Appendices

Overview:

- In Appendix A, we present more results of the generalization impairments of NTL models on third-party domains.
- In Appendix B, we present detailed formula derivation.
- In Appendix C, we provide more details of the implementations of our experiments.
- In Appendix D, we present additional experiments and analyses for attacking NTL models.
- In Appendix E, we show more results of the proposed defense method.

A. More Empirical Results

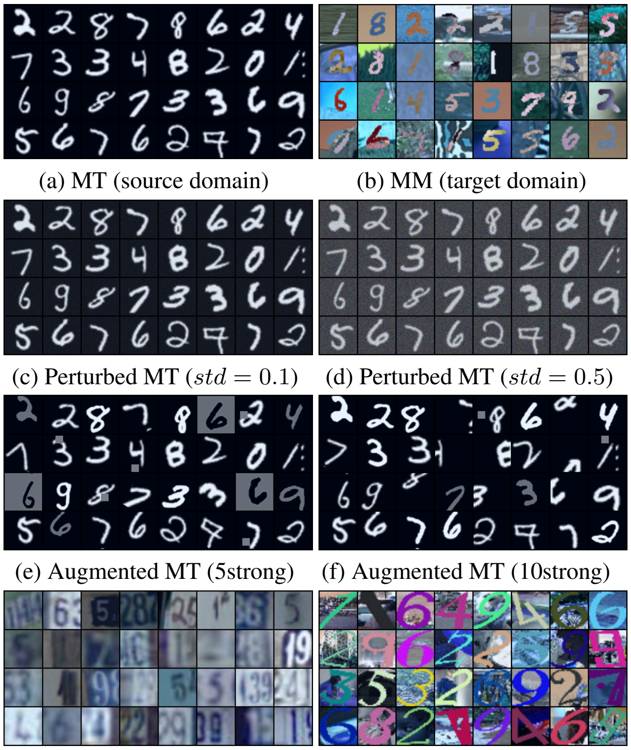
In this section, we present more empirical results to support our findings of the generalization impairments of NTL models. In Appendix A.1, we provide more details about the experiment of generalization impairments shown in our main paper. We also show the generalization impairments on other datasets. In Appendix A.2, we present more results on the two impairment patterns. In Appendix A.3, we present more results of the loss landscape of NTL models.

A.1. Generalization Impairments on Third-Party Domains

Complementary details for the experiment of generalization impairments. This paper starts by exploring the performance of target-specified NTL models on unseen *third-party domains*. In the experiment of Fig. 2 in the main paper, MNIST (MT) [10] and MNIST-M (MM) [13] are serviced as the source and target domain (see Fig. 5 (a-b)). SL model is trained on MT, and NTL methods (NTL [49] and CUTI [50]) are trained on MT and MM. Specifically, we involve three kinds of third-party domains:

- *Perturbed source domain*: we perturbed the source domain by adding Gaussian noise on the image with different standard deviations (i.e., $std = \{0.1, 0.5, 1.0, 2.0\}$). Typical perturbed images are shown in Fig. 5 (c-d)
- *Augmented source domain*: we augment the image by using RandAugment [8]. In detail, we show the results of weak augmentation (i.e., RandomCrop), 5 kinds of augmentations randomly sampled from the RandAugment, and 10 kinds of randomly sampled augmentations. Examples of the augmented images are shown in Fig. 5 (e-f).
- *Real domains collected from different environments.* We consider the SVHN (SN) [39] and SYN-D (SD) [43] as third-party domains collected from the real world. Images in the SN and SD are shown in Fig. 5 (g-h).

Generalization impairments on other datasets. In addition, to verify the wide existence of the generalization impairments of NTL methods, we also show the results on CIFAR10→STL10 and VisDA-T→VisDA-V in Fig. 6.


(a) MT (source domain) (b) MM (target domain)

(c) Perturbed MT ($std = 0.1$) (d) Perturbed MT ($std = 0.5$)

(e) Augmented MT (5strong) (f) Augmented MT (10strong)

(g) Read domain (SN) (h) Read domain (SD)

Figure 5. Images in each domain. (a-b) and (g-h) are four real domains, where (a) MT and (b) MM are the source domain and target domain in our experiments, respectively. (g) SN and (h) SD service as third-party domains collected from the real world. (c-d) show the perturbed source domain images. (e-f) show the augmented source domain images.

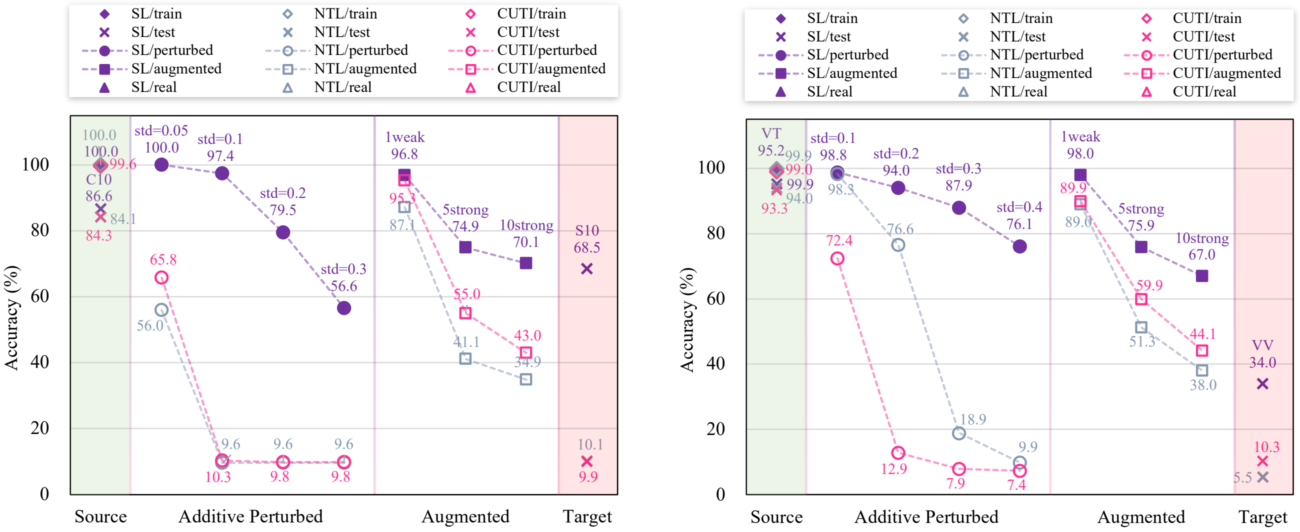
It is worth noting that for CIFAR10→STL10 and VisDA-T→VisDA-V, there are no more public-available real domains with the same labels but collected from different environments. Thus, we only show results on perturbed source domains and augmented source domains. The results in Fig. 6 further confirm that *the generalization of target-specified NTL models are impaired with varying degrees on third-party domains (compared to SL models)*.

A.2. Impairment Patterns

A.2.1 Over-confident prediction

Briefly, the pattern of "over-confident prediction" means that *NTL models exhibit over-confident predictions on third-party domains as well as the target domain.* In this section, we first show more results on the distribution of per-sample confidence. Besides, we show that NTL models will also make over-confident predictions on other datasets.

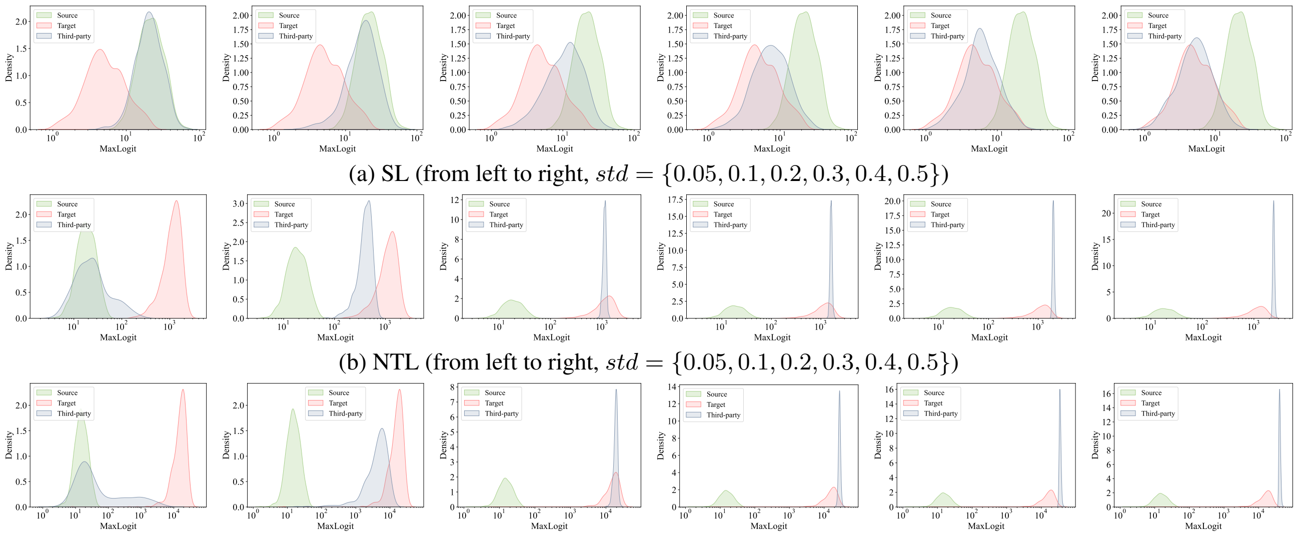
More results of confidence distribution. As shown in Fig. 7, we plot the distribution of prediction confidences on the source domain $\mathcal{D}_s$ (CIFAR10), the target domain $\mathcal{D}_t$ (STL10), and perturbation-based third-party domains



(a) CIFAR10 (C10) → STL10 (S10)

(b) VisDA-T (VT) → VisDA-V (VV)

Figure 6. The accuracies of SL and target-specified NTL (NTL [49] and CUTI [50]) on third-party domains, including pertubed source domains, augmented source domains. (a) Results on CIFAR10 (C10) → STL10 (S10). (b) Results on VisDA-T (VT) → VisDA-V (VV).



(a) SL (from left to right, $std = \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$)

(b) NTL (from left to right, $std = \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$)

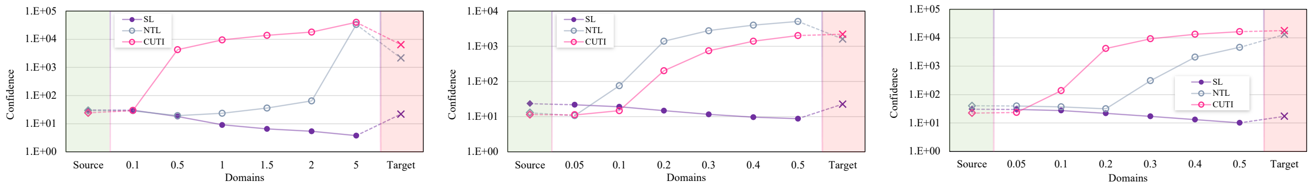
(c) CUTI (from left to right, $std = \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$)

Figure 7. Distribution of per-sample confidence of models on the source domain (CIFAR10), the target domain (STL10), and third-party domains. (a) SL model. (b) NTL [49] model. (c) CUTI [50] model. Domain-averaged confidence is shown in Fig. 3 (a) in the main paper.

$\{\hat{\mathcal{D}}_s^g\}_{g=1}^G$. Specifically, the perturbation is performed by adding Gaussian noise on the source domain images. We plot results on 6 perturbation-based third-party domains, with the standard derivation (std) of Gaussian noise $= \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

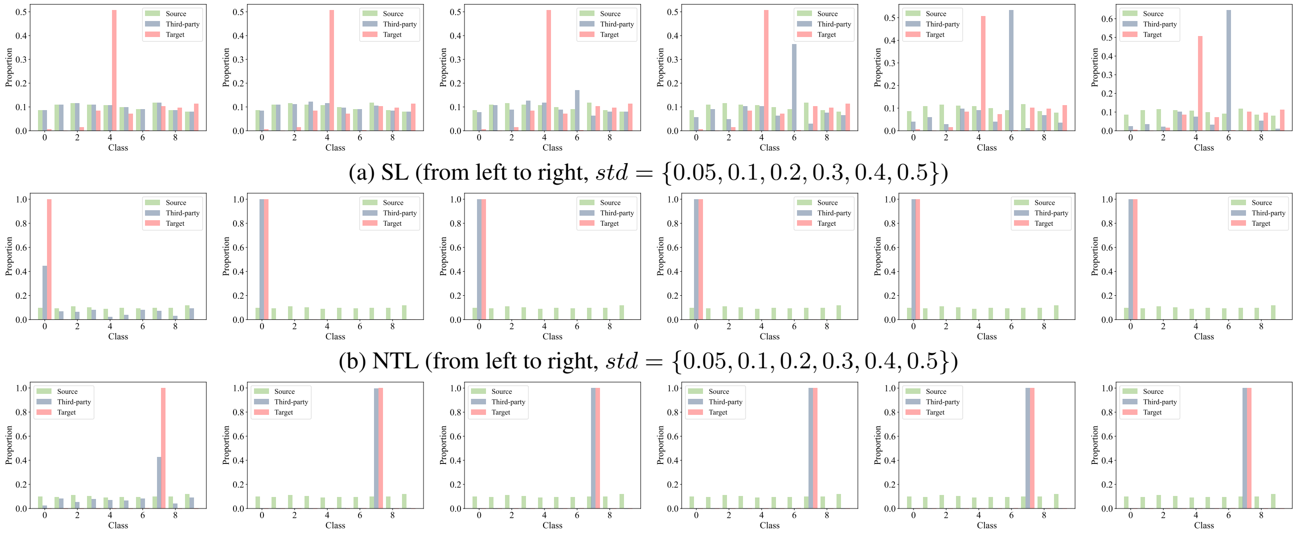
In Fig. 7 (a), we show results of the SL model trained on the source domain. We can see that the SL model makes lower confident predictions when facing unseen distribution shifts (e.g., the target domain and any third-party domains). In Fig. 7 (b-c), we show results of NTL models (NTL [49] and CUTI [50]). NTL models predict the target domain data with a significantly high confidence. Moreover, when facing unseen distribution shifts (i.e., perturbation-based third-party domains), NTL models also predict them with high confidence. Particularly, with the distribution shifts increasing (i.e., the std of the Gaussian noise increasing), NTL and CUTI predict the third-party domain data with more confidence. This is obviously opposite to the behavior of the SL model, which makes more unconfident predictions when the distribution shifts increase.

Results on more datasets. Furthermore, we present results on additional datasets to verify the existence of *over-*



(a) MNIST→MNIST-M (b) SVHN→SYN-D (c) VisDA-T→VisDA-V

Figure 8. Domain-averaged confidence of SL/NTL/CUTI models on the source domain, the target domain, and third-party domains. (a) Results on MNIST→MNIST-M. (b) Results on SVHN→SYN-D. (c) Results on VisDA-T→VisDA-V.



(a) SL (from left to right, $std = \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$)

(b) NTL (from left to right, $std = \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$)

(c) CUTI (from left to right, $std = \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$)

Figure 9. Prediction proportions of models on the source domain (CIFAR10), the target domain (STL10), and third-party domains. (a) SL model. (b) NTL [49] model. (c) CUTI [50] model.

confident prediction. In Fig. 8, we show the domain-averaged confidence of NTL models on MNIST→MNIST-M, SVHN→SYN-D, and VisDA-T→VisDA-V. We can see that NTL models also exhibit over-confident predictions on third-party domains as well as the target domain. These results confirm the wide existence of *over-confident prediction* in NTL models.

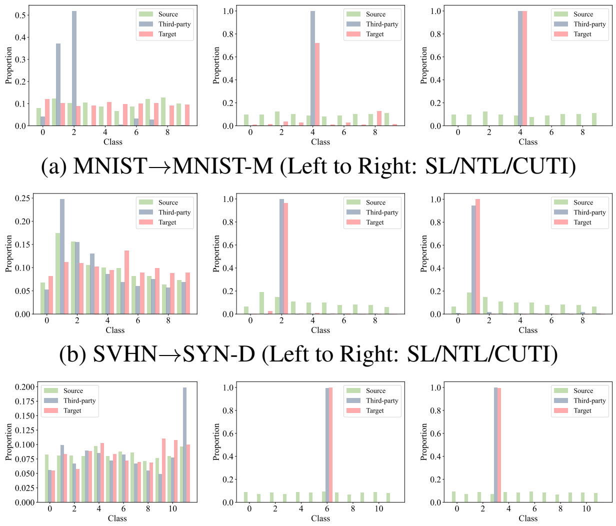
A.2.2 Implicit target domain class

Briefly, the pattern of "implicit target domain class" means that *NTL models tend to predict the "implicit target domain class" on third-party domains.* In this section, we first show more results on the proportion of different classes predicted by NTL models. Then, we show results on other datasets to further verify the existence of the pattern of "implicit target domain class".

More results of confidence distribution. As shown in Fig. 9, we plot the proportion of different classes predicted by SL and NTL models on the source domain $\mathcal{D}_s$ (CIFAR10), the target domain $\mathcal{D}_t$ (STL10), and perturbation-based third-party domains $\{\hat{\mathcal{D}}_s^g\}_{g=1}^G$. The same as Appendix A.2.1, the perturbation is performed by adding Gaussian noise on the source domain images, and we plot results on 6 perturbation-based third-party domains, with the standard derivation (std) of Gaussian noise $= \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

In Fig. 9 (a), we show results of the SL model trained on the source domain. We can see that the SL model makes diverse predictions for the data in third-party domains and the target domain. In Fig. 9 (b-c), we show results of NTL models (NTL [49] and CUTI [50]). These NTL models, although trained in a maximization term on the target domain (Eq. (1) in the main paper), predict all the target domain data to one class (denoted as the *implicit target-domain class*). Moreover, for the third-party domain obtained by slightly perturbing the source domain, the NTL model also tends to predict the label of the *implicit target-domain class*.

Results on different datasets. We also show results on other datasets, thus further verifying the impairment pattern of *implicit target-domain class*. In Fig. 10, we show the prediction proportions of models on MNIST→MNIST-M, SVHN→SYN-D, and VisDA-T→VisDA-V. From the results, it is widely existent that NTL models tend to predict the *implicit target-domain class* on third-party domains.



(a) MNIST→MNIST-M (Left to Right: SL/NTL/CUTI)

(b) SVHN→SYN-D (Left to Right: SL/NTL/CUTI)

(c) VisDA-T→VisDA-V (Left to Right: SL/NTL/CUTI)

Figure 10. Prediction proportions of models on the source domain, the target domain, and third-party domains. (a) Results on MNIST→MNIST-M. (b) Results on SVHN→SYN-D. (c) Results on VisDA-T→VisDA-V. Each row contains the results of the SL model, the NTL model, and the CUTI model (from left to right).

A.3. Loss Landscapes

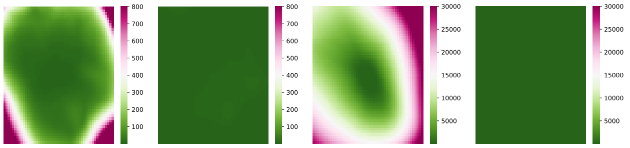
To explore the causes of the above-mentioned two impairment patterns, we separately explore the generalization of the source domain learning task $\mathcal{T}_{\text{src}}$ and the non-transferable task $\mathcal{T}_{\text{tgt}}$ by plotting the loss landscape on the source domain and the target domain, respectively. In the main paper, we show the results on CIFAR10→STL10 (i.e., the Fig. 4 in the main paper). In this section, we show more evidence across different datasets to confirm our findings. The loss landscapes of NTL models on MNIST→MNIST-M, SVHN→SYN-D, and VisDA-T→VisDA-V are shown in Fig. 11, Fig. 12, and Fig. 13, respectively. We can see that across different datasets, NTL models (either NTL or CUTI) are always optimized to an extremely sharp minima on the source domain (the left landscape in each subfigure), but are optimized to a relatively flat minima on the target domain (the right landscape in each subfigure). These results further support the explanation that due to the dominant generalization of non-transferable task, NTL models tend to make target-domain-consistent predictions on third-party domains.

B. Formulation Derivation

The total objective of TransNTL can be formulated as the following bi-level optimization problem:

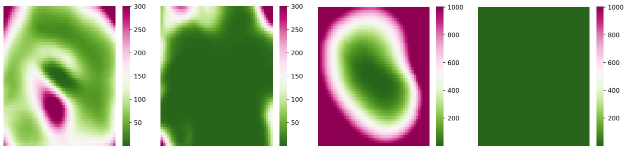
$$\min_{\theta} \max_{\|T_\theta^{-1}\epsilon\|_2 \le \rho} \mathcal{L}_{\text{irft}}(\theta + \epsilon), \tag{12}$$

where $\theta \in \mathbb{R}^d$ is the model parameters, $\epsilon \in \mathbb{R}^d$ is the perturbation, ρ is a hyper-parameter to control the perturbation magnitude, and the $T_\theta = \text{diag}(|\theta_1|, |\theta_2|, \dots, |\theta_d|) \in \mathbb{R}^{d\times d}$ is introduced to set element-wise adaptive weight for each parameter in θ [31, 61].



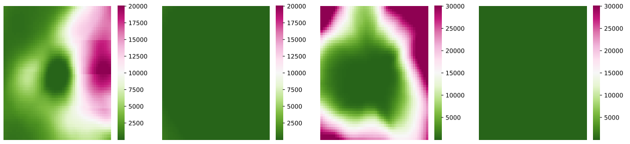
(a) NTL (L: source, R: target) (b) CUTI (L: source, R: target)

Figure 11. Loss landscapes of models on MNIST→MNIST-M.



(a) NTL (L: source, R: target) (b) CUTI (L: source, R: target)

Figure 12. Loss landscapes of models on SVHN→SYN-D.



(a) NTL (L: source, R: target) (b) CUTI (L: source, R: target)

Figure 13. Loss landscapes of models on VisDA-T→VisDA-V.

To solve the minimax problem in Eq. (12), we first follow [31] to find the optimal ϵ to maximize the $\mathcal{L}_{\text{irft}}(\theta + \epsilon)$:

$$\epsilon^* = \underset{\|T_\theta^{-1}\epsilon\|_2 \le \rho}{\arg\max}\ \mathcal{L}_{\text{irft}}(\theta + \epsilon). \tag{13}$$

By letting $\tilde{\epsilon} = T_\theta^{-1}\epsilon$, the Eq. (13) can be converted to:

$$\tilde{\epsilon}^* = \underset{\|\tilde{\epsilon}\|_2 \le \rho}{\arg\max}\, \mathcal{L}_{\text{irft}}(\theta + T_\theta\tilde{\epsilon}). \tag{14}$$

Specifically, to solve the maximization problem, we can approximate the $\mathcal{L}_{\text{irft}}(\theta + T_\theta\tilde{\epsilon})$ by using a first-order Taylor expansion [27, 31, 61], thus we have:

$$\begin{aligned} \tilde{\epsilon}^* &\approx \underset{\|\tilde{\epsilon}\|_2 \le \rho}{\arg\max}\, \mathcal{L}_{\text{irft}}(\theta) + \tilde{\epsilon}^\top T_\theta \nabla \mathcal{L}_{\text{irft}}(\theta) \\ &= \underset{\|\tilde{\epsilon}\|_2 \le \rho}{\arg\max}\, \tilde{\epsilon}^\top T_\theta \nabla \mathcal{L}_{\text{irft}}(\theta) \\ &= \rho \frac{T_\theta \nabla \mathcal{L}_{\text{irft}}(\theta)}{\|T_\theta \nabla \mathcal{L}_{\text{irft}}(\theta)\|_2}. \end{aligned} \tag{15}$$

Finally, the optimal perturbation ϵ can be derived as:

$$\epsilon^* = T_\theta \tilde{\epsilon} = \rho \frac{T_\theta^2 \nabla \mathcal{L}_{\text{irft}}(\theta)}{\|T_\theta \nabla \mathcal{L}_{\text{irft}}(\theta)\|_2}, \tag{16}$$

which is the Eq. (9) in the main paper.

C. Complementary Experimental Details

In this section, we provide complementary experimental details. Appendix C.1 contains introduction of datasets. In

Appendix C.2, we present more implementation details, including NTL baselines, backbones, NTL pre-training details, attacking details, evaluation metrics, and running environments. The implementations of NTL-based IP protection are provided in Appendix C.3.

C.1. Datasets

By following [49, 50], we conduct experiments on (1) *Digits*, (2) *CIFAR10 & STL10*, and (3) *VisDA-2017*. We provide more details of each dataset:

- ***Digits:*** We involve four Digit datasets, including: MNIST [10], MNIST-M [13], SVHN [39], and SYN-D [43]. Each dataset contains 10 digits collected from real scenes or artificially constructed.
- ***CIFAR10 & STL10:*** Both CIFAR10 [30] and STL10 [7] are 10-class classification datasets, which contain 6 animal categories and 4 transportation categories. Following [49, 50], we run experiments on both CIFAR10→STL10 and STL10→CIFAR10.
- ***VisDA-2017:*** VisDA-2017 [42] contains a training set VisDA-T and a validation set VisDA-V of 12 object categories. Following [49, 50], we consider the non-transferable task from VisDA-T to VisDA-V.

C.2. Implementation Details

NTL Baselines. We involve all of the previously proposed NTL methods as baselines, including the first method: NTL [49], and the state-of-the-art (SOTA) method: CUTI [50]. We also use supervised learning (SL) as a baseline. We provide brief introductions for each baseline:

- **SL:** We use a standard supervised learning pipeline with cross-entropy loss.
- **NTL:** NTL [49] adds two statistical dependence relaxation terms on standard supervised learning to resist transferability: (i) maximizing the Kullback-Leible (KL) divergence between target domain representation and label, and (ii) maximizing the maximum mean discrepancy (MMD) between the distribution of source and target domain representations.
- **CUTI:** CUTI [50] improves the NTL by introducing style transfer [23]. They augment target domain images by transferring their styles to the source domain style, thus obtaining a CUTI-domain. Then, CUTI trains a model by maximizing the KL divergence between labels and the representations on both the target domain and the CUTI-domain.

Backbones. We follow the same backbones as previous NTL methods [49, 50]. Specifically, we apply VGG-11 [45] for Digits datasets, VGG-13 for CIFAR10 & STL10, and VGG-19 for VisDA. All backbones are initialized as the pre-trained version of ImageNet-1K [9]. Besides, in Appendix D.3, we adopt the experiments on more backbones and analyze the influence of backbones for NTL.

NTL pre-training details. Following Wang et al. [49], we randomly select 8,000 samples as training data and 1,000 samples as testing data. All images are resized to 64×64. For training SL models, we employ the SGD as an optimizer with lr = 0.001 and set the batch size to 32. For training NTL [49] and CUTI [50] models, we use their released code and the same hyperparameters settings reported in their paper. Besides, the detailed implementations of NTL-based ownership verification and applicability authorization are illustrated in Appendix C.3.

Attacking details. For attacking NTL models, we train the proposed TransNTL up to 200 epochs, with Adam serving as an optimizer. We set batchsize = 32 and learning rate = 0.0001. The self-distillation weight λ_{sd} is set to 0.2, and the magnitude ρ is set to 0.5. For simplicity, we only consider a perturbation collection $\mathcal{P}$ with three perturbation functions: $\mathcal{P} = \{p_1, p_2, p_3\}$, where $p_1(x) = x + \delta_1$, $p_2(x) = x \odot (1 + \delta_2)$, and $p_3(x) = x \otimes k$. Specifically, δ_1 and $\delta_2 \sim \mathcal{N}(0, 0.1)$, and k is a Gaussian kernel with size = 5. The influences of each hyper-parameter are shown in Appendix D.1.

We also seek possible attack methods for comparison. We involve SOTA backdoor defense methods and watermark removal methods, including: FTAL [1], RTAL [1], FP [37], NAD [35], i-BAU [57], and FT-SAM [61]. We re-implement the FTAL/RTAL and follow the implementations in [54] for other methods. All attack baselines are also trained up to 200 epochs.

We mainly focus on the setting that all attacking methods can access 10% source domain data. The results of other proportions of available data are shown in Appendix D.2.

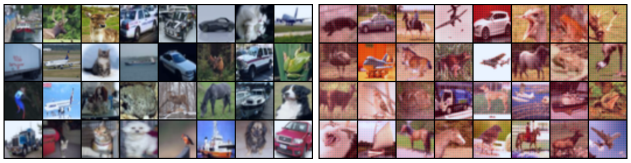
Evaluation metric. We show Top-1 classification accuracy (Acc) on the source domain and the target domain. We also calculate the accuracy drop (compared to the pre-trained model) on the source domain and the target domain.

Environment. Our code is implemented in Python 3.10.12 and PyTorch 2.0.1. All experiments are conducted on a server running Ubuntu 20.04 LTS, equipped with an NVIDIA GeForce RTX 4090 GPU.

C.3. NTL-based IP Protection

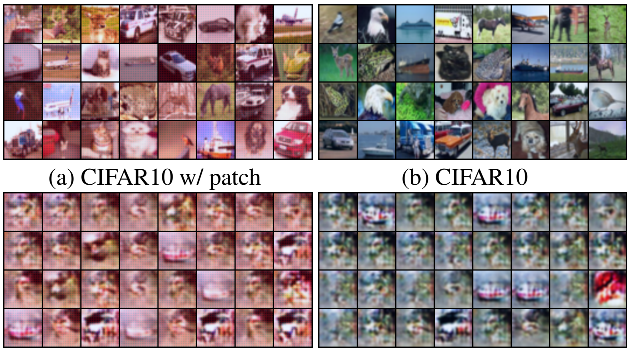
In this section, we introduce the detailed implementations of NTL-based IP protection (i.e., ownership verification and applicability authorization).

NTL-based ownership verification. Ownership verification aims to verify the ownership of a deep learning model [6, 32, 49]. NTL methods [49, 50] provide solutions for ownership verification by triggering misclassification on the target domain. Specifically, we add a pre-defined trigger patch (only known by the model owner) on the original dataset and see them as the target domain (e.g., Fig. 14 (b)). The original data without the patch is regarded as the source



(a) CIFAR10 (b) CIFAR10 w/ patch

Figure 14. An example of NTL-based ownership verification. (a) The original source data (e.g., CIFAR10) is regarded as the source domain. (b) We add a pre-defined trigger patch on the source data and see them as the target domain (e.g., CIFAR10 w/ patch).



(a) CIFAR10 w/ patch (b) CIFAR10

(c) AugCIFAR10 w/ patch (d) AugCIFAR10

Figure 15. An example of NTL-based applicability authorization. (a) The original source data with the pre-defined trigger patch (e.g., CIFAR10 w/ patch) is regarded as the source domain. (b-d) We see the union of the original data (CIFAR10), the augmented data with the patch (AugCIFAR10 w/ patch), and the augmented data without the patch (AugCIFAR10) as the target domain.

domain (e.g., Fig. 14 (a)). It is worth noting that such a pre-defined trigger patch can be controlled to be shallow so that normal SL models trained on the original source domain can still have good performance on the patched source domain. Then, we train a deep learning model by using NTL method [49, 50] on these two domains. After that, the trained model will perform poorly on the data with the patch but have a good performance on the data without the patch. Thus, by observing the performance difference of a trained model on the source domain data with and without the pre-defined trigger patch, we can verify whether a deep learning model belongs to the model owner.

NTL-based applicability authorization. Applicability authorization aims at authorizing models to certain data for preventing their usage on unauthorized data [49], which can be solved by applying source-only NTL method [49, 50] to restrict the model generalization ability to only the authorized domain. Specifically, we add a pre-defined authorized patch to the original data and see them as the source domain (e.g., Fig. 15 (a)). We regard *the union of the original data (without the authorized patch), the augmented original data with and without the authorized patch* as the target domain (e.g., Fig. 15 (b-d)). Then, we train a deep learning model by using NTL method [49, 50] on these two domains. After that, the trained model will only perform well on the authorized data (i.e., the original data with the authorized patch). For unauthorized data (e.g., the original data without the authorized patch, the data from other domains with or without the authorized patch), the trained model has a poor performance. Therefore, we achieve the model applicability authorization.

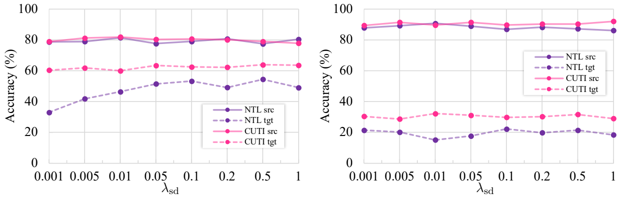
D. Additional Experimental Results

This section contains additional experiments and analyses. In Appendix D.1, we analyze the influence of main hyperparameters in the proposed TransNTL. In Appendix D.2, we conduct experiments on attacking NTL models by using fewer source domain data. Appendix D.3 analyses the influence of backbones to NTL models. In Appendix D.4, we show visualization results of attacked NTL models, including confidence distributions, prediction proportions, and t-SNE feature visualizations.

D.1. Influences of Hyperparameters

In this section, we analyze the influence of major hyperparameters in the proposed TransNTL, including the self-distillation weight λ_{sd}, the magnitude ρ, and the perturbation functions $\mathcal{P}$.

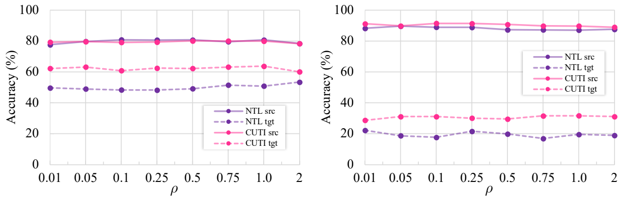
Self-distillation weight. The self-distillation weight λ_{sd} controls the importance of the loss term $\mathcal{L}_{\text{sd}}$ in the impairment-repair fine-tuning framework. We change the value of λ_{sd} from 0.001 to 1 and conduct experiments on CIFAR10→STL10 and VisDA-T→VisDA-V, thus analyzing its influence. As shown in Fig. 16, a too-small λ_{sd} value will limit the recoverment of target domain performance (e.g, NTL and CUTI on CIFAR10→STL10). For VisDA-T→VisDA-V, the performance of TransNTL is generally not sensitive to the value of λ_{sd}.



(a) CIFAR10→STL10 (b) VisDA-T→VisDA-V

Figure 16. Influence of the self-distillation weight λ_{sd}.

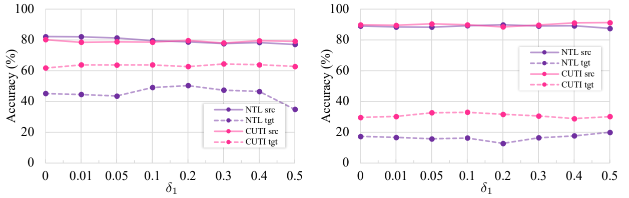
Magnitude. The hyperparameter ρ controls the magnitude of network-parameter-perturbations in the sharpness term $\mathcal{L}_{\text{sharp}}$. To explore its influence, we conduct experiments on CIFAR10→STL10 and VisDA-T→VisDA-V with different values of ρ. The results are shown in Fig. 17. In general, TransNTL is not sensitive to the value of ρ (especially on VisDA-T→VisDA-V). On CIFAR10→STL10, a too-large value of ρ will slightly degrade the source domain performance of both NTL and CUTI.



(a) CIFAR10→STL10 (b) VisDA-T→VisDA-V

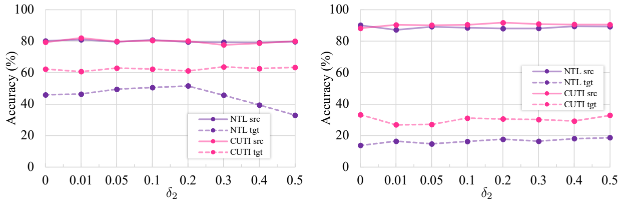
Figure 17. Influence of the magnitude ρ.

Perturbation functions. The collection $\mathcal{P}$ contains a group of perturbation functions. These perturbation functions are used for perturbing the source domain, thus obtaining third-party domains for repairing impairments. For simplicity, the perturbation collection $\mathcal{P}$ in our main experiments contains only three perturbation functions: $\mathcal{P} = \{p_1, p_2, p_3\}$, where $p_1(x) = x + \delta_1$, $p_2(x) = x \odot (1 + \delta_2)$, and $p_3(x) = x \otimes k$. Specifically, δ_1 and δ_2 are Gaussian noise and k is a Gaussian kernel. We analyze the influence of parameters in these perturbation functions, i.e., the standard derivation of Gaussian noise and the kernel size. Results are shown in Fig. 18, Fig. 19, and Fig. 20. We can see that CIFAR10→STL10 is more sensitive to VisDA-T→VisDA-V. When we set a too large value for each perturbation, the performance of TransNTL will be limited (especially for NTL [49]). This is because a too-large perturbation leads to an over-worst third-party domain, which is not necessary and even has negative influences.



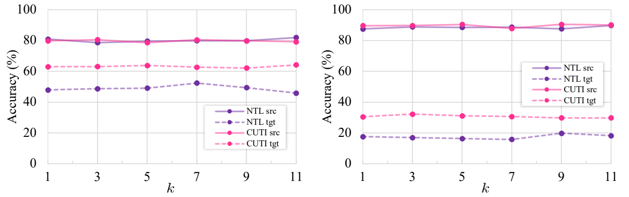
(a) CIFAR10→STL10 (b) VisDA-T→VisDA-V

Figure 18. Influence of the additive perturbation (δ_1).



(a) CIFAR10→STL10 (b) VisDA-T→VisDA-V

Figure 19. Influence of the multiplicative perturbation (δ_2).



(a) CIFAR10→STL10 (b) VisDA-T→VisDA-V

Figure 20. Influence of the convolutional perturbation (k).

Table 6. Transferring the NTL with fewer source domain data. The source domain accuracy and target domain accuracy (%) are reported. The accuracy drop compared to the pre-trained model is shown in brackets. The best result is highlighted in **bold**.

(a) **5%** source domain data are available for attack

	CIFAR10→STL10 (SL: 86.6 / 68.5)		**VisDA-T→VisDA-V** (SL: 95.2 / 34.0)	
	NTL	**CUTI**	**NTL**	**CUTI**
Pre-train	84.1 / 10.1	84.3 / 9.9	94.0 / 5.5	93.3 / 10.3
FTAL	83.9 (-0.2) 10.1 (+0.0)	83.7 (-0.6) 9.9 (+0.0)	91.8 (-2.2) 5.6 (+0.1)	92.8 (-0.5) 10.4 (+0.1)
RTAL	80.5 (-3.6) 10.1 (+0.0)	80.8 (-3.5) 9.9 (+0.0)	91.5 (-2.5) 8.3 (+2.8)	90.6 (-2.7) 11.0 (+0.7)
FP	80.4 (-3.7) 10.1 (+0.0)	82.0 (-2.3) 9.7 (-0.2)	90.1 (-3.9) 6.4 (+0.9)	90.2 (-3.1) 12.3 (+2.0)
NAD	79.2 (-4.9) 17.3 (+7.2)	82.2 (-2.1) 9.9 (+0.0)	9.0 (-85.0) 7.7 (+2.2)	88.4 (-4.9) 18.5 (+8.2)
i-BAU	82.3 (-1.8) 10.1 (+0.0)	81.7 (-2.6) 10.9 (+1.0)	85.7 (-8.3) 5.6 (+0.1)	89.0 (-4.3) 10.7 (+0.4)
FT-SAM	78.3 (-5.8) 20.2 (+10.1)	54.7 (-29.6) 43.0 (+33.1)	8.5 (-85.5) 8.4 (+2.9)	9.0 (-84.3) 7.5 (-2.8)
TransNTL (Ours)	**80.1 (-4.0)** **41.5 (+31.4)**	**77.2 (-7.1)** **60.2 (+50.3)**	**87.5 (-6.5)** **18.3 (+12.8)**	**89.8 (-3.5)** **29.6 (+19.3)**

(b) **1%** source domain data are available for attack

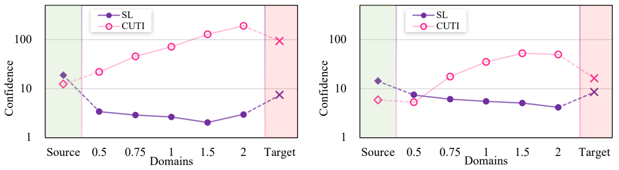
	CIFAR10→STL10 (SL: 86.6 / 68.5)		**VisDA-T→VisDA-V** (SL: 95.2 / 34.0)	
	NTL	**CUTI**	**NTL**	**CUTI**
Pre-train	84.1 / 10.1	84.3 / 9.9	94.0 / 5.5	93.3 / 10.3
FTAL	83.5 (-0.6) 10.1 (+0.0)	83.8 (-0.5) 9.9 (+0.0)	93.4 (-0.6) 5.6 (+0.1)	92.3 (-1.0) 10.3 (+0.0)
RTAL	65.5 (-18.6) 10.1 (+0.0)	38.8 (-45.5) 10.1 (+0.2)	85.5 (-8.5) 5.9 (+0.4)	84.1 (-9.2) 9.4 (-0.9)
FP	79.3 (-4.8) 10.5 (+0.4)	81.0 (-3.3) 10.1 (+0.2)	23.8 (-70.2) 9.2 (+3.7)	88.1 (-5.2) 11.4 (+1.1)
NAD	76.8 (-7.3) 11.3 (+1.2)	82.8 (-1.5) 9.9 (+0.0)	9.0 (-85.0) 7.7 (+2.2)	64.1 (-29.2) 15.1 (+4.8)
i-BAU	72.7 (-11.4) 10.2 (+0.1)	74.5 (-9.8) 9.9 (+0.0)	72.4 (-21.6) 5.5 (+0.0)	80.4 (-12.9) 11.5 (+1.2)
FT-SAM	75.7 (-8.4) 24.6 (+14.5)	78.6 (-5.7) 17.7 (+7.8)	8.2 (-85.8) 9.3 (+3.8)	7.5 (-85.8) 7.2 (-3.1)
TransNTL (Ours)	**79.0 (-5.1)** **33.1 (+23.0)**	**76.0 (-8.3)** **57.2 (+47.3)**	**76.0 (-18.0)** **12.9 (+7.4)**	**83.6 (-9.7)** **26.2 (+15.9)**

D.2. Attack with Fewer Source Domain Data

In this section, we analyze the influence of the amount of available source domain data for attacking. The results of using 5% and 1% source domain data to attack NTL models are shown in Tab. 6 (a) and (b), respectively. We can see that when fewer source domain data are available, all attacking methods face significant performance degradation. Most attack baselines totally fail to recover the target domain performance with fewer source domain data available. Compared to attack baselines, the proposed TransNTL still reaches the best attacking performance. However, TransNTL also inevitably has poor attacking performance. Specifically, TransNTL sacrifices more source domain per-

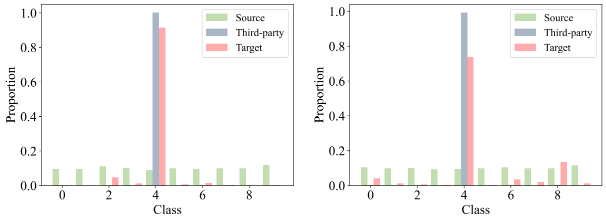
Table 7. Transfering the NTL with different backbones. We assume 10% source domain data is available for attack. We report the source domain accuracy (%) in blue and target domain accuracy (%) in red. The accuracy drop compared to the pre-trained model is shown in brackets.

	CIFAR10→STL10		**VisDA-T→VisDA-V**	
	VGG13bn	**ResNet34**	**VGG19bn**	**ResNet50**
SL	87.1 / 65.1	82.8 / 60.3	95.4 / 17.0	94.2 / 21.5
CUTI	87.6 / 14.5	82.4 / 10.1	96.5 / 10.6	89.4 / 9.6
TransNTL	85.3 (-2.3) 66.2 (+51.7)	80.7 (-1.7) 60.3 (+50.2)	92.1 (-4.4) 28.9 (+18.3)	89.2 (-0.2) 23.6 (+14.0)



(a) VGG13bn (b) ResNet34

Figure 21. Domain-averaged confidence of models with different backbones on the source domain (CIFAR10), the target domain (STL10), and third-party domains.



(a) VGG13bn (b) ResNet34

Figure 22. Prediction proportions of models with different backbones on the source domain (CIFAR10), the target domain (STL10), and a third-party domain.

formance to recover the target domain performance, and the recovered target domain accuracies are lower than using 10% data for attacking. This is because when fewer data are available, TransNTL is more easily overfitting to the fewer source domain data and the derived third-party domains, thus limiting both the source-domain-performance maintenance and target-domain-performance recovering.

D.3. Influence of Backbone

Previous NTL methods [49, 50] mainly use VGG [45] as backbone, and thus, we follow them to conduct major experiments with VGG. In this section, we additionally explore the influence of backbone to NTL pre-training and attacking. We consider two additional backbones: VGG with batch normalization [28] (VGG-bn), and ResNet [16].

It is worth noting that through our experiments, VGG-bn and ResNet trained by the NTL method [49] cannot reach the expected performance. It fails to simultaneously maintain the source domain performance and degrade the target domain performance. Thus, we only consider CUTI [50] in this section. The results of pre-trained SL and CUTI models are shown in Tab. 7. We further explore the generalization impairments and find that CUTI with VGG-bn or ResNet still exhibits the same impairment patterns as we observed on CUTI with VGG. Specifically, as shown in Fig. 21, CUTI models still exhibit over-confident predictions on third-party domains as well as the target domain. One difference is that the prediction confidence of CUTI models are significantly suppressed by the batch normalization layers in VGG-bn and ResNet. Moreover, as shown in Fig. 22, CUTI models also tend to predict the "implicit target domain class" on third-party domains.

The attacking results on CIFAR10→STL10 and VisDA-T→VisDA-V are shown in Tab. 7. We can see that the proposed TransNTL is effective for attacking CUTI with different backbones. These results further indicate the risk of using CUTI in practical IP protection.

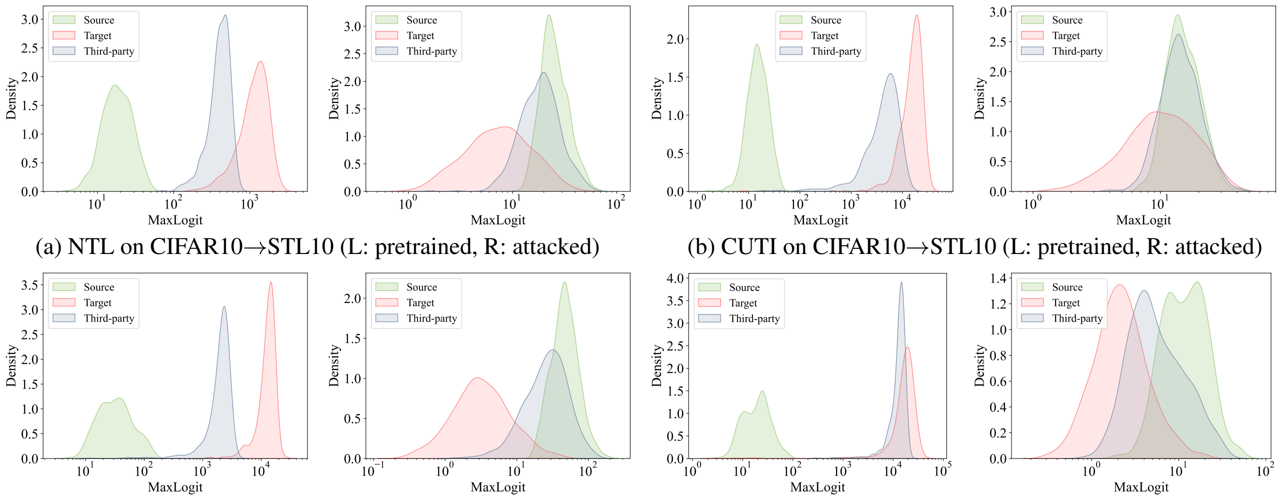
D.4. Visualization Results

In this section, we present visualization results of attacked NTL models, including the confidence distribution, prediction proportions, and t-SNE feature visualization.

Confidence distribution. We first analyze the confidence distribution of attacked NTL models. As shown in Fig. 23, we plot the confidence distribution of both pretrained and attacked NTL models (NTL [49] and CUTI [50]) on CIFAR10→STL10 and VisDA-T→VisDA-V. We can see that pretrained NTL models, as we mentioned in Appendix A.2, predict the third-party domain and the target domain with more confidence. In contrast, after being attacked by TransNTL, such abnormal behaviors of NTL models are repaired, with both the target domain and third-party domain being predicted with lower confidence than the source domain. The attacked NTL models thus behave more like normal SL models (normal SL models, as shown in Fig. 7 (a), make lower confident predictions when facing unseen distribution shifts).

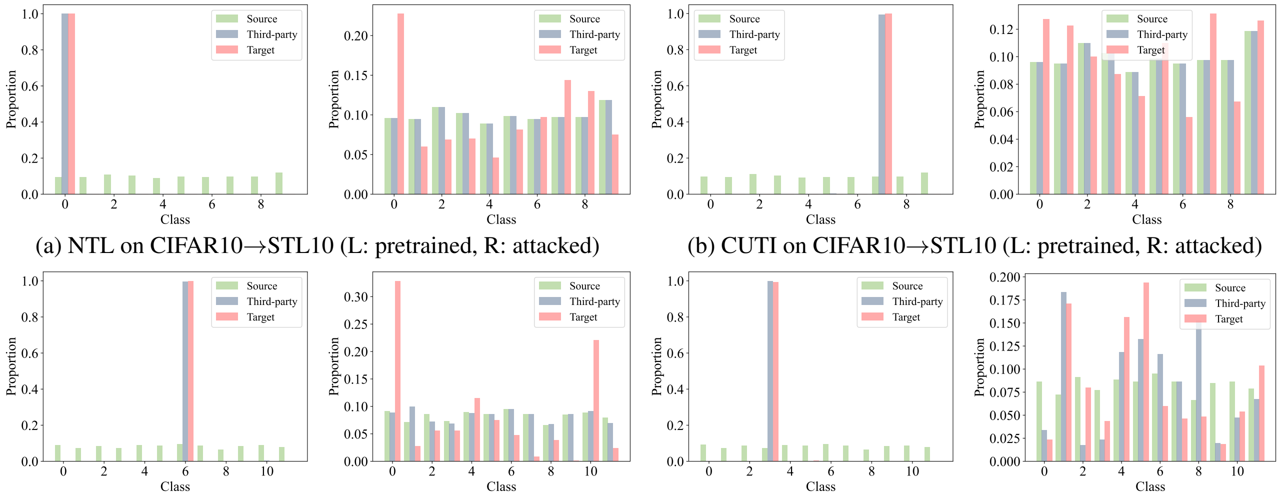
Prediction proportion. Another impairment pattern of NTL models is the *implicit target domain class*. In Fig. 24, we plot the proportion of different classes predicted by pretrained and attacked NTL models (NTL [49] and CUTI [50]) on CIFAR10→STL10 and VisDA-T→VisDA-V. After being attacked by TransNTL, we can see that the pattern of predicting the implicit target domain class on third-party domains is well-repaired. The attacked models make diverse predictions on third-party domains and the target domain, which is more similar to normal SL models.

t-SNE feature visualization. In addition, we plot the t-SNE visualization [47] of SL models (Fig. 25) and NTL models (Fig. 26 for CIFAR10→STL10 and Fig. 27 for



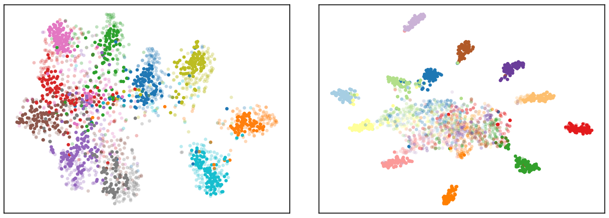
(a) NTL on CIFAR10→STL10 (L: pretrained, R: attacked)
(b) CUTI on CIFAR10→STL10 (L: pretrained, R: attacked)
(c) NTL on VisDA-T→VisDA-V (L: pretrained, R: attacked)
(d) CUTI on VisDA-T→VisDA-V (L: pretrained, R: attacked)

Figure 23. Prediction proportions of models on the source domain, the target domain, and third-party domains.



(a) NTL on CIFAR10→STL10 (L: pretrained, R: attacked)
(b) CUTI on CIFAR10→STL10 (L: pretrained, R: attacked)
(c) NTL on VisDA-T→VisDA-V (L: pretrained, R: attacked)
(d) CUTI on VisDA-T→VisDA-V (L: pretrained, R: attacked)

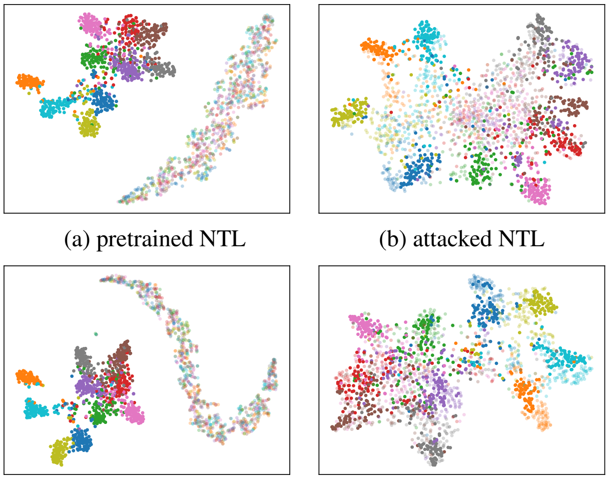
Figure 24. Prediction proportions of models on the source domain, the target domain, and third-party domains.



(a) CIFAR10→STL10
(b) VisDA-T→VisDA-V

Figure 25. t-SNE visualization of SL models. Opaque dots represent source domain features, and transparent dots represent target domain features.

VisDA-T→VisDA-V). Particularly, different color represents different class. Opaque dots in each figure represent source domain features, and transparent dots represent target domain features. As shown in Fig. 25, the target domain features of SL models overlap with the source domain features and maintain a certain discriminability. For pretrianed NTL models, as shown in Fig. 26 (a)(c) and Fig. 27 (a)(c), the source and target domain features are clearly separated by a certain distance. The source domain features are discriminable, but the target domain features are randomly distributed. Thus, the source-to-target generalization is limited, and the target domain performance is degraded to the random-classification accuracy. After being attacked, as shown in Fig. 26 (b)(d), Fig. 27 (b)(d), the transferability barriers are successfully broken. Due to the impairment repairing, the target domain features are re-distributed around the source domain features. Like SL models, these target domain features are overlapped with the source domain features and maintain a certain discriminability. Thus, the target domain performance is recovered.



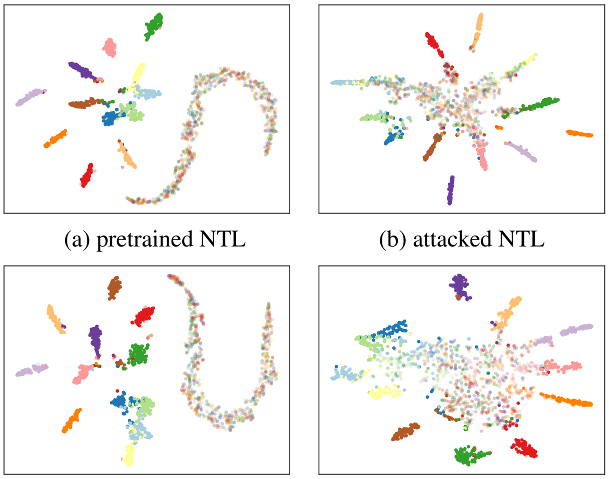
(a) pretrained NTL

(b) attacked NTL

(c) pretrained CUTI

(d) attacked CUTI

Figure 26. t-SNE visualization of NTL models (NTL [49] and CUTI [50]) on CIFAR10→STL10. Opaque dots in each figure represent source domain features, and transparent dots represent target domain features.



(a) pretrained NTL

(b) attacked NTL

(c) pretrained CUTI

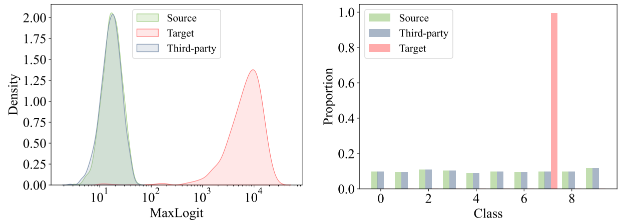
(d) attacked CUTI

Figure 27. t-SNE visualization of NTL models (NTL [49] and CUTI [50]) on VisDA-T→VisDA-V. Opaque dots in each figure represent source domain features, and transparent dots represent target domain features.

E. More Analyses of Defending

In this section, we present more visualization results of the proposed defending method. Briefly, the proposed defense method aims at pre-fixing the identified bugs by leveraging TransNTL in NTL training, thus eliminating the risk of NTL models in practical deployment.

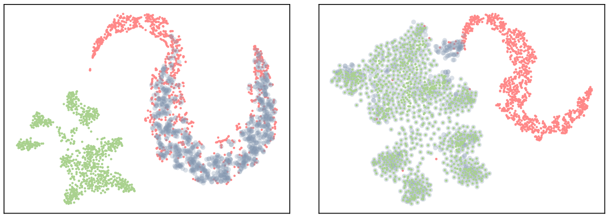
Following the defending experiments in the main paper (i.e., Tab. 5 in the main paper), we consider the CUTI [50] and its robust version obtained by our defense strategy: R-CUTI. In Fig. 28, we plot the confidence distribution and predicted proportion of R-CUTI on each domain. We can see that both impairment patterns on third-party domains are pre-repaired. Specifically, to implement the non-transferable learning, R-CUTI still predicts the target domain data to one class with high confidence, but for third-party domain data, R-CUTI has SL-like normal predictions. Moreover, the t-SNE visualization of features in each domain is shown in Fig. 29 (in this figure, different domain has different color). For the original CUTI (Fig. 29 (a)), the distribution of the third-party domain is overlapped with the target domain, thus leaving bugs for the third-party-domain-based TransNTL attack. In contrast, as shown in Fig. 29 (b), the R-CUTI pre-fix such bugs, with the target domain distribution being consistent with the source domain. As a result, the R-CUTI can effectively resist the TransNTL attack.



(a) CIFAR10→STL10

(b) VisDA-T→VisDA-V

Figure 28. Visualization of (a) confidence distribution (b) prediction proportion of R-CUTI on the source domain, the target domain, and third-party domains



(a) CUTI

(b) R-CUTI

Figure 29. t-SNE visualization of (a) CUTI and (b) R-CUTI on the source domain, the target domain, and third-party domains.

References

[1] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1615–1631, 2018. 6, 16

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 3

[3] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:22405–22418, 2021. 2, 4

[4] Shiming Chen, Wenjin Hou, Ziming Hong, Xiaohan Ding, Yibing Song, Xinge You, Tongliang Liu, and Kun Zhang. Evolving semantic prototype improves generative zero-shot learning. In *International Conference on Machine Learning (ICML)*, pages 4611–4622. PMLR, 2023. 3

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016. 3

[6] Yushi Cheng, Xiaoyu Ji, Lixu Wang, Qi Pang, Yi-Chao Chen, and Wenyuan Xu. {mID}: Tracing screen photos via {Moiré} patterns. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2969–2986, 2021. 16

[7] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 3, 6, 16

[8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) workshops*, pages 702–703, 2020. 1, 12

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009. 16

[10] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 2, 6, 12, 16

[11] N Benjamin Erichson, Soon Hoe Lim, Winnie Xu, Francisco Utrera, Ziang Cao, and Michael W Mahoney. Noisymix: Boosting model robustness to common corruptions. *arXiv preprint arXiv:2202.01263*, 2022. 5

[12] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 2, 6, 12, 16

[14] Saurabh Garg, Sivaraman Balakrishnan, and Zachary Lipton. Domain adaptation under open set label shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 22531–22546, 2022. 2

[15] Junfeng Guo, Yiming Li, Lixu Wang, Shu-Tao Xia, Heng Huang, Cong Liu, and Bo Li. Domain watermark: Effective and harmless dataset copyright protection is closed at hand. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 1

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 19

[17] Yingzhe He, Guozhu Meng, Kai Chen, Xingbo Hu, and Jinwen He. Towards security threats of deep learning systems: A survey. *IEEE Transactions on Software Engineering*, 48 (5):1743–1770, 2020. 1

[18] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019. 5

[19] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022. 4

[20] Ziming Hong, Shiming Chen, Guo-Sen Xie, Wenhan Yang, Jian Zhao, Yuanjie Shao, Qinmu Peng, and Xinge You. Semantic compression embedding for generative zero-shot learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 956–963, 2022. 3

[21] Ziming Hong, Zhenyi Wang, Li Shen, Yu Yao, Zhuo Huang, Shiming Chen, Chuanwu Yang, Mingming Gong, and Tongliang Liu. Improving non-transferable representation learning by harnessing content and style. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. 1

[22] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3635–3649, 2021. 2

[23] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1501–1510, 2017. 3, 16

[24] Zhuo Huang, Xiaobo Xia, Li Shen, Bo Han, Mingming Gong, Chen Gong, and Tongliang Liu. Harnessing out-of-distribution examples via augmenting content and style. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[25] Zhuo Huang, Muyang Li, Li Shen, Jun Yu, Chen Gong, Bo Han, and Tongliang Liu. Winning prize comes from los-

ing tickets: Improve invariant learning by exploring variant parameters for out-of-distribution generalization. *arXiv preprint arXiv:2310.16391*, 2023. 2

[26] Zhuo Huang, Li Shen, Jun Yu, Bo Han, and Tongliang Liu. Flatmatch: Bridging labeled data and unlabeled data with cross-sharpness for semi-supervised learning. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[27] Zhuo Huang, Miaoxi Zhu, Xiaobo Xia, Li Shen, Jun Yu, Chen Gong, Bo Han, Bo Du, and Tongliang Liu. Robust generalization against photon-limited corruptions via worst-case sharpness minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16175–16185, 2023. 2, 15

[28] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456. pmlr, 2015. 19

[29] Takuhiro Kaneko and Tatsuya Harada. Blur, noise, and compression robust generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13579–13589, 2021. 5

[30] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3, 6, 16

[31] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning (ICML)*, pages 5905–5914. PMLR, 2021. 6, 15

[32] Isabell Lederer, Rudolf Mayer, and Andreas Rauber. Identifying appropriate intellectual property protection mechanisms for machine learning models: A systematization of watermarking, fingerprinting, model access, and attacks. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 1, 16

[33] Bo Li, Xinge You, Jing Wang, Qinmu Peng, Shi Yin, Ruinan Qi, Qianqian Ren, and Ziming Hong. Ias-net: Joint intraclassly adaptive gan and segmentation network for unsupervised cross-domain in neonatal brain mri segmentation. *Medical Physics*, 48(11):6962–6975, 2021. 2

[34] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018. 2

[35] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2021. 6, 16

[36] Soon Hoe Lim, N. Benjamin Erichson, Francisco Utrera, Winnie Xu, and Michael W. Mahoney. Noisy feature mixup. In *International Conference on Learning Representations (ICLR)*, 2022. 5

[37] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pages 273–294. Springer, 2018. 6, 16

[38] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3

[39] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 2, 6, 12, 16

[40] Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. I know what you trained last summer: A survey on stealing machine learning models and defences. *ACM Computing Surveys*, 2023. 1

[41] Poojan Oza, Vishwanath A Sindagi, Vibashan Vishnukumar Sharmini, and Vishal M Patel. Unsupervised domain adaptation of object detectors: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[42] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 6, 16

[43] Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures. *arXiv preprint arXiv:1807.10108*, 2018. 2, 6, 12, 16

[44] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 16282–16292, 2020. 2

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 16, 19

[46] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6924–6932, 2017. 3

[47] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (11), 2008. 19

[48] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 3

[49] Lixu Wang, Shichao Xu, Ruiqi Xu, Xiao Wang, and Qi Zhu. Non-transferable learning: A new approach for model ownership verification and applicability authorization. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2, 3, 4, 6, 7, 12, 13, 14, 16, 17, 18, 19, 21

[50] Lianyu Wang, Meng Wang, Daoqiang Zhang, and Huazhu Fu. Model barrier: A compact un-transferable isolation domain for model intellectual property protection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20475–20484, 2023. 1, 2, 3, 4, 6, 7, 12, 13, 14, 16, 17, 19, 21

[51] Meng Wang, Weifang Zhu, Kai Yu, Zhongyue Chen, Fei Shi, Yi Zhou, Yuhui Ma, Yuanyuan Peng, Dengsen Bao, Shuanglang Feng, et al. Semi-supervised capsule cgan for

speckle noise reduction in retinal oct images. *IEEE Transactions on Medical Imaging*, 40(4):1168–1183, 2021. 5

[52] Qizhou Wang, Junjie Ye, Feng Liu, Quanyu Dai, Marcus Kalander, Tongliang Liu, Jianye HAO, and Bo Han. Out-of-distribution detection with implicit outlier transformation. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 4

[53] Zhenyi Wang, Li Shen, Tongliang Liu, Tiehang Duan, Yanjun Zhu, Donglin Zhan, David Doermann, and Mingchen Gao. Defending against data-free model extraction by distributionally robust defensive training. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. 1

[54] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:10546–10559, 2022. 6, 16

[55] Mingfu Xue, Yushu Zhang, Jian Wang, and Weiqiang Liu. Intellectual property protection for deep learning models: Taxonomy, methods, attacks, and evaluations. *IEEE Transactions on Artificial Intelligence*, 3(6):908–923, 2021. 1

[56] Chaojian Yu, Bo Han, Li Shen, Jun Yu, Chen Gong, Mingming Gong, and Tongliang Liu. Understanding robust overfitting of adversarial training and beyond. In *International Conference on Machine Learning (ICML)*, pages 25595–25610. PMLR, 2022. 5

[57] Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations (ICLR)*, 2022. 6, 16

[58] Jie Zhang, Dongdong Chen, Jing Liao, Weiming Zhang, Huamin Feng, Gang Hua, and Nenghai Yu. Deep model intellectual property protection via deep watermarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4005–4020, 2021. 1

[59] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2737–2746, 2020. 5

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

[61] Mingli Zhu, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4466–4477, 2023. 6, 15, 16

[62] Yingtian Zou, Kenji Kawaguchi, Yingnan Liu, Jiashuo Liu, Mong-Li Lee, and Wynne Hsu. Towards robust out-of-distribution generalization bounds via sharpness. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. 2, 4