Model Barrier: A Compact Un-Transferable Isolation Domain for Model Intellectual Property Protection

Lianyu Wang[1]*, Meng Wang[2]*, Daoqiang Zhang[1]†, Huazhu Fu[2]†

[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China.

[2]Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore 138632.

Abstract

As scientific and technological advancements result from human intellectual labor and computational costs, protecting model intellectual property (IP) has become increasingly important to encourage model creators and owners. Model IP protection involves preventing the use of well-trained models on unauthorized domains. To address this issue, we propose a novel approach called Compact Un-Transferable Isolation Domain (CUTI-domain), which acts as a barrier to block illegal transfers from authorized to unauthorized domains. Specifically, CUTI-domain blocks cross-domain transfers by highlighting the private style features of the authorized domain, leading to recognition failure on unauthorized domains with irrelevant private style features. Moreover, we provide two solutions for using CUTI-domain depending on whether the unauthorized domain is known or not: target-specified CUTI-domain and target-free CUTI-domain. Our comprehensive experimental results on four digit datasets, CIFAR10 & STL10, and VisDA-2017 dataset demonstrate that CUTI-domain can be easily implemented as a plug-and-play module with different backbones, providing an efficient solution for model IP protection.

1. Introduction

The recent success of deep learning models heavily relies on massive amounts of high-quality data, specialized training resources, and elaborate manual fine-tuning [4, 10, 26, 35]. Obtaining a well-trained deep model is both time-consuming and labor-intensive [22]. Therefore, it should be protected as a kind of scientific and technological achievement intellectual property (IP) [5, 50], thereby stimulating innovation enthusiasm in the community and further promoting the development of deep learning. As shown in Fig. 1, in supervised learning (SL), the model owner uses the overall features of the authorized domain (pink square) for training, obtains a high-performance model, and grants the right to use it to a specific user. Authorized users can use the model on the authorized domain to obtain correct predictions. However, since the model is trained with overall features, its potential feature generalization region is large and may cover some unauthorized domains. Therefore, there is a natural pathway between the authorized domain and the unauthorized domain, and the released high-performance model obtained by SL can be illegally transferred to the unauthorized domain (blue squares) through methods such as domain adaptation [30, 51], and domain generalization [44, 52], to obtain correct prediction results. This presents a challenge in protecting well-trained models. One of the most concerning threats raised is "Will releasing

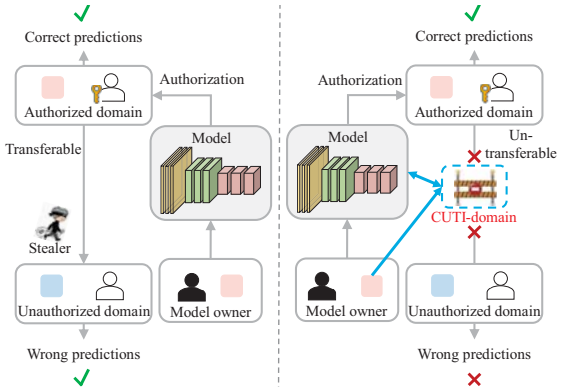

Figure 1. Model IP protection with our proposed CUTI-domain. **Left**: In standard supervised learning (SL), the model owner trains a high-performance model on the authorized domain (pink square) and then authorizes a specific user. Authorized users have the right to use the model on the authorized domain to get the correct prediction. However, a stealer can easily access the model on the unauthorized domain (blue square), which violates the legitimate rights and interests of the model owner. **Right**: Our method constructs a CUTI-domain between the authorized and unauthorized domains, which could block the illegal transferring and lead to a wrong prediction for unauthorized domains.

*L. Wang and M. Wang contributed equally to this work.

†Corresponding author: D. Zhang (dqzhang@nuaa.edu.cn) and H. Fu (hzfu@ieee.org).

the model make it easy for the main competitor to copy this new feature and hurt owner differentiation in the market?" Thus, the model IP protection has been proposed to defend against model stealing or unauthorized use.

A comprehensive intellectual property (IP) protection strategy for deep learning models should consider both ownership verification and applicability authorization [45, 47]. Ownership verification involves verifying who has permission to use the deep model by embedding watermarks [37, 48], model fingerprint [31], and predefined triggers [14]. The model owner can grant usage permission to a specific user, and any other users will be infringing on the owner's IP rights. However, an authorized user can easily transfer the model to an unauthorized user, so the model owner must add special marks during training to identify and verify ownership. Moreover, these methods are vulnerable to fine-tuning, classifier retraining, elastic weight consolidation algorithms, and watermark overwriting, which can weaken the model's protection. On the other hand, applicability authorization involves verifying the model's usage scenarios. Users with permission can apply the deep model for the tasks specified by the model owner, and it is an infringement to use it for unauthorized tasks [45]. However, users can easily transfer high-performance models to other similar tasks to save costs, which is a common and hidden infringement. Therefore, if the performance of the model can be limited to the tasks specified by the owner and reduced on other similar tasks, unauthorized users will lose confidence in stealing and re-authoring the model. To achieve this, a non-transferable learning (NTL) method is proposed [45], which uses an estimator with a characteristic kernel from Reproducing Kernel Hilbert Spaces to approximate and increase the maximum mean difference between two distributions on finite samples. However, the authors only considered using limited samples to increase the mean distribution difference of features between domains and ignored outliers. The convergence region of NTL is not tight enough. Moreover, the calculation of the maximum mean difference is class-independent, which reduces the model's feature recognition ability in the authorized domain to a certain extent.

To address the challenges outlined above, we **first** propose a novel approach called the Compact Un-Transferable Isolation (CUTI) domain to prevent illegal transferring of deep models from authorized to unauthorized domains. Our approach considers the overall feature of each domain, consisting of two components: shared features and private features. Shared features refer to semantic features, while private features include stylistic cues such as perspective, texture, saturation, brightness, background environment, and so on. We emphasize the private features of the authorized domain and construct a CUTI-domain as a model barrier with similar private style features. This approach prevents illegal transfers to unauthorized domains with new private style features, thereby leading to wrong predictions. **Furthermore**, we also provide two CUTI-domain solutions for different scenarios. When the unauthorized domain is known, we propose the target-specified CUTI-domain, where the model is trained with a combination of authorized, CUTI, and unauthorized domains. When the unauthorized domain is unknown, we use the target-free CUTI-domain, which employs a generator to synthesize unauthorized samples that replace the unauthorized domain in model training. **At last**, our comprehensive experimental results on four digit datasets, CIFAR10 & STL10, and VisDA-2017 demonstrate that our proposed CUTI-domain effectively reduces the recognition ability on unauthorized domains while maintaining strong recognition on authorized domains. Moreover, as a plug-and-play module, our CUTI-domain can be easily implemented within different backbones and provide efficient solutions.*

2. Related Work

2.1. Model IP Protection

There are currently two main categories of methods for IP protection, including ownership verification and applicability authorization. For ownership verification, the most classic method is watermarking embedding [38]. Kuribayashi *et al*. [25] proposed a quantifiable watermark embedding method, which reduces the variation caused by embedding watermarks. Adi *et al*. [1] proposed a tracking mechanism in a black-box way. However, such watermark embedding approaches have been proved to be susceptible to some watermark removal and watermark overwriting methods. In our experiments, a simple watermark is embedded into the model for ownership verification by triggering mis-classification. Comprehensive experimental results demonstrate that the proposed CUTI-domain is resistant to the common watermark removal methods.

Applicability authorization is derived from usage authorization. Usage authorization usually uses a preset private key to encrypt the whole/part of the network. Only authorized users can obtain the private key and then use the model. There are many advanced methods for usage authorization. For example, Alam *et al*. [2]. proposed an explicit locking mechanism for a lightweight deep neural network, utilizing S-Boxes with good cryptographic properties to lock each training parameter of a DNN model. Without knowledge of the legitimate private key, unauthorized access can severely degrade the accuracy of the model. Song *et al*. [39] analyzed and calculated the critical weight of the deep neural network model, and significantly reduced the time cost by encrypting the critical weight to lock the deep neural network model against unauthorized use.

*https://github.com/LyWang12/CUTI-Domain.

Wang *et al*. [45] proposed data-based applicability authorization named NTL, which preserves model performance on authorized data while degrading model performance in other data domains. Compared to the above, we construct a new class-dependent CUTI-domain with infinite samples whose features are more similar with the source domain. By decreasing the performance of the model on the CUTI-domain and the target domain, the generalization bound of the model can be compacter, thereby constraining the model performance within the authorized source domain.

2.2. Domain Transferring

In practice, domain gaps may arise due to different data collection scenarios in different domains. Domain adaptation (DA) and domain generalization (DG) are common solutions used to alleviate domain gaps [46, 53]. DA refers to transferring a model from a labeled source domain to an unlabeled but relevant target domain where the target domain's data is accessible during the training process [16]. DG differs from DA in that the target domain is inaccessible during model training [12, 49]. DANN [17] s a classic DA method that introduced a gradient inversion layer and a domain discriminator to confuse the feature distributions of the two domains. Subsequently, CDAN [29] further introduced categorical information entropy into the domain discriminator to alleviate the class mismatch problem. For DG, Tobin *et al*. [42] used domain randomization to generate more training data from simulated environments for generalization in real environments. Prakash *et al*. [34] further considered the structure of the scene when randomly placing objects for data generation, enabling the neural network to learn how to utilize context when detecting objects. Recently, some methods have been proposed and successfully applied to cross-domain applications by seeking an intermediate state between the source domain and the target domain, emphasizing the similarity between domains to improve the model's transferability [9, 19, 20, 27, 40]. In contrast, this paper aims to seek an intermediate state to highlight the difference between the authorized source domain and the unauthorized target domain, constraining the feature transferability of the model and protecting the scientific and technological achievements IP of the model owner.

3. Methodology

In this section, we first present our proposed CUTI-domain with the aim of developing a solution that limits the performance of the model to authorized source domains and reduces feature recognition capabilities on the unauthorized target domain. Then, depending on whether the unauthorized target domain is known or not, we provide two solutions, target-specified CUTI-domain and target-free CUTI-domain, for protecting the model IP.

3.1. Compact Un-Transferable Isolation Domain

In the deep neural network model, the overall features extracted by the feature extractor include two abstract components, *i.e.*, semantic features, and style features. Semantic features reflect the structural information of samples and play a leading role in sample recognition; while style feature refers to a series of weakly related clues, such as perspective, texture, saturation, brightness, and background environment. For different domains with the same task, semantic features are shared, while style features are private. Most of the previous DA and DG works have been devoted to improving feature transferability between domains, *i.e.*, strengthening the focus of the model on shared features while suppressing seemingly disturbing private style features. However, to protect the intellectual property of the model, this paper aims to limit the feature recognition ability of the model by highlighting private style features of the source domain through style transfer, thus leading to the failure of recognition on target domains that contain irrelative private style.

Style transfer techniques suggest that styles are homogeneous and composed of repeated structural motifs. Two images can be considered similar in style if the features extracted by a trained classifier share the same statistics [23, 43]. First- and second-order statistics are often used as style features due to their computational efficiency. These statistics refer to the mean and variance of the extracted features. On the other hand, semantic features only contain pure semantic information and exclude any style information. Similar to [13], the semantic feature f_s of the extracted feature f can be obtained by removing the style features, as follows:

$$f_s = \frac{f - \mu(f)}{\sigma(f)}, \tag{1}$$

where $\mu(f)$ and $\sigma(f)$ denote the mean and variance of f. Furthermore, style can be re-assign by $f_s \cdot \gamma + \beta$, where γ and β are learned parameters. Afterward, Huang *et al*. [23] further explored adapting f to arbitrarily given style by using style features of another extracted feature instead of learned parameters.

Drawing inspiration from these ideas, we propose a novel CUTI-domain design that incorporates style features from the source domain. Specifically, we randomly extract style features from the source domain and fuse them with the overall features of the CUTI-domain, making the private style features of the CUTI-domain more similar to those of the source domain. By doing so, we reduce the model's ability to recognize features on both the CUTI-domain and the target domain, thereby implicitly blocking the pathway between the source and target domains and limiting the model's performance to the source domains.

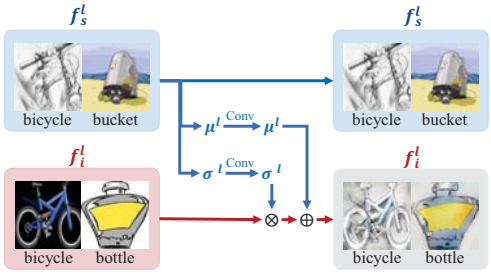

Figure 2. Illustration of our proposed CUTI-domain generator. The output feature of the source domain f_s^l and the CUTI-domain f_i^l in the i-th feature extractor block are sent to the CUTI-domain generator, and then the mean μ^l and variance σ^l of f_s^l is fused with f_i^l, so that the private style features of the updated f_i^l are closer to the feature of source domain f_s^l while the original semantics features are maintained.

Our CUTI-domain is generated by the CUTI-domain generator, which is a lightweight, and plug-and-play module, as shown in Fig. 2. f_s^l and f_i^l represent the deep features of the i-th feature extractor block in the source domain and the CUTI-domain, respectively. First, f_s^l and f_i^l are sent into CUTI-domain generator in parallel, and then the mean μ^l and variance σ^l of f_s^l are calculated according to the channel as private style features, followed by a 1×1 convolution layer $Conv$. Next, the μ^l and σ^l are multiplied and added channel-wisely by f_i^l as:

$$f_i^l \leftarrow (f_i^l \bigotimes Conv(\sigma^l)) \bigoplus Conv(\mu^l). \tag{2}$$

As can be seen in Fig. 2, the private style features of updated f_i^l are closer to those of f_s^l, while retaining its original semantic features. Through continuous computation, CUTI-domain generators can construct a labeled CUTI-domain containing a similar private style to the source domain.

3.2. Model IP Protection with CUTI-domain

3.2.1 Target-Specified CUTI-Domain

We first introduce how to utilize our CUTI-Domain with a given unauthorized target domain. Fig. 3 illustrates the whole framework trained with our proposed CUTI-domain, which consists of L feature extractor blocks, L CUTI-domain generators and a classifier. x_s, x_i and x_t denote the data of the source domian, CUTI-domain and target domain respectively. During training, when $epoch = 2e$, *i.e.*, epoch is equal to an even number, x_s and x_i are fed into the feature extractor blocks in parallel, followed by a CUTI-domain generator. The classifier at the end of the network is used to predict the category of the sample, and the prediction results are denoted by p_s and p_i, respectively. When $epoch = 2e + 1$, *i.e.*, epoch is equal to an odd number, x_s and x_t are input into the network parallelly without CUTI-domain generators, p_s and p_t denote their predicted results.

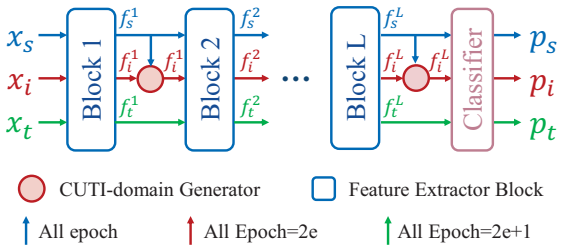

Figure 3. Illustration of the framework trained with our proposed CUTI-domain. The whole framework consists of L feature extractor blocks, L CUTI-domain generators, and a classifier. When the epoch is equal to an even/odd number, samples of the source domain x_s and CUTI-domain x_i / target domain x_t are fed into the feature extractor block in parallel. p_s, p_i and p_t denote their prediction results.

Based on the designed framework, an alternative loss function $\mathcal{L}$ according to the epoch is utilized:

$$\mathcal{L} = \begin{cases} KL(p_s||y_s) - KL(p_i||y_i), & \text{if } epoch = 2e, \\ KL(p_s||y_s) - KL(p_t||y_t), & \text{if } epoch = 2e+1, \end{cases} \tag{3}$$

where $KL(\cdot)$ stands for Kullback–Leibler divergence. By reducing the $KL(\cdot)$ between prediction results and the labels on the source domain, and expanding the $KL(\cdot)$ prediction results and the labels on the CUTI-domain/target domain, the feature recognition ability of the model in the source domain can be gradually improved, and the ability in the CUTI-domain/target domain can be gradually reduced, thus effectively constrain the performance of the model within authorized source domains for IP protection. Finally, we summarize the strategy of our proposed target-specified CUTI-domain in Algorithm 1.

During training, we initialize the CUTI-domain with the target domain training set, and feed data of source domain training set x_s, CUTI-domain x_i, and target domain training set x_t into the network in parallel to train the model as in Algorithm 1. At test time, model performance is evaluated on the source domain test set and the target domain test set. In an ideal situation, the model can maintain high sample recognition ability on the authorized source domain, but achieve poor performance on the unauthorized target domain.

3.2.2 Target-Free CUTI-Domain

Sometimes, the unauthorized target domain is unknown. In this case, we cannot directly feed the target domain and CUTI-domain into the network for model training. To solve this, a synthesized target domain could be utilized. For example, Wang *et al*. [45] designed a GAN-based method by freezing parameters to generate synthesized samples in different directions in place of the target domain train set. Al-

Algorithm 1 Target-Specified CUTI-Domain.

Require: The source domain x_s, CUTI-Domain x_i, the target domain x_t, number of feature blocks L, the model parameters θ.

Initialize CUTI-domain with the target domain.
For $epoch = 1$ **to** Max_{epochs} **do**
  If $epoch = 2i + 1$ **do**
    Calculate the output of x_s, x_i in the l-th feature block: f_s^l, f_i^l.
    For $l = 1$ **to** L **do**
      Update f_i^l according to Eq. (2).
    End For
  If $epoch = 2i$ **do**
    Calculate the output of x_s, x_t in the l-th feature block: f_s^l, f_t^l.
  Update model parameters θ by Eq. (3).
End For
Return model parameters θ.

though their method is able to generate high-quality synthesized samples, the direction of the generator is specified and there are certain omissions. Huang *et al*. [23] designed an adaptive instance normalization (AdaIN) method based on GAN, which can generate synthesized images with a specific style for a content image.

To take full advantage, we add Gaussian noise to AdaIN to obtain synthesized samples with random styles. Finally, we mix the synthesized samples generated by the above two methods (*e.g.*, GAN and AdaIN) to replace the target domain training set and initialize the CUTI-domain. The framework is consistent with the target-specified CUTI-domain and the training process is detailed in the Appendix. During testing, the model is evaluated on source domain test set and other unknown domain with the same task. Our goal is not to design GANs to generate high-quality synthesized images, but to evaluate the ability of CUTI-domain to block illegal feature transfer in the context of synthesized images. We still focus on reducing the feature recognition ability of the model on unauthorized target domains and maintaining it on the authorized source domains.

4. Experiment

4.1. Implementation Details

We evaluate our method on seven popular DA/DG benchmarks, *i.e.*, MNIST (MT) [11], USPS (US) [24], SVHN (SN) [32] and MNIST-M (MM) [18] are commonly used digit datasets, containing ten digits from 0 to 9 extracted from vary scenes; CIFAR10 [3] and STL10 [8] are all ten-class classification datasets. We follow the procedure of French et al. [15] to process the dataset such that the correspondence between them holds. VisDA-2017 [33] is a Synthetic-to-Real dataset containing training (T) and validation (V) sets from 12 categories. Following the general setup, we adopt accuracy (%) as the performance metric of each task.

For tasks with different complexity, different backbones are adopted to compare with the method proposed by Wang (NTL) *et al*. [45]. We used VGG-11 [36] for digit datasets, VGG-13 [36] for CIFAR10 and STL10, and VGG-19 [36] for VisDA. Since VGG contains five feature extractor blocks, L is set to 5 in this paper, and CUTI-domain generator is deployed after each max-pooling layer in the block. Pre-trained models are used for all backbones for a fair comparison. The implementation of our comprehensive experiments is based on the public platform Pytorch and an NVIDIA GeForce RTX 3090 GPU with 24GB of memory. The batch size for each domain is set to 32.

4.2. Result of Target-Specified CUTI-Domain

We selected one as the source domain and one as the target domain from the digital datasets of four different domains, and constructed 16 transfer tasks, as shown in Table 1. The left of $\Rightarrow$ represents the accuracy of the model trained on the source domain dataset using supervised learning (SL), and the right of $\Rightarrow$ is the accuracy of CUTI-domain. CUTI source/target drops represent the drop (relative drop) of the proposed CUTI-domain in the source and target domains, respectively. As can be seen, the average drop of CUTI-domain on the target and source domains is 55.94 (84.94%) and 0.13 (0.13%), respectively. The last two columns represent the average performance degradation of NTL with values of 46.48 (76.34%) and 1.30 (1.39%), respectively. Compared with NTL, the decline of CUTI-domain in the target domain is higher, and the negative impact on the source domain is smaller. It can be inferred that CUTI-domain can better reduce the sample recognition ability of the model for the target domain, and the decreases on the source domain are slight.

Fig. 4 shows the results on CIFAR10 $\rightarrow$ STL10, STL10 $\rightarrow$ CIFAR10 and T $\rightarrow$ V. The bars with different colors in each subgraph represent the accuracy of the corresponding method in the source domain, the accuracy in the target domain, and the degradation (relative degradation) of the model performance, respectively. Where the results of NTL are reproduced by its source code to obtain comparable experimental data. SL has the largest generalization region, resulting in the highest classification accuracy on the target domain. By blocking the pathway with model locker, we observe successful target domain reduction in accuracy for NTL and CUTI-domain, with higher degradation than SL. Meanwhile, regardless of the task, the degradation of CUTI-domain is higher than that of NTL, indicating that CUTI-domain can better compress the generalization region of the model.

Table 1. The accuracy (%) of target-specified CUTI-domain on digit datasets. The left of '⇒' represents the accuracy of the model trained on the source domain dataset with SL, and the right of '⇒' is the accuracy of CUTI-domain. CUTI source/target drop represent the average degradation (relative degradation) of the proposed CUTI-domain relative to SL on the source/target domains. NTL source/target drop is calculated from the original paper. The bold numbers indicate the best performance.

Source/Target	MT	US	SN	MM	CUTI Source Drop↓	CUTI Target Drop↑	NTL Source Drop↓	NTL Target Drop↑
MT	99.2 ⇒ 99.1	98.0 ⇒ 6.7	38.2 ⇒ 5.6	67.8 ⇒ 8.7	**0.10 (0.10%)**	**61.00 (88.56%)**	1.00 (1.01%)	46.57 (75.60%)
US	92.6 ⇒ 10.0	99.7 ⇒ 99.6	25.5 ⇒ 6.8	41.2 ⇒ 8.4	**0.10 (0.10%)**	**44.70 (80.72%)**	1.00(1.00%)	38.67 (75.55%)
SN	66.7 ⇒ 9.2	70.5 ⇒ 6.7	91.2 ⇒ 90.9	34.6 ⇒ 10.9	**0.30 (0.33%)**	**48.33 (81.73%)**	1.10(1.23%)	40.60 (77.25%)
MM	98.4 ⇒ 9.5	88.4 ⇒ 6.8	46.3 ⇒ 7.6	95.4 ⇒ 95.4	**0.00 (0.00%)**	**69.73 (88.75%)**	2.10(2.30%)	60.10 (76.95%)
Mean	/	/	/	/	**0.13 (0.13%)**	**55.94 (84.94%)**	1.30 (1.39%)	46.48 (76.34%)

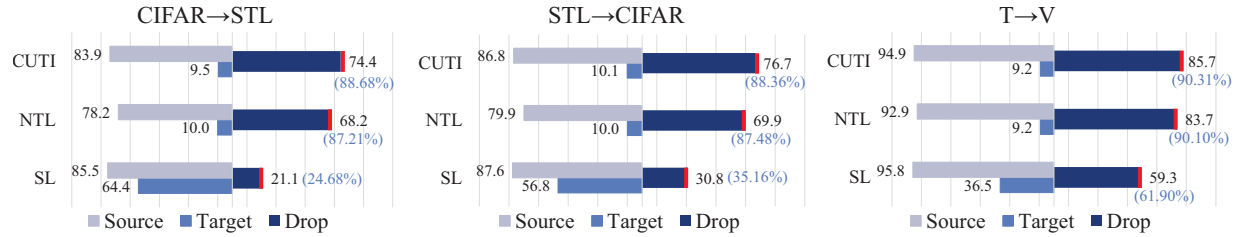

Figure 4. The accuracy (%) of SL, target-specified NTL and target-specified CUTI-domain on CIFAR10, STL10 and VisDA-2017. The left of '→' represents the source domain and the right of '→' is the target domain. The bars with different colors in each subgraph represent the accuracy of the corresponding method in the source domain, the target domain, and the degradation (relative degradation) of the model performance, respectively. The data of NTL is obtained by reproducing its source code.

Table 2. The accuracy (%) of ownership verification by SL and CUTI-domain. FTAL [1], RTAL [1], EWC [6], AU [6] and Overwriting are state-of-the-art watermark removal methods. Avg drop represents the average degradation of the test dataset with watermark patch relative to test dataset without a watermark patch. The bold numbers indicate the best performance.

Source without Patch	Training Methods		Watermark Removal Approaches on CUTI					Avg Drop↑	
	SL	CUTI	FTAL [1]	RTAL [1]	EWC [6]	AU [6]	Overwriting	CUTI	NTL
	[Test w/o Watermark (%)]		[Test w/o Watermark (%)]						
MT	99.0 / 99.3	11.3 / 99.1	9.0 / 100.0	9.4 / 100.0	9.7 / 100.0	9.0 / 100.0	9.4 / 96.2	**89.9**	88.4
US	99.8 / 99.8	7.7 / 99.8	8.0 / 100.0	8.7 / 100.0	9.4 / 100.0	9.4 / 100.0	8.7 / 98.6	**90.9**	85.7
SN	91.3 / 92.3	9.9 / 92.1	9.4 / 98.3	13.9 / 97.6	10.8 / 100.0	8.7 / 100.0	10.4 / 95.8	**87.7**	79.0
MM	96.6 / 96.0	16.8 / 96.0	14.3 / 95.4	24.0 / 98.6	14.6 / 100.0	14.6 / 100.0	14.9 / 95.8	**81.5**	77.3
CIFAR	83.3 / 75.1	10.7 / 86.8	14.9 / 97.9	14.9 / 93.8	14.9 / 100.0	9.4 / 97.2	16.7 / 90.3	**81.7**	74.6
STL	87.9 / 93.2	22.0 / 88.2	20.0 / 96.9	26.4 / 93.8	13.9 / 100.0	22.9 / 94.1	21.2 / 89.6	**74.0**	**74.0**
VisDA	93.6 / 92.2	13.1 / 94.1	15.0 / 95.5	20.5 / 95.1	15.3 / 100.0	21.9 / 95.1	19.4 / 96.2	**78.0**	76.8
Mean	/	/	/	/	/	/	/	**83.4**	79.4

4.3. Result of Ownership Verification

In this section, we conduct ownership verification of model by triggering classification errors. Specifically, we add a regular backdoor-based model watermark patch on the authorized source domain dataset followed by NTL [45], and treat the processed source domain as the new unauthorized target domain. The model classification accuracy of SL and CUTI-domain on the authorized source domain without watermark patch and the unauthorized target domain with watermark patch are shown in Table 2. As shown in the second and third columns in the table, for SL, there is little difference in the accuracy before and after embedding the watermark patch, so the model is not sensitive to the watermark patch. While for CUTI-domain, after watermark patch embedding, the accuracy on unauthorized target domain is greatly reduced, and this difference in performance can be used to verify the ownership of the model. In addition, we also test the robustness of CUTI-domain using FTAL [1], RTAL [1], EWC [6], AU [6] and watermark overwriting, which are state-of-the-art model watermark re-

Table 3. The accuracy (%) of target-free CUTI-domain on digit datasets. The left of '⇒' represents the accuracy of the model trained on the source domain dataset with SL, and the right of '⇒' is the accuracy of CUTI-domain. CUTI source/target drop represent the average degradation (relative degradation) of the proposed CUTI-domain relative to SL on the source/target domains. The data of NTL is obtained by reproducing its source code. The bold numbers indicate the best performance.

Source/Target	MT	US	SN	MM	CUTI Source Drop↓	CUTI Target Drop↑	NTL Source Drop↓	NTL Target Drop↑
MT	99.2 ⇒ 98.8	98.0 ⇒ 6.7	38.2 ⇒ 6.7	67.8 ⇒ 13.1	**0.40 (0.40%)**	**59.17 (85.43%)**	0.70 (0.71%)	57.30 (84.06%)
US	92.6 ⇒ 9.1	99.7 ⇒ 99.7	99.1 ⇒ 25.5	6.8 ⇒ 8.5	**0.60 (0.60%)**	**44.97 (80.96%)**	**0.60 (0.60%)**	42.90 (74.71%)
SN	66.7 ⇒ 11.9	70.5 ⇒ 14.3	91.2 ⇒ 88.7	34.6 ⇒ 13.6	**2.50 (2.74%)**	**44.00 (74.19%)**	3.20 (3.51%)	41.53 (64.09%)
MM	98.4 ⇒ 19.6	88.4 ⇒ 6.8	46.3 ⇒ 9.5	95.4 ⇒ 95.1	**0.30 (0.31%)**	**65.73 (83.96%)**	2.00 (2.10%)	63.03 (77.62%)
Mean	/	/	/	/	**0.95 (1.02%)**	**53.47 (81.13%)**	1.63 (1.73%)	51.19 (75.12%)

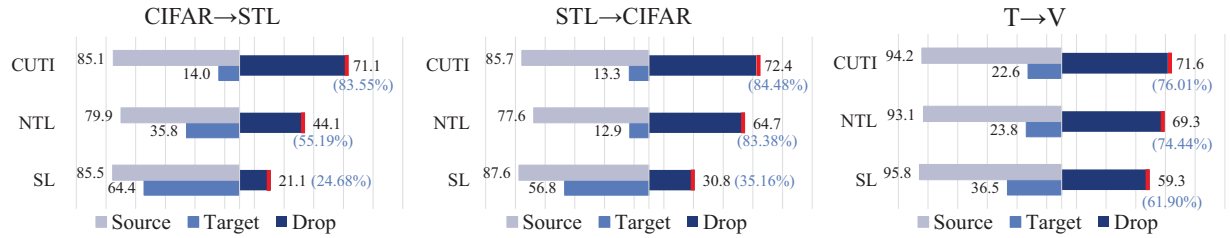

Figure 5. The accuracy (%) of SL, target-free NTL and target-free CUTI-domain on CIFAR10, STL10, and VisDA-2017. The left of '→' represents the source domain and the right of '→' is the target domain. The bars with different colors in each subgraph represent the accuracy of the corresponding method in the source domain, the target domain, and the degradation (relative degradation) of the model performance, respectively. The data of NTL is obtained by reproducing its source code.

moval methods. For a fair comparison, the settings of these watermark removal methods are all consistent with NTL. The last two columns of Table 2 are the drop in accuracy for watermarked versus un-watermarked data. It can be observed that both CUTI-domain and NTL can effectively resist the attack of watermark removal method, and the performance of CUTI-domain is about 4% higher.

4.4. Result of Target-free CUTI-Domain

As detailed in Section 3.3.2, when the target domain is unknown, we use synthesized samples to replace the target domain training set, and use other unknown domain datasets as the target domain test set, as shown in Table 3 and Fig. 5. For a fair comparison, the data of NTL is obtained by reproducing its source code. It can be observed that the average drop of CUTI-domain on the target domain is higher than that of NTL, and the drop on the source domain is lower, indicating better IP protection ability of our proposed CUTI-domain on unauthorized domains.

Fig. 5 shows the results for CIFAR10 → STL10, STL10 → CIFAR10 and T → V. Where the results of NTL are reproduced by its source code to obtain comparable experimental data. Consistent with the previous text, the degradation of CUTI-domain and NTL are higher than SL, and CUTI-domain is the highest, which implies that CUTI-domain can better compress the generalization region of the model. Meanwhile, considering the complexity of the VisDA-2017 dataset, it is more difficult to extract representative features to build CUTI-domain, so the drop of T → V of CUTI-domain is only slightly higher than NTL.

4.5. Result of Applicability Authorization

In this section, we validate the applicability of model by restricting its generalization ability to the authorized source domain. Specifically, similar to section 4.3, we add an authorized watermark patch on the source domain as a new authorized source domain training set. Then the original source domain, synthesized samples, and synthesized samples with authorized watermark patches are mixed as the unauthorized target domain training set. During testing, other unknown domains are used as the test set. The experimental results of CUTI-domain are shown in Table 4, where the results of NTL are reproduced by its source code. It can be observed that the model performs better on the source domain with authorized watermark patch, but performs poorly on other unknown domains with or without watermark patch. This is consistent with our expectation that the generalization ability of the model is restricted to the source domain with the authorized watermark patch. Meanwhile, the average drop rate of our proposed CUTI-domain is 83.75 (84.27%), which is higher than NTL with 81.03 (81.81%). This is mainly because NTL directly utilizes the limited features of the source and target domains for distance maximization, while we construct a CUTI-

Table 4. The accuracy (%) of applicability authorization on digit datasets, CIFAR10, STL10 and VisDA-2017. CUTI authorized/other domain represent the average accuracy of CUTI-domain on the authorized/other domain, CUTI drop denote the degradation (relative degradation). The results of NTL are reproduced by its source code. The bold numbers indicate the best performance.

Source with Path	Test with Path(%)				Test without Path(%)				CUTI Authorized Domain↑	CUTI Other Domain↓	CUTI Drop ↑	NTL Authorized Domain↑	NTL Other Domain↓	NTL Drop ↑
	MT	US	SN	MM	MT	US	SN	MM						
MT	100.0	14.3	17.6	12.9	10.3	8.6	18.3	14.1	**100.0**	**13.7**	**86.27(86.27%)**	99.8	14.5	85.31(85.49%)
US	9.6	99.2	14.9	10.7	10.2	6.7	8.6	10.6	**99.2**	**10.2**	**89.01(89.73%)**	98.5	13.3	85.20(86.50%)
SN	10.8	13.5	99.1	23.0	9.6	9.2	17.7	8.9	99.1	**13.2**	**85.86(86.64%)**	**99.3**	15.8	83.51(84.10%)
MM	8.9	9.1	15.0	99.5	10.0	9.1	11.9	25.9	**99.5**	**12.8**	**86.66(87.09%)**	**99.5**	14.0	85.49(85.92%)
	CIFAR		STL		CIFAR		STL		/					
CIFAR	97.9		42.5		13.1		11.6		**97.9**	**22.4**	**75.50(77.12%)**	97.5	24.6	72.90(74.77%)
STL	29.0		99.9		13.3		15.2		**99.9**	**19.2**	**80.73(80.81%)**	98.6	20.5	78.10(79.21%)
	T		V		T		V		/					
T	100.0		22.9		20.3		10.1		**100.0**	**17.8**	**82.23(82.23%)**	**100.0**	23.3	76.70(76.70%)
Mean	/								**99.4**	**15.5**	**83.75(84.27%)**	99.0	18.0	81.03(81.81%)

domain with infinite samples similar to the source domain, whose generalization boundary is more compact, and thus the model IP protection ability is stronger.

4.6. Ablation Study

Backbone: In this section, we verify the ability of IP protection of CUTI-domain combined with other backbones on the VisDA-2017 dataset. As shown in the left of Fig. 6, compared with SL, CUTI-domain can further reduce the recognition ability on the target domain when implemented with VGG-19 [36], ResNet-34 [21], Inceptionv3 [41], Xception [7] and SWIM [28], Meanwhile, the accuracy of the CUTI-domain on target domain is lower when combined with Xception [7] and SWIM [28], since they have stronger feature extraction ability on complex datasets than VGG-19 [36], ResNet-34 [21] and Inceptionv3 [41], lead to building a CUTI-domain that is more similar to the source domain, thus better compacting the model performance within the source domain, implying stronger model IP protection ability.

Loss Function: We further explore the contribution of various parts of our proposed alternative loss function $\mathcal{L}$ in Eq. (3). The variants of $\mathcal{L}$ are designed as:

$$\mathcal{L}_1 = KL(p_s||y_s) - KL(p_t||y_t), \tag{4}$$

$$\mathcal{L}_2 = KL(p_s||y_s) - KL(p_i||y_i), \tag{5}$$

$$\mathcal{L}_3 = KL(p_s||y_s) - KL(p_t||y_t) - KL(p_i||y_i). \tag{6}$$

We validate the performance of different loss function variants on three random tasks, as shown in the right of Fig. 6. Due to the different complexity of the datasets, the difficulties of feature extraction and CUTI-domain construction are varying. On datasets with simple features (*i.e.*, MT → SN), $\mathcal{L}$ performs better, while on slightly more complex datasets, $\mathcal{L}$ only slightly outperforms other loss functions. It can be seen that the accuracy scores of different loss function variants on the target domain are relatively close, and $\mathcal{L}$ has the lowest accuracy on each dataset, implying the validity of our proposed alternative loss function $\mathcal{L}$.

Figure 6. **Left:** The accuracy (%) of SL and target-specified CUTI-domain combined with different backbones on VisDA-2017. **Right:** The accuracy (%) of SL and target-specified CUTI-domain with different loss functions on three random tasks.

5. Conclusion

In the field of artificial intelligence, protecting well-trained models as a form of IP poses challenges. To address this issue, we propose a novel CUTI-domain that acts as a barrier to constrain model performance to authorized domains. Our approach involves creating an isolation domain with features similar to those in the authorized domain, effectively blocking the model's pathway between authorized and unauthorized domains and leading to recognition failure on unauthorized domains with new private style features. We also offer two versions of the CUTI-domain, *e.g.*, target-specified and target-free, depending on whether the unauthorized domain is known. Our experimental results on seven popular cross-domain datasets demonstrate the efficacy of our lightweight, plug-and-play CUTI-domain module. We hope that our work could promote the research of model IP protection and security, which should be taken seriously in real-world applications.

Acknowledge: This work was supported by the National Natural Science Foundation of China (Nos. 62136004, 62276130), the Key Research and Development Plan of Jiangsu Province (No. BE2022842), and Huazhu Fu's A*STAR Central Research Fund and Career Development Fund (C222812010).

References

[1] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium*, pages 1615–1631, 2018. 2, 6

[2] Manaar Alam, Sayandeep Saha, Debdeep Mukhopadhyay, and Sandip Kundu. Deep-lock: Secure authorization for deep neural networks. *arXiv preprint arXiv:2008.05966*, 2020. 2

[3] Krizhevsky Alex and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. 5

[4] Rishi Bommasani, Drew A. Hudson, et al. On the Opportunities and Risks of Foundation Models. *arXiv*, aug 2021. 1

[5] Abhishek Chakraborty, Ankit Mondai, and Ankur Srivastava. Hardware-assisted intellectual property protection of deep learning models. In *57th ACM/IEEE Design Automation Conference*, pages 1–6, 2020. 1

[6] Xinyun Chen, Wenxiao Wang, Chris Bender, Yiming Ding, Ruoxi Jia, Bo Li, and Dawn Song. Refit: a unified watermark removal framework for deep learning systems with limited data. *arXiv preprint arXiv:1911.07205*, 2019. 6

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1800–1807, 2017. 8

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011. 5

[9] Yongxing Dai, Jun Liu, Yifan Sun, Zekun Tong, Chi Zhang, and Ling-Yu Duan. Idm: An intermediate domain module for domain adaptive person re-id. In *ICCV*, pages 11864–11874, 2021. 3

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1

[11] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 5

[12] Jiahua Dong, Yang Cong, Gan Sun, Zhen Fang, and Zhengming Ding. Where and how to transfer: knowledge aggregation-induced transferability perception for unsupervised domain adaptation. *IEEE TPAMI*, pages 1–1, 2021. 3

[13] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016. 3

[14] Lixin Fan, Kam Woh Ng, and Chee Seng Chan. Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. *Advances in Neural Information Processing Systems 32*, 2019. 2

[15] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017. 5

[16] WaCsurka Gabriela. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017. 3

[17] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 3

[18] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Franc¸ois Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 5

[19] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012. 3

[20] Raghuraman Gopalan, Ruonan Li, and R. Rama Chellappa. Unsupervised adaptation across domain shifts by generating intermediate data representations. *IEEE TPAMI*, 36(11), 2013. 3

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 8

[22] Kaiming He, Xiangyu Zhang, n Shaoqing Re, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[23] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1501–1510, 2017. 3, 5

[24] Jonathan J Hull. A database for handwritten text recognition research. *IEEE TPAMI*, 16(5):550–554, 1994. 5

[25] Minoru Kuribayashi, Takuro Tanaka, and Nobuo Funabiki. Deepwatermark: Embedding watermark into dnn model. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1340–1346, 2020. 2

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, may 2015. 1

[27] Shuang Li, Mixue Xie, Kaixiong Gong, Chi Harold Liu, Yulin Wang, and Wei Li. Transferable semantic augmentation for domain adaptation. In *CVPR*, pages 11516–11525, 2021. 3

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, tephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 9992–10002, 2021. 8

[29] Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. *Advances in Neural Information Processing Systems 31*, 2018. 3

[30] Wang Mei and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 1

[31] Erwan Le Merrer, Patrick Perez, and Gilles Tredan. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32(13):9233–9244, 2020. 2

[32] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop*, 2011. 5

[33] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 5

[34] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation*, pages 7249–7255, 2019. 3

[35] Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. Transformers in Medical Imaging: A Survey. *arXiv*, jan 2022. 1

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5, 8

[37] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 587–601, 2017. 2

[38] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 587–601, 2017. 2

[39] Ziwei Song, Wei Jiang, Jinyu Zhan, Xiangyu Wen, and Chen Bian. Critical-weight based locking scheme for dnn ip protection in edge computing: work-in-progress. In *Proceedings of the 2021 International Conference on Hardware/Software Codesign and System Synthesis*, 2021. 2

[40] Yunzhe Sun, Gang Yang, Dayong Ding, Gangwei Cheng, Jieping Xu, and Xirong Li. A gan-based domain adaptation method for glaucoma diagnosis. 2020. 3

[41] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 8

[42] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30, 2017. 3

[43] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*, pages 6924–6932, 2017. 3

[44] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2022. 1

[45] Lixu Wang, Shichao Xu, Ruiqi Xu, Xiao Wang, and Qi Zhu. Non-transferable learning: A new approach for model ownership verification and applicability authorization. In *ICLR*, 2021. 2, 3, 4, 5, 6

[46] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 3

[47] Run Wang, Jixing Ren, Boheng Li, Tianyi She, Chenhao Lin, Liming Fang, Jing Chen, Chao Shen, and Lina Wang. Free Fine-tuning: A Plug-and-Play Watermarking Scheme for Deep Neural Networks. *arXiv*, oct 2022. 2

[48] Ping Wah Wong and N. Memon. Secret and public key image watermarking schemes for image authentication and ownership verification. *IEEE Transactions on Image Processing*, 10(10):1593–1601, 2001. 2

[49] Shichao Xu, Lixu Wang, Yixuan Wang, and Qi Zhu. Weak adaptation learning: Addressing crossdomain data insufficiency with weak annotator. In *ICCV*, pages 8917–8926, 2021. 3

[50] Mingfu Xue, Yushu Zhang, Jian Wang, and Weiqiang Liu. Intellectual Property Protection for Deep Learning Models: Taxonomy, Methods, Attacks, and Evaluations. *IEEE Transactions on Artificial Intelligence*, pages 1–1, 2021. 1

[51] Kaichao You, Mingsheng Long, Zhangjie Cao andJianmin Wang, and Michael I. Jordan. Universal domain adaptation. In *CVPR*, pages 2720–2729, 2019. 1

[52] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Deep visual domain adaptation: A survey. *IEEE TPAMI*, pages 1–20, 2022. 1

[53] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022. 3