Your Transferability Barrier is Fragile: Free-Lunch for Transferring the Non-Transferable Learning

Ziming Hong[1] Li Shen[2] Tongliang Liu[1*]
[1]Sydney AI Centre, The University of Sydney [2]JD Explore Academy

Abstract

Recently, non-transferable learning (NTL) was proposed to restrict models' generalization toward the target domain(s), which serves as state-of-the-art solutions for intellectual property (IP) protection. However, the robustness of the established "transferability barrier" for degrading the target domain performance has not been well studied. In this paper, we first show that the generalization performance of NTL models is widely impaired on third-party domains (i.e., the unseen domain in the NTL training stage). We explore the impairment patterns and find that: due to the dominant generalization of non-transferable task, NTL models tend to make target-domain-consistent predictions on third-party domains, even though only a slight distribution shift from the third-party domain to the source domain. Motivated by these findings, we uncover the potential risks of NTL by proposing a simple but effective method (dubbed as TransNTL) to recover the target domain performance with few source domain data. Specifically, by performing a group of different perturbations on the few source domain data, we obtain diverse third-party domains that evoke the same impairment patterns as the unavailable target domain. Then, we fine-tune the NTL model under an impairment-repair self-distillation framework, where the source-domain predictions are used to teach the model itself how to predict on third-party domains, thus repairing the impaired generalization. Empirically, experiments on standard NTL benchmarks show that the proposed TransNTL reaches up to ∼72% target-domain improvements by using only 10% source domain data. Finally, we also explore a feasible defense method and empirically demonstrate its effectiveness.

1. Introduction

Well-trained deep learning models are the core of Machine-Learning-as-a-Service (MLaaS), which are being provided in a wide range of applications closely related to our daily life [40, 55]. The training process of deep learning models requires massive well-annotated training data, expensive hardware resources, and often takes weeks or even months, which requires high cost, and thus leading to the high business value [55]. As such, how can the model owners protect the intellectual property (IP) [15, 17, 49, 53, 55, 58] of deep learning models is waiting to be solved.

Recently, Non-Transferable Learning (NTL) [49] was proposed as a novel technology in IP protection. NTL aims to restrict the generalization of a deep learning model toward a certain target domain (target-specified NTL) or all other domains except the source domain (source-only NTL). To this end, built upon the source-domain supervised learning (SL) paradigm, existing methods [21, 49, 50] impose a *non-transferable task* to maximize the target domain representations and correct labels. Target-specified NTL and source-only NTL serve as promising solutions of two types of IP protection techniques: *ownership verification* [32] and *applicability authorization* [49], respectively.

Despite the success of NTL in restricting source-to-target knowledge transferring, the robustness of the "transferability barrier" established in NTL models has not been well studied. Earlier evaluations in [49, 50] show that NTL models are still resistant to state-of-art watermark removal attacks when up to 30% source domain data are available for attack. An intriguing inquiry is thus: *How can an attacker break the transferability barrier, thus effectively recovering target domain performance?* In this work, we seek to test the robustness of the transferability barrier in NTL models and show that it is possible to recover target domain knowledge using only few source domain data.

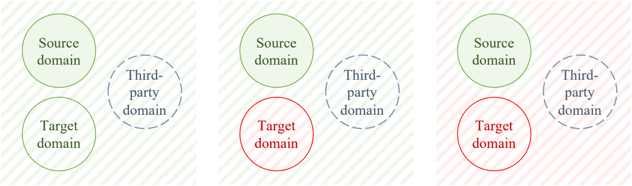
We start by exploring the performance of target-specified NTL models on unseen *third-party domains* (i.e., the domain with distribution gaps to both the source and target domain[1], as shown in Fig. 1). We involve three kinds of third-party domains: perturbed source domain[2], augmented source domain[3], and real domains collected from different environments. As shown in Fig. 2, we observe that *although the intention is to degrade the target domain performance, the generalization of target-specified NTL models are im-*

[*] Correspondence to Tongliang Liu (tongliang.liu@sydney.edu.au)

[1] In NTL scenarios, the defined third-party domain shares the same contents (i.e., class labels) with the source and the target domain.

[2] We perturbed the image by adding Gaussian noise with different std.

[3] We augment the image by using RandAugment [8].



(a) SL (b) target-specified NTL (c) source-only NTL

Figure 1. The source domain, the target domain, and the third-party domain in the paradigm of (a) supervised learning (SL), (b) target-specified NTL, and (c) source-only NTL. The green shadow represents the *ideal* generalization area of the source domain. The red shadow represents the non-transferable area.

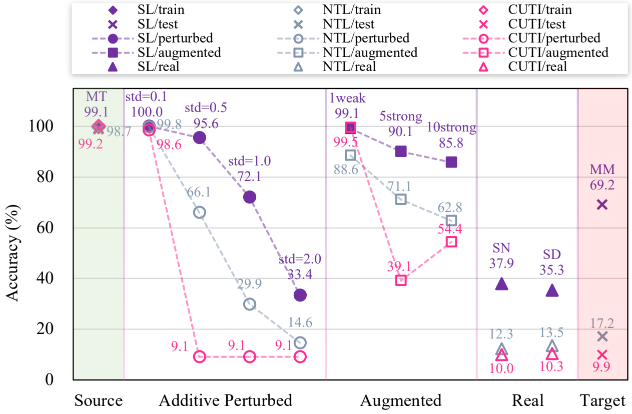


Figure 2. The accuracies of SL and target-specified NTL (NTL [49] and CUTI [50]) on third-party domains, including pertubed source domains, augmented source domains, and real-world datasets. Compared to SL, NTL models encounter varying degrees of generalization impairments on third-party domains.

paired with varying degrees on third-party domains (compared to SL models)[4]. We investigate the impairment patterns on third-party domains and identify that NTL models: (1) easily make an over-confident prediction, (2) tend to predict target-domain label on third-party domains. *Above impairment patterns are exactly consistent with the counterpart on the target domain and occurred even though only a slight distribution shift from the third-party domain to the source domain, thus being a security risk for NTL models.* Through the lens of the flatness [3, 12, 25–27, 34, 62] of loss landscapes, we find such impairment patterns are caused by the dominant generalization of the *non-transferable task* in the whole data space, which is reflected on the more flat loss landscape of the target domain distribution.

Motivated by these findings, we uncover the potential risks of NTL by proposing a simple but effective method (dubbed as TransNTL) that enables an attacker to Transferring the NTL (i.e., recover the target domain performance) with few source domain data. To begin with, by performing a group of different perturbations on the few source domain data, we obtain third-party domains with diverse distribution shifts from the original source domain. These source-domain-derived third-party domains, although making no assumptions about the target domain knowledge, evoke the same impairment patterns as the unavailable target domain, thus serving as a "free lunch" for breaking the transferability barrier. Then, we fine-tune the NTL model under an impairment-repair self-distillation framework, where the source-domain prediction is used to teach the model itself how to predict on third-party domains, thus repairing the impairment patterns and breaking the transferability barrier in the NTL model. Besides, to suppress the dominant generalization of non-transferable task, we penalize the sharpness of the source and third-party distributions when fine-tuning the NTL model. Accordingly, a flat loss landscape around the source distributions will be produced, thus further enhancing the generalization of source domain and prompting the impairment repairments.

Empirically, we conduct experiments on standard NTL benchmarks (i.e., Digits, CIFAR10 & STL10, and VisDA) and show that the proposed TransNTL can effectively break the transferability barriers for both target-specified NTL models and source-only NTL models, with the target domain accuracy increasing by up to ∼72%. Additionally, considering that the proposed TransNTL reveals the potential risk of NTL, we further propose a feasible defense method that leverages TransNTL in NTL training, and empirically, we validate its effectiveness.

Our contributions are summarized as follows:

- By analyzing the performance on third-party domains, we observe that NTL models exhibit varying degrees of generalization impairments compared to SL models. We further identify its impairment patterns and the underlying cause, thus providing insights for attacking NTL models.
- We are the first to reveal the risk of NTL by proposing an effective attack method (dubbed as TransNTL). TransNTL breaks the transferability barrier by leveraging the observation that slight-perturbed source domain exhibits the same impairment patterns as the target domain.
- Extensive experiments on NTL benchmarks demonstrate the effectiveness of TransNTL in attacking target-specified and source-only NTL models and NTL-based ownership verification and applicability authorization.
- We also propose a defense method against TransNTL, and empirically, we validate its effectiveness.

2. Related Work

Target-specified NTL. Target-specified NTL aims to restrict the generalization of a deep learning model toward a certain target domain, which can be seen as an anti-task to domain adaptation (DA) [14, 22, 33, 41, 44]. As the target domain data is accessible, existing methods [49, 50] directly

[4] In Fig. 2, MNIST (MT) [10] and MNIST-M (MM) [13] are serviced as the source and target domain, respectively. SVHN (SN) [39] and SYN-D (SD) [43] are third-party domains collected from real world. More results of the impairments on third-party domains are shown in Appendix A.1.

reduce the statistical dependence between the source domain representations and the target domain representations, thus resisting the target-specified transferability. Wang et al. [49] first propose the NTL task. They design an NTL framework that adds two statistical dependence relaxation terms on standard supervised learning: (i) maximizing the Kullback-Leible (KL) divergence between target domain representations and labels, and (ii) maximizing the maximum mean discrepancy (MMD) between the distribution of source and target domain representations. Further, CUTI [50] improves the NTL by introducing style transfer [23, 46]. They augment target domain images by transferring their styles to the source domain style, thus obtaining a CUTI-domain. Then, they train a model by maximizing the KL divergence between labels and the representations on both the target domain and the CUTI-domain.

In this work, we show that although aiming at degrading the target domain performance, the target-specified NTL models inevitably result in significant generalization impairments on third-party domains. Such impairments can be used to recover the target domain performance, thus leading to the unreliable ownership verification deployed by target-specified NTL.

Source-only NTL. The intention of source-only NTL is to degrade the performance in all other domains except the source domain, which is opposite to the purpose of domain generalization (DG) [2, 24, 48, 60]. Due to the assumption of only the source domain data being available, existing methods [49, 50] introduce generative adversarial network (GAN) [4, 5, 20, 38] to synthesize fake images from the source domain and see them as the target domain. Thus, target-specified NTL methods can be leveraged to solve it.

However, due to the risk of target-specified NTL methods, existing source-only NTL-based applicability authorizations are also unreliable.

3. Investigating Generalization Impairments in NTL Models

In this section, we present an empirical study to investigate the generalization impairments of NTL models on third-party domains. In Sec. 3.1, we first review the general framework of existing NTL methods, and then, we formally present the definition of *third-party domain* in the NTL scenarios. In Sec. 3.2, we investigate the patterns of generalization impairments. Subsequently, in Sec. 3.3, we explore the underlying reasons behind these patterns.

3.1. Preliminary

General framework of NTL. Considering an image classification task. Let $\mathcal{D}_s$ and $\mathcal{D}_t$ represent the source domain and the target domain, respectively. Considering a neural network $f_\theta : \mathcal{X} \to \mathcal{Y}$ with parameters θ, NTL aims to train the f_θ to degrade performance on the target domain $\mathcal{D}_t$ and simultaneously maintain performance on the source domain $\mathcal{D}_s$. To reach this goal, existing methods (NTL [49] and CUTI [50]) follow the framework that imposes a regularization term on the SL to maximize the target domain representations and the correct labels:

$$\min_\theta \Big\{ \mathcal{L}_{\text{ntl}} := \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}_s}\left[\mathcal{L}_{\text{src}}(f_\theta(x), y)\right]}_{\mathcal{T}_{\text{src}}} - \lambda \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}_t}\left[\mathcal{L}_{\text{tgt}}(f_\theta(x), y)\right]}_{\mathcal{T}_{\text{tgt}}} \Big\}, \quad (1)$$

where λ is a trade-off weight, $\mathcal{L}_{\text{src}}$ and $\mathcal{L}_{\text{tgt}}$ represent the loss function (e.g., Kullback-Leible divergence) for the source and the target domain, respectively. Intuitively, the general NTL framework can be split into two tasks: (1) a source domain learning task $\mathcal{T}_{\text{src}}$ to maintain the source domain performance, and (2) a non-transferable task $\mathcal{T}_{\text{tgt}}$ to degrade the target domain performance.

Third-party domain. Beyond the source domain and the target domain, we pay more attention to third-party domains in the data space. Formally, we define the third-party domain in the NTL scenarios as follows:

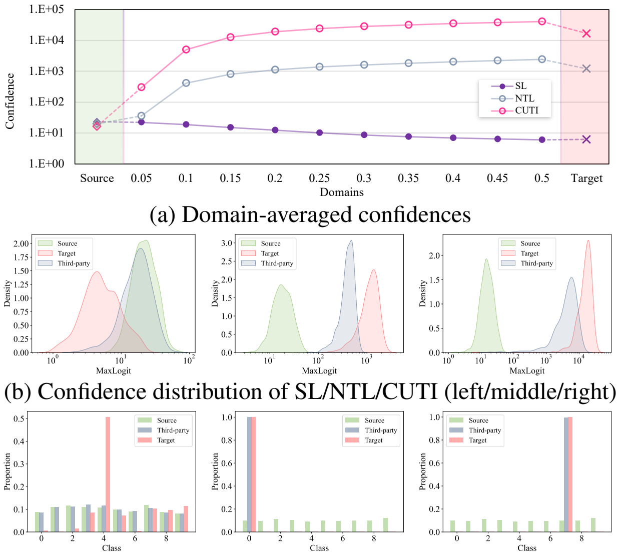
Definition 1 (Third-party domain) *In the NTL scenarios, if a domain $\mathcal{D}$ shares the same contents (i.e., class labels) with but has distribution gaps to both the source domain $\mathcal{D}_s$ and target domain $\mathcal{D}_t$, we call it as the third-party domain.*

An intuitive illustration between the source domain, the target domain, and the third-party domain is shown in Fig. 1. In general, SL models trained on $\mathcal{D}_s$ and target-specified NTL models trained on $\{\mathcal{D}_s, \mathcal{D}_t\}$ are expected to have similar classification accuracies on third-party domains. This is because third-party domains share the same contents with the source domain $\mathcal{D}_s$ and has distribution gap to the target domain $\mathcal{D}_t$. However, as shown in Fig. 2, we observe that despite the intention of target-specified NTL is to degrade the target domain performance, *the generalization of target-specified NTL models are widespreadly impaired on third-party domains (compared to normal SL models)*. Motivated by such phenomena, we further explore the underlying patterns of these impairments.

3.2. Impairment Patterns on Third-Party Domains

We focus on third-party domains obtained by perturbing the source domain $\mathcal{D}_s$ through Gaussian additive noise, thus making the distribution shift between third-party domains and the source domain controllable. Specifically, we perturb source domain images[5] by adding Gaussian noise with different standard deviations (i.e., std), thus obtaining a group

[5] For illustration, we consider the task that CIFAR10 [30] is the source domain and STL10 [7] is the target domain. Due to limited space, more implementation details and results on other datasets to support the identified impairment patterns are shown in Appendix A.2.



(a) Domain-averaged confidences

(b) Confidence distribution of SL/NTL/CUTI (left/middle/right)

(c) Predictions proportion of SL/NTL/CUTI (left/middle/right)

Figure 3. Impairment patterns of NTL models. (a) Domain-averaged confidence of SL, NTL [49] and CUTI [50] on the source domain, the target domain, and third-party domains obtained by perturbing the source domain with different *std*. (b) Distribution of per-sample confidence of SL/NTL/CUTI on the source domain, the target domain, and a typical third-party domain ($std = 0.1$). (c) Prediction proportions of SL/NTL/CUTI on the source domain, the target domain, and the typical third-party domain ($std = 0.1$).

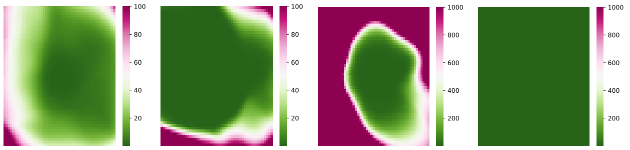
of third-party domains $\{\hat{\mathcal{D}}_s^g\}_{g=1}^G$ with different distribution shifts from the source domain, where G is the number of standard deviations. Totally, we identify two impairment patterns, which are shown as follows:

Pattern 1: Over-confident prediction. *NTL models exhibit over-confident predictions on third-party domain as well as the target domain.* Specifically, the confidence of prediction is quantified by the MaxLogit scoring [19, 52], which can be represented as:

$$s_{\text{ML}}(x; f) = \max_k f_k(x), \tag{2}$$

where $f_k(\cdot)$ denotes the k-th element of the output logits. As shown in Fig. 3 (a), we plot the prediction confidences of NTL models (NTL [49] and CUTI [50]) and a SL model on the source domain $\mathcal{D}_s$, the target domain $\mathcal{D}_t$, and third-party domains $\{\hat{\mathcal{D}}_s^g\}_{g=1}^G$. Particularly, typical distributions of per-sample confidence of each model are shown in Fig. 3 (b). These results illustrate that with the noise severity increasing, NTL and CUTI predict the third-party domain with more confidence. It is worth noting that such phenomenons are opposite to the SL model which makes lower confident predictions when facing unseen distribution shifts (e.g., the target domain and any third-party domains).

Pattern 2: Implicit target domain class. *NTL models tend to predict the "implicit target domain class" on third-party domains.* As shown in Fig. 3 (c), we plot the proportion of different classes predicted by a SL model and NTL models (NTL [49] and CUTI [50]). The SL model makes diverse predictions for the data in the third-party domain and the target domain, However, NTL models, although trained in a maximization term on the target domain (refer to Eq. (1)), 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*.



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

Figure 4. Loss landscapes of an NTL [49] model and a CUTI [50] model. (a) Loss landscape of the NTL model on the source domain (left) and the target domain (right). (b) Loss landscape of the CUTI model on the source domain (left) and the target domain (right).

Overall, NTL models tend to make an over-confident decision to predict the label of the "*implicit target-domain class*" on third-party domains. Particularly, such *target-domain-consistent predictions* can still occur even though only a slight distribution shift from the third-party domain to the source domain, thus being a loophole in NTL models.

3.3. Causes for Generalization Impairments

Although the impairment patterns of NTL models are identified, the underlying causes are still unclear. In this section, we explore what's leading to the generalization impairments of NTL models and why the NTL models exhibit target-domain-like impairment patterns on third-party domains.

According to Eq. (1), the general objective of NTL can be divided into two parts: *a source domain learning task* $\mathcal{T}_{\text{src}}$ to maintain the source domain performance, and *a non-transferable task* $\mathcal{T}_{\text{tgt}}$ to degrade the target domain performance. Motivated by the fact that the performance of out-of-domain generalization is closely related to the flatness of loss landscape [3, 62], 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. The results are shown in Fig. 4, and we have two major findings:

- *NTL models are optimized to an extremely sharp minima on the source domain, thus limiting the generalization of the source domain learning task.*
- *NTL models are optimized to a relative flat minima on the target domain, thus leading to the well-generalization of the non-transferable task.*

Accordingly, the sharp source domain landscape and the flat target domain landscape co-lead to the dominance of the generalization of the non-transferable task $\mathcal{T}_{\text{tgt}}$ in the whole data space. Thus, the performance of NTL models on third-party domains will be mainly influenced by the non-transferable task $\mathcal{T}_{\text{tgt}}$ rather than the normal source domain learning task $\mathcal{T}_{\text{src}}$. Particularly, the $\mathcal{T}_{\text{tgt}}$ is designed to degrade classification performance. As a result, NTL models tend to exhibit degraded performance (i.e., generalization impairments) on third-party domains, with the models' predictions following the objective of the non-transferable task $\mathcal{T}_{\text{tgt}}$ (i.e., target-domain-like impairment patterns).

More empirical results to support the explanations can be found in Appendix A.3.

4. Transferring the NTL

Motivated by our findings, in this section, we uncover the potential risks of NTL by proposing a simple but effective method (dubbed as TransNTL) that enables an attacker Transferring the NTL with only a small proportion of source domain data. In Sec. 4.1, we first formula the problem of attacking NTL. Then, in Sec. 4.2 and Sec. 4.3, we introduce main components of the proposed TransNTL. In Sec. 4.4, we illustrate the overall training process of TransNTL.

4.1. Problem Formulation

Pre-trained NTL model. We assume the attack target is an NTL model f_θ (with parameter $\theta \in \mathbb{R}^d$) trained on a source domain $\mathcal{D}_s = \{(x_i, y_i)\}_{i=1}^{N_s}$ and a target domain $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{N_t}$[6]. The NTL model f_θ has normal accuracy (comparable to SL) on the source domain $\mathcal{D}_s$ and poor performance on the target domain $\mathcal{D}_t$.

Attacking goal. We consider the scenarios that the attacker is given a white-box NTL model f_θ with few source domain data $\mathcal{D}_c$, where $\mathcal{D}_c = \{(x_i, y_i)\}_{i=1}^{N_c}$ and $N_c \ll N_s$. The attacker's goal is to recover the normal transferability from the source domain to the target domain in the NTL model f_θ (i.e., recovering the target domain accuracy) and also maintain the source domain performance.

4.2. Impairment-Repair Fine-Tuning

Fine-tuning paradigm. The proposed TransNTL is built upon the fine-tuning paradigm. Specifically, we use the few source domain data $\mathcal{D}_c$ to fine-tune all the parameters in the NTL model f_θ. Formally, the objective of fine-tuning is formulated as:

$$\min_\theta \left\{\mathcal{L}_{\text{ft}} := \mathbb{E}_{(x,y)\sim\mathcal{D}_c}\left[\mathcal{L}_{\text{ce}}(f_\theta(x), y)\right]\right\}, \tag{3}$$

where $\mathcal{L}_{\text{ce}}$ represents the Cross-Entropy loss.

Impairment-repair self-distillation. According to our findings in Sec. 3, third-party domains which are derived from perturbing the source domain, although making no assumptions about the target domain knowledge, evoke the same impairment patterns as the unavailable target domain. Therefore, we are motivated to start breaking the transferability barrier by repairing the impairment patterns on perturbation-based third-party domains.

For effectively attacking, we consider a group of different perturbation functions $p_g(\cdot), g \in \mathcal{G} = \{1, 2, \cdots, G\}$, where each p_g is sampled from the perturbation collection $\mathcal{P}$. Particularly, we let the collection $\mathcal{P}$ contains three kinds of perturbation sets: additive perturbation set $\mathcal{P}_\oplus$ (e.g., Gaussian noise [18], adversarial noise [56]), multiplicative perturbation set $\mathcal{P}_\odot$ (e.g., speckle noise [11, 36, 51]), and convolution perturbation set $\mathcal{P}_\otimes$ (e.g., motion blur [29, 59]). Regarding perturbations with different types, the perturbed image $\hat{x} = p(x)$ can be represented as:

$$\hat{x} = p(x) = \begin{cases} x + \delta, & p \in \mathcal{P}_\oplus \\ x \odot (1 + \delta), & p \in \mathcal{P}_\odot \\ x \otimes k, & p \in \mathcal{P}_\otimes \end{cases} \tag{4}$$

where $\odot$ and $\otimes$ represent the dot-product and convolution operation, respectively. The δ represents a random perturbation in which elements are independently drawn from a pre-defined probability distribution (e.g., Gaussian distribution). The k is a pre-defined kernel (e.g., Gaussian kernel).

By performing the group of perturbations on $\mathcal{D}_c$, we obtain a group of third-party domains $\{\hat{\mathcal{D}}_c^g\}_{g=1}^G$ with diverse distribution shifts from the original source domain, where $\hat{\mathcal{D}}_c^g = \{(p_g(x), y) \mid p_g \in \mathcal{P}, (x, y) \sim \mathcal{D}_c\}$. To mitigate the over-confident target-domain-consistent predictions on third-party domains, we propose an *impairment-repair self-distillation* framework, in which we use the model predictions on the source domain to teach the model itself how to predict on third-party domains. The self-distillation framework can be formulated as follows:

$$\min_\theta \left\{\mathcal{L}_{\text{sd}} := \max_{p\in\mathcal{P}} \mathbb{E}_{(x,y)\sim\mathcal{D}_c}\left[\mathcal{L}_{\text{kl}}(f_\theta(p(x)), f_\theta(x))\right]\right\}, \tag{5}$$

where $\mathcal{L}_{\text{kl}}$ represents the Kullback-Leible (KL) divergence. The loss term $\mathcal{L}_{\text{sd}}$ finds the most-impaired third-party domain based on the KL divergence between the perturbed distribution and the source domain distribution. By minimizing $\mathcal{L}_{\text{sd}}$, the NTL model f_θ is taught by itself to learn source-domain-consistent predictions on the group of third-party domains. Consequently, the impairment patterns on third-party domains (i.e., over-confidence and implicit target-domain class) will be repaired.

The term $\mathcal{L}_{\text{sd}}$ is used as a regularization in the fine-tuning paradigm. Accordingly, the total impairment-repair fine-tuning framework can be formulated as:

$$\min_\theta \left\{\mathcal{L}_{\text{irft}} := \mathcal{L}_{\text{ft}} + \lambda_{\text{sd}}\mathcal{L}_{\text{sd}}\right\}, \tag{6}$$

where λ_{sd} is a weight to balance the self-distillation loss.

[6] For target-specified NTL, the $\mathcal{D}_t$ is the real target domain. For source-only NTL, the $\mathcal{D}_t$ is the fake domain synthesized from the $\mathcal{D}_s$ using GAN.

Algorithm 1 Training TransNTL

```
Input: An NTL model $f_\theta$ with parameters $\theta$, a few source domain data $\mathcal{D}_c = \{(x_i, y_i)\}_{i=1}^{N_c}$; perturbation collection $\mathcal{P}$, total fine-tuning epochs $E$, batchsize $B$, self-distillation loss weight $\lambda_{\text{sd}}$, radius of maxmization region $\rho$.
for $e = 1$ to $E$ do                                              ▷ Start training
    Sample a mini-batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{B}$ from $\mathcal{D}_c$;
    $\mathcal{L}_{\text{irft}}(\theta)$ = ImpairRepairFTLoss($f_\theta$, $\mathcal{B}$, $\mathcal{P}$, $\lambda_{\text{sd}}$);
    Compute the $\epsilon^*$ through $\rho\frac{T_\theta^2\nabla\mathcal{L}_{\text{irft}}(\theta)}{\|T_\theta\nabla\mathcal{L}_{\text{irft}}(\theta)\|_2}$;          ▷ Eq. (9)
    $\mathcal{L}_{\text{irft}}(\theta+\epsilon^*)$ = ImpairRepairFTLoss($f_{\theta+\epsilon^*}$, $\mathcal{B}$, $\mathcal{P}$, $\lambda_{\text{sd}}$);
    Updating parameters $\theta$ through minimizing $\mathcal{L}_{\text{irft}}(\theta+\epsilon^*)$;
end for                                                            ▷ End training
Output: the attack result $f_\theta$

function ImpairRepairFTLoss($f_\theta$, $\mathcal{B}$, $\mathcal{P}$, $\lambda_{\text{sd}}$);
    $\mathcal{L}_{\text{ft}} = \frac{1}{B}\sum_{i=1}^{B}\mathcal{L}_{\text{ce}}(f_\theta(x_i), y_i)$;                    ▷ Eq. (3)
    $\mathcal{L}_{\text{sd}} = \max_{p\in\mathcal{P}}\frac{1}{B}\sum_{i=1}^{B}[\mathcal{L}_{\text{kl}}(f_\theta(p(x_i)), f_\theta(x_i))]$;   ▷ Eq. (5)
    return $\mathcal{L}_{\text{irft}} = \mathcal{L}_{\text{ft}} + \lambda_{\text{sd}}\mathcal{L}_{\text{sd}}$
```

4.3. Sharpness-Aware Minimization

According to Sec. 3.3, the source domain landscape of NTL model is sharper than the landscape of the target domain. This leads to the limited generalization of source domain learning task $\mathcal{T}_{\text{src}}$ and the dominant generalization of non-transferable task $\mathcal{T}_{\text{tgt}}$. In order to alleviate such a situation, we propose to produce a flat loss landscape over the source domain distributions by minimizing the sharpness (i.e., loss changes around the neighbor of model parameters) of the source domain and the third-party domains. Formally, following [31], the sharpness term $\mathcal{L}_{\text{sharp}}$ can be formulated as:

$$\mathcal{L}_{\text{sharp}} := \max_{\|T_\theta^{-1}\epsilon\|_2\leq\rho} \left\{\mathcal{L}_{\text{irft}}(\theta+\epsilon) - \mathcal{L}_{\text{irft}}(\theta)\right\}, \tag{7}$$

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

Intuitively, the $\mathcal{L}_{\text{sharp}}$ indicates the largest loss change when model parameter θ is perturbed with ϵ, which is seen as the loss sharpness. By minimizing $\mathcal{L}_{\text{Sharp}}$, the model will find a more flat minima on the source domain and the third-party domain. Therefore, the generalization performance of the source domain learning task $\mathcal{T}_{\text{src}}$ will be further enhanced, enabling better repairment of NTL impairments.

4.4. Overall Training Process for TransNTL

By combining the self-distillation framework in Eq. (5) and the sharpness term in Eq. (7), we get the total objective of TransNTL, which can be formulated as:

$$\min_\theta\Big\{\mathcal{L}_{\text{TransNTL}} := \mathcal{L}_{\text{irft}} + \mathcal{L}_{\text{sharp}} = \max_{\|T_\theta^{-1}\epsilon\|_2\leq\rho}\mathcal{L}_{\text{irft}}(\theta+\epsilon)\Big\}. \tag{8}$$

Specifically, the inner maximization of the above bi-level optimization problem can be solved as follows (the derivation is shown in Appendix B):

$$\epsilon^* = \mathop{\arg\max}_{\|T_\theta^{-1}\epsilon\|_2\leq\rho}\mathcal{L}_{\text{irft}}(\theta+\epsilon) \approx \rho\frac{T_\theta^2\nabla\mathcal{L}_{\text{irft}}(\theta)}{\|T_\theta\nabla\mathcal{L}_{\text{irft}}(\theta)\|_2}. \tag{9}$$

Overall, the complete algorithm for training TransNTL is shown in Algorithm 1.

5. Experiments

We first describe our experimental setups. Then, in Sec. 5.1, we verify the effectiveness of the proposed TransNTL in attacking NTL and NTL-based IP protection (i.e., ownership verification and applicability authorization). In Sec. 5.2, we conduct ablation studies. Finally, in Sec. 5.3, we discuss a defense method and verify its effectiveness.

Experimental setups. **For datasets**, by following [49, 50], we conduct experiments on (1) *Digits*: MNIST [10], MNIST-M [13], SVHN [39], and SYN-D [43]; (2) *CIFAR10 & STL10* [7, 30]; (3) *VisDA-2017* [42]. **For pretraining NTL models**, we involve all NTL methods, including the first proposed method NTL [49] and the state-of-the-art (SOTA) method CUTI [50]. We pre-train NTL and CUTI by using their released codes and following their parameters. We also follow the same data split, preprocessing and backbones. **For attacking NTL**, we 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 the methods can access 10% source domain data. **For evaluation**, we report Top-1 Accuracy on both the source domain and the target domain. Due to the limited space, more implementation details can be found in Appendix C.

5.1. Transfering the NTL

Effectiveness of TransNTL in recovering target domain performance. We first conduct experiments on target-specified NTL with natural target domains, thus verifying the effectiveness of TransNTL in recovering target domain performance. We carry out experiments on two digits tasks (MNIST→ Digits, SVHN→ Digits) and three more complex tasks (CIFAR10→STL10, STL10→CIFAR10 and VisDA-T→VisDA-V). As shown in Tab. 1, the proposed TransNTL effectively recovers the target domain performance on each task by using only 10% source domain data, with the accuracy increasing by up to 49.2% for NTL and 55.0% for CUTI. Besides, TransNTL significantly outperforms all attacking baselines, with most of them failing to recover the target domain performance for either NTL or CUTI. Moreover, CUTI models always recover to better target domain performance compared to NTL models. This indicates that CUTI is more vulnerable to TransNTL.

Table 1. Transfering the NTL with 10% source domain data. 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. The best result[7] is highlighted in **bold**.

	MNIST→Digits (SL: 99.1 / 47.5)		**SVHN→Digits** (SL: 91.1 / 54.7)		**CIFAR10→STL10** (SL: 86.6 / 68.5)		**STL10→CIFAR10** (SL: 91.0 / 62.4)		**VisDA-T→VisDA-V** (SL: 95.2 / 34.0)	
	NTL	**CUTI**	**NTL**	**CUTI**	**NTL**	**CUTI**	**NTL**	**CUTI**	**NTL**	**CUTI**
Pre-train	98.7 / 12.2	99.3 / 8.7	85.2 / 10.6	89.8 / 10.9	84.1 / 10.1	84.3 / 9.9	83.9 / 11.1	88.1 / 10.2	94.0 / 5.5	93.3 / 10.3
FTAL	97.2 (-1.5) 12.4 (+0.2)	99.3 (+0.0) 8.7 (+0.0)	84.0 (-1.2) 10.8 (+0.2)	89.1 (-0.7) 10.0 (-0.9)	84.6 (+0.5) 10.1 (+0.0)	84.6 (+0.3) 9.9 (+0.0)	84.7 (+0.8) 11.1 (+0.0)	88.2 (+0.1) 10.2 (+0.0)	93.2 (-0.8) 5.6 (+0.1)	93.1 (-0.2) 10.3 (+0.0)
RTAL	95.2 (-3.5) 14.9 (+2.7)	99.0 (-0.3) 8.5 (-0.2)	82.5 (-2.7) 13.2 (+2.6)	87.4 (-2.4) 11.6 (+0.7)	82.2 (-1.9) 10.1 (+0.0)	82.6 (-1.7) 9.9 (+0.0)	83.5 (-0.4) 11.0 (-0.1)	85.6 (-2.5) 10.2 (+0.0)	92.8 (-1.2) 8.3 (+2.8)	91.0 (-2.3) 10.7 (+0.4)
FP	96.8 (-1.9) 14.2 (+2.0)	98.6 (-0.7) 9.9 (+1.2)	83.8 (-1.4) 10.7 (+0.1)	87.2 (-2.6) 12.4 (+1.5)	82.3 (-1.8) 10.4 (+0.3)	82.7 (-1.6) 10.0 (+0.1)	82.6 (-1.3) 9.0 (-2.1)	84.4 (-3.7) 10.5 (+0.3)	91.0 (-3.0) 7.3 (+1.8)	91.1 (-2.2) 13.3 (+3.0)
NAD	97.0 (-1.7) 12.8 (+0.6)	99.0 (-0.3) 8.8 (+0.1)	82.5 (-2.7) 10.4 (-0.2)	87.2 (-2.6) 10.9 (+0.0)	81.2 (-2.9) 10.1 (+0.0)	83.9 (-0.4) 9.9 (+0.0)	73.3 (-10.6) 10.7 (-0.4)	84.3 (-3.8) 50.4 (+40.2)	9.0 (-85.0) 7.7 (+2.2)	89.0 (-4.3) 25.4 (+15.1)
i-BAU	96.4 (-2.3) 10.9 (-1.3)	99.2 (-0.1) 8.9 (+0.2)	81.2 (-4.0) 13.9 (+3.3)	86.7 (-3.1) 11.0 (+0.1)	80.5 (-3.6) 10.2 (+0.1)	77.0 (-7.3) 17.7 (+7.8)	83.6 (-0.3) 11.0 (-0.1)	83.5 (-4.6) 36.5 (+26.3)	90.3 (-3.7) 5.9 (+0.4)	88.4 (-4.9) 15.1 (+4.8)
FT-SAM	90.9 (-7.8) 11.7 (-0.5)	94.8 (-4.5) 34.6 (+25.9)	72.5 (-12.7) 22.6 (+12.0)	87.3 (-2.5) 13.6 (+2.7)	76.1 (-8.0) 14.7 (+4.6)	78.9 (-5.4) 26.7 (+16.8)	78.5 (-5.4) 11.1 (+0.0)	86.3 (-1.8) 11.4 (+1.2)	12.0 (-82.0) 9.5 (+4.0)	28.4 (-64.9) 11.0 (+0.7)
TransNTL (Ours)	**97.9 (-0.8)** **37.6 (+25.4)**	**99.0 (-0.3)** **48.4 (+39.7)**	**84.8 (-0.4)** **48.2 (+37.6)**	**87.9 (-1.9)** **55.3 (+44.4)**	**82.5 (-1.6)** **49.7 (+39.6)**	**80.2 (-4.1)** **62.3 (+52.4)**	**83.4 (-0.5)** **60.3 (+49.2)**	**85.1 (-3.0)** **65.2 (+55.0)**	**89.4 (-4.6)** **22.5 (+17.0)**	**91.0 (-2.3)** **31.2 (+20.9)**

Table 2. Risks for NTL-based ownership verification. "(P)" represents the patched domain. We report the source domain accuracy (%) and target domain accuracy (%). The relative accuracy drop compared to the pre-trained NTL method is shown in brackets.

	CIFAR10→CIFAR10(P) (SL: 86.6 / 61.5)		**VisDA-T→VisDA-T(P)** (SL: 95.2 / 94.0)	
	NTL	**CUTI**	**NTL**	**CUTI**
Pre-train	81.2 / 9.5	86.2 / 10.5	94.3 / 16.3	95.1 / 17.6
FTAL	82.3 (+1.1) 9.5 (+0.0)	85.4 (-0.8) 10.5 (+0.0)	94.0 (-0.3) 17.9 (+1.6)	94.3 (-0.8) 16.7 (-0.9)
RTAL	81.3 (+0.1) 9.5 (+0.0)	82.4 (-3.8) 11.4 (+0.9)	92.9 (-1.4) 33.4 (+17.1)	93.4 (-1.7) 26.8 (+9.2)
FP	79.6 (-1.6) 9.6 (+0.1)	82.3 (-3.9) 8.2 (-2.3)	92.3 (-2.0) 53.9 (+37.6)	92.9 (-2.2) 27.4 (+9.8)
NAD	79.8 (-1.4) 10.1 (+0.6)	83.8 (-2.4) 10.5 (+0.0)	92.8 (-1.5) 58.6 (+42.3)	92.4 (-2.7) 61.7 (+44.1)
i-BAU	78.3 (-2.9) 10.0 (+0.5)	84.6 (-1.6) 10.4 (-0.1)	88.5 (-5.8) 24.8 (+8.5)	90.8 (-4.3) 21.0 (+3.4)
FT-SAM	78.7 (-2.5) 9.5 (+0.0)	84.2 (-2.0) 19.5 (+9.0)	91.1 (-3.2) 58.9 (+42.6)	87.5 (-7.6) 83.2 (+65.6)
TransNTL (Ours)	**77.3 (-3.9)** **38.3 (+28.8)**	**82.6 (-3.6)** **54.0 (+43.5)**	**91.3 (-3.0)** **85.3 (+69.0)**	**90.3 (-4.8)** **89.3 (+71.7)**

Risk of NTL-based ownership verification. Further, we uncover the risk of NTL-based ownership verification. In the pre-training stage, we follow [49, 50] to add a trigger patch on the source domain data and see them as the target domain. Thus, NTL-based ownership verification can be achieved by observing the performance difference of a trained model on the data with and without trigger patch.

The attacking results on CIFAR10 and VisDA-T are shown in Tab. 2. We can see that the proposed TransNTL successfully recovers the performance on patched data, with improvements by up to 71.7% and exceeding all attack baselines. By performing TransNTL, the NTL-based ownership verification will be cracked as the attacked model behaves more like an SL model, with the performance nearly the same on the data with and without the patch.

Risk of NTL-based applicability authorization. We also reveal the risk of NTL-based applicability authorization, which aims to restrict the model generalization ability to only the authorized domain. We first pretrain NTL and CUTI to employ applicability authorization by following [49, 50]. Specifically, *the source data with an authorized patch* is regarded as the source domain, and *the union of the original source data, the generated neighborhood data*[8] *with and without the patch* is seen as the target domain.

Then, we attack the NTL-based applicability authorization models and show results in Tab. 2. By performing TransNTL, the restricted generalization of the pretrained NTL and CUTI is unleashed to unauthorized domains, with the performance increasing by up to 28.7%. TransNTL also outperforms all the attacking baselines, becoming the most serious threat to NTL-based applicability authorization.

5.2. Ablation Studies

In this section, we explore the effectiveness of main components in TransNTL by conducting ablation studies. As shown in Tab. 4, performing the vanilla fine-tuning (i.e., $\mathcal{L}_{\text{ft}}$) on 10% source domain data fails to attack both NTL and CUTI. When adding the self-distillation loss (i.e., $\mathcal{L}_{\text{ft}} + \mathcal{L}_{\text{sd}}$), the impairments on third-party domains are repaired, and the transferability barrier in NTL models begin to be broken, with the target domain performance increasing by up to 49.4%. However, the source domain performance has significant degradation (e.g., 6.7% dropping of NTL on VisDA-T→VisDA-V). As we continue to add the sharpness

[7] In each table, we highlight the best overall accuracy (i.e., the sum of accuracies of the source domain and the target domain)

[8] We follow the GAN-based method proposed in [49] to generate the neighborhood data from the original source data.

Table 3. Risks of NTL-based applicability authorization. "(P)" represents the authorized domain. "unAuths" represents the unauthorized domains. We report the source domain accuracy (%) in blue and target domain accuracy (%) in red. The accuracy drop compared to the pre-trained NTL method is shown in brackets.

	CIFAR10(P)→unAuths (SL: 83.7 / 71.9)		**VisDA-T(P)→unAuths** (SL: 95.7 / 55.1)	
	NTL	**CUTI**	**NTL**	**CUTI**
Pre-train	80.7 / 30.4	80.3 / 37.2	93.9 / 17.1	94.0 / 20.0
FTAL	80.5 (-0.2) 31.5 (+1.1)	79.9 (-0.4) 39.8 (+2.6)	93.6 (-0.3) 17.3 (+0.2)	93.2 (-0.8) 19.6 (-0.4)
RTAL	79.6 (-1.1) 34.1 (+3.7)	76.8 (-3.5) 48.1 (+10.9)	92.5 (-1.4) 18.6 (+1.5)	92.0 (-2.0) 21.5 (+1.5)
FP	80.0 (-0.7) 35.5 (+5.1)	76.2 (-4.1) 51.3 (+14.1)	91.7 (-2.2) 29.9 (+12.8)	90.0 (-4.0) 26.9 (+6.9)
NAD	79.7 (-1.0) 37.5 (+7.1)	77.4 (-2.9) 46.0 (+8.8)	93.7 (-0.2) 16.4 (-0.7)	86.3 (-7.7) 44.2 (+24.2)
i-BAU	82.1 (+1.4) 32.2 (+1.8)	76.4 (-3.9) 36.7 (-0.5)	90.2 (-3.7) 36.6 (+19.5)	87.1 (-6.9) 38.7 (+18.7)
FT-SAM	80.3 (-0.4) 36.9 (+6.5)	77.0 (-3.3) 55.0 (+17.8)	60.9 (-33.0) 27.5 (+10.4)	92.7 (-1.3) 37.7 (+17.7)
TransNTL (Ours)	**80.3 (-0.4)** **42.0 (+11.6)**	**77.6 (-2.7)** **63.4 (+26.2)**	**88.4 (-5.5)** **45.8 (+28.7)**	**89.1 (-4.9)** **47.8 (+27.8)**

term (i.e., the full TransNTL), the attack becomes more effective because the sharpness term further enhances the generalization of both the fine-tuning on the source domain and the self-distillation on third-party domains. Thus, the full TransNTL achieves the best attacking performance.

More results and analyses are shown in Appendix D.

5.3. Defending the TransNTL

Previously, our experiments reveal the risks of existing NTL methods. For completeness, in this section, we discuss how to defend the proposed TransNTL attack. We first illustrate the main principle of the defense method, and then we conduct basic experiments to show its effectiveness.

Defense method. To defend the TransNTL, we propose to *pre-repair the impairments in perturbation-based third-party domains*. Thus, by simply modifying the impairment-repair self-distillation loss term in Eq. (5), we have:

$$\mathcal{L}_{\text{Defense}} := \max_{p\in\mathcal{P}} \mathbb{E}_{(x,y)\sim\mathcal{D}_s}\left[\mathcal{L}_{\text{kl}}(f_\theta(p(x)), f_\theta(x))\right], \tag{10}$$

where $\mathcal{P}$ is the perturbation collection, $\mathcal{D}_s$ is the source domain, and $\mathcal{L}_{\text{kl}}$ is the KL divergence. We add the $\mathcal{L}_{\text{Defense}}$ to the general NTL framework and get the total objective:

$$\mathcal{L}_{\text{RobustNTL}} := \mathcal{L}_{\text{NTL}} + \lambda_{\text{df}}\mathcal{L}_{\text{Defense}}, \tag{11}$$

where λ_{df} is a trade-off weight. Intuitively, by minimizing $\mathcal{L}_{\text{RobustNTL}}$, we let the NTL model exhibits source-domain-consistent behaviours on perturbed source domain data, thus pre-fixing bugs before deployment.

Defense results. We here consider the more vulnerable method CUTI. According to Eq. (11), we revise CUTI to its robust version: R-CUTI. Experiments are conducted on CIFAR10→STL10 and VisDA-T→VisDA-V. As shown in Tab. 5, the R-CUTI is more robust against the TransNTL than CUTI, with the post-attack target domain accuracy increasing by 0.0% and 16.6% on CIFAR10→STL10 and VisDA-T→VisDA-V, respectively (significantly lower than 52.4% and 20.9% of the original CUTI). Due to the limited space, more analyses of defense are shown in Appendix E.

Table 4. Ablation studies of the proposed TransNTL. We report the source domain accuracy (%) in blue and target domain accuracy (%) in red. In brackets, we show the relative accuracy drop compared to the corresponding pre-trained NTL method.

	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
$\mathcal{L}_{\text{ft}}$	84.6 (+0.5) 10.1 (+0.0)	84.6 (+0.3) 9.9 (+0.0)	93.2 (-0.8) 5.6 (+0.1)	93.1 (-0.2) 10.3 (+0.0)
$\mathcal{L}_{\text{ft}} + \mathcal{L}_{\text{sd}}$	82.6 (-1.5) 44.8 (+34.7)	79.7 (-4.6) 59.3 (+49.4)	87.3 (-6.7) 16.9 (+11.4)	90.6 (-2.7) 28.3 (+18.0)
TransNTL (full)	**82.5 (-1.6)** **49.7 (+39.6)**	**80.2 (-4.1)** **62.3 (+52.4)**	**89.4 (-4.6)** **22.5 (+17.0)**	**91.0 (-2.3)** **31.2 (+20.9)**

Table 5. Defending the TransNTL. The source domain accuracy and target domain accuracy (%) are reported, with the accuracy drop compared to the pre-trained NTL model shown in brackets.

	CIFAR10→STL10 (SL: 86.6 / 68.5)		**VisDA-T→VisDA-V** (SL: 95.2 / 34.0)	
	CUTI	**R-CUTI**	**CUTI**	**R-CUTI**
Pre-train	84.3 / 9.9	83.6 / 10.0	93.3 / 10.3	95.4 / 7.0
TransNTL	80.2 (-4.1) 62.3 (+52.4)	83.5 (-0.1) 10.0 (+0.0)	91.0 (-2.3) 31.2 (+20.9)	92.7 (-2.7) 23.6 (+16.6)

6. Conclusion

In this paper, we focus on the robustness of NTL. We observe that the generalization of NTL models is widely impaired outside the target domain, which patterns are identified as over-confident predictions on the implicit target domain class. Motivated by these findings, we propose a TransNTL method to reveal the risk of existing NTL methods. TransNTL attacks NTL based on the finding that the slight-perturbed source domain exhibits the same impairment patterns as the target domain. The effectiveness of TransNTL is verified through extensive experiments. Finally, we discuss a feasible defense method against TransNTL, and empirically, we validate its effectiveness.

Acknowledgements

Li Shen is supported by STI 2030—Major Projects (No. 2021ZD0201405). Ziming Hong is supported by JD Technology Scholarship for Postgraduate Research in Artificial Intelligence No. SC4103. Tongliang Liu is partially supported by the following Australian Research Council projects: FT220100318, DP220102121, LP220100527, LP220200949, IC190100031. We would like to thank Dr. Zhenyi Wang from University of Maryland, College Park for providing suggestions and technical support.

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