AdaptGuard: Defending Against Universal Attacks for Model Adaptation

Lijun Sheng[1,2], Jian Liang[2,3]*, Ran He[2,3], Zilei Wang[1], Tieniu Tan[2,4]
[1] University of Science and Technology of China
[2] CRIPAC & MAIS, Institute of Automation, Chinese Academy of Sciences
[3] University of Chinese Academy of Sciences [4] Nanjing University
slj0728@mail.ustc.edu.cn, liangjian92@gmail.com

Abstract

Model adaptation aims at solving the domain transfer problem under the constraint of only accessing the pretrained source models. With the increasing considerations of data privacy and transmission efficiency, this paradigm has been gaining recent popularity. This paper studies the vulnerability to universal attacks transferred from the source domain during model adaptation algorithms due to the existence of malicious providers. We explore both universal adversarial perturbations and backdoor attacks as loopholes on the source side and discover that they still survive in the target models after adaptation. To address this issue, we propose a model preprocessing framework, named AdaptGuard, to improve the security of model adaptation algorithms. AdaptGuard avoids direct use of the risky source parameters through knowledge distillation and utilizes the pseudo adversarial samples under adjusted radius to enhance the robustness. AdaptGuard is a plug-and-play module that requires neither robust pretrained models nor any changes for the following model adaptation algorithms. Extensive results on three commonly used datasets and two popular adaptation methods validate that AdaptGuard can effectively defend against universal attacks and maintain clean accuracy in the target domain simultaneously. We hope this research will shed light on the safety and robustness of transfer learning. Code is available at https://github.com/TomSheng21/AdaptGuard.

1. Introduction

Over the last decade, great efforts have been dedicated to deep neural networks (DNN) [20, 16], which have made significant progress in computer vision, natural language processing, and many other applications. Since the data distribution during deployment often differs from the training set, unsupervised domain adaptation [5, 11, 34, 30, 23] is widely studied to improve the model performance across different domains and save the cost of extra annotations. Growing attention is devoted to a new situation, model adaptation [29, 24], which restricts algorithms to only access the pretrained model from the source domain and the unlabeled target data. This new paradigm [28, 61, 32, 22, 2, 55, 10, 46] quickly becomes popular among the transfer learning field thanks to the appealing privacy-preserving property and the competitive performance to source-data-dependent methods. Nowadays, the model adaptation scheme has been widely applied to various tasks, *e.g.*, image classification [29, 61, 32, 19], semantic segmentation [33, 21, 62], object detection [59, 25].

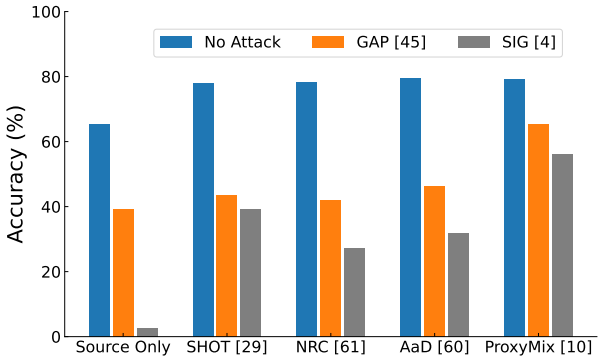


Figure 1. Target performance of five model adaptation methods under universal adversarial perturbation (*i.e.*, GAP), and backdoor attack (*i.e.*, SIG) on the task **A**→**P** from **OfficeHome** [53] dataset.

As model adaptation methods always trust the source providers unconditionally [29], users may ignore the potential risks inside the source models. In fact, providers would use inherent properties of models and the transferability of loopholes to attack clients' models which have been re-trained by adaptation algorithms. This vulnerability may cause great security breaches when the models are deployed in real-world scenarios. For example, if the street sign recognition system is under attack, it will make wrong decisions during driving. And attacking the digital recogni-

*To whom correspondence should be addressed.

tion network will cause great losses, especially in the financial systems and security fields. The widely-used robustness metrics [13, 35] are based on image-specific adversarial perturbations. As image-specific perturbations need multiple queries to the victim model per image, for the sake of simplicity and practicality, we choose to investigate image-agnostic perturbations, aka universal perturbations. In our framework, universal perturbations are generated from the source side and directly used to execute attacks on the model after adaptation in the target domain. This framework is more suitable and flexible and can be extended to many situations of different attacks.

This paper considers two kinds of universal perturbations from the source domain, *i.e.*, universal adversarial perturbation (UAP) and backdoor attack. UAP [37, 45] aims to generate a quasi-imperceptible image-agnostic perturbation, which is able to alter the predictions on a large percentage of test samples. Backdoor attack [14, 56, 7, 4] poisons the training dataset with a designed trigger to embed the backdoor in the model, and when the trigger appears during testing, the prediction will be disturbed following the preset mode. We empirically test the transferability of the selected universal attacks [45, 4] from the source domain towards existing model adaptation methods [29, 10, 61, 60], and results are shown in Fig. 1. It is obvious that the performance under perturbation is much lower than the normal value, indicating that the model adaptation method will indeed inherit the risk from the source side.

To defend against the well-transferred universal attacks for model adaptation, we propose a model preprocessing framework, named AdaptGuard. In particular, we conjecture the source model parameters to be risky and avoid using them directly as initialization. Thereby, we extract the effective information from the source model via knowledge distillation [17]. To against local perturbation, we introduce adversarial examples by projected gradient descent (PGD) [35] into distillation. Further, we develop a radius-adjusting strategy that constrains adversarial examples to gradually become stronger, mitigating the negative impact on student network training at an early stage. In this manner, the student model is expected to provide reliable initialization for the following model adaptation methods. In the experiment, we validate the effectiveness of AdaptGuard for two popular model adaptation methods (*i.e.*, SHOT [29] and NRC [61]) on three datasets, including Office, OfficeHome, and DomainNet126. Besides, we provide the result under image-specific attack to show the flexibility of our framework. Our contributions are summarized as follows:

- We examine the vulnerability of model adaptation against well-transferred universal attacks from malicious source providers, providing a new but critical perspective to existing methods.
- We propose a simple yet effective framework, named AdaptGuard, to enhance the defense ability of model adaptation methods against universal attacks.
- Extensive empirical results show that AdaptGuard performs effective defense and meanwhile maintains the domain transfer capacity.

2. Related Work

2.1. Model Adaptation

Model adaptation [28, 29, 19], also known as source-free unsupervised domain adaptation [61, 46, 46, 10, 60, 9], aims to transfer knowledge from the well-trained source model to the unlabeled target domain. Self-supervised learning-based methods [10, 58] are popular in the model adaptation task. SHOT [29], as a representative work, provides an effective framework to align the target domain to the source hypothesis with information maximization and self-training strategy. NRC [61] exploits the target distribution structure and encourages strong consistency between the adjacent samples for better alignment. SHOT++ [32] employs MixMatch [6] to alleviate the noise caused by less-confident samples and uses a rotation prediction task to improve domain transfer.

Another mainstream idea to solve the model adaptation problem is source domain generation [24, 22, 46]. SDDA [22] uses the conditional GAN [36] under the supervision of the source model to generate samples to replace absent source data. CPGA [46] generates source avatar prototypes via contrastive learning and achieves adaptation with target pseudo labels. SFIT [18] designs a two-branch framework to achieve image translation and fine-tunes the target model with the generated images. However, generation-based methods always require additional models and are difficult to train.

2.2. Universal Adversarial Perturbation

Previous works [52, 13] observe deep neural networks are vulnerable to imperceptible perturbations, which can change the prediction of the networks. Researchers [37] find that an image-agnostic perturbation is able to fool most samples during inference, which is termed as universal adversarial perturbation [65]. UAP [37] generates the perturbation by employing DeepFool [38] per image iteratively until the fooling rate reaches the desired value. Some methods [39, 45] use the generative ability of GANs [12] to avoid directly optimizing the perturbation. For short training time and simplicity of the framework, the following works [64, 51] improve the efficiency of searching universal perturbations without using generative models.

Corresponding to attack methods, various defense strategies [37, 41, 51, 35, 47] against UAP have also been explored. Fine-tuning with perturbed images [37] is employed

(a) Visualize UAP. (b) Original 'clock'. (c) + UAP [37].

(d) + GAP [45]. (e) + Blended [7]. (f) + SIG [4].

Figure 2. Visualization of perturbations and examples of an artistic image 'clock' from **OfficeHome** [53] under four attacks. The red border is only used to distinguish the attacked images here.

and proposes a shared adversarial training strategy [41] is designed to handle the trade-off between accuracy on clean examples and robustness against UAP. Shafahi et al. [51] use universal adversarial training by optimizing a min-max problem using the alternating or simultaneous stochastic gradient. There are also some works [1, 3] study about robust domain adaptation, but they focus on the image-specific attack and always need additional information (*i.e.*, robust ImageNet pretrained model). Different from them, we consider UAP which is more suitable for model adaptation and design defense strategy with external resources.

2.3. Backdoor Attack

Backdoor attacks [14, 56, 26, 67] poison the training dataset with a well-designed trigger to embed a backdoor into the model. The victim model behaves normally on clean samples, but when the trigger appears, it makes predictions expected by attackers. BadNets [14] first proposes the backdoor attack by putting a pattern of bright pixels in the corner of the MNIST images. Blended [7] achieves backdoor poisoning attacks with a fixed image trigger through a blended injection strategy. The following poison methods [4, 50, 63] design invisible and natural triggers with effective poisoning manners. Besides using a single pattern as the trigger, there are also backdoor attacks [43, 42] that control the training process and learn the trigger during training.

We focus on backdoor defense in the post-training stage [56, 57, 54, 15] which mitigates the impact of the backdoor on a well-trained model. NAD [27] utilizes a teacher network to guide the fine-tuning of the victim model. ANP [57] prunes sensitive neurons under adversarial neuron perturbations to remove the injected backdoor. These defense methods always require the part of clean training data, so they can not be utilized in the model adaptation scenarios.

3. Universal Attacks in Model Adaptation

Most model adaptation methods follow the framework proposed in SHOT [29] and train the source model f_s using empirical risk minimization with label smoothing technique [40] on labeled source data $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$. Then, users utilize the well-trained source model and unlabeled target data $\mathcal{D}_t = \{x_i^t\}_{i=1}^{N_t}$ to build the target model f_t through their own adaptation algorithms.

Existing adaptation algorithms always ignore the risk from the source domain, whose training procedure is actually uncontrolled in real scenarios. In this section, we discuss two types of universal attacks that are easy to implement by source domain providers.

Remark. We focus on universal attacks from the source side in this section, which are also the main attack methods considered in our framework. Different from most image-specific attacks, universal attacks *need no queries to the victim model per image*, which are more convenient to calculate and implement.

3.1. Universal Adversarial Perturbation

Previous work [37] observes the existence of universal adversarial perturbation v which misleads the classifier on almost test images. The ℓ_p norm of the perturbation v is less than a predefined value, guaranteeing its stealthiness and semantic invariance. v is calculated from the well-trained model and can be generalized well on the whole data. The attacker obtains the perturbation through directly optimizing the perturbation [37] or training a generative network [45] instead.

In model adaptation problems, model providers can calculate universal perturbations using the model f_s and dataset $\mathcal{D}_s$ in the source domain. The perturbation v will satisfy the following requirements:

$$\mathop{\mathbb{P}}_{x,y\sim\mathcal{D}_s}(f_s(x) \neq f_s(x+v)) \geq \delta, \; s.t. \; \|v\|_p \leq \xi, \qquad (1)$$

where $\mathbb{P}$ denotes probability calculation and $\delta < 1.0$ represents the desired fooling rate. The constraint means that the ℓ_p norm of v is not greater than ξ. As shown in Fig. 2 (c-d), the images after adding universal perturbations are quasi-imperceptible to humans. UAP is a powerful attack on current model adaption techniques due to its great stealthiness and natural generation process.

3.2. Backdoor Attack

Backdoor attack methods poison part of the training set with the designed trigger and the predefined label to embed backdoors into networks.

In the domain adaptation task, malicious model providers randomly select a portion of the source data to poison and specify a targeted category label y^b. Blended [7] takes a fixed image as a trigger that has the same size

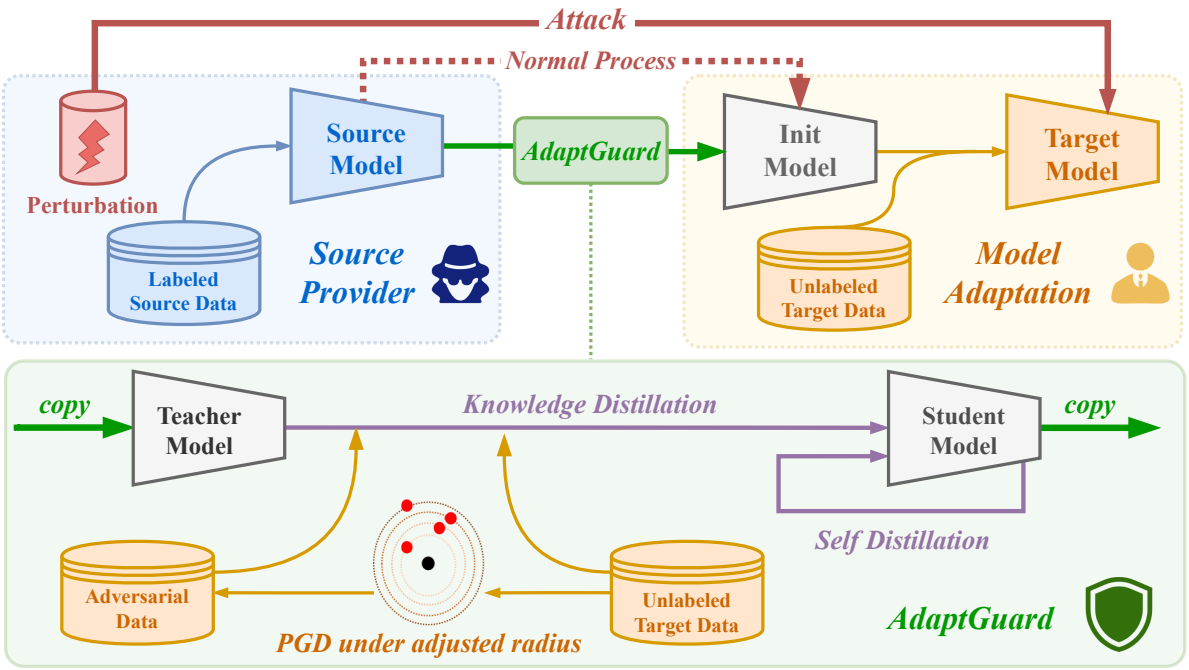


Figure 3. An overview of the proposed AdaptGuard framework. Knowledge distillation strategy is used to extract information about unlabeled data from source models. Adversarial examples under adjusted radius are introduced during distillation to further improve the model defense ability. Finally, the student network can be regarded as a reliable initialization for downstream model adaptation methods.

as the training data x_i^s. The backdoor source data $x_i^{b,s}$ is calculated through a mixup manner as follows,

$$\textbf{Blended}: x_i^{b,s} = (1-\alpha)\cdot x_i^s + \alpha \cdot I_{tri}, \tag{2}$$

where I_{tri} denotes the designed trigger and α controls the mix ratio. Another pattern of trigger [4] uses a designed perturbation that can be directly added to the training data. This perturbation needs to have a bounded amplitude and the backdoor source data must be clamped to the meaningful interval as,

$$\textbf{SIG}: x_i^{b,s} = clamp(x_i^s + \Delta_{tri}, 0, 255), \tag{3}$$

where Δ_{tri} represents the trigger perturbation and $clamp$ denotes the clamp operation to make sure all pixel values are in the range between 0 and 255.

The providers replace the selected set with its poisoned version $\{(x_i^{b,s}, y^b)\}_{i=1}^{N_b}$ and train the backdoored source model f_s^b. Fig. 2 (e-f) shows the attacked images under the above two backdoor attacks. As the victim model maintains normal performance on clean samples, backdoor attacks are difficult to detect. Even if users re-train the victim model with the target data, the backdoor can still exist and result in a successful attack sometimes.

4. Defending Against Universal Attacks

As common image-agnostic attack methods, universal adversarial perturbations and backdoor attacks are still threats to the target model after model adaptation. We proposed a model preprocessing defense framework termed AdaptGuard, to defend against potential attacks from the source provider.

4.1. Knowledge Distillation from the Source Model

To avoid directly using the risky source model as initialization during adaptation, we choose to extract information correlative with the target domain via knowledge distillation (KD) [17]. Let f be the network and $f(x)$ be the logit of input x given by f. The source model f_s is regarded as the teacher network. We init the student network f_{kd} with ImageNet [8] pretrained parameters and init the last fully-connected layer randomly. KD optimizes the student network by minimizing the discrepancy between the predictions from the two networks. In the target domain $\mathcal{D}_t = \{x_i^t\}_{i=1}^{N_t}$, we take the KL divergence as the discrepancy criterion and the optimization process can be expressed as follows:

$$\mathcal{L}_{kd}(f_{kd}) = \mathbb{E}_{x_t \in \mathcal{D}_t} \mathcal{KL}\left(\sigma(f_s(x_t)) \parallel \sigma(f_{kd}(x_t))\right), \tag{4}$$

where $\mathcal{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence of two probability distributions and σ is the softmax operation.

Table 1. Accuracies (%) of various defense methods against four attacks on **Office** [48] dataset for model adaptation (ResNet-50). Best accuracy of clean samples (**bold**), best accuracy of attacked samples (**bold red**).

Attack		UAP [37]				GAP [45]				Blended [7]				SIG [4]			
Task		A→	D→	W→	Avg	A→	D→	W→	Avg	A→	D→	W→	Avg	A→	D→	W→	Avg
Clean (Source Only)		78.8	78.4	80.1	79.1	78.8	78.4	80.1	79.1	76.6	77.0	79.9	77.9	77.3	77.1	79.2	77.8
Attack (Source Only)		37.6	47.0	42.5	42.4	13.1	46.4	43.2	34.2	0.3	0.6	1.3	0.7	1.1	13.5	4.0	6.2
Clean (SHOT)	SHOT [29]	92.6	**85.9**	**88.0**	**88.8**	92.6	**85.9**	**88.0**	**88.8**	**92.7**	**85.4**	**86.9**	**88.4**	**93.7**	**85.4**	87.0	**88.7**
	+ANP [57]	90.7	**85.9**	87.1	87.9	90.7	**85.9**	87.1	87.9	90.0	85.2	83.9	86.4	89.0	85.0	**87.2**	87.0
	+TRADES [66]	91.0	85.7	86.7	87.8	91.0	85.7	86.7	87.8	91.4	84.9	85.6	87.3	91.9	84.7	84.9	87.2
	+PGD [35]	**92.8**	85.4	86.3	88.2	**92.8**	85.4	86.3	88.2	92.5	84.6	85.4	87.5	93.6	84.6	85.5	87.9
	+AdaptGuard	88.2	83.7	85.4	85.8	88.2	83.7	85.4	85.8	88.4	83.7	85.2	85.8	88.1	83.0	84.6	85.2
Attack (SHOT)	SHOT [29]	70.6	72.0	64.9	69.2	40.5	68.0	64.2	57.6	2.9	27.7	19.9	16.8	15.2	60.6	33.5	36.4
	+ANP [57]	71.6	71.5	67.4	70.2	44.1	64.7	64.6	57.8	6.5	36.5	11.3	18.1	14.8	60.5	36.1	37.1
	+TRADES [66]	85.4	**85.1**	81.2	83.9	**67.0**	**83.6**	76.4	75.6	23.5	50.3	31.2	35.0	38.9	75.0	47.8	53.9
	+PGD [35]	**87.6**	84.7	81.9	84.7	65.5	83.0	79.1	75.8	25.5	49.2	36.7	37.1	50.8	**77.8**	52.8	**60.5**
	+AdaptGuard	86.4	83.7	**85.1**	**85.1**	61.8	**83.6**	**82.8**	**76.1**	**60.3**	**63.1**	**61.0**	**61.5**	**59.7**	61.4	**60.4**	**60.5**
Clean (NRC)	NRC [61]	90.9	**86.2**	**87.1**	88.1	90.9	**86.2**	**87.1**	88.1	91.3	**86.8**	**86.8**	**88.3**	90.1	**86.7**	86.9	**87.9**
	+ANP [57]	91.5	**86.2**	86.9	**88.2**	91.5	**86.2**	86.9	**88.2**	91.1	86.2	86.3	87.9	89.8	86.5	**87.3**	**87.9**
	+TRADES [66]	89.3	84.9	84.0	86.1	89.3	84.9	84.0	86.1	89.5	84.9	83.7	86.1	87.7	83.7	82.6	84.7
	+PGD [35]	**92.6**	82.8	84.2	86.5	**92.6**	82.8	84.2	86.5	**92.2**	82.4	84.5	86.4	**92.0**	81.5	84.2	85.9
	+AdaptGuard	91.0	85.3	85.2	87.2	91.0	85.3	85.2	87.2	91.0	84.9	85.8	87.2	89.6	82.7	85.3	85.9
Attack (NRC)	NRC [61]	62.0	63.8	59.9	61.9	35.6	63.4	60.5	53.2	1.3	19.4	14.9	11.8	9.4	41.8	20.1	23.8
	+ANP [57]	64.4	64.2	60.5	63.0	39.3	59.1	59.6	52.6	2.9	23.6	10.3	12.2	12.8	46.7	26.2	28.5
	+TRADES [66]	86.9	**84.1**	80.1	83.7	**74.8**	82.9	75.1	77.6	28.2	46.9	31.9	35.6	43.0	73.1	46.6	54.3
	+PGD [35]	**90.1**	82.7	83.1	**85.3**	74.5	81.5	79.8	**78.6**	31.5	52.3	42.6	42.1	58.0	76.2	62.1	65.4
	+AdaptGuard	86.4	83.7	**85.1**	85.1	63.6	**84.1**	**80.5**	76.1	**58.7**	**61.3**	**59.5**	**59.8**	**84.2**	**78.7**	**81.7**	**81.5**

We follow [31] and use a self-distillation strategy to mitigate the effects of noise in the teacher network. This paradigm uses a dynamic supervised signal $P(x_t)$ init with the teacher network prediction $f_s(x_t)$. The student network f_{kd} updates $P(x_t)$ with its prediction through exponential moving average (EMA) every few iterations. The self-distillation strategy introduces the self-training pattern to the standard distillation process. The final loss function is given by,

$$\begin{aligned} \mathcal{L}_{kd} &= \mathbb{E}_{x_t \in \mathcal{D}_t} \mathcal{KL}\left(\sigma(P(x_t)) \parallel \sigma(f_{kd}(x_t))\right), \\ P(x_t) &\leftarrow \gamma P(x_t) + (1-\gamma) f_{kd}(x_t), \end{aligned} \tag{5}$$

where γ is a hyperparameter representing the weight of past prediction in EMA.

4.2. Adversarial Examples under Adjusted Radius

Introducing adversarial examples into the training stage is an effective method to improve robustness. After training with adversarial examples, the decision boundary is far away from the data point, so that the neighbor block of the sample in the input space will have the same prediction. Based on this, we incorporate adversarial samples in the distillation process. Adversarial examples x_{adv} are calculated by projected gradient descent (PGD) [35] on the negative loss function in a multi-step manner:

$$x_{t,adv}^{(i+1)} = \Pi_{x+\epsilon}\left(x_{t,adv}^{(i)} + \alpha \cdot sgn(\nabla_x L(x,y))\right), \tag{6}$$

where α represents the step size in each iteration, i is the current iteration number and ϵ controls the size of the searching space to ensure the similarity to the original. For the loss function L in the formula, PGD [35] adopts the cross entropy loss which requires the ground truth of the input sample. We choose to replace it with the pseudo label in the model adaptation task. The pseudo label is also updated with the model during the distillation process. The optimization objective of KD considering both the original sample and the adversarial sample is as follows:

$$\begin{aligned} \mathcal{L}_{kd} = \mathbb{E}_{x_t \in \mathcal{D}_t} &\frac{1}{2}\mathcal{KL}\left(\sigma(P(x_t)) \parallel \sigma(f_{kd}(x_t))\right) + \\ &\frac{1}{2}\mathcal{KL}\left(\sigma(P(x_t)) \parallel \sigma(f_{kd}(x_{t,adv}))\right). \end{aligned} \tag{7}$$

Adversarial examples can help us defend against attacks, they also disturb the training process. In the early stage of the distillation process, the student network does not perform well in the target domain. In this period, strong adversarial examples will interfere with learning the target domain knowledge, resulting in performance degradation on the target clean samples. To maintain network performance and not introduce extra hyperparameters, we employ

Table 2. Accuracies (%) of various defense methods against four attacks on **OfficeHome** [53] dataset for model adaptation (ResNet-50).

Attack		UAP [37]					GAP [45]					Blended [7]					SIG [4]				
Task		A→	C→	P→	R→	Avg	A→	C→	P→	R→	Avg	A→	C→	P→	R→	Avg	A→	C→	P→	R→	Avg
Clean (Source Only)		60.7	59.3	55.0	62.4	59.4	60.7	59.3	55.0	62.4	59.4	60.5	56.0	54.0	60.9	57.9	59.7	56.4	53.6	60.7	57.6
Attack (Source Only)		29.7	13.8	12.4	15.6	17.9	37.0	16.1	16.2	22.0	22.8	2.8	1.0	0.9	1.1	1.4	3.4	4.2	3.6	1.3	3.1
Clean (SHOT)	SHOT [29]	**71.9**	**74.6**	**68.6**	**72.3**	**71.9**	**71.9**	**74.6**	**68.6**	**72.3**	**71.9**	**71.5**	**73.2**	**67.6**	**72.3**	**71.1**	**71.5**	**73.7**	**67.8**	**71.7**	**71.2**
	+ANP [57]	70.0	72.8	66.9	71.6	70.3	70.0	72.8	66.9	71.6	70.3	69.7	66.5	65.7	2.0	51.0	69.1	65.1	66.8	2.4	50.8
	+TRADES [66]	70.6	73.0	67.5	71.2	70.6	70.6	73.0	67.5	71.2	70.6	70.3	72.3	67.1	71.0	70.2	70.3	72.8	67.1	70.4	70.2
	+PGD [35]	68.9	72.0	66.1	70.1	69.3	68.9	72.0	66.1	70.1	69.3	68.7	70.7	66.1	70.3	68.9	68.6	71.3	66.0	69.9	68.9
	+AdaptGuard	70.2	70.9	65.0	68.3	68.6	70.2	70.9	65.0	68.3	68.6	70.1	69.4	64.4	68.6	68.1	69.6	70.0	64.1	68.1	68.0
Attack (SHOT)	SHOT [29]	50.6	52.9	40.4	35.7	44.9	47.6	44.2	38.1	32.0	40.5	37.2	31.6	27.7	25.7	30.6	32.3	36.6	35.1	25.4	32.3
	+ANP [57]	55.8	55.8	41.9	38.3	48.0	52.9	47.9	37.7	38.0	44.1	37.9	38.2	29.8	1.1	26.7	50.9	47.0	45.7	1.3	36.2
	+TRADES [66]	**69.8**	70.8	**65.1**	66.5	68.1	**68.8**	67.5	61.7	64.0	65.5	48.3	46.4	37.8	40.6	43.3	56.6	56.5	55.7	53.8	55.6
	+PGD [35]	68.8	**71.2**	65.0	**68.2**	**68.3**	68.6	**70.2**	**62.9**	**65.9**	**66.9**	52.6	52.1	46.1	48.2	49.7	62.1	64.8	**61.3**	60.2	62.1
	+AdaptGuard	70.0	70.1	64.0	67.2	67.8	68.6	68.1	61.4	64.3	65.6	**53.4**	**54.0**	**47.8**	**51.8**	**51.7**	**62.3**	**65.8**	59.6	**61.7**	**62.4**
Clean (NRC)	NRC [61]	**72.4**	**75.0**	**67.8**	**71.8**	**71.8**	**72.4**	**75.0**	**67.8**	**71.8**	**71.8**	**71.6**	**74.3**	**67.6**	**70.9**	**71.1**	**71.7**	**74.1**	**67.7**	**71.0**	**71.1**
	+ANP [57]	71.1	73.5	67.0	71.3	70.7	71.1	73.5	67.0	71.3	70.7	70.4	70.9	66.5	5.8	53.4	70.8	70.5	66.6	4.6	53.1
	+TRADES [66]	64.9	65.8	58.7	66.1	63.9	64.9	65.8	58.7	66.1	63.9	64.6	64.0	57.9	65.1	62.9	63.2	64.1	56.6	64.5	62.1
	+PGD [35]	65.1	68.3	61.6	66.5	65.4	65.1	68.3	61.6	66.5	65.4	64.3	67.4	61.5	65.4	64.6	63.9	68.6	62.0	65.3	64.9
	+AdaptGuard	70.9	70.8	64.5	68.1	68.6	70.9	70.8	64.5	68.1	68.6	70.6	68.3	63.6	67.5	67.5	69.8	68.5	63.0	67.7	67.2
Attack (NRC)	NRC [61]	45.6	47.4	33.2	29.2	38.8	44.8	40.2	33.8	31.5	37.6	25.1	32.7	22.2	20.2	25.0	21.2	31.7	24.9	19.1	24.2
	+ANP [57]	51.4	52.0	37.2	31.1	42.9	49.6	44.0	35.2	35.7	41.1	28.8	36.4	24.6	2.7	23.1	38.5	38.6	27.1	3.4	26.9
	+TRADES [66]	61.8	58.7	50.5	55.9	56.7	59.6	52.8	44.9	53.3	52.6	34.4	30.4	22.1	27.9	28.7	34.3	32.9	29.3	28.8	31.3
	+PGD [35]	64.1	65.3	58.0	62.4	62.5	63.4	62.1	54.3	59.4	59.8	47.4	43.2	33.9	37.7	40.5	50.7	54.8	46.2	44.9	49.2
	+AdaptGuard	**70.4**	**68.5**	**63.3**	**66.8**	**67.3**	**69.5**	**66.7**	**61.5**	**64.4**	**65.5**	**53.0**	**51.5**	**45.6**	**50.8**	**50.3**	**61.1**	**63.2**	**55.9**	**62.1**	**60.6**

a radius-adjusting strategy. We restrict the radius ϵ of the adversarial example's searching space to grow linearly from 0 to the maximum value. Adversarial examples also follow a change from weak to strong. The initial adversarial examples are similar to the original and will have fewer side effects during the distillation process. Later when the student model already has stable performance on the target domain, strong adversarial examples can be exploited to further improve the defense capability.

Discussions of the overall defense process. We provide a summary of AdaptGuard here. Users of model adaptation methods should not directly take the well-trained model from the source domain as the initialization of the follow-up algorithm. For the sake of safety, it is recommended to employ AdaptGuard as a model preprocessing step to extract effective information within the source model and discard its risky parameters. Then take the student network as initialization for their own model adaptation algorithm. With no restrictions on source training or target adaptation and no requirements about the network structure, AdaptGuard can be a general solution applied to many scenarios.

5. Experiment

5.1. Setup

Datasets. Our experiments are based on three popular image classification datasets in the model adaptation task. **Office** [48] is a standard benchmark whose images are collected from the office environment. It contains 31 categories across three domains (*i.e.*, Amazon (A), DSLR (D), and Webcam (W)). **OfficeHome** [53] is a medium-size dataset consisting of 65 categories and four domains (*i.e.*, Art (A), Clipart (C), Product (P), and Real World (R)). OfficeHome has a more balanced sample number across domains and is now the most used dataset. **DomainNet126** [49], a subset version of DomainNet [44], is a large-size benchmark that consists of 126 categories and 4 domains (*i.e.*, Clipart (C), Painting (P), Real (R), and Sketch (S)). Different from the above datasets, it has a separate test set, and we only choose four tasks in a loop (*i.e.*, CP, PR, RS, and SC) due to a large number of experiments.

Implementation Details. We choose two popular model adaptation methods, SHOT [29] and NRC [61], as our base methods. Also, we decide to use two universal adversarial perturbations (*i.e.*, UAP [37] and GAP [45]) and two backdoor attacks (*i.e.*, Blended [7] and SIG [4]) as universal attack methods from the source domain. As UAP and GAP generate perturbation based on a well-trained model, they share the same source models with the original model adaptation algorithms. Blended and SIG embed the backdoor into the model using different patterns, so we train a set of source models for each one. L_{inf} norm of universal perturbation generated using UAP and GAP is required to be no greater than 10/255. In Blended and SIG, the poisoning rate

Table 3. Accuracies (%) of various defense methods against four attacks on **DomainNet126** [44] dataset for model adaptation (ResNet-50).

Attack		UAP [37]					GAP [45]					Blended [7]					SIG [4]				
Task		CP	PR	RS	SC	Avg	CP	PR	RS	SC	Avg	CP	PR	RS	SC	Avg	CP	PR	RS	SC	Avg
Clean (Source Only)		46.7	74.1	48.6	56.8	56.5	46.7	74.1	48.6	56.8	56.5	44.7	73.3	48.1	54.6	55.2	44.6	73.2	48.1	55.3	55.3
Attack (Source Only)		42.3	49.1	6.8	37.4	33.9	14.2	42.4	11.6	34.5	25.7	0.4	0.4	0.0	0.6	0.4	0.9	1.3	0.0	0.2	0.6
Clean (SHOT)	SHOT [29]	**61.0**	**79.8**	**56.3**	**69.8**	**66.8**	**61.0**	**79.8**	**56.3**	**69.8**	**66.8**	**60.2**	**79.9**	**55.7**	**69.4**	**66.3**	**60.5**	**79.8**	55.1	**69.4**	**66.2**
	+ANP [57]	52.9	78.6	49.7	66.9	62.0	52.9	78.6	49.7	66.9	62.0	52.5	77.4	52.5	64.8	61.8	53.5	76.9	52.3	66.4	62.3
	+TRADES [66]	57.7	41.7	53.2	69.4	55.5	57.7	41.7	53.2	69.4	55.5	56.7	35.6	51.2	68.9	53.1	56.9	53.4	51.7	68.3	57.6
	+PGD [35]	54.7	66.1	53.0	68.4	60.5	54.7	66.1	53.0	68.4	60.5	53.8	40.1	53.9	69.0	54.2	54.2	37.9	53.3	68.0	53.4
	+AdaptGuard	54.7	77.7	55.8	67.6	64.0	54.7	77.7	55.8	67.6	64.0	53.1	77.5	**55.7**	66.0	63.1	53.1	77.6	**56.4**	66.8	63.5
Attack (SHOT)	SHOT [29]	**58.8**	61.7	41.4	67.9	57.4	39.3	51.0	37.3	57.5	46.3	31.9	54.5	30.6	38.0	38.7	47.2	71.6	47.1	54.4	55.1
	+ANP [57]	50.8	64.9	39.2	65.1	55.0	38.2	48.3	35.7	57.8	45.0	35.9	50.2	35.5	49.8	42.9	50.7	72.8	47.1	56.7	56.8
	+TRADES [66]	57.0	42.4	52.8	**69.1**	55.3	**55.2**	43.2	52.1	**67.9**	54.6	37.4	21.5	41.2	55.9	39.0	**53.8**	49.8	50.6	66.3	55.1
	+PGD [35]	54.2	65.3	52.8	68.2	60.1	53.2	64.7	52.3	67.8	59.5	**39.9**	26.4	42.3	**59.1**	41.9	52.7	35.7	52.2	**67.4**	52.0
	+AdaptGuard	54.3	**76.6**	**55.2**	67.4	**63.3**	52.0	**70.7**	**54.4**	64.9	**60.5**	39.5	**61.1**	**45.4**	54.9	**50.2**	51.5	**75.5**	**54.7**	65.0	**61.6**
Clean (NRC)	NRC [61]	**63.5**	**82.0**	**60.7**	**72.1**	**69.6**	**63.5**	**82.0**	**60.7**	**72.1**	**69.6**	**61.9**	**81.5**	**60.7**	**71.6**	**68.9**	**62.5**	**81.4**	**60.8**	**71.3**	**69.4**
	+ANP [57]	62.4	81.3	60.3	71.8	68.9	62.4	81.3	60.3	71.8	68.9	61.7	80.7	59.8	69.9	68.0	61.3	80.8	60.4	70.2	68.4
	+TRADES [66]	32.2	55.5	35.6	63.7	46.7	32.2	55.5	35.6	63.7	46.7	30.9	32.0	35.5	62.3	40.2	27.9	49.3	33.4	56.5	44.9
	+PGD [35]	49.6	64.6	50.9	66.9	58.0	49.6	64.6	50.9	66.9	58.0	46.3	65.4	51.0	66.0	57.2	46.0	64.6	51.8	66.4	58.6
	+AdaptGuard	56.8	78.1	59.8	67.7	65.6	56.8	78.1	59.8	67.7	65.6	55.2	78.3	60.5	66.2	65.0	55.5	78.0	60.6	66.3	65.3
Attack (NRC)	NRC [61]	**61.7**	59.0	38.4	69.3	57.1	40.4	47.2	33.9	58.5	45.0	36.6	44.5	34.0	48.6	40.9	51.4	65.2	32.4	36.4	45.8
	+ANP [57]	60.5	62.7	41.1	**69.7**	58.5	41.7	53.2	36.0	59.5	47.6	37.5	45.1	36.6	48.8	42.0	56.7	73.5	44.6	60.2	54.3
	+TRADES [66]	32.1	47.9	34.4	63.2	44.4	22.0	37.2	31.2	60.9	37.8	10.6	7.6	21.1	43.4	20.7	7.1	16.5	24.7	39.2	25.3
	+PGD [35]	49.4	64.3	50.7	66.7	57.8	47.3	64.0	50.0	66.1	56.9	31.8	52.0	41.9	**59.5**	46.3	39.4	62.4	50.5	65.0	53.8
	+AdaptGuard	54.3	**76.6**	**55.2**	67.4	**63.3**	**55.5**	**76.9**	**58.8**	**66.5**	**64.4**	**41.5**	**61.1**	**49.1**	57.8	**52.4**	**54.1**	**75.5**	**59.4**	**65.5**	**60.8**

in the training dataset is set to 0.1 and we choose 0 as the target category in all experiments. Note that as the dataset in model adaptation is relatively small, we select the source poison data from the whole categories in SIG. Finally, we use the accuracies of the clean samples and attacked samples (*i.e.*, Clean and Attack) as evaluation metrics to evaluate the security of the algorithms.

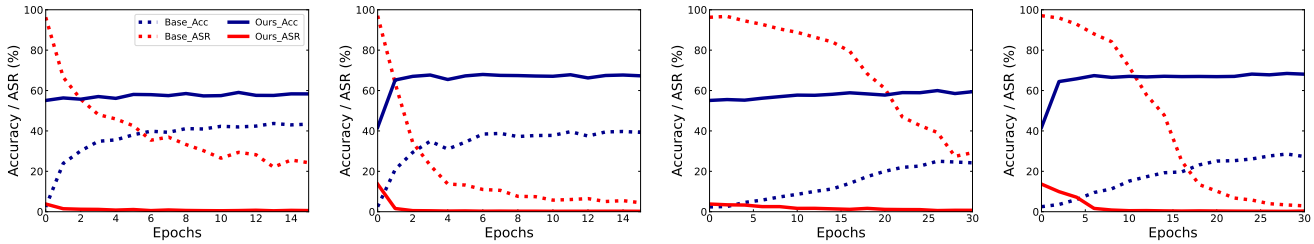
Hyperparameters. We use the same hyperparameters of AdaptGuard in all experiments. The default epoch number of knowledge distillation is 50. The weight of EMA progress in Eq. (5) is set to 0.6. And as shown in Eq. (7), both weights in loss functions are 0.5. About PGD [35] algorithm, we search 7 iterations in the area whose L_{inf} norm is less than 4/255 and calculate step size in the way recommended by PGD [35]. For the hyperparameters in SHOT [29] and NRC [61], we follow their official settings.

Baselines. Since there is no previous work about the universal attack on model adaptation tasks, we use several robust training strategies as comparison methods. Considering backdoor defense, we use ANP [57] to cut the sensitive BatchNorm layers from the source model and then employ model adaptation algorithms. As ANP fails under unlabeled data, we use half of the clean source dataset in its optimize process, which is not allowed in our framework. Besides, we add PGD [35] and TRADES [66] terms to model adaptation methods as an additional loss function. The trade-off hyperparameters of PGD and TRADES are sensitive in joint training, so we use 0.2 in Office and OfficeHome and 0.1 in DomainNet126 to ensure convergence. In addition, SHOT can also be regarded as a fine-tune-based defense method.

5.2. Results

We evaluate AdaptGuard on three image classification datasets against four universal attacks and the results are shown in Table 1, 2, 3. Due to space limitations, for Office and OfficeHome, we only report the average value according to the source domain and leave the detailed results in the **supplementary material**. And we also evaluate AdaptGuard under image-specific attacks from the source side to show the flexibility of our framework, whose results are provided in Table 4.

Analysis about universal adversarial perturbations. Universal adversarial perturbations generated from the source domain can transferred to the target domain. As shown in Table 2, the performance of the source-only model drops 41.5% when attacked by UAP. Model adaptation methods also have the ability to defend against perturbations. When employing SHOT on the source-only model, this drop gap narrows to 27.0%. PGD and TRADES can provide an effective defense against adversarial perturbations, but they make a huge impact on the clean sample performance. AdaptGuard achieves satisfactory defense against universal adversarial perturbations and maintains the transfer capability of target samples.



(a) SHOT under Blended attack. (b) SHOT under SIG attack. (c) NRC under Blended attack. (d) NRC under SIG attack.

Figure 4. Analysis of Performance trend of backdoor defense on **A→P** from **OfficeHome** [53].

Table 4. Accuracies (%) of various defense methods against **image-specific attack** on three benchmarks for model adaptation.

Datasets	OfficeHome	Office	DomainNet126
SHOT [29]	13.7	11.1	26.1
+ANP [57]	27.2	19.5	31.7
+TRADES [66]	61.6	57.1	52.5
+PGD [35]	67.0	63.9	59.2
+AdaptGuard	**67.1** (+0.1)	**84.7** (+20.8)	**60.5** (+1.3)
NRC [61]	5.5	6.7	17.0
+ANP [57]	16.1	9.9	24.3
+TRADES [66]	41.9	70.1	25.1
+PGD [35]	57.5	83.8	55.6
+AdaptGuard	**65.6** (+8.1)	**85.5** (+1.7)	**63.3** (+7.7)

Table 5. **Ablation studies** on three tasks (*i.e.*, **A→P** in OfficeHome, **A→W** in Office, and **P→R** in DomainNet126).

Attack		UAP [37]			Blended [7]		
Task		A→P	A→W	P→R	A→P	A→W	P→R
Clean	SHOT [29]	78.0	**91.6**	78.8	**77.9**	**90.8**	79.9
	+AdaptGuard (w/o adv)	**78.2**	88.7	**80.5**	76.5	88.7	**80.4**
	+AdaptGuard (w/o adjust)	74.0	83.1	74.9	73.4	82.5	75.2
	+AdaptGuard	77.0	83.9	77.7	76.8	84.0	77.5
Attack	SHOT [29]	52.1	74.5	61.7	43.4	4.1	54.5
	+AdaptGuard (w/o adv)	47.9	71.8	63.7	56.7	56.7	**64.4**
	+AdaptGuard (w/o adjust)	73.7	76.7	74.1	57.1	59.4	59.9
	+AdaptGuard	**76.6**	**83.9**	**76.6**	**58.3**	**61.6**	61.1

Analysis about backdoor attacks. The backdoor embedded in the source model can achieve attacks on the target domain. As shown in Table 3, for source-only models, the accuracy on the poisoned images is less than 1%, but performance on the clean test set is about 55%. Similar phenomena are also found in other benchmarks. Model adaptation methods can defend against backdoor attacks to some extent. It is found that SHOT achieves a better defense ability than NRC from the same victim source model. We think SHOT employs fine-tuning with pseudo labels which is regarded as a direct backdoor defense while NRC only exploits the target domain structure. PGD and TRADES can defend against backdoor attacks in some tasks. They generate adversarial examples to provide more diverse samples for the victim model, promoting forgetting of training set knowledge, and weakening the effectiveness of backdoor attacks. ANP acts as a good defense method against backdoor attacks, but may not be suitable for model preprocessing in transfer problems because pruning has side effects on the subsequent model adaptation methods. From the experiment results, AdaptGuard defends against backdoor attacks better than compared methods. Take OfficeHome as an example, as shown in Table 2, AdaptGuard achieves 51.7% and 62.4% accuracy based on SHOT, and 50.3% and 60.6% accuracy based on NRC under Blend and SIG respectively.

Analysis about accuracy and ASR during the backdoor defense. Attack success rate (ASR) refers to the proportion of samples that can be classified into the target class after the backdoor attack. It is a common metric for evaluating backdoor defenses. We provide the curves of accuracy and ASR in the target during model adaptation shown in Fig. 4. For model adaptation methods, ASR drops rapidly at the beginning of training. This phenomenon proves that more samples being misled to the target class are misclassified to other wrong classes. These misclassified samples make ASR value better but contribute nothing to the adaptation task, so it is necessary to consider more accuracy metrics. Our method, as shown in the curves, supports a robust initialization to downstream adaptation methods and achieves higher and stabler accuracy of attacked images.

Defend against image-specific attack. For a deeper discussion, we also validate AdaptGuard under the image-specific attack. We use a PGD-7 [35] with L_{inf} norm 4/255 from the source domain to generate an adversarial sample for each test instance, and report the attack accuracy of the target model in Table 4. From the results, the performance of model adaptation algorithms drops a lot under image-specific attacks from the source side, and AdaptGuard can also provide a strong defense in various benchmarks. This implies that our framework has attractive flexibility.

5.3. Ablation Study

To further study the contribution of terms in our proposed method, we investigate the effectiveness of introducing adversarial samples and the radius-adjusting strat-

egy used in AdaptGuard. When adversarial examples are not used, we adjust the weight of the loss function to 1.0 as shown in Eq. (5), in order to ensure that the value of the loss function does not change much. As shown in Table 5, AdaptGuard significantly improves the model's defense against backdoor attacks as a network preprocessing process. The introduction of adversarial examples improves the adversarial robustness of the model but also affects the performance of clean samples. The radius-adjusting strategy further enhances the defense ability against two kinds of attacks and mitigates the negative impact of adversarial examples on performance.

6. Conclusion

In this paper, we study the vulnerability of model adaptation methods toward universal attacks across domains. Existing model adaptation methods fail under the universal adversarial attack and the backdoor attack from the source model providers. We propose a defense framework named AdaptGuard to defend against potential attacks from the source domain and maintain the target performance without requiring additional resources. AdaptGuard employs knowledge distillation to avoid direct copying source parameters and utilizes adversarial examples with a gradually adjusted searching radius. Experiments on common datasets validate that AdaptGuard can effectively defend against universal attacks in the model adaptation scenario. And we plan to investigate various model adaptation tasks (e.g., segmentation) under more complex image-specific attacks (e.g., transfer-based attacks) in future work to extend our framework in broader application scenarios.

Acknowledgment

This work was partially funded by National Natural Science Foundation of China under Grants (62276256 and U21B2045) and Beijing Nova Program under Grant Z211100002121108. The authors would like to thank Zi Wang (AHU) and Jiyang Guan (CASIA) for their valuable discussions.

References

[1] Peshal Agarwal, Danda Pani Paudel, Jan-Nico Zaech, and Luc Van Gool. Unsupervised robust domain adaptation without source data. In *Proc. WACV*, 2022. 3

[2] Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *Proc. CVPR*, pages 10103–10112, 2021. 1

[3] Muhammad Awais, Fengwei Zhou, Hang Xu, Lanqing Hong, Ping Luo, Sung-Ho Bae, and Zhenguo Li. Adversarial robustness for unsupervised domain adaptation. In *Proc. ICCV*, pages 8568–8577, 2021. 3

[4] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *Proc. ICIP*, pages 101–105, 2019. 2, 3, 4, 5, 6, 7

[5] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010. 1

[6] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Proc. NeurIPS*, 2019. 2

[7] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 2, 3, 5, 6, 7, 8

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255, 2009. 4

[9] Yuhe Ding, Jian Liang, Bo Jiang, Aihua Zheng, and Ran He. Maps: A noise-robust progressive learning approach for source-free domain adaptive keypoint detection. *arXiv preprint arXiv:2302.04589*, 2023. 2

[10] Yuhe Ding, Lijun Sheng, Jian Liang, Aihua Zheng, and Ran He. Proxymix: Proxy-based mixup training with label refinery for source-free domain adaptation. *arXiv preprint arXiv:2205.14566*, 2022. 1, 2

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Machine Learning*, 17(1):2096–2030, 2016. 1

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *Proc. NeurIPS*, 2014. 2

[13] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proc. ICLR*, 2015. 2

[14] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 2, 3

[15] Jiyang Guan, Zhuozhuo Tu, Ran He, and Dacheng Tao. Few-shot backdoor defense using shapley estimation. In *Proc. CVPR*, pages 13358–13367, 2022. 3

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

[17] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. In *Proc. NeurIPS Workshops*, 2014. 2, 4

[18] Yunzhong Hou and Liang Zheng. Visualizing adapted knowledge in domain transfer. In *Proc. CVPR*, pages 13824–13833, 2021. 2

[19] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsu-

pervised domain adaptation without source data. In *Proc. NeurIPS*, pages 3635–3649, 2021. 1, 2

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NeurIPS*, 2012. 1

[21] Jogendra Nath Kundu, Akshay Kulkarni, Amit Singh, Varun Jampani, and R Venkatesh Babu. Generalize then adapt: Source-free domain adaptive semantic segmentation. In *Proc. ICCV*, pages 7046–7056, 2021. 1

[22] Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *Proc. WACV*, pages 615–625, 2021. 1, 2

[23] Junjie Li, Yixin Zhang, Zilei Wang, and Keyu Tu. Probabilistic contrastive learning for domain adaptation. *arXiv preprint arXiv:2111.06021*, 2021. 1

[24] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proc. CVPR*, pages 9641–9650, 2020. 1, 2

[25] Shuaifeng Li, Mao Ye, Xiatian Zhu, Lihua Zhou, and Lin Xiong. Source-free object detection by learning to overlook domain style. In *Proc. CVPR*, pages 8014–8023, 2022. 1

[26] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 3

[27] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *Proc. ICLR*, 2021. 3

[28] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *arXiv preprint arXiv:2303.15361*, 2023. 1, 2

[29] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proc. ICML*, pages 6028–6039, 2020. 1, 2, 3, 5, 6, 7, 8

[30] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *Proc. CVPR*, pages 16632–16642, 2021. 1

[31] Jian Liang, Dapeng Hu, Jiashi Feng, and Ran He. Dine: Domain adaptation from single and multiple black-box predictors. In *Proc. CVPR*, pages 8003–8013, 2022. 5

[32] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 44(11):8602–8617, 2021. 1, 2

[33] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *Proc. CVPR*, pages 1215–1224, 2021. 1

[34] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Proc. NeurIPS*, 2018. 1

[35] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. ICLR*, 2018. 2, 5, 6, 7, 8

[36] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[37] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proc. CVPR*, pages 1765–1773, 2017. 2, 3, 5, 6, 7, 8

[38] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proc. CVPR*, pages 2574–2582, 2016. 2

[39] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R Venkatesh Babu. Nag: Network for adversary generation. In *Proc. CVPR*, pages 742–751, 2018. 2

[40] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Proc. NeurIPS*, 2019. 3

[41] Chaithanya Kumar Mummadi, Thomas Brox, and Jan Hendrik Metzen. Defending against universal perturbations with shared adversarial training. In *Proc. ICCV*, pages 4928–4937, 2019. 2, 3

[42] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. In *Proc. ICLR*, 2021. 3

[43] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *Proc. NeurIPS*, pages 3454–3464, 2020. 3

[44] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proc. ICCV*, pages 1406–1415, 2019. 6, 7

[45] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proc. CVPR*, 2018. 2, 3, 5, 6, 7

[46] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. In *Proc. IJCAI*, 2021. 1, 2

[47] Min Ren, Yun-Long Wang, and Zhao-Feng He. Towards interpretable defense against adversarial attacks via causal inference. *Machine Intelligence Research*, 19(3):209–226, 2022. 2

[48] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proc. ECCV*, pages 213–226, 2010. 5, 6

[49] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proc. ICCV*, pages 8050–8058, 2019. 6

[50] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Proc. NeurIPS*, 2018. 3

[51] Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein. Universal adversarial training. In *Proc. AAAI*, pages 5636–5643, 2020. 2, 3

[52] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proc. ICLR*, 2014. 2

[53] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proc. CVPR*, pages 5018–5027, 2017. 1, 3, 6, 8

[54] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Proc. S&P*, 2019. 3

[55] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *Proc. ICLR*, 2021. 1

[56] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, Chao Shen, and Hongyuan Zha. Backdoorbench: A comprehensive benchmark of backdoor learning. In *Proc. NeurIPS*, pages 10546–10559, 2022. 2, 3

[57] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *Proc. NeurIPS*, pages 16913–16925, 2021. 3, 5, 6, 7, 8

[58] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *Proc. ICCV*, pages 9010–9019, 2021. 2

[59] Lin Xiong, Mao Ye, Dan Zhang, Yan Gan, Xue Li, and Yingying Zhu. Source data-free domain adaptation of object detector through domain-specific perturbation. *International Journal of Intelligent Systems*, 36(8):3746–3766, 2021. 1

[60] Shiqi Yang, Shangling Jui, Joost van de Weijer, et al. Attracting and dispersing: A simple approach for source-free domain adaptation. In *Proc. NeurIPS*, pages 5802–5815, 2022. 2

[61] Shiqi Yang, Joost van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *Proc. NeurIPS*, pages 29393–29405, 2021. 1, 2, 5, 6, 7, 8

[62] Fuming You, Jingjing Li, Lei Zhu, Zhi Chen, and Zi Huang. Domain adaptive semantic segmentation without source data. In *Proc. ACM MM*, pages 3293–3302, 2021. 1

[63] Yi Zeng, Won Park, Z Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proc. ICCV*, pages 16473–16481, 2021. 3

[64] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In So Kweon. Understanding adversarial examples from the mutual influence of images and perturbations. In *Proc. CVPR*, pages 14521–14530, 2020. 2

[65] Chaoning Zhang, Philipp Benz, Chenguo Lin, Adil Karjauv, Jing Wu, and In So Kweon. A survey on universal adversarial attack. *arXiv preprint arXiv:2103.01498*, 2021. 2

[66] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proc. ICML*, pages 7472–7482, 2019. 5, 6, 7, 8

[67] Zhengyan Zhang, Guangxuan Xiao, Yongwei Li, Tian Lv, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Xin Jiang, and Maosong Sun. Red alarm for pre-trained models: Universal vulnerability to neuron-level backdoor attacks. *Machine Intelligence Research*, 20(2):180–193, 2023. 3