models, which can largely ease the burden of adversarial training, especially for those models with limited

- *J. Jia, Y. Wu, and A. Li are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798. E-mail: {jia.ju, anran.li}@ntu.edu.sg, wuyueming21@gmail.com*
- *S. Ma is with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2612, Australia. E-mail: siqi.ma@unsw.edu.au*
- *Y. Liu is with the Zhejiang Sci-Tech University, China and Nanyang Technological University, Singapore. E-mail: yangliu@ntu.edu.sg*

Manuscript received XX, 2022; revised XX, 2022.
(Corresponding authors: Yueming Wu and Yang Liu.)

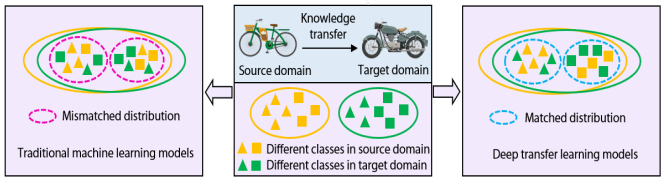


Fig. 1: Comparison of deep transfer learning models with traditional machine learning models. Deep transfer learning models achieve knowledge transfer by matching distributions between the source and target domain, while traditional machine learning models fail to match distributions across domains.

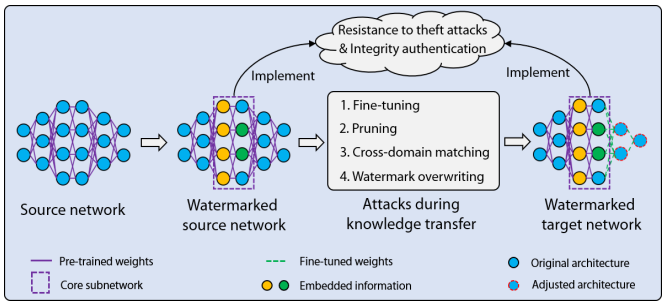


Fig. 2: Our scheme embeds the watermark by seeking a core subnetwork from the perspective of the matching relationship between the data and model. The watermark embedded on the core subnetwork can not only help the whole model resist common theft attacks, but also resist various other attacks during the knowledge transfer. Moreover, the core subnetwork can be recovered losslessly under the condition of no malicious jamming attack.

computational capabilities [21], [22], [23], [24]. Fig. 1 gives an intuitive example of deep transfer learning models. Therefore, it is considered as a promising machine learning service technique in the real applications for the sake of work efficiency [25]. However, deep knowledge transfer operations would bring great challenges to the IP protection for DNN models. Compared to common deep learning models, deep transfer learning models tend to adopt knowledge distillation or domain adaptation techniques to find a shared distribution between the source and target domains, which can easily lead to the traditional DNN watermarking to lose the ownership information in such cases. Moreover, ensuring the integrity of conventional DNN watermarking will limit the generalization capability of the common deep learning model on a specific task, which makes it difficult to achieve the desired knowledge transfer from the seen domain (i.e., source domain) to the unseen domain (i.e., target domain).

For DNNs, the functionality of the model is characterized by its weight parameters. To avoid permanent loss in robust watermarking schemes, researchers exploit the redundancy among the model weights to embed and extract watermarks without compromising any weights. Such a technique, called reversible watermarking (e.g., also known as reversible data hiding or cover-lossless information hiding), can successfully restore the original neural network in case of no attack [26]. Guan et al. [27] utilized the pruning theory of network compression to build a host sequence guided for embedding reversible watermarking information by histogram shift, which indicates that the reversible watermarking can be used for the fragile authentication. However, it is challenging in most practical applications due to its sensitivity to the weight pruning or updating operations. In the transfer learning settings that DNN models require high fidelity in these areas, such as medical image diagnosis, multimedia digital forensics, hyperspectral data processing, and high-precision navigation tracking, the data loss and performance degradation caused by information embedding are intolerable. As a result, a cover-lossless robust DNN watermarking against hostile theft attacks is developed to avoid losing the model integrity due to embedding operations, which can be well applied to deep transfer learning scenarios.

On the basis of the above analysis, we propose a novel subnetwork-lossless robust DNN watermarking (SRDW) framework for the copyright protection of deep transfer learning models. The motivation comes from the fact that the mapping relationships between the data and the model can be leveraged to promote the reversibility and robustness of watermarking through the sample generation (i.e., key samples) and the subnetwork selection (i.e., core subnetwork). Fig. 2 illustrates our solution for the IP protection of deep transfer learning models. This scenario differs significantly from the IP protection of traditional DNN models, i.e., a subnetwork-lossless robust watermarking is required for both source and target networks throughout the knowledge transfer process. Specifically, considering the impact of embedding operation and model capacity on the generalization behavior in the specific tasks, we use modular risk minimization to hunt the most rational modification subnetwork as a robust guarantee for watermark embedding. To guarantee the reversibility of the watermark, we combine the contrastive alignment error and the corresponding hash value to design the reversible compensation information. In this way, the embedding of the watermark can be flexibly controlled by exploiting the reversible compensation information to generate the final watermarked network. Extensive experiments show that our proposed subnetwork-lossless robust watermarking scheme can be effectively utilized for the IP protection of deep transfer learning models. The contributions of our paper are summarized as follows.

1) We present an OOD-guided data augmentation technique based on constrained variational autoencoder (CVAE) to generate synthetic samples, which promotes not only the robustness of the embedded watermark but also the task-related performance of the model.
2) We introduce the module risk minimization to explore the most suitable modification architecture (i.e., core subnetwork), which achieves the watermark embedding without sacrificing the model accuracy in an alternating optimization manner.
3) We design the reversible compensation information consisting of the contrastive alignment error and the corresponding hash value to guarantee the reversibility of the watermark. Moreover, the experimental results in Section 6.4 indicate that the core subnetwork can be recovered losslessly under the condition of no malicious jamming attack.
4) To the best of our knowledge, this is the first work to propose a subnetwork-lossless robust DNN watermark-

ing for the IP protection of deep transfer learning models. We conduct extensive experiments to evaluate our proposed SRDW on various deep transfer learning models. The promising results clearly show the reversibility and robustness of the proposed watermarking scheme.

The rest of the paper is organized as follows. Section 2 presents the related work and motivation. Section 3 introduces the problem statement of this work. Section 4 describes the proposed SRDW for the copyright protection of deep transfer learning models. The experimental setting and implementation are given in Section 5. The effectiveness of our watermarking scheme is demonstrated through experimental evaluation results in Section 6. Finally, the conclusions and the future research directions are summarized in Section 7.

2 Related Work

2.1 DNN IP Protection

The powerful data processing and representation description capabilities of DNNs pose a potential security threat to the IP of deep learning models. Early researchers tended to focus on how to access the DNN model sufficiently to enable a flexible watermark embedding and extraction process [1], [5]. Following this, Fan et al. [6] presented an interesting DNN ownership verification method to embed the watermarks by exploiting the information in the "passport" layers, where the corresponding network parameters are essential to maintain the performance of the model. As another representative work, Lin et al. [13] developed a chaotic weight framework based on the chaotic map theory to protect the IP of DNN with very low overhead. Since the structures of DNN models are usually invisible in practical applications, most existing approaches employ the idea of backdoor attacks to force the triggers into the models [28], which may undermine the integrity of the model and lead to the performance degradation [2], [3], [8], [9], [11]. Inspired by the reversible digital image watermarking techniques, a reversible DNN watermarking framework was designed to achieve model integrity authentication [27]. Despite their success, most of these copyright protection schemes have been proposed for classification or recognition tasks in the matched distribution scenarios (i.e., simple modification-based attacks). By contrast, few studies have designed IP protection watermarks specifically for deep transfer learning models in the mismatched distribution scenarios (i.e., complex modification-based attacks). The recently proposed null embedding technique to construct the piracy-resistant DNN watermark shows its effectiveness in deep transfer learning models to a certain extent [29]. Due to the large number of fine-tuning operations in the process of knowledge transfer, robustness and reversibility are crucial for the watermarking of transfer learning models. Motivated by these facts, we design a controllable watermarking framework to ensure that specific patterns and embedded information are associated to make it robust and reversible throughout the whole transfer learning process.

2.2 Data-Centric Robustness Enhancement

Data-centric learning is a promising technique based on data augmentation that improves the robustness and generalization of DNNs by generating high-quality datasets. For instance, Ng et al. [30] proposed a data augmentation method based on self-supervised manifolds to generate synthetic training samples and improve out-of-domain robustness. In addition, Hua et al. [31] dynamically allocated distinct amounts of computation for generating corrupted samples by their importance to facilitate the acquisition of robust DNNs. Notably, the effect of pre-training data size, model scale and data pre-processing pipeline was investigated in the literature [32], whose results indicated that increasing the training set and model sizes or simple changes in the pre-processing (e.g., modifying the image resolution) can promote the robustness and generalization of the model in some cases. Another line of research [33] aimed to build a generalizable model from biased domain knowledge, which can reinforce the learning ability using a target dataset close to the essence of the desired test data. Our proposed scheme exploits data-centric learning algorithms to enhance the OOD generalization of the model by preventing shortcut cues, which can facilitate the implementation of watermarking in specific regions (i.e., the core subnetwork). Moreover, the robustness of DNN watermarking can be improved by adjusting the generation of key samples and the magnitude of modification operations. Based on the above observations, we draw on the data-centric perspectives to generate key samples to improve the robustness of watermarking by exploiting the relationships between source and target samples in transfer learning.

2.3 Spurious Correlation Suppression

Deep transfer learning models (DTLMs) aim to transfer knowledge from different but related source domains to enhance the performance on target tasks by exploiting the powerful generalization ability of DNNs. DTLMs have been successfully applied in many real-world scenarios [34], [35], [36]. It is worth noting that the knowledge transfer does not always bring the positive effect on new tasks. If there is little common ground between the source and target domains, the knowledge transfer may be unsuccessful. Moreover, the correlations across domains do not necessarily promote the performance of transfer learning, because sometimes models are misled by spurious correlations. The literature [37] has revealed that DNNs are prone to learn superficial representations to make overconfident predictions, such as relying on spurious correlations rather than the intrinsic mechanisms of the task of interest. This phenomenon has caused widespread concern among researchers because the performance of the model deteriorates under the interference of these spurious correlations. Therefore, it is desirable to avoid the influence of spurious modes on DNNs while training models with data-centric techniques. Since Arjovsky et al. [38] brought invariant predictors into more realistic practical situations, a large number of studies have achieved tremendous success in suppressing spurious correlations and exploring stable representations. Following this trend, Zhang et al. [39] exploited deep stable learning to get rid of spurious correlations and thus concentrate more on the true correlations between discriminative features and labels. Recently, Zhan et al. [40] designed a boundary-aware rectified linear unit (ReLU) to avoid spurious correlations by

improving the reliability of DNNs, where an upper bound of ReLU is selected to ensure the correctness of the final result. Therefore, we adopt OOD awareness to remove spurious correlations so that the generated samples are within a reasonable region. In other words, the generated samples need to satisfy the condition that they are neither easily perceived (i.e., similarity) nor easily misclassified (i.e., discrepancy).

3 Problem Statement

Considering the protection of source model M_s and target model M_t in deep transfer learning, we adopt an L-layer feed-forward DNN to describe a general model M as follows:

$$M(\cdot;\boldsymbol{\alpha}) := \left(M^{L}_{\boldsymbol{\alpha}_L} \circ M^{L-1}_{\boldsymbol{\alpha}_{L-1}} \circ \cdots M^{1}_{\boldsymbol{\alpha}_1}\right)(\cdot), \tag{1}$$

which is parameterized by $\boldsymbol{\alpha} := \{\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_L\}$. We utilize M^b to denote the b-th layer of the model M and employ $M^{(b_1,b_2)}$ to represent the layers ranging from b_1 to b_2, i.e., $M^{(b_1,b_2)} := M^{b_1} \circ \cdots \circ M^{b_2}$. The first b layers is denoted as $M^{(b)}$ for shorthand. This paper mainly investigates the deep knowledge transfer scenarios, where both the source and target models may be subject to surrogate model attacks and fine-tuning attacks during the entire transfer learning process. To this end, we design a robust and reversible watermark, which can provide a guarantee for the realization of flexible and controllable integrity authentication. Specifically, the goal of reversible watermarking is to embed a T-bit vector $\mathbf{e}$ (i.e., the substantive information of watermark) into the source model M_s to generate the embedded source model M'_s. Then, the source model with robust watermark can be manipulated to obtain the target model with robust watermark. In addition, the embedding information of the watermark can be extracted completely without causing damage to the model. Thus, the task can be expressed as:

$$M'_s = Emb(M_s, \mathbf{e}), \tag{2}$$

$$M'_t = Ada(M'_s), \tag{3}$$

$$\begin{cases} (M_s, \mathbf{e}) = Ext(M'_s), \\ (M_t, \mathbf{e}) = Ext(M'_t), \end{cases} \tag{4}$$

where $Emb(\cdot)$ and $Ext(\cdot)$ denote the embedding and extraction algorithms, M'_s and M'_t represent the embedded source and target models, and $Ada(\cdot)$ denotes an adaptation function that transfers knowledge from a source model to a target model. A proof is provided in Appendix A to demonstrate that the embedded content is not affected by Eq. (3) in the process of knowledge transfer.

The working principle of this paper is based on the hypothesis that if a model M can perform well in the source and target domains, then if a series of trigger samples can be generated in the OOD region (i.e., neither the source nor the target domain), M will also learn the knowledge in this region as a specific pattern of the watermark, which can be adopted for forensics. We summarize the characteristics of copyright protection for deep transfer learning models:

- Against piracy: Both the source and target models need to be protected because the target model may be easily obtained according to the source model.
- Against corruption: The process of adapting the watermarked source model to the target domain requires resistance to various attacks, including fine-tuning, pruning, and cross-domain matching.
- Against takeover: In the whole process of deep transfer learning, it is possible to be subjected to replacement attacks. Therefore, the watermark needs to be reversible to provide integrity authentication.

In transfer learning scenarios, we make three assumptions about the adversary. First, the adversary intends to put its own watermark on the source and target models or to destroy the legitimate watermark. Second, the adversary is not willing to sacrifice the functionality of the deep transfer learning model, that is, if the attack significantly degrades the performance of the model, it means that the attack fails. Third, the adversary has limited source and target data, as well as limited computational resources; otherwise, the adversary can train its own deep transfer learning model from scratch, which will cause the IP protection of the model to become unimportant. Our goal is to make it sufficiently difficult to break the watermark so that it is more cost-effective for the adversary to pay reasonable licensing fees.

4 Proposed SRDW Scheme

Based on the above analysis, we propose a subnetwork-lossless robust watermarking scheme for deep transfer learning models (DTLMs) as shown in Fig. 3, which can effectively realize the IP protection of DTLMs. The primary goal is to generate key samples and embed them successfully without updating the parameters related to the inference performance of normal input data. Specifically, the key samples have to satisfy three criteria:

1) Distinguishability of key samples: A reasonable degree of differentiation (DOD) should be maintained between the key and task samples. As a high DOD makes it easy to perceive embedding operations, while a low DOD is prone to misclassification.
2) Preservation of model functionality: The watermark should have no negative impacts on the functionality of DNNs in the process of embedding and extracting.
3) Control of labels: The labels of key samples should be easily manipulated by the legitimate DNN model.

To fulfill these criteria, we present a novel OOD-guided data augmentation, which can explore inherent distribution relationships in the source and target data and guarantee the synthetic samples located in reasonable regions. Therefore, we generate new samples close to the boundary regions of the source and target domains as key samples so that the models can easily handle their labels with minor modifications. In addition, we design the reversible compensation information to ensure that the watermark can be extracted from the embedded source and target networks without harming the carrier in the absence of malicious jamming attacks. For a quick reference, Table 1 presents the notations and their definitions used in this article. The details of each step will be provided in the following sections.

4.1 OOD-Guided Data Augmentation

To obtain robust trigger samples and promote useful knowledge transfer, we propose an OOD-guided data augmenta-

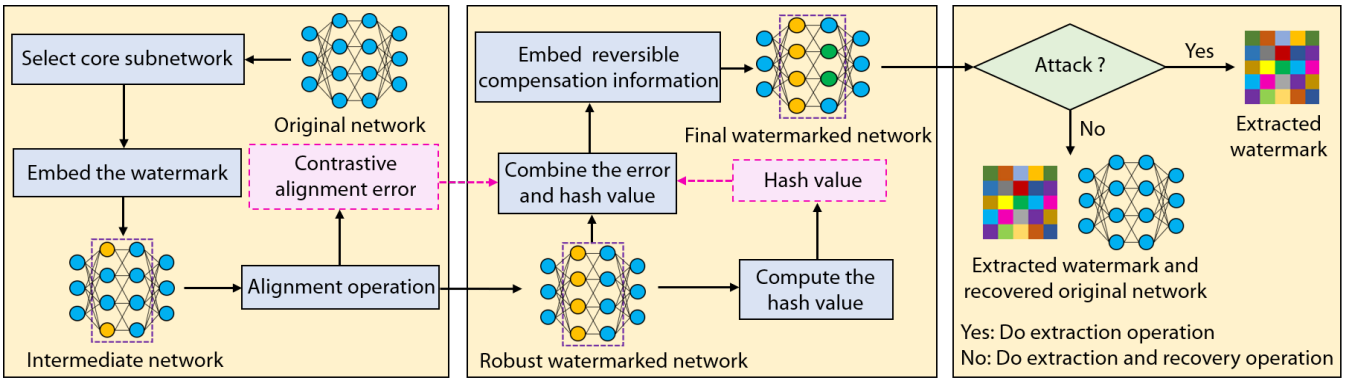


Fig. 3: A novel subnetwork-lossless robust DNN watermarking framework for deep transfer learning models in this paper.

TABLE 1: Descriptions of notations used in this article.

Notation	Definition
M_s	Source model
M_t	Target model
$\mathbf{x}_s$	Source sample
y_s	Source data label
$\mathbf{x}_t$	Target sample
y_t	Target data label
$\hat{\mathbf{x}}$	Generated sample
$\hat{y}$	Generated data label
$\mathbf{e}$	Substantive information vector
M'_s	Embedded source model
M'_t	Embedded target model
$\mathbf{z}$	Potential coding vector
τ	Domain label condition
θ	Distribution parameter
ϕ	Condition distribution parameters
$\mu_{\mathbf{x}}$	Mean value of samples
$\sigma_{\mathbf{x}}$	Standard deviation of samples
n_s	Total number of source samples
n_t	Total number of target samples
ψ	Variational posterior distribution parameters
γ_1, γ_2	Adjustable thresholds
$\boldsymbol{\eta}$	Modification function
$\boldsymbol{\Theta}$	Model parameters
$\boldsymbol{\alpha}$	Neural network layer parameters
λ	Adjustable control parameter
E_{ca}	Contrastive alignment error
β_1, β_2	Balance parameter
ξ	Margin value
H	Hash value
N_d	Original core subnetwork
N_w	Original watermarked core subnetwork
$\widetilde{N}_w$	Current watermarked core subnetwork
N_w^a	Attacked watermarked core subnetwork

tion model based on constrained variational autoencoder (CVAE) to generate synthetic features through conditioned domain labels rather than simple class labels.

As illustrated in Fig. 4, for an input sample $\mathbf{x}$ from either the source or target data, the goal of the encoder is to simulate a distribution $p_\theta(\mathbf{z})$ (i.e., approximated by $q_\phi(\mathbf{z}|\mathbf{x}, \tau)$), where the potential coding vector $\mathbf{z}$ can be selected and then fed into the decoder to obtain the reconstructed input feature $\hat{\mathbf{x}}$. Concretely, θ and ϕ are the parameters that describes the corresponding data distribution, and τ represents the domain label condition (i.e. $\tau \in \{s, t\}$). The decoder can be parameterized through $p_\theta(\mathbf{x}|\mathbf{z}, \tau)$, so that the model is expected to guide the generation of new data with the prior knowledge of source and target samples. In basic CVAE, the loss function consists of the following two components:

$$\mathcal{L}_c(\mathbf{x}; \phi, \theta) = \mathcal{L}_{re}(\mathbf{x}, \hat{\mathbf{x}}) + D_{KL}(\mathcal{N}(\mu_{\mathbf{x}}, \sigma_{\mathbf{x}})||\mathcal{N}(0, I)), \quad (5)$$

where $\mu_{\mathbf{x}}$ and $\sigma_{\mathbf{x}}$ denote the mean and the standard deviation, respectively. The first term $\mathcal{L}_{re}$ estimates the reconstruction error and the second term D_{KL} calculates the KL divergence between the empirical distribution and the standard normal distribution. The KL divergence is a powerful constraint term that can force the learned potential coding vector $\mathbf{z}$ to follow the standard normal distribution. This additional constraint allows the learned model to obtain the ability to generate the desired data from the potential coding vector $\mathbf{z}$ collected from the standard normal distribution.

To implement the OOD-guided data generation, we modify the loss function and introduce the evaluation mechanism for synthetic samples. To this end, the improved loss function can be given in the following form:

$$\begin{aligned}\mathcal{L}_{ic}(\mathbf{x}; \phi, \theta) = \mathcal{L}_{re}(\mathbf{x}_s, \hat{\mathbf{x}}_s) + \mathcal{L}_{re}(\mathbf{x}_t, \hat{\mathbf{x}}_t) \\ + \mathcal{L}_{re}(\mathbf{x}_s, \hat{\mathbf{x}}_{st}) + \mathcal{L}_{re}(\mathbf{x}_t, \hat{\mathbf{x}}_{ts}),\end{aligned} \quad (6)$$

where $\hat{\mathbf{x}}_{st}$ denotes the reconstructed source sample from the target domain, and $\hat{\mathbf{x}}_{ts}$ is the reconstructed target sample from the source domain. The first two terms calculate the non-cross-domain reconstruction errors for the source and target samples, while the last two terms measure the cross-domain reconstruction errors. Subsequently, we evaluate the generated data by introducing likelihood gap (LG), which measures the underlying log likelihood improvement of model configuration that makes the likelihood of a single sample larger than the population likelihood.

Motivated by the following observations, if a generative model is well trained, given in-distribution (ID) test samples, the current model configuration may be close to the optimal one and the improvement of the likelihood is relatively small, so that the LG will be low; however, given OOD test samples, since the pre-trained model has not seen similar samples, the current model configuration may differ considerably from the optimal one and the improvement of the likelihood is relatively large, so that the LG will be high. Consequently, LG can serve as a basis for the region division of synthetic samples. For a specific input sample $\mathbf{x}$,

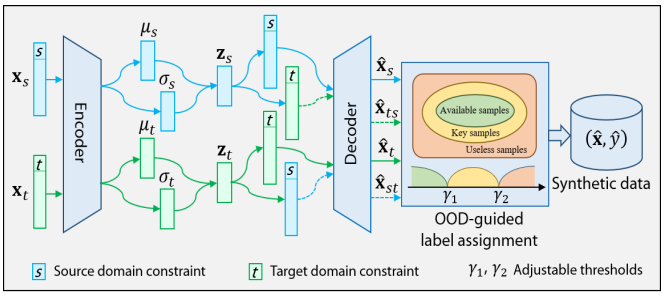


Fig. 4: The diagram of OOD-guided data augmentation based on constrained variational autoencoder. The encoder and decoder are constrained by the domain label s or t. Given the source sample $\mathbf{x}_s$ or the target sample $\mathbf{x}_t$ as the input, the model generates reconstructed samples $\hat{\mathbf{x}}_s$, $\hat{\mathbf{x}}_{ts}$, $\hat{\mathbf{x}}_t$, and $\hat{\mathbf{x}}_{st}$, which can be divided into three categories (i.e., available samples, key samples, and useless samples). The available samples can be utilized to improve the performance of the task model, and the key samples are employed to implement watermark embedding.

we use $\psi(\mathbf{x}, \phi)$ to represent the sufficient statistics $(\mu_{\mathbf{x}}, \sigma_{\mathbf{x}})$ of $q_\phi(\mathbf{z}|\mathbf{x}, \tau)$. In this way, the loss function in Eq. (6) can be expressed as $\mathcal{L}_{ic}(\mathbf{x}; \psi(\mathbf{x}, \phi), \theta)$ to reveal the dependence on $\psi(\mathbf{x}, \phi)$. According to the variational evidence lower bound (ELBO) [41], we train the CVAE to obtain optimal parameters (ϕ^*, θ^*) by minimizing the improved loss function instead of maximizing the likelihood, i.e.,

$$(\phi^*, \theta^*) = \arg\min_{(\phi,\theta)} \frac{1}{n_s + n_t} \sum_{i=1}^{n_s+n_t} \mathcal{L}_{ic}(\mathbf{x}_i; \psi(\mathbf{x}_i, \phi), \theta), \quad (7)$$

where n_s and n_t are the total number of source and target samples, respectively. Moreover, if the decoder parameters θ^* are fixed, the optimal configuration of the variational posterior distribution parameters $\hat{\psi}(\mathbf{x}, \phi)$ can be found by minimizing the improved loss function as follows:

$$\hat{\psi}(\mathbf{x}, \phi) = \arg\min_{\psi} \mathcal{L}_{ic}(\mathbf{x}; \psi, \theta^*). \quad (8)$$

In other words, $\hat{\psi}(\mathbf{x}, \phi)$ is the optimal posterior distribution of the potential coding vector $\mathbf{z}$ derived from the training data in the case of the specific input $\mathbf{x}$ and the optimal decoder θ^*. Therefore, the likelihood gap (LG) of the input sample $\mathbf{x}$ can be defined as:

$$LG(\mathbf{x}) = \mathcal{L}_{ic}(\mathbf{x}; \hat{\psi}(\mathbf{x}, \phi), \theta^*) - \mathcal{L}_{ic}(\mathbf{x}; \phi^*, \theta^*). \quad (9)$$

LG can be interpreted as the difference between the likelihood obtained from the generation model (CVAE) with the optimal configuration and the likelihood obtained from the CVAE learned on the training set.

The synthetic samples are mapped from a high-dimensional feature space to a label space including three categories, including available samples (e.g., denoted by "0"), key samples (e.g., denoted by "1"), and useless samples (e.g., denoted by "2"). The corresponding label can be recorded as:

$$\hat{y} = \begin{cases} 0, & if \quad LG \le \gamma_1, \\ 1, & if \quad \gamma_1 < LG \le \gamma_2, \\ 2, & if \quad \gamma_2 < LG, \end{cases} \quad (10)$$

where γ_1 and γ_2 are the two adjustable thresholds. In practice, the distribution of the generated data is compared with the distribution of the available data to adaptively adjust the threshold so that the generated samples fall in a plausible region.

4.2 Alternating Optimization Embedding

In order to maintain the predictive capability of the original model, we propose an alternating optimization embedding based on module risk minimization to explore the most suitable modification space for robust watermark embedding. The robust watermark is reflected in both the specific patterns and the substantive contents. Specifically, the specific patterns of the watermark are realized by key samples and copyright verification labels. The substantive contents of the watermark are composed of the embedding modification and the compensation information. The embedding modification aims to implement the specific trigger-response patterns, while the compensation information is designed to achieve the lossless recovery of core subnetwork for integrity authentication.

In functional lottery ticket hypothesis, the findings suggest that a randomly initialized full network contains an initialized subnetwork, and it can achieve better OOD performance (i.e., a wider recognition region) than the original full network for a given functionality (e.g., handwritten digit classification) when trained individually [42]. Our objective is to ensure that the OOD performance can be generalized to key sample regions.

Concretely, for a source model $M_s(\mathbf{x}; \boldsymbol{\alpha})$, there exists a modification function $\boldsymbol{\eta}$ guiding the model $M_s(\mathbf{x}; \boldsymbol{\eta}(\boldsymbol{\alpha}'))$ with initialization parameter $\boldsymbol{\alpha}'$ to expand the OOD performance to the region where the key samples are located. Therefore, an ideal alternative is to characterize the best modification function, i.e., the mapping from $\boldsymbol{\alpha}$ to the optimal parameter $\boldsymbol{\eta}^*(\boldsymbol{\alpha})$. We capture the changes in $\boldsymbol{\alpha}$ through scaling and shifting the corresponding hidden units of $M_s(\mathbf{x}; \boldsymbol{\eta}(\boldsymbol{\alpha}))$, which in turn requires the additional parameters $\boldsymbol{\eta}'$, denoted as $\boldsymbol{\Theta} = \{\boldsymbol{\eta}(\boldsymbol{\alpha}), \boldsymbol{\eta}'\}$. Given a loss function l (e.g., cross entropy loss) and a regularization term $R(\boldsymbol{\eta}, \boldsymbol{\alpha})$ (e.g., squared L_2-norm distance), we are interested in

$$\boldsymbol{\eta}^*(\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{\eta}} \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_{s\cup k}}[\mathcal{L}(\mathbf{x}, y; \boldsymbol{\eta}, \boldsymbol{\alpha})], \quad (11)$$

$$\mathcal{L}(\mathbf{x}, y; \boldsymbol{\eta}, \boldsymbol{\alpha}) = l(M_s(\mathbf{x}; \boldsymbol{\alpha}), y) + R(\boldsymbol{\eta}, \boldsymbol{\alpha}), \quad (12)$$

where $\mathcal{D}_{s\cup k}$ is the data region where the source and key samples are located, and $\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_{s\cup k}}[\cdot]$ represents the expectation that is uniformly distributed over $\mathcal{D}_{s\cup k}$.

By replacing the typical objective in Eq. (11), our desired training objective can be defined as follows:

$$\min_{\boldsymbol{\Theta}} \mathbb{E}_{\boldsymbol{\alpha}\sim p(\boldsymbol{\alpha}),(\mathbf{x},y)\in\mathcal{D}_{s\cup k}}[\mathcal{L}(\mathbf{x}, y; \boldsymbol{\Theta}, \boldsymbol{\alpha})], \quad (13)$$

where $p(\boldsymbol{\alpha})$ denotes the distribution of the parameter $\boldsymbol{\alpha}$. The main benefit of Eq. (13) is that a single training learns a parametric relationship $\boldsymbol{\alpha} \rightarrow M_s^{\hat{\boldsymbol{\Theta}}}(\mathbf{x}; \boldsymbol{\alpha})$ between the modification and performance, which approximates the description of $M_s^{\hat{\boldsymbol{\Theta}}}$ for the parameters in the support of $p(\boldsymbol{\alpha})$. We adopt $p(\boldsymbol{\alpha}|\boldsymbol{\epsilon})$ to represent a distribution with a bound of $\boldsymbol{\epsilon}$ containing the range $\boldsymbol{\alpha}$. The training process first performs a stochastic gradient search on $\boldsymbol{\Theta}$ and a joint sampling of $\boldsymbol{\alpha} \sim p(\boldsymbol{\alpha}|\boldsymbol{\epsilon})$ and $(\mathbf{x}, y) \in \mathcal{D}_{s\cup k}$ in Eq. (13). Subsequently, the

tuning step implements a stochastic gradient update on $\boldsymbol{\epsilon}$ by minimizing the validation objective. Inspired by the variational inference [43], we introduce an entropy regularization term $\mathcal{T}[\cdot]$ controlled by λ to prevent $p(\boldsymbol{\alpha}|\boldsymbol{\epsilon})$ from changing into a degenerate distribution, so the following equation can be obtained:

$$\min_{\boldsymbol{\epsilon}} \mathbb{E}_{\boldsymbol{\alpha}\sim p(\boldsymbol{\alpha}|\boldsymbol{\epsilon}),(\mathbf{x},y)\in\mathcal{D}_{s\cup k}^{val}}\left[l_{val}(M_s^{\hat{\boldsymbol{\Theta}}}(\mathbf{x};\boldsymbol{\alpha}),y)-\lambda\mathcal{T}[p(\boldsymbol{\alpha}|\boldsymbol{\epsilon})]\right], \tag{14}$$

where $\mathcal{D}_{s\cup k}^{val}$ is the region where the source and key samples are located in the validation data, l_{val} denotes the loss function (e.g., cross entropy loss) on the validation data, and $\lambda \geq 0$ is an adjustable control parameter.

By investigating the trigger response rules between the data and the model, the regularized representations of key samples are utilized to identify the core subnetwork, while the embedding objective function is optimized to minimize the modification loss. The success rate of watermark embedding is enhanced by explicitly exploring the discriminative direction of the core subnetwork instead of randomly selecting from the candidate key samples. Some hyperparameters of the neural network (e.g., regularization parameters) can be optimized by estimating the best response function. In other words, a matching trigger response is implemented by constructing a mapping from the hyperparameters to the parameters of the neural network. Therefore, Eq. (14) learns this mapping by tuning the parameters of the model on the validation set, and the embedding of the watermark is finally achieved by alternating optimization iterations. Furthermore, we observe that the model subnetworks can be regulated to achieve adaptive optimization while avoiding fitting spurious correlations by imposing the necessary parametric constraints, which have been verified in Section 6.6.

4.3 Reversible Compensation Information Acquisition

The reversible compensation information is obtained by calculating the contrastive alignment error and the corresponding hash value, which is an important guarantee to achieve watermark reversibility. The contrastive alignment error is utilized to measure the difference between the original network and the watermarked network. We minimize the contrastive alignment error to make the representations of the aligned sublayers have a very small distance while those of the unaligned sublayers have a large distance:

$$E_{ca}(\boldsymbol{\alpha}) = \sum_{(i,i')\in\mathcal{M}^+} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_{i'}\|_2 + \sum_{(j,j')\in\mathcal{M}^-} \beta_1[\|\boldsymbol{\alpha}_j - \boldsymbol{\alpha}_{j'}\|_2 - \xi]_+, \tag{15}$$

where $\mathcal{M}^+$ and $\mathcal{M}^-$ denote the set of layers aligned and unaligned after the training of key samples, respectively. $[\nu]_+$ means to compare the element ν with 0 and take the larger element as its value, i.e., $[\nu]_+ = \max(0,\nu)$. β_1 is a balance parameter. The distance of unaligned layers is supposed to be larger than a margin ξ, i.e., $\|\boldsymbol{\alpha}_j - \boldsymbol{\alpha}_{j'}\|_2 > \xi$. In this way, the specific patterns and the substantive contents of the watermark can be associated with each other to achieve reversible embedding. In addition, the mathematical proof for the recovery of the original parameters in the core subnetwork can be seen in Appendix B.

In the knowledge transfer from source domains to target domains, the watermarked source models would be attacked in an unavoidable way (e.g., fine-tuning the pre-trained model parameters or pruning the redundant convolution channels) or suffer from malicious design manipulations such as removing the watermark through overwriting. Therefore, it is necessary to authenticate the integrity of the deep transfer learning model at the user terminal. To this end, the robust watermarked network N_{w_1} (e.g., source or target network) is hashed to produce a hash value H, which is offered as part of the reversible compensation information for fragile authentication. The reversible compensation information is first compressed in a lossless way and then embedded into the edge of the core subnetwork, which can be isolated from the robust watermark embedding area. Such a way is beneficial to ensure the reversibility and reduce the impact on watermarking performance. However, when the reversible compensation information is too large to be fully embedded into the edge area, the remaining part will be embedded into the frozen shallow layers (i.e., non-core subnetwork).

The lossless compression aims to find a smaller network that is locally equivalent to the original network. In other words, the local equivalence for the underlying input data is sufficient to guarantee that both networks have the same performance in any test environments. To make the core subnetwork more compact, we need to explore the relationships between data inputs, network connections and final outputs to embed more reversible compensation information at the edge of the core subnetwork through the lossless compression technique. We introduce a probabilistic connection importance evaluation to determine whether the connections are correlated with their outputs in a DNN. Specifically, we adopt the probabilistic tensor product decomposition [44] to split the association of two connected nodes into two components, including correlated outputs and uncorrelated outputs. If the strength of the component is high, this component is retained. Otherwise, it is removed. Meanwhile, the weights and biases of these layers are adjusted accordingly.

Since the reversible embedding process can be regarded as an additive blending operation while the watermark message is embedded into the core subnetwork, the embedding operation of reversible compensation information in the non-core subnetwork has little effect on the watermark. Our proposed subnetwork-lossless robust watermarking is summarized in Algorithm 1. Specifically, to preserve the functionality of the model, we find the most suitable embedding modification space based on module risk minimization to update the space-specific parameters, instead of all parameters. We adjust the sample sizes between key samples and task samples during training to rectify the learning bias to achieve the robust DNN watermarking, as shown in steps 8-14. Afterwards, we verify the existence of the watermark in the model by observing the specific trigger-response patterns and by extracting the substantive contents. On the one hand, we generate key samples through the OOD-guided data augmentation technique to make the model output the corresponding copyright verification labels, which can be used to verify whether the model contains a watermark. On the other hand, a more accurate approach is to effectively

Algorithm 1 Subnetwork-Lossless Robust Watermarking

Input: source datasets $\mathcal{D}_s = \{(\mathbf{x}_s^i, y_s^i)\ (i = 1, 2, ..., n_s)\}$, target datasets $\mathcal{D}_t = \{(\mathbf{x}_t^j, y_t^j)\ (j = 1, 2, ..., n_t)\}$, pre-training parameters $\mathbf{\Theta}_{pre}$, number of iteration T;
Output: number of key samples n_k, watermarked model M_w;

1: Initialize $\delta = 0$;
2: **While** $\delta < T$ **do;**
3: $\delta \leftarrow \delta + 1$;
4: Train the encoder and decoder using $\mathcal{D}_s$ and $\mathcal{D}_t$ according to Eq. (6);
5: Search the optimal configuration of variational parameters by minimizing the loss function in Eq. (7);
6: Assign the corresponding labels for synthetic samples using Eq. (10);
7: **End while**
8: **For** $\zeta = 1, 2, ..., n_k$ (Optimization Embedding) **do**
9: If $\zeta = 0$, initialize parameters $\mathbf{\Theta}' = \mathbf{\Theta}_{pre}$, otherwise, $\mathbf{\Theta}' = \mathbf{\Theta}_{\zeta-1}$, where $\mathbf{\Theta}_{\zeta-1}$ represents the set of parameters for the model when the embedding number of key samples is $\zeta - 1$;
10: Explore the most suitable modification space for robust watermark embedding by Eq. (11) and Eq. (12);
11: Guide the update of parameters during the watermark embedding based on Eq. (14);
12: Calculate the contrastive alignment error E_{ca} and the corresponding hash value H;
13: Embed the reversible compensation information into the edge of the core subnetwork through lossless compression;
14: **End for**

extract the substantive information of the watermark and then determine the existence of the watermark by integrity authentication. Moreover, we show how to verify whether the model contains a watermark through experiments in Section 6.2 and Section 6.4.

4.4 Content Integrity Authentication

In this section, we explore the feasibility to realize the integrity authentication by the application of hash value. Note that we verify the integrity between the source and target core subnetworks, not the entire model. We record $\widetilde{N}_{w_2}$ as the current version of the final watermarked network N_{w_2}. To authenticate the integrity of the core network after knowledge transfer, we first obtain the hash value H_1 using the same reversible watermarking technique. Subsequently, the core subnetwork $\widetilde{N}_{w_1}$ is recovered and the corresponding hash value H_2 will be calculated. If the hash values H_1 and H_2 are identical, we assume that there is no malicious attack during knowledge transfer or model deployment, and the original core subnetwork can be recovered losslessly. Otherwise, it means that the network N_{w_2} is subject to a malicious attack, which results in the permanent loss of the network. In this case, the watermark can be directly extracted from the network $\widetilde{N}_{w_2}$ for robust authentication. The details of extracting the robust watermark and recovering the original core subnetwork will be presented.

a) Extraction under non-attack conditions

As mentioned above, we can effectively extract the robust watermark and simultaneously recover the original core subnetwork if the model is not attacked. The process of watermark extraction and original core subnetwork recovery is described as follows.

1) Extraction of the reversible compensation information: At the user terminal, we have $\widetilde{N}_{w_2} = N_{w_2}$ if there is no malicious jamming attack. In this way, we can accurately extract the reversible compensation through the same watermarking technique. Subsequently, the robust watermarked subnetwork N_{w_1} can be recovered.

2) Extraction of the watermark: Under the condition of no malicious jamming attacks, the watermark information can be extracted by using Eq. (14) in alternating optimization to determine the embedding position and calculate the substantive content.

3) Recovery of the original core subnetwork: After extracting the robust watermark, the reversible compensation information is utilized to recover the original core subnetwork from watermarked network according to Eq. (15). Following the description in Section 4.3, the extracted contrastive alignment error E_{ca} will be applied to compensate the network $\widetilde{N}_d$ ($d \in \{s, t\}$) to recover the original core subnetwork N_d, which can be expressed as:

$$\widetilde{N}_d = N_d + E_{ca}. \tag{16}$$

b) Extraction under attack conditions

We denote the network $N_{w_2}^a$ as the attacked version of the watermarked network N_{w_2}. Although the original core subnetwork cannot be recovered under the attack scenarios, the embedded watermark is still be effectively extracted because the subnetwork determined by alternate optimization based on module risk minimization has strong robustness against the attacks of fine-tuning and pruning operations.

Similar to the embedding process of watermarking, the watermark extraction is performed using the differences between before and after the embedding modifications by Eq. (14). Through the calculation of reversible compensation information via Eq. (15), we estimate the magnitude of the contrastive alignment error to accomplish accurate extraction of the watermark. In this way, we effectively extract the embedded substantive watermark information in case of malicious jamming attacks, which can be applied for the content integrity authentication. Specifically, by imposing the necessary parametric constraints, the model subnetwork is adjusted using Eq. (14) to achieve adaptive optimization, while eliminating the risk of fitting spurious correlations. Subsequently, the contrastive alignment error contained in the reversible compensation information needs to be computed by Eq. (15) during the watermark embedding process. In the process of watermark extraction, the legitimate copyright owner can exploit the prior knowledge to obtain the reversible compensation information in a specific area. Moreover, the original parameters of the core subnetwork can be recovered by the contrastive alignment error, for which we provide a detailed mathematical proof in Appendix B. After recovering the original parameters, it is easy to obtain the substantive embedding information of the watermark to complete the extraction of the watermark. Therefore, both the extraction of the watermark by Eq. (14) and the calculation of the reversible compensation information by Eq. (15) do not require the original model.

5 Experimental Setting and Implementation

5.1 Datasets and Models

In our experiments, we evaluate the SRDW on the popular datasets, including the Rotated MNIST, CIFAR-10 & STL-10, and PACS. The detailed information about these datasets is summarized in Table 2. We adopt four typical deep transfer learning architectures (i.e., standard knowledge distillation (SKD) [45], domain-adversarial neural network (DANN)[1] [46], multi-source distilling domain adaptation (MDDA)[2] [47], and robust feature adaptation (RFA)[3] [48]) on these datasets to complete knowledge transfer.

In the OOD-guided data augmentation process, we generate key samples, as well as auxiliary source and target samples, by the procedure described in Section 4.1. In the robust reversible watermark embedding process, we seek the most rational modification structure (i.e., core subnetwork) by the alternating optimization mechanism based on module risk minimization to achieve the watermark embedding. In the process of adapting the source model to the target domain tasks, we execute the corresponding attack strategies according to different transfer learning schemes. Ultimately, we perform integrity authentication by extracting the substantive content and recovering the core subnetwork in the presence or absence of attacks. Here, we provide the implementation details of the datasets and related models in our evaluation:

- Rotated MNIST: It is a modified version of MNIST produced by rotating the original grayscale hand-written digits from 0° to 90° with an interval of 15°. As a baseline comparison, we first train a deep transfer learning network using standard knowledge distillation techniques [45]. Specifically, we take LeNet-5 as the teacher model. The shadow model contains two convolution layers (i.e., 32 and 64 channels) and two fully-connection layers (i.e., 1024 and 10 channels) to construct the student model developed by the user. Subsequently, following the setting of [46], we compose the feature extractor with two or three convolutional layers and then pick their exact configurations by parameter optimization. We use the domain adapter with three fully connected layers and the stochastic gradient descent (SGD) with a momentum of 0.9. The initial learning rate is set to 10^{-3} which is scaled by a factor of 10 at 50k iterations.
- CIFAR-10 & STL-10: Both CIFAR-10 and STL-10 are 10-class image datasets consisting of 60,000 32×32 color images and 13,000 96×96 color images, respectively. These two datasets are similar but only one class is different. Therefore, we employ the common nine classes in our experiments. In the data preprocessing phase, we resize the image in STL-10 to the size of 32×32. We adopt Alexnet as our backbone. The last layer is employed as a classifier and the other layers are utilized as encoders. According to [47], we set the balancing coefficient to 10 empirically. We use these deep transfer learning settings to evaluate our proposed SRDW scheme.
- PACS: It contains seven categories (dog, elephant, giraffe, guitar, horse, house, person) and four different domains: i.e., Art paintings (A), Cartoon (C), Photo (P) and Sketch (S). We use the ResNet-50 architecture as the backbone model and RFA [48] as the knowledge transfer algorithm, with the exception of the weight decay setting as 10^{-5} and the learning rate setting as 10^{-4}. The training iteration is determined as 1,200 with a batch size of 128 samples, and the learning rate is decayed by a factor of 0.1 at 900 iterations. We implement our scheme using this deep transfer learning network trained on PACS.

1. https://github.com/fungtion/DANN/
2. https://github.com/daoyuan98/MDDA/
3. https://awaisrauf.github.io/robust_uda

TABLE 2: Summary of the datasets in our evaluation experiments.

Datasets	Source Domains	Target Domains
Rotated MNIST	15°-30°-45°-60°	0°-90°
	15°-45°-60°-75°	0°-90°
	30°-45°-60°-75°	0°-90°
CIFAR-10 & STL-10	CIFAR-10	STL-10
	STL-10	CIFAR-10
PACS	C-P-S	A
	A-P-S	C
	A-C-S	P
	A-C-P	S

5.2 Hostile Attacks

We evaluate the effectiveness of the proposed scheme against the following four commonly-considered hostile attacks in deep transfer learning scenarios.

Fine-Tuning. The adversary can remove the watermark while maintaining the model accuracy by fine-tuning part of the network layers on the original data (i.e., task samples). In our experiment, the watermarked models are fine-tuned using the corresponding validation data.

Pruning. The pruning is a powerful technique to compress well-trained models, which can be misused by adversaries to alter the embedded watermarks. We employ the technique from [49] to implement the pruning attack.

Cross-Domain Matching. Cross-domain matching aims to bridge the distribution gap between the source and target domains by using the original watermarked network as a pre-trained model. In this way, the watermarked network will suffer from various complex processing operations based on the corresponding distribution matching techniques. In Section 6.4, we verify that it is feasible to extract the watermarked substantive content and recover the core subnetwork completely in the absence of malicious jamming attacks.

Watermark Overwriting. The adversary attempts to select a new set of key samples and then use the proposed scheme to embed a second watermark for the purpose of overwriting the first watermark without compromising the inference accuracy. The second watermark is randomly selected in our experiments.

5.3 Evaluation Metrics

We adopt the following four evaluation metrics to discuss the performance of the proposed scheme.

Fidelity is measured by the embedding success rate R_e, the accuracy loss R_l, and the number of updated parameters. Specifically, R_e calculates the percentage of

key samples that are successfully embedded in the deep transfer learning networks. We desire a high embedding success rate R_e and a low accuracy loss rate R_l, so that the watermarked network preserves the performance on normal test data. Concretely, the corresponding calculation equations are defined as follows:

$$R_e = \frac{N_{se}}{N_k}, \tag{17}$$

$$R_l = Acc_{be} - Acc_{ae}, \tag{18}$$

where N_{se} denotes the number of key samples successfully embedded, N_k is the total number of key samples, Acc_{be} represents the accuracy before embedding (i.e., accuracy of original model), and Acc_{ae} denotes the accuracy after embedding (i.e., accuracy of watermarked model).

Robustness is evaluated against various hostile attacks. We use the pattern retention rate R_p to quantify the preserved embedding capability of watermark. Similarly, the functionality retention rate R_f is employed to estimate the retained prediction capability of model, which is tested on the validation dataset. The embedded watermark should be difficult to remove when the R_f of the task-related inputs remains high. To be specific, the relevant evaluation equations are expressed as follows:

$$R_p = \frac{N_e}{N_{se}}, \tag{19}$$

$$R_f = \frac{Acc_{aa}}{Acc_{ae}}, \tag{20}$$

where N_e is the number of key samples that remain effective after the hostile attack, and Acc_{aa} denotes the model accuracy after the hostile attack.

Capacity is the maximum amount of information (i.e., number of key samples) that the proposed scheme can embed into a deep transfer learning network without violating other constraints.

Reversibility is an important guarantee used for integrity authentication. Our scheme can extract the watermark and losslessly recover the core subnetwork in case of no malicious jamming attack.

6 Experimental Results

In the experiments, our evaluation aims to answer the following research questions:

- **RQ1:** What are the embedding success rate and accuracy loss rate of SRDW under various attacks in the process of knowledge transfer?
- **RQ2:** Whether our SRDW is robust against the common hostile attacks, such as fine-tuning, pruning, cross-domain matching, and watermark overwriting.
- **RQ3:** What is the capacity of our proposed watermarking scheme on different deep transfer learning networks?
- **RQ4:** Whether the integrity authentication can be effectively performed by extracting the watermark and recovering the core subnetwork?

6.1 Evaluation on Fidelity (RQ1)

We validate the fidelity of the proposed SRDW on a variety of datasets and models. Table 3 shows the average embedding success rate and accuracy loss rate calculated from multiple runs of the experiments. From the results, it can be seen that most of the selected key samples can be successfully identified under the condition of different numbers of given key samples. For example, our DNN watermarking scheme can effectively embed 40 key samples into Rotated MNIST (SKD) with an accuracy loss rate of less than 0.09%. In addition, Fig. 5 shows the percentage of modified parameters when the watermark embedding is performed on the deep transfer learning networks. We observe that the algorithm tunes less than 0.05% weights on the core subnetwork of the RFA model trained on the PACS dataset, while achieving a high embedding success rate and a low prediction accuracy loss rate.

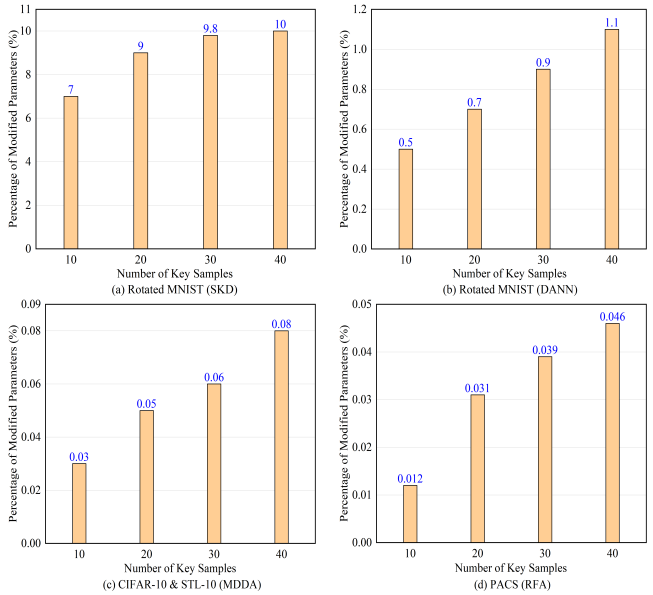


Fig. 5: Percentage of modified parameters for deep transfer learning networks under different number of key samples.

6.2 Evaluation on Robustness (RQ2)

We utilize the four hostile attacks mentioned in Section 5.2 to evaluate the robustness of the proposed scheme as follows.

1) Fig. 6 shows the performance of SRDW against fine-tuning attacks. The results indicate that our scheme exhibits robustness in the fine-tuning process. Specifically, although the functionality retention rate decreases after several fine-tuning tests, the pattern retention rate still remains stable during the entire process.

2) Fig. 7 presents the effect on the embedding ability of the watermark and the inference ability of the model with an increasing pruning rate. It is revealed that there is no loss in the prediction accuracy even using a pruning rate of 40%. However, as the pruning rate continues to increase, the prediction accuracy starts to decrease sharply. Moreover, we observe that SRDW performs better on a more complicated deep transfer learning network, which is attributed to the larger parameter space that can provide a more suitable and robust core subnetwork.

TABLE 3: Results of embedding success rate and accuracy loss rate in various deep transfer learning models.

Number	Rotated MNIST (SKD)		Rotated MNIST (DANN)		CIFAR-10 & STL-10 (MDDA)		PACS (RFA)	
	R_e	R_l	R_e	R_l	R_e	R_l	R_e	R_l
10	100%	0.00%	100%	0.01%	100%	0.05%	100%	0.04%
20	100%	0.04%	100%	0.03%	100%	0.07%	100%	0.06%
30	100%	0.06%	100%	0.07%	100%	0.06%	100%	0.08%
40	100%	0.09%	100%	0.11%	99.2%	0.12%	98.5%	0.14%

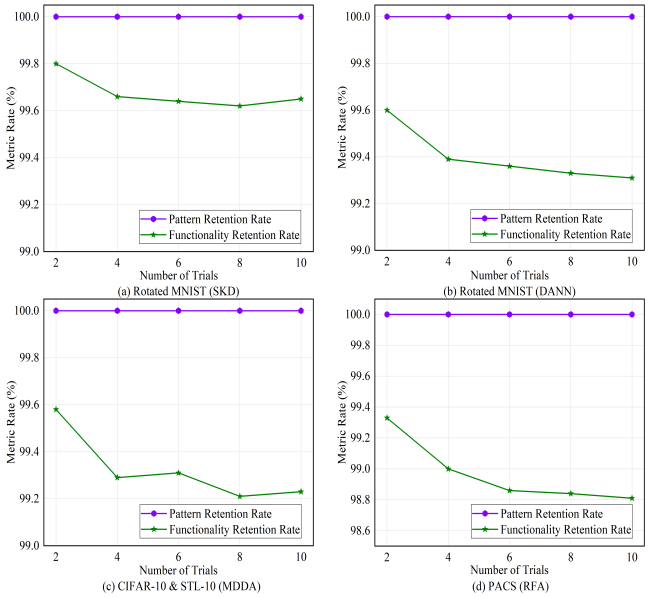


Fig. 6: Pattern retention rate and functionality retention rate in the process of fine-tuning on validation datasets.

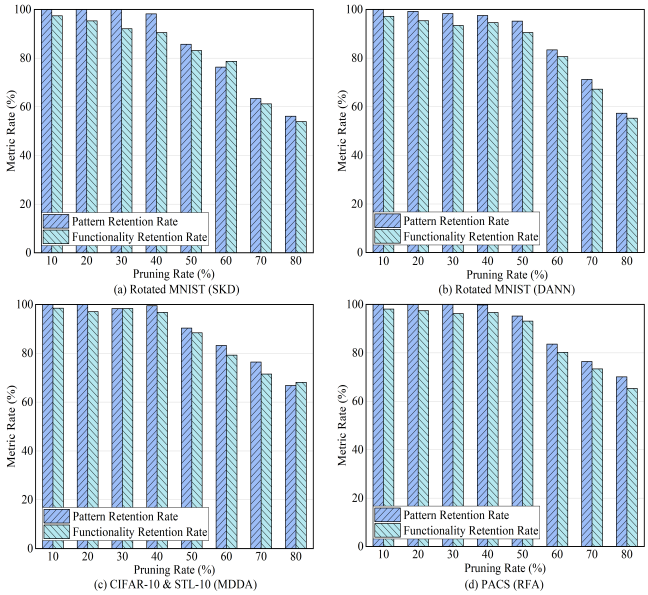


Fig. 7: Pattern retention rate and functionality retention rate under the condition of various pruning rates.

3) Table 4 demonstrates the impact on the embedding ability of the watermark (i.e., measured by pattern retention rate R_p) and the prediction ability of the model (i.e., measured by functionality retention rate R_f) during the process of knowledge transfer using various cross-domain matching strategies (e.g., knowledge distillation and distribution adaptation). From the results, it can be seen that the embedding ability of the watermark and the prediction ability of the model are not affected before and after the knowledge transfer. This is due to the fact that our scheme utilizes the distribution relationships between source and target samples to guide the generation of key samples, which is a data-centric perspective that can essentially match the data and the model perfectly to achieve robust deep transfer learning network watermarking.

TABLE 4: Robustness of our scheme against cross-domain matching manipulations in the process of knowledge transfer.

Models	Source→Target	Expanded Domains	R_p	R_f
SKD	15°-30°-45°→0°-90°	60°	100%	100%
DANN	30°-45°-60°→0°-90°	75°	100%	100%
RFA	A-P → C	S	100%	100%
RFA	C-P → S	A	100%	100%

4) We evaluate the robustness of the proposed scheme against watermark overwriting attacks, where an adversary attempts to insert an additional watermark into the model that disables the original legitimate watermark. From the results in Fig. 8, it can be found that our scheme is consistently stable against overwriting attacks on different deep transfer learning models. Moreover, the watermark embedding is more robust on relatively complex networks. Notably, the opposite results in Fig. 6 and Fig. 8 are caused by two aspects. On the one hand, the difference in the type of attack causes the discrepancy in the results. For example, Fig. 6 shows the results of the proposed scheme against the fine-tuning attack, while Fig. 8 shows the results of the proposed scheme against the watermark overwriting attack. On the other hand, the difference in the unit scale of the vertical coordinates also leads to a significant impact on the results.

6.3 Capacity Measurement (RQ3)

We present the variation trend of embedding success rate and functionality retention rate with different numbers of key samples in Fig. 9. Specifically, as the number of embedded key samples increases, the embedding success rate and functionality retention rate decrease, which is due to the fact that embedding more key samples requires modifying more weights. However, our proposed OOD-guided key sample generation considers the distribution relationships between the source and target domains to explore the suitable core subnetwork for watermark embedding through module risk minimization. In this way, the proposed watermarking algorithm has little influence on the prior learned knowledge of the network. Therefore, our scheme can achieve a satisfactory embedding success rate (i.e., more than 92%)

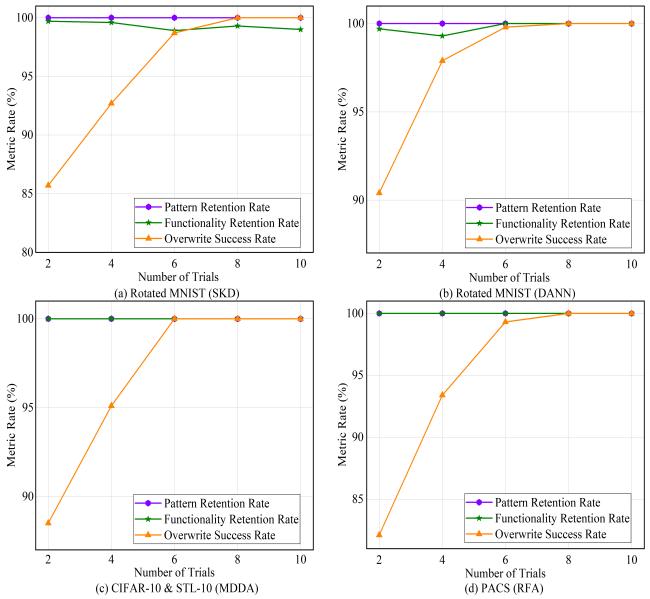


Fig. 8: Pattern retention rate and functionality retention rate in the process of overwriting attacks.

and functionality retention rate (i.e., at least 97.5% when embedding 90 key samples).

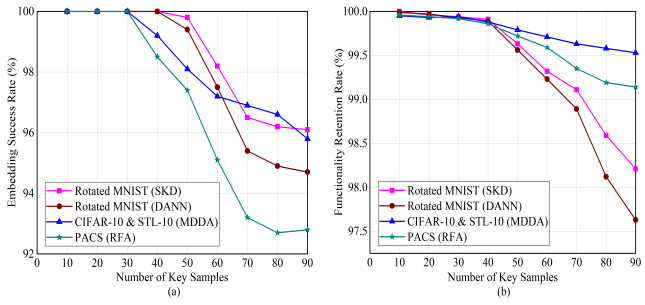


Fig. 9: Embedding success rate and functionality retention rate with different numbers of key samples.

6.4 Content Integrity Authentication (RQ4)

We evaluate the application of the proposed SRDW scheme for content integrity authentication. We first determine the location of the core subnetwork, and then use the hash algorithm (e.g., secure hash algorithm 256) to obtain the representation of the entire core subnetwork. The characteristics of the hash algorithm are employed to judge whether the model is subject to malicious jamming attacks, i.e., the hash value will change no matter where the attacker modifies the core subnetwork. In the absence of malicious jamming attacks, we adopt reversible compensation information to extract the substantive watermark content and recover the core subnetwork without loss. Similarly, in the case of malicious jamming attacks, only the watermark substantive content is extracted, and the results are provided in Table 5. All experiments indicate that our scheme can effectively extract the embedded watermark. Moreover, the core subnetwork can be recovered losslessly under the condition of no malicious jamming attack.

6.5 Impact of Reversible Compensation Information

We perform a set of experiments to evaluate the influence of reversible compensation information on the performance of the model. The simulation experiments are conducted using the lossless compression technique to observe the trend of model performance with the embedding amount of reversible compensation information. Fig. 10 shows the variation curve of model performance with the embedding amount of reversible compensation information. From the experimental results, it can be seen that as the embedding amount of reversible compensation information increases, the change curve of model performance can be divided into three stages. Namely, the model performance remains stable at first, then the model performance decreases slowly, and finally the model performance decreases sharply. Therefore, the embedding amount of reversible compensation information would not affect the performance of the model within a certain range. Moreover, when the embedding amount keeps increasing and the embedding cannot be completed in the edge area of core subnetwork, the remaining part will be embedded in the frozen shallow layer of non-core subnetwork, which will cause the performance of the model to degrade slowly. However, if the embedding amount is too large to embed in the edge area and the frozen shallow layer, the performance of the model decreases sharply. In addition, it can also be observed from the experiments that the embedding capacity of reversible compensation information using the lossless compression technique is about 300 bits to 600 bits.

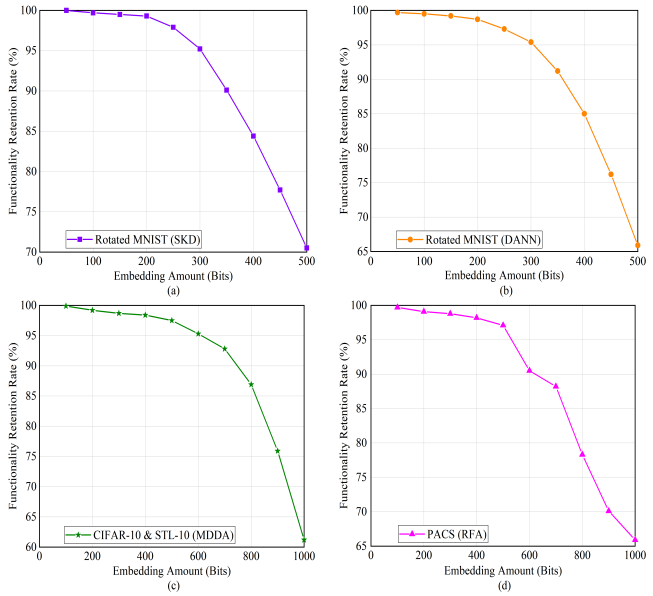


Fig. 10: Results of functionality retention rate with different embedding amounts of reversible compensation information.

6.6 Ablation Study

We conduct ablation experiments to investigate the following contents: 1) the influence of the threshold selection on our proposed scheme; 2) the impact of the batch normalization (BN) layer modification rate on the resistance to hostile attacks.

TABLE 5: Evaluation results of content integrity authentication under 30 key samples. "0" indicates without attack, while "1" denotes with attack. "Yes" means that the core subnetwork can be losslessly recovered, while "No" represents that the core subnetwork is irreversible.

Models	Source Domains	Target Domains	Attacks	Reversibility	Length of Watermark (Bits)
SKD	15°-30°-45°-60°	0°-90°	0	Yes	7648
			1	No	
DANN	30°-45°-60°-75°	0°-90°	0	Yes	5290
			1	No	
MDDA	STL-10	CIFAR-10	0	Yes	9735
			1	No	
RFA	A-C-S	P	0	Yes	13260
			1	No	

Effect of threshold settings. We observed that the threshold setting is crucial for OOD-guided key sample generation. In the following experiments, we investigate the impact of key samples generated by different threshold selections on embedding success rate and functionality retention rate with MDDA model on CIFAR-10 & STL-10 dataset. Our thresholds can divide the synthesized samples into three categories, including available samples (i.e., source or target samples), key samples, and useless samples (i.e., interference samples). We select five threshold settings under a given number of embedded key samples (e.g., 30 key samples), as shown in Fig. 11. From the experimental results, it can be seen that the appropriate threshold selection is beneficial to improving the embedding success rate and functionality retention rate. The reason is that the proper threshold setting can promote the quality of the generated samples, which can keep the key samples and task samples with a reasonable distinguishability. Specifically, for CIFAR-10 → STL-10 task, a threshold setting of (0.4, 0.8) enables our scheme to exhibit superior performance.

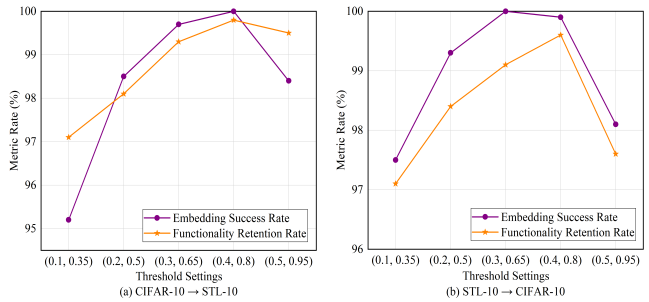


Fig. 11: Results of embedding success rate and functionality retention rate with different threshold settings on CIFAR-10 & STL-10 dataset.

Effect of BN layer modification rate. For the models using knowledge distillation strategies, we evaluate the impact of the BN layer modification rate on the resistance to hostile attacks (e.g., pruning attacks). We draw the curve of pattern retention rate and functionality retention rate with BN layer modification rate on three tasks (i.e., Task1: 15°-30°-45°-60°→0°-90°, Task2: 15°-30°-45°-60°→0°-90°, and Task3: 15°-30°-45°-60°→0°-90°) using a fixed pruning rate (e.g., 30% or 60%), as shown in Fig. 12. It can be found that as the increase of the BN layer modification rate, the performance of our scheme to resist the pruning attack gradually decreases. The reason may be attributed to the fact that the BN layers are correlated with the discriminative power of spurious correlations, which will result in a decrease in the inference ability of the model due to the excessive modification of BN layers.

6.7 Analysis and Discussion

We summarize a comparison table according to the properties of DNN watermarks, as shown in Table 6. Here we classify the watermarks into four categories, including non-robust irreversible, robust irreversible, non-robust reversible, and robust reversible. For non-robust irreversible watermark, which is similar to steganographic techniques [50], [51], the DNN model can be regarded as a carrier to achieve covert communications. For robust irreversible watermark, it is mainly applied for copyright protection. For non-robust reversible watermark, it can be utilized for global integrity authentication (i.e., the entire network can be recovered losslessly). For robust reversible watermarking, it can be deployed for local integrity authentication (i.e., the core subnetwork can be recovered losslessly).

Most of the existing approaches aim to embed key samples into specific regions to achieve robust watermarking, which ignores reversibility and results in a failure of integrity authentication. In this way, it is easily corrupted by attacks during knowledge transfer because these methods cannot seek the most suitable parameter space for embedding. Although it is intuitive to consider white-box-based embedding for reversible watermarking, the poor robustness restricts its practical application. Moreover, there are multiple complex network modification attacks during knowledge transfer, which can severely damage the global integrity of the model. Therefore, the watermarking of deep transfer learning model requires to be able to achieve the local integrity authentication. Our scheme essentially implements robust and reversible embedding of watermarks by considering the matching relationship between the data and model. From the experimental results in Section 6.2, it is clear that our scheme is effective for cross-domain matching attacks (e.g., knowledge distillation and deep domain adaptation) during the knowledge transfer. Meanwhile, in order to realize the reversibility, our scheme needs to restrict the embedding operations into the core subnetwork, which causes the proposed scheme to exhibit some limitations in terms of watermarking capacity. However, as analyzed and discussed above, our watermarking scheme can provide competitive results, such as fidelity, robustness and reversibility on deep transfer learning models.

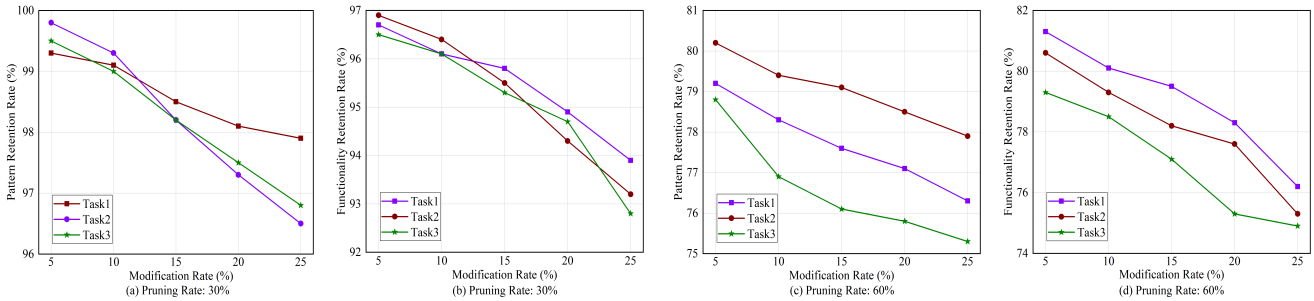


Fig. 12: Results of pattern retention rate and functionality retention rate with different BN layer modification rates under the condition of pruning attacks.

TABLE 6: Qualitative comparison for four different DNN watermarks according to the characteristics of DNN watermarks.

Properties of DNN Watermarks	Robustness	Capacity	Reversibility	Deep Transfer Learning Scenarios	Application
Non-Robust & Irreversible	×	Large	×	×	Secret Communications
Robust & Irreversible	√	Small	×	×	Intellectual Property Protection
Non-Robust & Reversible	×	Medium	√	×	Global Integrity Authentication
Robust & Reversible (Ours)	√	Medium	√	√	Local Integrity Authentication

7 Conclusion

Intellectual property protection for deep transfer learning models is an intriguing but challenging issue due to the uncertainty of knowledge transfer. Motivated by the reversible media watermarking and the excellent generalizability of DNNs, we innovatively propose a subnetwork-lossless robust watermarking for hostile theft attacks in deep transfer learning models. The following conclusions can be derived from this research work: 1) the OOD-guided data augmentation based on constrained variational autoencoder can generate samples to reveal the predictive relationships between the data and the model, which provides a crucial guidance to ensure that the synthetic key samples fall in a reasonable region; 2) a module risk minimization is introduced to find the most suitable modification space for robust watermark embedding, which can preserve the predictive capability of the original model in alternating optimization manner; 3) this paper presents a new DNN watermarking solution with reversibility and robustness for deep transfer learning models that can losslessly recover the core subnetwork for content integrity authentication in the absence of malicious jamming attacks; and 4) extensive empirical results indicate that our proposed scheme achieves superior performance on a wide range of settings and deep transfer learning architectures, and can further the facilitate robustness against hostile attacks by reducing the embedding modifications in batch normalization layers.

There are still some interesting issues that need to be addressed in the near future. For example, although a large number of samples are generated with an OOD-guided technique, it is worthy to investigate how to evaluate these samples to improve both the robustness of the watermark and the generalization of the model. In addition, since the size of the core subnetwork limits the embedding length of the watermark, it is necessary to design an adaptive subnetwork selection criterion that can enhance the embedding capability to a certain extent.

Acknowledgments

This work was supported in part by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-019), Singapore National Cybersecurity R&D Program No. NRF2018NCR-NCR005-0001, National Satellite of Excellence in Trustworthy Software System No.NRF2018NCR-NSOE003-0001, and NRF Investigatorship No. NRFI06-2020-0022-0001. We gratefully acknowledge the support of NVIDIA AI Tech Center (NVAITC) and AWS Cloud Credits for Research Award.

References

[1] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, 2017, pp. 269–277.

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

[3] J. Zhang, Z. Gu, J. Jang, H. Wu, M. P. Stoecklin, H. Huang, and I. M. Molloy, "Protecting intellectual property of deep neural networks with watermarking," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018, pp. 159–172.

[4] H. Chen, B. D. Rouhani, C. Fu, J. Zhao, and F. Koushanfar, "Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models," in *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, 2019, pp. 105–113.

[5] B. D. Rouhani, H. Chen, and F. Koushanfar, "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*, 2019, pp. 485–497.

[6] L. Fan, K. Ng, and C. S. Chan, "Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks," in *Annual Conference on Neural Information Processing Systems*, 2019, pp. 4716–4725.

[7] Z. He, T. Zhang, and R. B. Lee, "Sensitive-sample fingerprinting of deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4729–4737.

[8] Z. Li, C. Hu, Y. Zhang, and S. Guo, "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*, 2019, pp. 126–137.

[9] J. Zhang, D. Chen, J. Liao, H. Fang, W. Zhang, W. Zhou, H. Cui, and N. Yu, "Model watermarking for image processing networks," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020, pp. 12 805–12 812.

[10] P. Yang, Y. Lao, and P. Li, "Robust watermarking for deep neural networks via bi-level optimization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 841–14 850.

[11] S. Szyller, B. G. Atli, S. Marchal, and N. Asokan, "DAWN: dynamic adversarial watermarking of neural networks," in *ACM Multimedia Conference*, 2021, pp. 4417–4425.

[12] K. Chen, S. Guo, T. Zhang, S. Li, and Y. Liu, "Temporal watermarks for deep reinforcement learning models," in *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems*, 2021, pp. 314–322.

[13] N. Lin, X. Chen, H. Lu, and X. Li, "Chaotic weights: A novel approach to protect intellectual property of deep neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 7, pp. 1327–1339, 2021.

[14] X. Lou, S. Guo, T. Zhang, Y. Zhang, and Y. Liu, "When nas meets watermarking: ownership verification of dnn models via cache side channels," *arXiv preprint arXiv:2102.03523*, 2021.

[15] S. Guo, T. Zhang, H. Qiu, Y. Zeng, T. Xiang, and Y. Liu, "The hidden vulnerability of watermarking for deep neural networks," *arXiv preprint arXiv:2009.08697*, vol. 3, 2020.

[16] S. Lee, W. Song, S. Jana, M. Cha, and S. Son, "Evaluating the robustness of trigger set-based watermarks embedded in deep neural networks," *arXiv preprint arXiv:2106.10147*, 2021.

[17] S. Guo, T. Zhang, H. Qiu, Y. Zeng, T. Xiang, and Y. Liu, "Fine-tuning is not enough: A simple yet effective watermark removal attack for DNN models," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021, pp. 3635–3641.

[18] Y. Li, H. Wang, and M. Barni, "A survey of deep neural network watermarking techniques," *Neurocomputing*, vol. 461, pp. 171–193, 2021.

[19] J. Jia, L. Zhai, W. Ren, L. Wang, Y. Ren, and L. Zhang, "Transferable heterogeneous feature subspace learning for JPEG mismatched steganalysis," *Pattern Recognition*, vol. 100, 2020, Art. no. 107105.

[20] J. Jia, M. Luo, S. Ma, L. Wang, and Y. Liu, "Consensus-clustering-based automatic distribution matching for cross-domain image steganalysis," *IEEE Transactions on Knowledge and Data Engineering*, to be published, doi: 10.1109/TKDE.2022.3155924.

[21] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[22] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Annual Conference on Neural Information Processing Systems*, 2014, pp. 3320–3328.

[23] Y. Wei, Y. Zhang, J. Huang, and Q. Yang, "Transfer learning via learning to transfer," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 5072–5081.

[24] J. Jia, M. Luo, S. Ma, and L. Wang, "Partial knowledge transfer in visual recognition systems via joint loss-aware consistency learning," *IEEE Transactions on Industrial Informatics*, to be published, doi: 10.1109/TII.2022.3168029.

[25] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[26] Z. Yin, Y. Peng, and Y. Xiang, "Reversible data hiding in encrypted images based on pixel prediction and bit-plane compression," *IEEE Transactions on Dependable and Secure Computing*, to be published, doi: 10.1109/TDSC.2020.3019490.

[27] X. Guan, H. Feng, W. Zhang, H. Zhou, J. Zhang, and N. Yu, "Reversible watermarking in deep convolutional neural networks for integrity authentication," in *The 28th ACM International Conference on Multimedia*, 2020, pp. 2273–2280.

[28] S. Li, M. Xue, B. Z. H. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2088–2105, 2021.

[29] H. Li, E. Wenger, S. Shan, B. Y. Zhao, and H. Zheng, "Piracy resistant watermarks for deep neural networks," *arXiv preprint arXiv:1910.01226*, 2019.

[30] N. Ng, K. Cho, and M. Ghassemi, "SSMBA: self-supervised manifold based data augmentation for improving out-of-domain robustness," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 1268–1283.

[31] W. Hua, Y. Zhang, C. Guo, Z. Zhang, and G. E. Suh, "Bullettrain: Accelerating robust neural network training via boundary example mining," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[32] J. Djolonga, J. Yung, M. Tschannen, R. Romijnders, L. Beyer, A. Kolesnikov, J. Puigcerver, M. Minderer, A. D'Amour, D. Moldovan, S. Gelly, N. Houlsby, X. Zhai, and M. Lucic, "On robustness and transferability of convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 458–16 468.

[33] M. M. Kamani, S. Farhang, M. Mahdavi, and J. Z. Wang, "Targeted data-driven regularization for out-of-distribution generalization," in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020, pp. 882–891.

[34] G. Lin, J. Zhang, W. Luo, L. Pan, O. Y. de Vel, P. Montague, and Y. Xiang, "Software vulnerability discovery via learning multi-domain knowledge bases," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2469–2485, 2021.

[35] R. Hu, T. Wang, Y. Zhou, H. Snoussi, and A. Cherouat, "Ft-mdnet: A deep-frozen transfer learning framework for person search," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4721–4732, 2021.

[36] S. Liu, G. Lin, L. Qu, J. Zhang, O. Y. de Vel, P. Montague, and Y. Xiang, "Cd-vuld: Cross-domain vulnerability discovery based on deep domain adaptation," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 438–451, 2022.

[37] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[38] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.

[39] X. Zhang, P. Cui, R. Xu, L. Zhou, Y. He, and Z. Shen, "Deep stable learning for out-of-distribution generalization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5372–5382.

[40] J. Zhan, R. Sun, W. Jiang, Y. Jiang, X. Yin, and C. Zhuo, "Improving fault tolerance for reliable dnn using boundary-aware activation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, to be published, doi: 10.1109/TCAD.2021.3129114.

[41] A. A. Alemi, B. Poole, I. Fischer, J. V. Dillon, R. A. Saurous, and K. Murphy, "Fixing a broken ELBO," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 159–168.

[42] D. Zhang, K. Ahuja, Y. Xu, Y. Wang, and A. C. Courville, "Can subnetwork structure be the key to out-of-distribution generalization?" in *Proceedings of the 38th International Conference on Machine Learning*, vol. 139, 2021, pp. 12 356–12 367.

[43] M. MacKay, P. Vicol, J. Lorraine, D. Duvenaud, and R. B. Grosse, "Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions," in *International Conference on Learning Representations*, 2019.

[44] X. Xing, L. Sha, P. Hong, Z. Shang, and J. S. Liu, "Probabilistic connection importance inference and lossless compression of deep neural networks," in *International Conference on Learning Representations*, 2020.

[45] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[46] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, pp. 59:1–59:35, 2016.

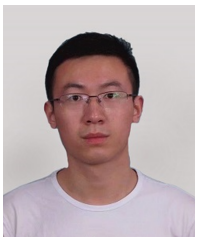
[47] S. Zhao, G. Wang, S. Zhang, Y. Gu, Y. Li, Z. Song, P. Xu, R. Hu, H. Chai, and K. Keutzer, "Multi-source distilling domain adaptation," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020, pp. 12 975–12 983.

[48] M. Awais, F. Zhou, H. Xu, L. Hong, P. Luo, S.-H. Bae, and Z. Li, "Adversarial robustness for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8568–8577.

[49] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," in *Proceedings of the 4th International Conference on Learning Representations*, 2016.

[50] J. Jia, Z. Xiang, L. Wang, and Y. Xu, "An adaptive JPEG double compression steganographic scheme based on irregular DCT coefficients distribution," *IEEE Access*, vol. 7, pp. 119 506–119 518, 2019.

[51] X. Zhou, W. Peng, B. Yang, J. Wen, Y. Xue, and P. Zhong, "Linguistic steganography based on adaptive probability distribution," *IEEE Transactions on Dependable and Secure Computing*, to be published, doi: 10.1109/TDSC.2021.3079957.
[52] S. Wang, Z. Ding, and Y. Fu, "Cross-generation kinship verification with sparse discriminative metric," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2783–2790, 2019.
[53] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems*, 2013, pp. 2787–2795.
[54] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, vol. 9, 2010, pp. 249–256.



Ju Jia received his Ph.D. degree in cyberspace security from Wuhan University, Wuhan, China, in 2021. He is currently a research fellow with the School of Computer Science and Engineering at Nanyang Technological University, Singapore. His research interests include: transfer learning, intellectual property protection of DNN, information hidding, and multimedia data security.

Yueming Wu received the B.E. degree in Computer Science and Technology at Southwest Jiaotong University, Chengdu, China, in 2016 and the Ph.D. degree in School of Cyber Science and Engineering at Huazhong University of Science and Technology, Wuhan, China, in 2021. He is currently a research fellow in the School of Computer Science and Engineering at Nanyang Technological University. His primary research interests lie in malware analysis, vulnerability analysis, and DNN watermarking.

Anran Li is a research fellow in the School of Computer Science and Engineering, Nanyang Technological University, Singapore. She received her Ph.D degree from University of Science and Technology of China, and Bachelor degree from Anhui University of Science and Technology. Her research interests mainly focus on data quality assessment, federated learning and mobile computing, and artificial intelligence in cybersecurity.

Siqi Ma received the B.S. degree in computer science from Xidian University, Xi'an, China in 2013 and Ph.D. degree in information system from Singapore Management University in 2018, respectively. She was a research fellow of distinguished system security group from CSIRO and then was a lecturer at University of Queensland. She is currently a senior lecturer of the University of New South Wales, Canberra Campus, Australia. Her research interests include data security, IoT security and software security.

Yang Liu (Senior Member, IEEE) received the B.Comp. degree (Hons.) from the National University of Singapore (NUS) in 2005 and the Ph.D. degree from NUS and MIT, in 2010. He started his postdoctoral work in NUS and MIT. In 2012, he joined Nanyang Technological University (NTU). He is currently a Full Professor and the Director of the Cybersecurity Laboratory, NTU. He specializes in software verification, security, and software engineering. His research has bridged the gap between the theory and practical usage of formal methods and program analysis to evaluate the design and implementation of software for high assurance and security. By now, he has more than 400 publications in top tier conferences and journals. He has received a number of prestigious awards including MSRA Fellowship, TRF Fellowship, Nanyang Assistant Professor, Tan Chin Tuan Fellowship, Nanyang Research Award 2019, ACM Distinguished Speaker, NRF Investigatorship, and 15 best paper awards and one most influence system award in top software engineering conferences like ASE, FSE and ICSE.