The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)

Protecting Model Adaptation from Trojans in the Unlabeled Data

Lijun Sheng[1,2], **Jian Liang**[2,3*], **Ran He**[2,3], **Zilei Wang**[1], **Tieniu Tan**[4,2,3]
[1] University of Science and Technology of China
[2] NLPR & 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, zlwang@ustc.edu.cn, {rhe, tnt}@nlpr.ia.ac.cn

Abstract

Model adaptation tackles the distribution shift problem with a pre-trained model instead of raw data, which has become a popular paradigm due to its great privacy protection. Existing methods always assume adapting to a clean target domain, overlooking the security risks of unlabeled samples. This paper for the first time explores the potential trojan attacks on model adaptation launched by well-designed poisoning target data. Concretely, we provide two trigger patterns with two poisoning strategies for different prior knowledge owned by attackers. These attacks achieve a high success rate while maintaining the normal performance on clean samples in the test stage. To defend against such backdoor injection, we propose a plug-and-play method named DIFFADAPT, which can be seamlessly integrated with existing adaptation algorithms. Experiments across commonly used benchmarks and adaptation methods demonstrate the effectiveness of DIFFADAPT. We hope this work will shed light on the safety of transfer learning with unlabeled data.

Code — https://github.com/TomSheng21/DiffAdapt

Introduction

Over recent years, deep neural networks (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016; Dosovitskiy et al. 2021) have gained substantial research interest and demonstrated remarkable capabilities across various tasks. However, distribution shift (Saenko et al. 2010) between the training set and deployment environment inevitably arises, leading to a significant drop in performance. To solve this issue, researchers propose domain adaptation (Ben-David et al. 2010; Ganin et al. 2016; Long et al. 2018) to improve the performance on unlabeled target domains by utilizing labeled source data. As privacy awareness grows, source providers restrict user's access to raw data. Instead, model adaptation (Liang, Hu, and Feng 2020), a novel paradigm only accessing pre-trained source models, has gained popularity (Liang, Hu, and Feng 2020; Li et al. 2020). Since its proposal, model adaptation has been extensively investigated across various visual tasks, including semantic segmentation (Fleuret et al. 2021; Liu, Zhang, and Wang 2021) and object detection (Li et al. 2021a; Huang et al. 2021).

Security problems in model adaptation are always ignored, only two recent works (Sheng et al. 2023; Ahmed et al. 2023) reveal its vulnerability to the neural trojans (also known as backdoors) (Gu, Dolan-Gavitt, and Garg 2017; Chen et al. 2017) embedded in the source model. A distillation framework (Sheng et al. 2023) and a model compression scheme (Ahmed et al. 2023) are proposed to eliminate threats from suspicious source providers, respectively. In this paper, we raise a similar question: Can we trust the unlabeled target data? Different from the source model, injecting trojans through unlabeled data faces several significant challenges. It is difficult for unsupervised algorithms to directly establish a strong connection between the trigger and the target class through poisoning unlabeled data. Nevertheless, we find that well-poisoned unlabeled datasets still achieve successful backdoor attacks on adaptation algorithms, as illustrated in Fig. 1.

We decompose unsupervised backdoor injection into two parts: trigger design and poisoning strategy. First, a non-optimization-based trigger and an optimization-based trigger are introduced. We adopt the Hello Kitty trigger in Blended (Chen et al. 2017) as the non-optimization-based trigger. The optimization-based one is an adversarial perturbation (Poursaeed et al. 2018) calculated with a surrogate model. As for the poisoning sample selection, we provide two strategies for different prior knowledge owned by attackers. In cases where the attackers have ground truth labels, samples belonging to the target class are directly selected. When labels cannot be accessed, samples to be poisoned are selected by querying the open-source CLIP model (Radford et al. 2021). Please note that after the poisoning stage, attackers release the unlabeled version of the dataset for downstream adaptation users. Experimental results have shown that the collaboration of designed triggers and poisoning strategies achieves successful backdoor attacks.

To defend model adaptation against the backdoor threat, we propose a plug-and-play method called DIFFADAPT. DIFFADAPT eliminates the mapping between the backdoor trigger and the target class by neglecting potentially poisoning target samples during optimization. First, we train a potential risk target model with unlabeled data following the common model adaptation algorithms (e.g., SHOT (Liang, Hu, and Feng 2020), NRC (Yang et al. 2021)). Since poisoning samples have both semantic features and backdoor

---

[*] To whom correspondence should be addressed.

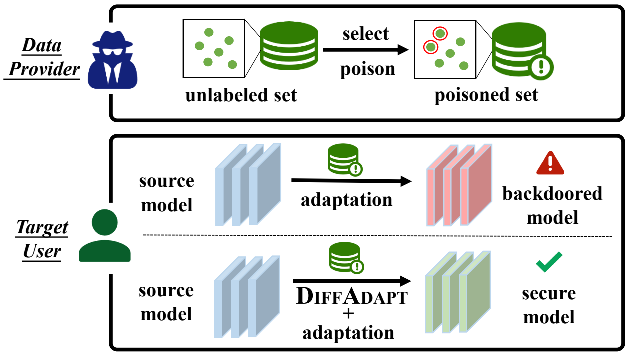


Figure 1: Backdoor attack and defense on model adaptation. With well-poisoned unlabeled data from malicious providers, target users suffer from the risks of backdoor injection. We propose DiffAdapt, a defense method against backdoor injection without sacrificing clean performance.

triggers connected with the target class, they tend to be less sensitive to random noise perturbation. We calculate the distance between the output prediction of the original and perturbed version for every unlabeled data to form the sample weight. Finally, the sample weight is averaged by the distance of all samples with the same pseudo label and a new secure target model is trained with the sample weight to avoid being injected with backdoors during unsupervised adaptation. Since no requirements for loss functions and network architectures, DiffAdapt can seamlessly integrate with existing adaptation algorithms. In the experiment section, we demonstrate the effectiveness of DiffAdapt on two popular model adaptation methods (i.e., SHOT (Liang, Hu, and Feng 2020), and NRC (Yang et al. 2021)) across three frequently used datasets (i.e., Office (Saenko et al. 2010), OfficeHome (Venkateswara et al. 2017), and DomainNet (Peng et al. 2019)). Our contributions are summarized as follows:

- We explore backdoor attacks on model adaptation through poisoning unlabeled target data. To the best of our knowledge, this is the first attempt at unsupervised backdoor attacks during adaptation tasks.
- We provide two poisoning strategies coupled with two trigger patterns capable of successfully embedding neural trojans into existing adaptation algorithms.
- We propose DiffAdapt, a flexible plug-and-play defense method against potential backdoor attacks while maintaining task performance on clean data.
- Extensive experiments involving two model adaptation methods across three benchmarks demonstrate the effectiveness of DiffAdapt.

Related Work

Model Adaptation

Model adaptation (Liang, Hu, and Feng 2020; Yang et al. 2021; Ding et al. 2023; Liang et al. 2021b), aims to transfer knowledge from a pre-trained source model to an unlabeled target domain, which is also called source-free domain adaptation or test-time domain adaptation (Liang, He, and Tan 2024; Yu et al. 2023). SHOT (Liang, Hu, and Feng 2020) first exploits this paradigm and employs information maximization loss and self-supervised pseudo-labeling to achieve source hypothesis transfer. NRC (Yang et al. 2021) captures target feature structure and promotes label consistency among high-affinity neighbor samples. Some methods (Li et al. 2020; Zhang et al. 2022; Tian et al. 2021) attempt to estimate the source domain or select source-similar samples to benefit knowledge transfer. Existing works also discuss many variants of model adaptation, such as black-box adaptation (Liang et al. 2022; Zhang et al. 2023a), open-partial (Liang et al. 2021a), and online (Wang et al. 2021; Yu et al. 2024) scenarios.

With widespread attention on the security topic, a series of works (Agarwal et al. 2022; Li et al. 2021b; Sheng et al. 2023; Ahmed et al. 2023) have studied the security of model adaptation. A robust adaptation method (Agarwal et al. 2022) is proposed to improve the adversarial robustness of model adaptation. ISFDA (Li et al. 2021b) focuses on adaptation on class-imbalanced target dataset. AdaptGuard (Sheng et al. 2023) investigates the vulnerability to image-agnostic attacks launched by the source side and introduces a model processing defense framework. SSDA (Ahmed et al. 2023) proposes a model compression scheme against source backdoor attacks. However, this paper focuses on the trojan attack on model adaptation through unlabeled poisoning data, which has not been studied so far.

Backdoor Attack and Defense

Backdoor attack (Gu, Dolan-Gavitt, and Garg 2017; Wu et al. 2022; Li et al. 2022; Zhang et al. 2023b; Liao et al. 2024) is an emerging security topic to plant a neural trojan or a backdoor associated with a trigger pattern in deep neural networks. Many well-designed backdoor triggers are proposed to achieve trojan injection. BadNets (Gu, Dolan-Gavitt, and Garg 2017) utilizes a pattern of bright pixels to attack digit classifiers and street sign detectors. Blended (Chen et al. 2017) achieves a strong invisible backdoor attack by mixing samples with a cartoon image. ISSBA (Li et al. 2021c) proposes a sample-specific trigger generated through an encoder-decoder network. In addition to poisoning-based solutions, some methods enhance their attack effects by controlling the training process (Nguyen and Tran 2021; Doan et al. 2021).

Recently, backdoor attacks have been studied in diverse scenarios besides supervised learning. Some works (Saha et al. 2022; Li et al. 2023) explore the backdoor attacks for victims who deploy self-supervised methods on unlabeled datasets. A repeat dot matrix trigger (Shejwalkar, Lyu, and Houmansadr 2023) is designed to attack semi-supervised learning methods by poisoning unlabeled data. Backdoor injection (Chou, Chen, and Ho 2023, 2024) also works on diffusion models (Dhariwal and Nichol 2021). Our work tries to launch the backdoor attack on model adaptation via poisoning unlabeled data, evaluating the danger of backdoor attacks from a new perspective.

With the emergence of trojan attack methods, various backdoor defense methods (Liu, Dolan-Gavitt, and Garg 2018; Wang et al. 2019; Guan et al. 2022; Guan, Liang, and He 2024) are proposed alternately. Fine-Pruning (Liu, Dolan-Gavitt, and Garg 2018) finds that a combination of pruning and fine-tuning can effectively weaken backdoors. NAD (Li et al. 2021d) optimizes the backdoored model using a distillation loss with a fine-tuned teacher model. ANP (Wu and Wang 2021) identifies and prunes backdoor neurons that are more sensitive to adversarial neuron perturbation. CLP (Zheng et al. 2022) removes risky channels with a high channel Lipschitz constant in a data-free way. However, those defense methods are deployed on in-distribution data and most of them require labeled samples, which are impractical for model adaptation.

Backdoor Attack on Model Adaptation

In this section, we focus on the backdoor attack on model adaptation through unsupervised poisoning. First, we review the model adaptation framework and introduce the challenge and attacker's knowledge of injecting trojans during adaptation. Subsequently, we decompose backdoor embedding into trigger design and data poisoning strategy, providing a detailed discussion respectively.

Preliminary Knowledge

Model adaptation (Liang, Hu, and Feng 2020), also known as source-free domain adaptation, aims to adapt a pre-trained source model f_s to a related target domain. Two domains share the same label space but follow different distributions with a domain gap. Model adaptation methods employ unsupervised learning techniques with the source model f_s and unlabeled data $\mathcal{D}_t = \{x_i^t\}_{i=1}^{N_t}$ to obtain a model f_t with better performance on the target domain.

Challenges of backdoor attacks on model adaptation. Unlike conventional backdoor embedding methods, backdoor attacks on model adaptation encounter several additional challenges.

Previous attackers always achieve backdoor embedding on supervised learning by adding the trigger on some samples and modifying their labels with the target class. Supervised victim learners using the poisoned dataset will capture the mapping from the trigger to the target class. However, for model adaptation algorithms, attackers are restricted to poison unlabeled data which establishes a weak connection. Moreover, the weak optimization ability of unsupervised fine-tuning also makes implanting new features very challenging.

The attacker's knowledge. In the scenario of backdoor attacks on model adaptation, attackers are allowed to control only target data, and in extremely challenging cases, only the data supply of the target class. As the target data owner, the attacker may have ground truth labels or obtain pseudo labels through the open-source basic model (e.g., CLIP (Radford et al. 2021)). Last but not least, the attacker is not allowed to access the source model and have no knowledge about the downstream adaptation learners.

Backdoor Triggers

To make adaptation algorithms capture the mapping from the trigger to the target class, we utilize the triggers with semantic information. Triggers with semantic information will be extracted by the pre-trained source model with a higher probability. Based on this requirement, we introduce a non-optimization-based trigger and an optimization-based perturbation trigger in the following.

▷ **A non-optimization-based (Blended) trigger.** Blended (Chen et al. 2017) is a strong backdoor attack technique that blends the samples with a Hello Kitty image. Blended trigger satisfies the requirements and no additional knowledge is required, making it a suitable choice as our non-optimization-based trigger.

▷ **An optimization-based (perturbation) trigger.** In addition to the hand-crafted trigger, we introduce an optimization-based method for trigger generation. Initially, a surrogate model is trained on the target dataset $\mathcal{D}_t = \{x_i^t, y_i\}_{i=1}^{N_t}$ using a cross-entropy loss function. With the surrogate model and the target data, we compute the universal adversarial perturbations (Poursaeed et al. 2018) for the target class which leads to the misclassification of the majority of samples. The perturbation has misleading semantics and is the same size as input samples which makes it a great optimization-based trigger. It is worth noting that the architecture of the surrogate model and the source model are always different, and the perturbation will not achieve such a high attack success rate on the source model due to the weak transferability between different architectures.

Data Poisoning

Previous backdoor attack methods employ data poison with a random sampling strategy. Due to the unsupervised nature of adaptation algorithms, attackers are unable to establish a connection between the trigger and the target class explicitly. Hence, a well-designed poison set selection strategy becomes critical backdoor embedding. Attackers are allowed to access either ground truth or open-source basic model predictions for the poisoning data selection. We provide a selection strategy for each condition below.

▷ **Ground-truth-based selection strategy.** When attackers hold the ground truth labels $\{y_i^t\}_{i=1}^{N_t}$ of all samples, samples belonging to the target class y_t are simply selected to construct a poisoning set $\mathcal{D}_t^{poison} = \{x_i^t\}_{i=1}^{P}$. To avoid interference for backdoor embedding, samples in other classes remain unchanged.

▷ **Pseudo-label-based selection strategy.** With no access to the ground truth labels, attackers first obtain pseudo-labels for all target data from the open-source basic model CLIP (Radford et al. 2021). Then, a base poisoning set $\mathcal{D}_t^{pl} = \{x_i^t\}_{i=1}^{P}$ consists of samples belonging to the target class y_t. In order to strengthen the poisoning set, attackers can choose to continually select samples outside of $\mathcal{D}_t^{pl}$ but with a high prediction probability for the target class y_t, creating a supplementary set $\mathcal{D}_t^{supp} = \{x_i^t\}_{i=1}^{P'}$. The final poisoning set $\mathcal{D}_t^{poison}$ is the union of the above two sets: $\mathcal{D}_t^{poison} = \mathcal{D}_t^{pl} \cup \mathcal{D}_t^{supp}$.

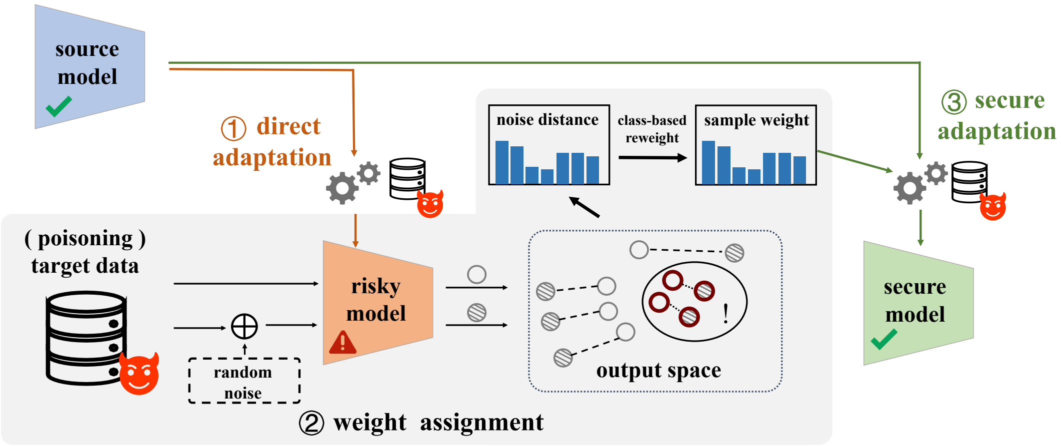


Figure 2: The framework of defense method DiffAdapt. We train a potentially risky target model and obtain the distance between the output prediction of the original and perturbed version for every unlabeled data. The sample weight is averaged by the distance of all samples with the same pseudo label and a new secure target model is trained with the sample weight.

DiffAdapt: A Secure Adaptation Method Against Backdoor Attacks

From the previous section, we learn an incredible fact: malicious target data providers can achieve a backdoor embedding on model adaptation algorithms through unsupervised poisoning. We introduce a straightforward defense method named DiffAdapt to mitigate such a risk. DiffAdapt is designed to defend against potential backdoor attacks while preserving the adaptation performance in the target domain. The framework of DiffAdapt is illustrated in Fig. 2.

The main idea inside DiffAdapt is intuitive, assigning low weights to samples that may contain backdoor triggers, instead of the uniform weights in existing adaptation methods. The key is to obtain accurate weights, that is, identify samples containing backdoor triggers. For risky target models that may contain backdoors, the outputs of poisoned samples in the unlabeled training set will be determined by their category semantics and the trigger. Since these samples have both features connected with the target class, they tend to be less sensitive to random noise perturbation, that is, there will be little change between the output of the original image and the perturbed version. Therefore, DiffAdapt assigns weights to samples based on their sensitivity to noise on potentially risky target models.

Here, we provide a detailed outline of the procedures involved in DiffAdapt. Firstly, a potentially risky target model f_t^{risk} is trained using the existing model adaptation algorithm with unlabeled target dataset $\mathcal{D}_t$. Note that "potentially" means that the target model f_t^{risk} may be secure. Our defense method does not assume that the unlabeled target domain has to contain poisoned samples. To obtain the sensitivity of each target sample, we randomly generate a Gaussian noise ϵ and construct a perturbed target domain dataset. The distance between the predictions of the original image and the perturbed version is calculated as follows:

$$\delta(x_{t,i}) = \|f_t^{risk}(\widetilde{x}_{t,i}) - f_t^{risk}(x_{t,i})\|, \tag{1}$$

where $\widetilde{x}_{t,i} = x_{t,i} + \epsilon$ represents the perturbed image. Since the risky target model f_t^{risk} has incorporated the knowledge of unlabeled training data, a certain proportion of the data tends to be less sensitive to noise like the poisoned samples. In addition, categories that are not related to backdoors contain some highly sensitive outlier samples, and the model's high dependence on them will decrease the stability of the adaptation process. Therefore, to focus on low-weight clean samples while reducing the dependence on high-weight samples, DiffAdapt reassigns sample weights according to pseudo labels. The final weight w_i is the average among all samples with the same pseudo label:

$$w(x_{t,i}) = \frac{\sum_{j=1}^{N_t} \mathbb{1}(\hat{y}_j = \hat{y}_i)\ \delta(x_{t,j})}{\sum_{j=1}^{N_t} \mathbb{1}(\hat{y}_j = \hat{y}_i)}, \tag{2}$$

where $\hat{y}_j = \mathbf{argmax}_c f_t^{risk}(x_{t,j})_c$ represents the pseudo label provided by the risk target model. Finally, the secure target domain model is obtained by retraining using the adaptation algorithm. The objective corresponding to each sample in the adaptation process will be replaced by a weighted version with w_i:

$$L(x_{t,i}) = \mathbb{E}_{x_{t,i} \in \mathcal{D}_t} w(x_{t,i})\ l(x_{t,i}), \tag{3}$$

where $l(x_{t,i})$ refers to the loss calculated on the sample $x_{t,i}$, for example, self-training loss and entropy minimization loss in SHOT (Liang, Hu, and Feng 2020) and class-consistency loss in NRC (Yang et al. 2021).

Discussion. Since DiffAdapt has no requirements on model adaptation algorithms and network architectures, it can be used as a plug-and-play defense strategy simply combined with existing adaptation algorithms (e.g., SHOT (Liang, Hu, and Feng 2020), and NRC (Yang et al. 2021)). Besides effectively defending against test-time backdoor attacks, DiffAdapt maintains the adaptation performance on the clean data.

	SHOT (Liang, Hu, and Feng 2020)										**NRC** (Yang et al. 2021)									
Task	A→W		D→A		D→W		W→A		Avg		A→W		D→A		D→W		W→A		Avg	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
Source Only	77.4	-	62.0	-	95.0	-	63.6	-	74.5	-	77.4	-	62.0	-	95.0	-	63.6	-	74.5	-
No Poisoning	92.5	-	76.4	-	97.5	-	76.4	-	85.7	-	93.7	-	78.3	-	98.7	-	77.1	-	87.0	-
Poisoning	91.2	41.9	75.8	48.4	98.1	60.0	76.4	40.9	85.4	47.8	92.5	58.1	76.6	80.3	98.1	90.3	77.4	82.3	86.1	77.7
+ CLP	90.6	28.4	76.0	**27.3**	97.5	52.9	75.0	**25.8**	84.8	33.6	91.2	23.2	75.3	55.3	96.9	76.8	74.6	72.9	84.5	57.0
+ FP	88.1	39.4	75.3	51.2	93.1	48.4	73.4	43.1	82.5	45.5	86.8	63.2	73.7	78.3	90.6	78.1	72.7	82.1	80.9	75.4
+ DiffAdapt	87.4	**9.0**	74.4	47.0	98.7	**34.8**	72.7	29.7	83.3	**30.1**	85.5	**0.0**	75.1	**49.5**	98.1	**53.6**	72.3	**23.8**	82.8	**31.7**
							Blended trigger ⇈				**Perturbation trigger** ⇊									
Poisoning	91.2	64.5	76.6	88.2	97.5	87.7	74.8	86.0	85.0	81.6	92.5	38.7	76.4	59.9	98.1	76.8	78.2	55.1	86.3	57.6
+ CLP	91.2	27.7	75.1	61.1	96.9	38.1	75.0	41.1	84.5	42.0	91.2	6.5	75.1	25.8	97.5	22.6	74.6	5.5	84.6	15.1
+ FP	88.7	52.3	74.6	70.2	94.3	74.8	73.4	53.4	82.7	62.7	86.8	26.5	74.4	30.4	92.5	71.0	73.5	37.2	81.8	41.3
+ DiffAdapt	87.4	**1.3**	73.0	**15.8**	98.1	**34.2**	72.1	**10.3**	82.7	**15.4**	88.1	**0.7**	74.6	**8.7**	98.7	**10.3**	75.0	**4.4**	84.1	**6.0**

Table 1: ACC (%) and ASR (%) of DiffAdapt against backdoor attacks on **Office** (Saenko et al. 2010) dataset for model adaptation (ResNet-50).

	SHOT (Liang, Hu, and Feng 2020)										**NRC** (Yang et al. 2021)									
Task	A→		C→		P→		R→		Avg		A→		C→		P→		R→		Avg	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
Source Only	60.2	-	58.6	-	54.7	-	62.4	-	59.0	-	60.2	-	58.6	-	54.7	-	62.4	-	59.0	-
No Poisoning	71.3	-	73.4	-	67.2	-	69.5	-	70.4	-	71.1	-	72.9	-	64.6	-	69.8	-	69.6	-
Poisoning	70.7	85.6	73.6	89.2	66.8	62.5	69.8	86.0	70.2	80.8	71.0	85.9	72.4	87.9	65.0	89.3	69.0	84.6	69.4	86.9
+ CLP	68.3	77.7	70.6	86.5	63.8	59.4	67.7	80.9	67.6	76.1	68.3	80.4	69.2	83.2	61.7	85.5	66.6	76.9	66.5	81.5
+ FP	66.8	**64.6**	68.9	72.0	61.3	**51.6**	64.4	**65.0**	65.3	**63.3**	67.6	69.6	67.7	67.2	60.1	**70.0**	63.9	**62.7**	64.8	67.4
+ DiffAdapt	69.2	68.0	71.4	**67.2**	63.3	60.6	68.6	82.5	68.1	69.6	69.1	**65.1**	71.1	**38.0**	63.4	79.8	68.1	83.3	67.9	**66.6**
							Blended trigger ⇈				**Perturbation trigger** ⇊									
Poisoning	71.7	80.2	74.2	70.7	66.4	78.2	70.4	74.0	70.7	75.8	71.4	51.9	72.2	36.0	64.8	51.8	69.3	54.7	69.4	48.6
+ CLP	69.0	46.6	70.9	36.6	62.5	70.5	68.4	54.3	67.7	52.0	68.9	23.5	69.4	8.9	61.6	38.5	67.1	33.9	66.7	26.2
+ FP	67.7	43.8	69.3	32.1	61.5	43.4	65.2	33.5	65.9	38.2	67.4	19.3	68.1	6.5	60.8	24.0	64.2	15.9	65.1	16.4
+ DiffAdapt	68.9	**1.5**	71.3	**0.9**	63.4	**4.8**	67.5	**3.5**	67.8	**2.6**	69.4	**4.5**	70.9	**0.1**	64.1	**2.0**	68.0	**0.2**	68.1	**1.7**

Table 2: ACC (%) and ASR (%) of DiffAdapt against backdoor attacks on **OfficeHome** (Venkateswara et al. 2017) dataset for model adaptation (ResNet-50).

Experiment

Setup

Datasets. We evaluate our framework on three commonly used model adaptation benchmarks from image classification tasks. **Office** (Saenko et al. 2010) is a classic model adaptation dataset containing 31 categories across three domains (i.e., Amazon (A), DSLR (D), and Webcam (W)). Since the small size of the data in the DSLR domain makes it difficult to poison a certain category, we remove two tasks whose target domain is DSLR and retain the remaining four (i.e., A→W, D→A, D→W, W→A). **OfficeHome** (Venkateswara et al. 2017) is a popular dataset whose images are collected from office and home environments. It consists of 65 categories across four domains (i.e., Art (A), Clipart (C), Product (P), and Real World (R)). **DomainNet** (Peng et al. 2019) is a large-size challenging benchmark with imbalanced classes and extremely difficult tasks. Following previous work (Tan, Peng, and Saenko 2020; Li et al. 2021b), we consider a subset version, **miniDomainNet** for convenience and efficiency. miniDomainNet contains four domains (i.e., Clipart (C), Painting (P), Real (R), and Sketch (S)) and 40 categories. For OfficeHome and miniDomainNet datasets, we use all 12 tasks to evaluate our framework.

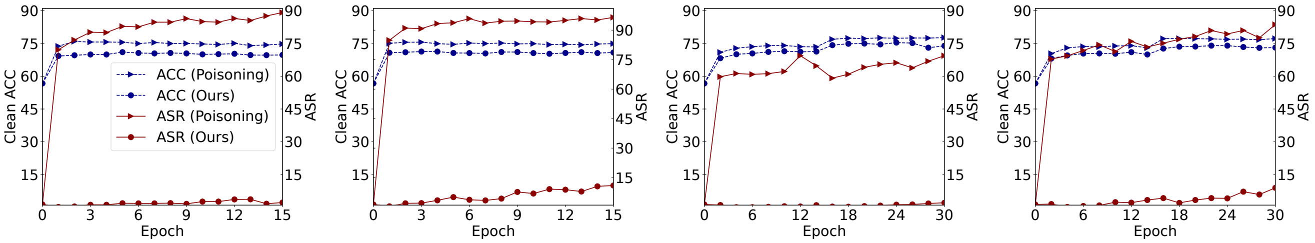
Evaluation metrics. In our experiments, we divide 80% of the target domain samples as the unlabeled training set for adaptation and the remaining 20% as the test set for metric calculation. In order to avoid loss of generality, we uniformly select class 0 in alphabetical order as the target class for backdoor attack and defense. We adopt accuracy on the clean samples (**ACC**) and attack success rate on the poison samples (**ASR**) to evaluate the effectiveness of our attack and defense method. A stealthy attack should achieve high ASR while maintaining accuracy on clean samples to keep the backdoor from being detected. Similarly, a great defense method should achieve both low ASR and high accuracy.

Baselines. Since there are no methods designed for defending backdoor injection during the adaptation stage, we select two pruning techniques that can be combined with the model adaptation algorithms and require no annotated clean samples. CLP (Zheng et al. 2022) performs data-free pruning on the potentially backdoored model based on the upper bound of the channel Lipschitz constant. FP (Liu, Dolan-Gavitt, and Garg 2018) weakens backdoors by pruning the neurons with high average activation values.

Implementation details. Different from supervised backdoor attacks, we choose two popular model adaptation methods, SHOT (Liang, Hu, and Feng 2020) and NRC (Yang et al. 2021), as victim algorithms. We use their official codes

	SHOT (Liang, Hu, and Feng 2020)										**NRC** (Yang et al. 2021)									
Task	C→		P→		R→		S→		Avg		C→		P→		R→		S→		Avg	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
Source Only	64.0	-	70.2	-	67.4	-	66.8	-	67.1	-	64.0	-	70.2	-	67.4	-	66.8	-	67.1	-
No Poisoning	80.5	-	80.3	-	77.5	-	80.9	-	79.8	-	80.7	-	81.6	-	77.5	-	83.2	-	80.7	-
Poisoning	78.8	63.3	79.6	44.5	76.3	26.0	80.6	46.9	78.8	45.2	80.6	59.2	80.8	43.1	77.1	20.9	83.1	78.1	80.4	50.3
+ CLP	76.2	65.2	77.9	44.4	73.9	26.2	78.7	56.3	76.7	48.0	78.4	62.7	78.7	47.4	74.1	28.2	80.7	82.4	78.0	55.2
+ FP	69.7	33.7	75.5	32.1	67.7	13.9	74.1	36.0	71.8	28.9	73.6	**32.6**	76.7	**29.9**	69.6	8.6	76.2	61.1	74.0	33.0
+ DIFFADAPT	75.5	**6.9**	76.5	**31.7**	72.3	**13.0**	77.0	**25.7**	75.3	**19.3**	78.9	35.3	80.4	44.9	73.9	**2.6**	79.0	**45.2**	78.1	**32.0**
								Blended trigger ⇈			**Perturbation trigger ⇊**									
Poisoning	78.3	80.0	79.7	73.0	75.6	87.8	80.3	69.1	78.5	77.5	80.2	55.0	80.7	44.1	76.6	62.0	83.2	44.4	80.2	51.4
+ CLP	75.9	60.5	78.2	52.5	73.7	63.4	78.3	61.7	76.5	59.5	77.7	49.5	78.6	29.8	74.0	37.9	80.9	46.7	77.8	41.0
+ FP	69.3	31.4	75.8	**14.4**	67.9	34.2	73.5	**29.7**	71.6	27.4	73.7	**9.1**	76.7	**3.7**	70.3	**6.5**	76.4	17.1	74.3	**9.1**
+ DIFFADAPT	74.0	**8.0**	76.7	26.1	72.7	**14.3**	77.0	32.5	75.1	**20.2**	78.0	12.1	80.1	15.3	74.7	14.2	78.7	**16.4**	77.9	14.5

Table 3: ACC (%) and ASR (%) of DIFFADAPT against backdoor attacks on **miniDomainNet** (Peng et al. 2019) dataset for model adaptation (ResNet-50).



(a) SHOT (GT Strategy). (b) SHOT (PL Strategy). (c) NRC (GT Strategy). (d) NRC (PL Strategy).

Figure 3: ACC and ASR curve of backdoor attack and DIFFADAPT on C→P from **miniDomainNet** (Peng et al. 2019).

and hyperparameters with ResNet-50 (He et al. 2016). For each adaptation algorithm, we report the results from four attack methods (two types of trigger with two poison selection strategies). For the non-optimization-based trigger, we use the Hello Kitty trigger in Blended (Chen et al. 2017) directly. The optimization-based trigger is implemented by GAP (Poursaeed et al. 2018) and L_{inf} norm is 0.5 in a 120×120 patch for Office and 100×100 patch for others.

Hyperparameters. For all experiments, we simply set the noise pixels to be sampled from a uniform distribution [-0.25, 0.25]. DIFFADAPT is a plug-and-play defense method for existing model adaptation algorithms, so no additional hyperparameters are introduced. Other details are consistent with the official settings of the adaptation algorithms.

Main Results

We evaluate two backdoor triggers with the ground truth poisoning strategy and our defense method against the above attacks. The results are shown in Table 1, 2, 3. Note that the results of the non-optimization-based (Blended) trigger are reported in the upper part of the tables and the results of the optimization-based backdoor (perturbation) trigger are provided in the lower part. Due to space limitations, for OfficeHome and miniDomainNet, we report the average result across tasks from the same source domain and leave the detailed results in the **supplementary material**[1].

Results about backdoor attacks on model adaptation. Non-optimization-based backdoor attacks (Blended trigger) obtain a high ASR across various benchmarks. Take results on OfficeHome in Table 2 as an example, Blended trigger achieves an average ASR of 80.8% on SHOT and 86.9% on NRC. Besides, the injection of Blended trigger maintains the target domain performance of the victim model which demonstrates its great concealment. As shown in Table 1, 2, compared with the clean training set, the poisoning set with Blended trigger only causes 0.6% and 0.2% decrease in accuracy on Office and OfficeHome datasets, respectively.

For optimization-based backdoor attacks, as shown in Table 1, the perturbation trigger achieves an average ASR of 81.6% and 57.6% on Office dataset on two adaptation algorithms. And on miniDomainNet dataset in Table 3, the average ASR of perturbation trigger on SHOT arrives at 77.5%. Also, the perturbation trigger keeps the model's performance, only reduces the clean accuracy of SHOT on miniDomainNet from 79.8% to 78.5% and causes a smaller 0.5% gap on NRC algorithm. Generally, the results demonstrate the effectiveness of two backdoor triggers, revealing the poisoning risk of unlabeled data during the adaptation.

Results about DIFFADAPT against backdoor attack. To defend against the backdoor attacks for the model adaptation, we further evaluate our proposed defense method DIFFADAPT on the above benchmarks, and the results are shown in Tables 1, 2, 3. It is obvious that DIFFADAPT effectively reduces ASR scores while maintaining the original classification ability. Take Office dataset with pertur-

---

[1] Supplementary material is available at https://github.com/TomSheng21/DiffAdapt/pdf/supp.pdf.

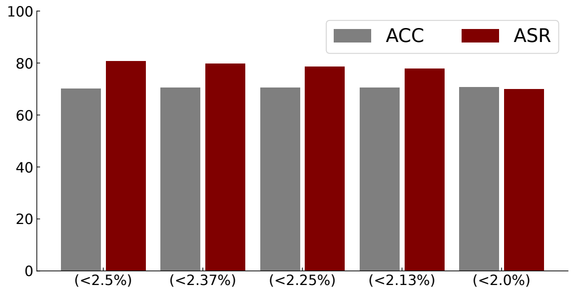


Figure 4: ACC (%) and ASR (%) of backdoor attacks under different poisoning rates for model adaptation.

Task	A→		C→		P→		R→		Avg	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
Poisoning	70.6	92.0	73.1	86.3	66.8	95.4	70.9	89.1	70.4	90.7
+ CLP	68.8	66.7	71.0	58.5	63.5	87.3	67.9	72.5	67.8	71.2
+ FP	66.9	60.3	68.3	43.7	62.7	80.4	65.7	51.0	65.9	58.9
+ DIFFADAPT	68.4	**1.4**	71.4	**9.2**	64.4	**7.4**	68.8	**10.3**	68.3	**7.1**

Table 4: ACC (%) and ASR (%) of DIFFADAPT against backdoor attacks (**pseudo label strategy**) on **OfficeHome** (Venkateswara et al. 2017) dataset for model adaptation.

bation trigger in Table 1 as an example, DIFFADAPT reduces ASR scores from 81.6% to 15.4% on SHOT and from 57.6% to 6.0% on NRC. At the same time, the clean accuracy of the target domain drops by 2.3% and 2.2% respectively, which is within an acceptable range. Compared with baseline defense methods, DIFFADAPT always achieves better ASR and clean accuracy. Although DIFFADAPT shows its superiority across most tasks, on OfficeHome data with the challenging Blended trigger, it is still slightly worse than FP whose defense is also weak. In addition, we record the curves of ASR score and accuracy of backdoor attack and DIFFADAPT on the C→P task from miniDomainNet in Fig. 3. As expected, as shown in curves of adaptation, DIFFADAPT effectively defends against backdoor injection without affecting the convergence of the base algorithm.

More Analysis

Analysis about pseudo labeling poisoning strategy. Besides the poisoning selection strategy based on ground truth labels, we also provide a pseudo-label poisoning strategy when the attacker can not access labels. The results on OfficeHome dataset with the perturbation trigger are shown in Table 4. Pseudo-label-based strategy achieves a high ASR score of 90.7% on SHOT algorithm on OfficeHome dataset. Also, DIFFADAPT outperforms baseline methods, it reduces ASR from 90.7% to 7.1% and causes only a 2.1% accuracy gap. Although the CLP obtains slightly better accuracy, it produces poor performance at backdoor removal. These results further demonstrate the flexibility of the proposed attack method and the effectiveness of DIFFADAPT.

Analysis about different network architectures. To assess the versatility of our attack framework, we evaluate our attack method on a variety of network architectures including VGG, ViT, ConvNext, MobileNet, and ResNet101.

Task	A→		C→		P→		R→		Avg	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
VGG16	64.4	76.8	67.2	52.1	55.9	26.1	62.3	69.7	62.5	56.2
+ DIFFADAPT	63.0	24.5	65.1	6.6	52.9	3.7	58.6	38.3	59.9	18.3
ViT-Base	78.3	50.7	56.1	32.5	73.2	65.5	75.4	44.7	70.7	48.4
+ DIFFADAPT	75.4	20.2	81.0	42.2	70.2	25.0	72.0	39.6	74.7	31.8
ConvNext	80.0	65.7	84.8	78.5	74.9	77.5	77.6	68.2	79.3	72.5
+ DIFFADAPT	79.7	47.5	81.4	18.7	73.2	44.9	74.8	48.6	77.3	39.9
MobileNet	63.6	80.7	62.8	56.1	57.2	26.8	63.7	77.1	61.8	60.2
+ DIFFADAPT	62.3	48.0	61.3	21.0	53.5	30.8	59.1	43.0	59.0	35.7
Resnet101	73.2	86.4	76.2	88.7	68.6	88.8	73.1	87.8	72.8	87.9
+ DIFFADAPT	70.6	24.7	71.8	41.9	65.8	34.4	69.1	50.5	69.3	37.9

Table 5: ACC (%) and ASR (%) with different backbones against backdoor attacks on **OfficeHome** dataset.

Since DIFFADAPT needs no well-designed modification for any networks, we also employ it on those backbones. The results of Blended trigger on SHOT algorithm are provided in Table 5. Take ConvNext for an example, our attack achieves an average ASR score of 72.5% while DIFFADAPT brings it down to 39.9%. It is shown that our attack method achieves a successful attack across different backbones and DIFFADAPT mitigates backdoor injection.

Analysis about different poisoning rates. In most experiments, we assume that the attacker as well as the target domain provider can control the whole unlabeled dataset. Here we study our attack method under various poison rates and report the results on OfficeHome with Blended trigger in Fig. 4. Please note that the rate in the figure refers to the poisoning rate of the whole target dataset and the samples of the target class are approximately equal to 2.5%. It is shown that a higher poisoning rate will result in a higher ASR score. Moreover, various poisoning rates share almost the same performance on the clean sample.

Conclusion

This paper discusses whether users can trust unlabeled data during model adaptation. Our study focuses on backdoor attacks during model adaptation and finds that a malicious data provider can achieve backdoor embedding through unsupervised poisoning. Furthermore, to reduce the risks of potential backdoor attacks, we propose DIFFADAPT, a plug-and-play defense method to protect adaptation algorithms. DIFFADAPT eliminates the association between triggers and target class by exchanging background areas among target samples. Extensive experiments conducted on commonly used adaptation benchmarks validate the efficacy of DIFFADAPT in effectively defending against backdoor attacks.

Limitation. It is worth noting that while our framework achieves successful attacks and defenses, some limitations still exist. We only explore the classification problem, which is a relatively basic task. Popular online adaptation algorithms employ minor optimization, making backdoor injection more difficult. As for defense, compared to direct adaptation, DIFFADAPT requires twice the computational cost. Improvements in these aspects can further improve the versatility of our framework.

Ethical Statement

In our work, we explore trojan attack and defense in model adaptation, a sub-field of machine learning. Given that model adaptation techniques are increasingly deployed in sensitive areas such as security systems and medical diagnostics, our proposed backdoor attack could potentially result in risky consequences, including accidents and security breaches. To address these risks, we propose a robust defense framework aimed at helping adaptation algorithms defend against trojan attacks. Our work highlights the vulnerability of model adaptation to such attacks, intending to raise awareness about these risks, particularly in high-stakes applications where the impact could be significant.

Acknowledgements

This work was funded by the National Natural Science Foundation of China under Grants (62276256, U2441251) and the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001).

References

Agarwal, P.; Paudel, D. P.; Zaech, J.-N.; and Van Gool, L. 2022. Unsupervised robust domain adaptation without source data. In *Proc. WACV*, 2009–2018.

Ahmed, S.; Al Arafat, A.; Rizve, M. N.; Hossain, R.; Guo, Z.; and Rakin, A. S. 2023. SSDA: Secure Source-Free Domain Adaptation. In *Proc. ICCV*, 19180–19190.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Machine Learning*, 79(1): 151–175.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*.

Chou, S.-Y.; Chen, P.-Y.; and Ho, T.-Y. 2023. How to backdoor diffusion models? In *Proc. CVPR*, 4015–4024.

Chou, S.-Y.; Chen, P.-Y.; and Ho, T.-Y. 2024. Villandiffusion: A unified backdoor attack framework for diffusion models. In *Proc. NeurIPS*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. In *Proc. NeurIPS*, 8780–8794.

Ding, Y.; Sheng, L.; Liang, J.; Zheng, A.; and He, R. 2023. Proxymix: Proxy-based mixup training with label refinery for source-free domain adaptation. *Neural Networks*, 167: 92–103.

Doan, K.; Lao, Y.; Zhao, W.; and Li, P. 2021. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proc. ICCV*, 11966–11976.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. ICLR*.

Fleuret, F.; et al. 2021. Uncertainty reduction for model adaptation in semantic segmentation. In *Proc. CVPR*, 9613–9623.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *Machine Learning*, 17(1): 2096–2030.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

Guan, J.; Liang, J.; and He, R. 2024. Backdoor Defense via Test-Time Detecting and Repairing. In *Proc. CVPR*, 24564–24573.

Guan, J.; Tu, Z.; He, R.; and Tao, D. 2022. Few-shot backdoor defense using shapley estimation. In *Proc. CVPR*, 13358–13367.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proc. CVPR*, 770–778.

Huang, J.; Guan, D.; Xiao, A.; and Lu, S. 2021. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In *Proc. NeurIPS*, 3635–3649.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Proc. NeurIPS*.

Li, C.; Pang, R.; Xi, Z.; Du, T.; Ji, S.; Yao, Y.; and Wang, T. 2023. An Embarrassingly Simple Backdoor Attack on Self-supervised Learning. In *Proc. ICCV*, 4367–4378.

Li, R.; Jiao, Q.; Cao, W.; Wong, H.-S.; and Wu, S. 2020. Model adaptation: Unsupervised domain adaptation without source data. In *Proc. CVPR*, 9641–9650.

Li, X.; Chen, W.; Xie, D.; Yang, S.; Yuan, P.; Pu, S.; and Zhuang, Y. 2021a. A free lunch for unsupervised domain adaptive object detection without source data. In *Proc. AAAI*, 8474–8481.

Li, X.; Li, J.; Zhu, L.; Wang, G.; and Huang, Z. 2021b. Imbalanced source-free domain adaptation. In *Proc. ACM MM*, 3330–3339.

Li, Y.; Jiang, Y.; Li, Z.; and Xia, S.-T. 2022. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1): 5–22.

Li, Y.; Li, Y.; Wu, B.; Li, L.; He, R.; and Lyu, S. 2021c. Invisible backdoor attack with sample-specific triggers. In *Proc. ICCV*, 16463–16472.

Li, Y.; Lyu, X.; Koren, N.; Lyu, L.; Li, B.; and Ma, X. 2021d. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *Proc. ICLR*.

Liang, J.; He, R.; and Tan, T. 2024. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 1–34.

Liang, J.; Hu, D.; and Feng, J. 2020. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proc. ICML*, 6028–6039.

Liang, J.; Hu, D.; Feng, J.; and He, R. 2021a. Umad: Universal model adaptation under domain and category shift. *arXiv preprint arXiv:2112.08553*.

Liang, J.; Hu, D.; Feng, J.; and He, R. 2022. Dine: Domain adaptation from single and multiple black-box predictors. In *Proc. CVPR*, 8003–8013.

Liang, J.; Hu, D.; Wang, Y.; He, R.; and Feng, J. 2021b. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 44(11): 8602–8617.

Liao, J.; Yi, L.; Shi, W.; Yang, W.; Fang, Y.; and Yang, X. 2024. Imperceptible backdoor watermarks for speech recognition model copyright protection. *Visual Intelligence*, 2(1): 23.

Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions and Defenses*, 273–294.

Liu, Y.; Zhang, W.; and Wang, J. 2021. Source-free domain adaptation for semantic segmentation. In *Proc. CVPR*, 1215–1224.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional adversarial domain adaptation. In *Proc. NeurIPS*.

Nguyen, A.; and Tran, A. 2021. WaNet–Imperceptible Warping-based Backdoor Attack. In *Proc. ICLR*.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment matching for multi-source domain adaptation. In *Proc. ICCV*, 1406–1415.

Poursaeed, O.; Katsman, I.; Gao, B.; and Belongie, S. 2018. Generative adversarial perturbations. In *Proc. CVPR*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *Proc. ICML*, 8748–8763.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *Proc. ECCV*, 213–226.

Saha, A.; Tejankar, A.; Koohpayegani, S. A.; and Pirsiavash, H. 2022. Backdoor attacks on self-supervised learning. In *Proc. CVPR*, 13337–13346.

Shejwalkar, V.; Lyu, L.; and Houmansadr, A. 2023. The perils of learning from unlabeled data: Backdoor attacks on semi-supervised learning. In *Proc. ICCV*, 4730–4740.

Sheng, L.; Liang, J.; He, R.; Wang, Z.; and Tan, T. 2023. AdaptGuard: Defending Against Universal Attacks for Model Adaptation. In *Proc. ICCV*.

Tan, S.; Peng, X.; and Saenko, K. 2020. Class-imbalanced domain adaptation: an empirical odyssey. In *Proc. ECCV Workshops*, 585–602.

Tian, J.; Zhang, J.; Li, W.; and Xu, D. 2021. VDM-DA: Virtual domain modeling for source data-free domain adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6): 3749–3760.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *Proc. CVPR*, 5018–5027.

Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; and Zhao, B. Y. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Proc. S&P*, 707–723.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; and Darrell, T. 2021. Tent: Fully test-time adaptation by entropy minimization. In *Proc. ICLR*.

Wu, B.; Chen, H.; Zhang, M.; Zhu, Z.; Wei, S.; Yuan, D.; and Shen, C. 2022. Backdoorbench: A comprehensive benchmark of backdoor learning. In *Proc. NeurIPS*, 10546–10559.

Wu, D.; and Wang, Y. 2021. Adversarial neuron pruning purifies backdoored deep models. In *Proc. NeurIPS*, 16913–16925.

Yang, S.; van de Weijer, J.; Herranz, L.; Jui, S.; et al. 2021. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *Proc. NeurIPS*, 29393–29405.

Yu, Y.; Sheng, L.; He, R.; and Liang, J. 2023. Benchmarking test-time adaptation against distribution shifts in image classification. *arXiv preprint arXiv:2307.03133*.

Yu, Y.; Sheng, L.; He, R.; and Liang, J. 2024. STAMP: Outlier-Aware Test-Time Adaptation with Stable Memory Replay. In *Proc. ECCV*, 375–392.

Zhang, J.; Huang, J.; Jiang, X.; and Lu, S. 2023a. Black-box Unsupervised Domain Adaptation with Bi-directional Atkinson-Shiffrin Memory. In *Proc. ICCV*, 11771–11782.

Zhang, Z.; Chen, W.; Cheng, H.; Li, Z.; Li, S.; Lin, L.; and Li, G. 2022. Divide and contrast: Source-free domain adaptation via adaptive contrastive learning. In *Proc. NeurIPS*, 5137–5149.

Zhang, Z.; Xiao, G.; Li, Y.; Lv, T.; Qi, F.; Liu, Z.; Wang, Y.; Jiang, X.; and Sun, M. 2023b. Red alarm for pre-trained models: Universal vulnerability to neuron-level backdoor attacks. *Machine Intelligence Research*, 20(2): 180–193.

Zheng, R.; Tang, R.; Li, J.; and Liu, L. 2022. Data-free backdoor removal based on channel lipschitzness. In *Proc. ECCV*, 175–191.