Jailbreaking the Non-Transferable Barrier via Test-Time Data Disguising

Supplementary Material

Overview:

- In Appendix A, we present the algorithms of JailNTL* and explain how JailNTL integrates with white-box attack methods.
- In Appendix B, we provide additional details on the experimental setup.
- In Appendix C, we conduct more experiments.
- In Appendix D, we discuss the assumption of the accessibility of authorized data during attack.
- In Appendix E, we demonstrate the limitations of the proposed JailNTL.

A. Algorithms

A.1. JailNTL* Algorithm

JailNTL* presents the basic JailNTL framework, which incorporates only data-intrinsic disguising without model-guided disguising. We present the algorithmic structure of JailNTL* in Algorithm 2.

Algorithm 2: Training JailNTL*

Data: A small partition of authorized domain data $\mathcal{D}_a$; Part of unauthorized domain data $\mathcal{D}_u$.
Input: The pre-trained NTL model f_{ntl}; Initial disguising models f_d and $\hat{f}_d$; Initial discriminators f_c and $\hat{f}_c$; Number of training epochs E.

for $e = 1$ **to** E **do**
- Sample mini-batch $\mathrm{B_u}$ and $\mathrm{B_a}$ from $\mathcal{D}_u$ and $\mathcal{D}_u$;
- Generate disguised domain $f_d(\mathrm{B_u})$, $\hat{f}_d(\mathrm{B_a})$, $\hat{f}_d(f_d(\mathrm{B_u}))$, $f_d(\hat{f}_d(\mathrm{B_a}))$;
- Compute L_{adv}, L^r_{adv}, L_{cs}, L^r_{cs} with Eq. 1, 5, 3, 6 in the main paper;
- Compute L by incorporating multiple losses with Eq. 7 in the main paper;
- Update parameters f_d and $\hat{f}_d$ by minimizing L_{total} and update f_c and $\hat{f}_c$ by maximizing L_{total} with Eq. 8 in the main paper;

end for;
Output: The well-trained f^*_d

A.2. Training TransNTL with JailNTL

We integrate our black-box attack method, JailNTL, with the state-of-the-art (SOTA) attack method, TransNTL [17]. The disguised domain data generated by JailNTL is utilized to enhance TransNTL's performance. Specifically, following the implementation of TransNTL, we incorporate the disguised domain as an unlabeled authorized domain and use it to generate third-party domains $\left\{\hat{D}^g_s\right\}_{g=1}^{G}$ with diverse distribution shifts $\mathcal{P}$ from the disguised domain $\mathcal{D}_s$, where $\hat{D}^g_s = \{(p_g(x), y) \mid p_g \in \mathcal{P}, (x, y) \sim \mathcal{D}_s\}$. These generated domains are then included in the calculation of the impairment-repair self-distillation loss for each optimization iteration. We present the algorithmic structure in Algorithm 3.

Algorithm 3: Training TransNTL with JailNTL

Data: A small partition of authorized domain data $\mathcal{D}_a$; Disguised unauthorized domain data $\mathcal{D}_s$.
Input: Pre-trained NTL model f_{ntl}; Perturbation collection $\mathcal{P}$; Impairment-repair self-distillation loss weight λ_{sd}; Number of training epochs E.

for $e = 1$ **to** E **do**
- Sample mini-batches $\mathrm{B_a}$ and $\mathrm{B_s}$ from $\mathcal{D}_a$ and $\mathcal{D}_s$, respectively;
- Compute fine-tuning loss $\mathcal{L}_{ft}$ using $\mathrm{B_a}$ and its corresponding labels;
- Generate third-party domains $\hat{B}_a$, $\hat{B}_s$ from $\mathrm{B_a}$, $\mathrm{B_s}$ by applying perturbations from $\mathcal{P}$;
- Calculate self-distillation loss $\mathcal{L}_{sd}$ using $\mathrm{B_a}$, $\hat{B}_a$, $\mathrm{B_s}$, and $\hat{B}_s$;
- Compute impairment-repair fine-tuning loss $\mathcal{L}_{irft} = \lambda_{sd}\mathcal{L}_{sd} + \mathcal{L}_{ft}$;
- Update parameters of f_{ntl} by minimizing $\mathcal{L}_{irft}$;

Output: Fine-tuned model f^*_{ntl}

B. Experiment Detail

B.1. Baseline

For pre-trained NTL methods, we include all open-source NTL methods as baselines, including the NTL [52] and CUTI [53] methods. For attack NTL methods, we incorporate white-box attack methods which have the same data setup as our JailNTL, including the basic fine-tuning methods FTAL and RTAL [1] and the state-of-the-art (SOTA) method TransNTL [17]. For all the experiments, we use the official implementations of NTL methods (NTL[4], CUTI[5]) and attack methods (FTAL, RTAL and TransNTL)[6].

[4] https://github.com/conditionWang/NTL
[5] https://github.com/LyWang12/CUTI-Domain
[6] https://github.com/tmllab/2024_CVPR_TransNTL

B.2. Datasets

Following the NTL baseline [52, 53], we conduct experiments on CIFAR10 [32], STL10 [8], and VisDA-2017 [41]. We present samples of these datasets as shown in Fig. 7. Details of these datasets are as follows:

- CIFAR10 & STL10: The CIFAR10 dataset comprises 32×32 color images in 10 classes, consisting of 6 animal classes and 4 vehicle classes. The STL10 dataset contains 96×96 color images in 10 classes, with a similar class distribution to CIFAR10. We conduct experiments on both CIFAR10 → STL10 and STL10 → CIFAR10 transfer tasks.
- VisDA-2017: VisDA-2017 is a simulation-to-real dataset containing 12 classes with distinct training, validation, and testing domains. The training images are synthetic renderings of 3D models under various conditions, while the validation images are collected from MSCOCO. We conduct experiments on the VisDA-T → VisDA-V.

Consistent with the NTL baseline [17, 52], we resize all images to a resolution of 64×64 pixels for the NTL tasks.

B.3. Implementation of the Disguising Network

We build the disguising model based on the ResNet [13, 61] structure which consists of two downsampling layers, nine residual blocks, and two upsampling layers, along with the instance normalization layers. We apply a kernel size of 3 within the ResNet blocks and a kernel size of 7 in the sampling layers, which allows for effective feature extraction. For the discriminators, we follow the PatchGAN of the pix2pix [27] method for efficiency.

B.4. Optimization

For the optimization of JailNTL, we employ the Adam optimizer with an initial learning rate of 0.0002. By employing zero-order gradient estimation via finite difference approximation [30], we apply model-guided loss to the disguising model without back-propagating through the NTL model, thereby following the setting of black-box attack.

C. More Experiment

We present more experimental results in this section. In Appendix C.1, we present the model's class balance and confidence across various datasets, NTL models, and network backbones. In Appendix C.2, we conduct an ablation study on data-intrinsic disguising. In Appendix C.3, we show the influence of hyperparameters on JailNTL. Then, in Appendix C.4, we provide additional model visualization results using t-SNE and GradCAM to analyze how JailNTL affects the NTL model. Finally, in Appendices C.5 and C.6, we evaluate JailNTL on different backbones and with less authorized domain data, demonstrating its effectiveness across scenarios.

C.1. Confidence and Classification Balance Discrepancies in NTL Models

This subsection presents a comprehensive analysis of the confidence and classification balance discrepancies exhibited by Non-Transferable Learning (NTL) models across various scenarios. We examine these discrepancies between authorized and unauthorized domains under different conditions, including diverse datasets (CIFAR10 [32], STL10 [8], and VisDA [41]), distinct methods (NTL [52] and CUTI [53]), and different network backbones (VGG, VGGbn [47], and ResNet34 [13]). Our observations consistently reveal significant differences in classification balance and confidence levels between authorized and unauthorized domains across all scenarios. These findings support the universality of our proposed model-guided disguise approach, which leverages these discrepancies.

Class Balance As shown in Fig. 8, we observed that the NTL model predicts unbalanced classes (preferring one or two classes) on the unauthorized domain, while predicting balanced classes on authorized domains. This phenomenon was consistently observed across different backbones (VGG, VGGbn, and ResNet34) in various datasets, including CIFAR10, STL10, and VisDA, for both NTL and CUTI methods.

Confidence We employ two types of metrics to evaluate the model's confidence: maximum logits [17] (Eq. 16) and the entropy of softmax logits [45] (Eq. 9 in the main paper). Figs. 9 and 10 illustrate the distribution of confidence for the NTL model, revealing a notable difference between the unauthorized and the authorized domain across different backbones (VGG, VGGbn, and ResNet34) in various tasks.

$$E_{cf}(x) = \max(f_{ntl}(x)) \tag{16}$$

C.2. More Ablation Studies

In this section, we conduct ablation studies to demonstrate the effectiveness of data-intrinsic disguising in JailNTL. The full data-intrinsic disguising includes a forward process, a feedback network, and a bidirectional structure.

As shown in the Tab. 4, JailNTL with model-guided disguising and only the forward process in data-intrisic disguising (i.e., without the feedback network and bidirectional structure, denoted as **Forward**) shows poor attack performance. Then, adding feedback network to JailNTL (denoted as **+ Feedback**) improves attack performance, with an increase in unauthorized domain accuracy from 14.9% to 49.2% in CIFAR10 → STL10 CUTI task. Further, the introduction of the bidirectional network to JailNTL (denoted as **Full**) achieves the highest accuracy in the unauthorized domain while maintaining performance in the authorized domain.

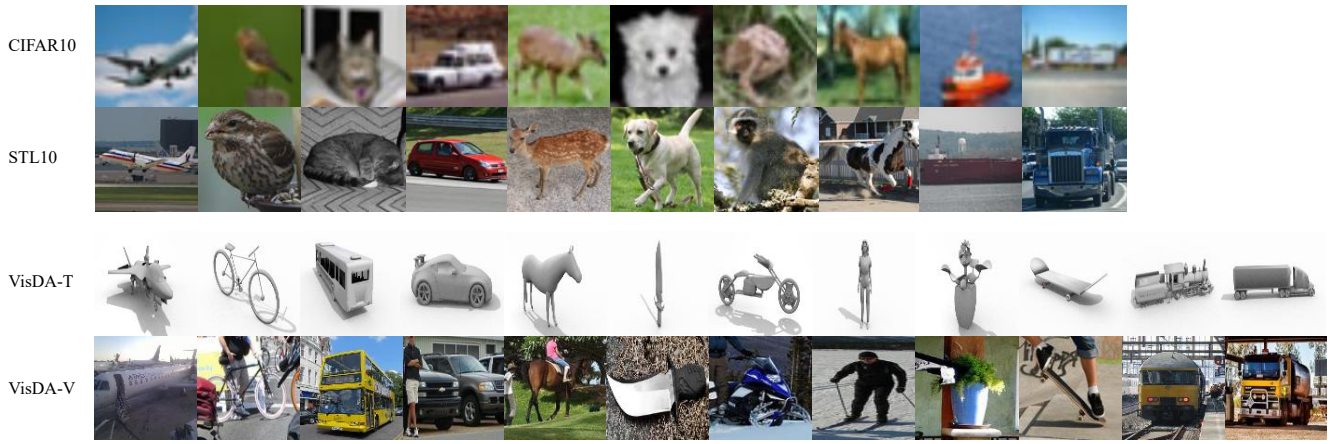


Figure 7. Examples of NTL tasks: From top to bottom, we present samples from CIFAR10, STL10, VisDA-Train, and VisDA-Validation datasets. These datasets serve as authorized or unauthorized domains in NTL tasks, exhibiting distinct style differences. Specifically, both CIFAR10 and STL10 contain photo-realistic or real-world images, with CIFAR10 having a lower resolution (32×32 pixels) compared to STL10 (96×96 pixels). VisDA-T consists of 2D images synthesized from 3D models with diverse viewing angles and lighting variations, while VisDA-V comprises photo-realistic or real-world photographs.

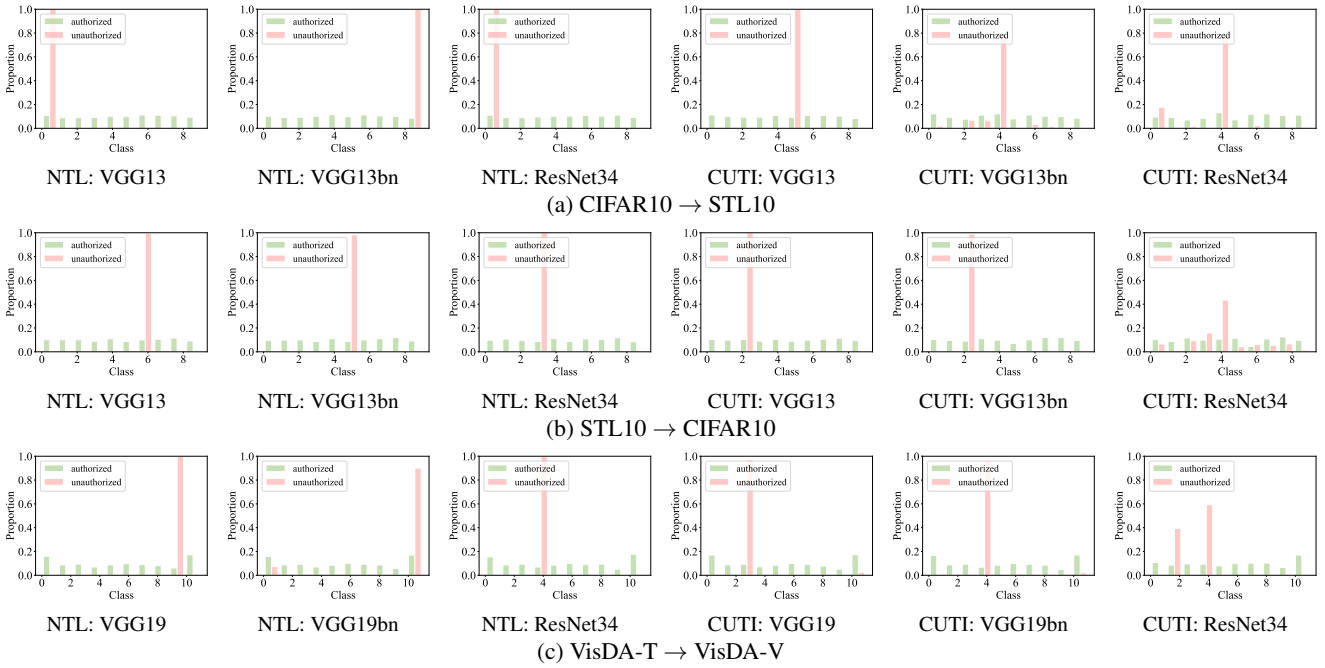
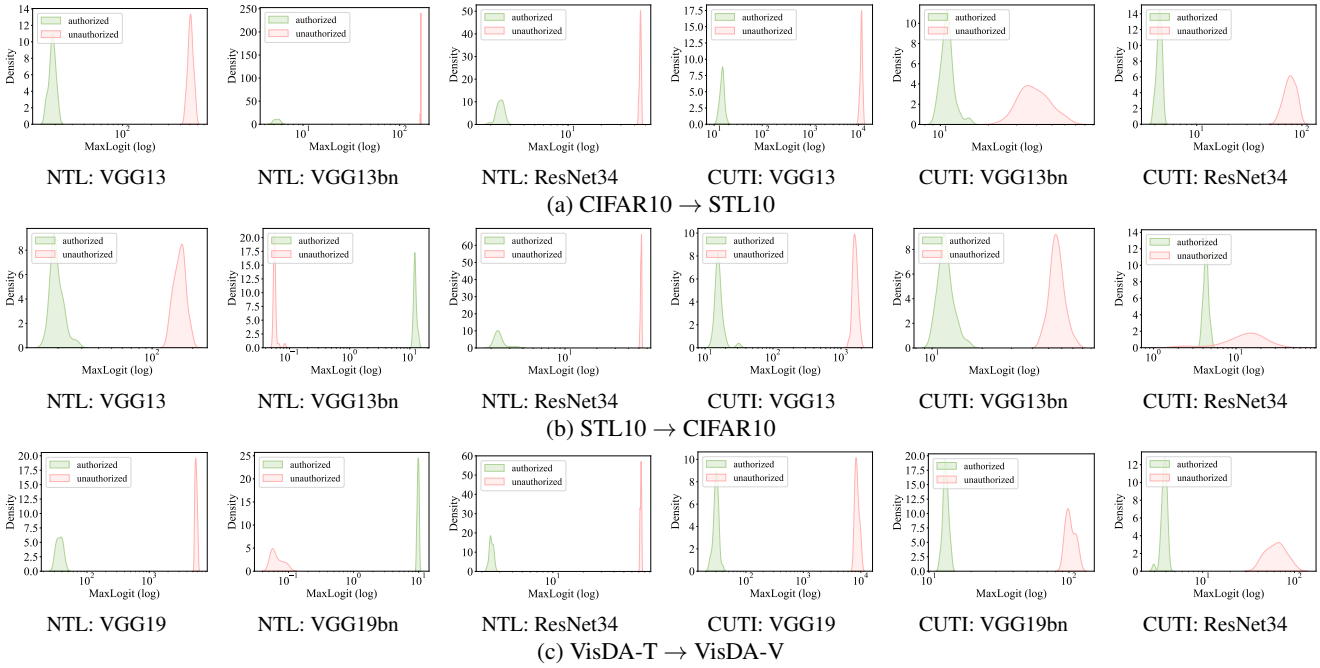


Figure 8. The analysis of class balance of NTL and CUTI across three different tasks. We present CIFAR10 → STL10 task in subfigure (a), STL10 → CIFAR10 task in subfigure (b), and VisDA-T → VisDA-V task in subfigure (c). For each task, we show results for both NTL and CUTI methods using different network architectures. We use green to represent the authorized domain, and red to represent the unauthorized domain.

C.3. Influences of Hyperparameters

In this section, we analyze the influence of the hyperparameters λ_{cf} and λ_{ba} in the JailNTL methods. These parameters control the importance of the confidence loss L_{cf} (Eq. (10)) and class balance loss L_{ba} (Eq. (13)), respectively. To evaluate their impact, we conducted two sets of experiments. First, we keep the value of λ_{ba} and assigned values to λ_{cf} from the set [0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0001]. Subsequently, we repeated the process by assigning the values above to λ_{ba}. As illustrated in Fig. 11, the performance of JailNTL remains stable across various values of λ_{cf} and λ_{ba}. This stability demonstrates the robustness of our method for these hyperparameters.



NTL: VGG13 NTL: VGG13bn NTL: ResNet34 CUTI: VGG13 CUTI: VGG13bn CUTI: ResNet34

(a) CIFAR10 → STL10

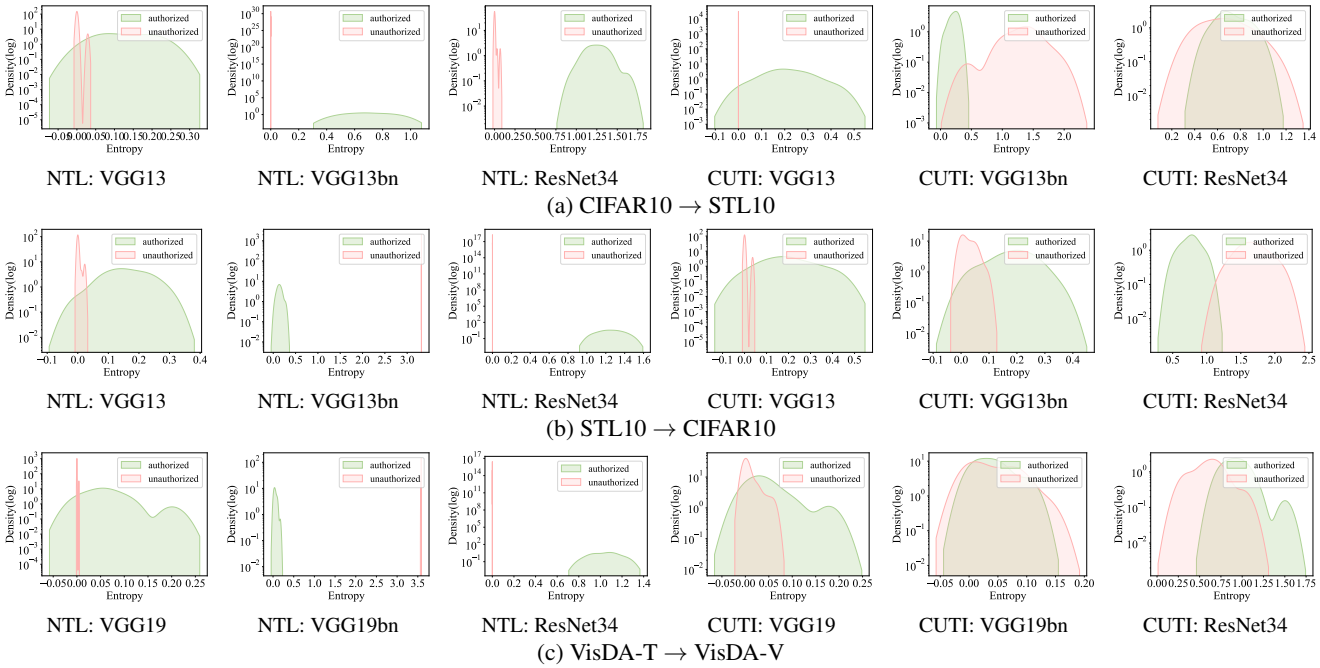
NTL: VGG13 NTL: VGG13bn NTL: ResNet34 CUTI: VGG13 CUTI: VGG13bn CUTI: ResNet34

(b) STL10 → CIFAR10

NTL: VGG19 NTL: VGG19bn NTL: ResNet34 CUTI: VGG19 CUTI: VGG19bn CUTI: ResNet34

(c) VisDA-T → VisDA-V

Figure 9. The maximum logits of NTL and CUTI in three different tasks. We employ maximum logits as a metric to assess the model's confidence. We present CIFAR10 → STL10 task in subfigure (a), STL10 → CIFAR10 task in subfigure (b), and VisDA-T → VisDA-V task in subfigure (c). For each task, we show results for both NTL and CUTI methods using different network architectures. We use green to represent the authorized domain and red to represent the unauthorized domain.



NTL: VGG13 NTL: VGG13bn NTL: ResNet34 CUTI: VGG13 CUTI: VGG13bn CUTI: ResNet34

(a) CIFAR10 → STL10

NTL: VGG13 NTL: VGG13bn NTL: ResNet34 CUTI: VGG13 CUTI: VGG13bn CUTI: ResNet34

(b) STL10 → CIFAR10

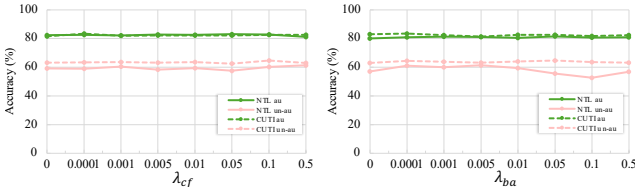
NTL: VGG19 NTL: VGG19bn NTL: ResNet34 CUTI: VGG19 CUTI: VGG19bn CUTI: ResNet34

(c) VisDA-T → VisDA-V

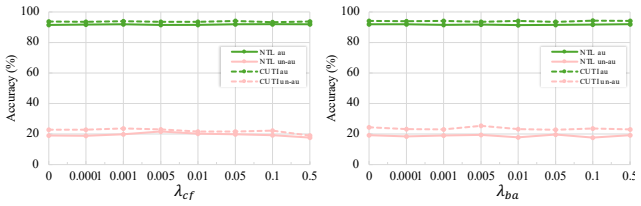
Figure 10. The confidence (entropy) of NTL and CUTI in three different tasks. We employ the entropy of softmax logits as a metric to assess the model's confidence. We present CIFAR10 → STL10 task in subfigure (a), STL10 → CIFAR10 task in subfigure (b), and VisDA-T → VisDA-V task in subfigure (c). For each task, we show results for both NTL and CUTI methods using different network architectures. We use green to represent the authorized domain and red to represent the unauthorized domain. Due to the significant differences in entropy distribution between the validation domain and the unvalidated domain, we apply *a logarithmic scale for the density axis* to clearly display the distributions of both.

Table 4. Ablation Studies of Data-intrinsic Disguising. We present authorized domain accuracy (%) in black, and unauthorized domain accuracy (%) in red. Change *vs* pre-trained NTL models are shown in brackets. "Full" represents the model-guided disguising and complete data-intrinsic disguising approach, which incorporates both the forward process and the feedback network, as well as a bidirectional structure.

Domain	NTL	Pre-Train	Forward	+ Feedback	Full
CIFAR10 → STL10	NTL	85.6 9.8	23.4 (-62.2) 28.3 (+18.5)	30.5 (-55.1) 25.5 (+15.7)	81.2 (-4.4) 61.4 (+51.6)
	CUTI domain	85.8 9.0	27.5 (-58.3) 14.9 (+5.9)	75.9 (-9.9) 49.2 (+40.2)	82.5 (-3.3) 64.7 (+55.7)
STL10 → CIFAR10	NTL	84.5 11.0	21.6 (-62.9) 10.9 (-0.1)	60.5 (-24.0) 16.1 (+5.1)	83.7 (-0.8) 39.8 (+28.8)
	CUTI domain	88.3 9.9	16.3 (-72.0) 10.0 (+0.1)	78.8 (-9.5) 11.3 (+1.4)	85.6 (-2.7) 43.5 (+33.6)
VisDA-T → VisDA-V	NTL	93.0 6.7	73.8 (-19.2) 9.1 (+2.4)	89.8 (-3.2) 14.8 (+8.1)	91.5 (-1.5) 21.7 (+15.0)
	CUTI domain	94.7 10.1	82.7 (-12.0) 8.5 (-1.6)	92.0 (-2.7) 17.3 (+7.2)	93.6 (-1.1) 25.4 (+15.4)



(a) CIFAR10 → STL10



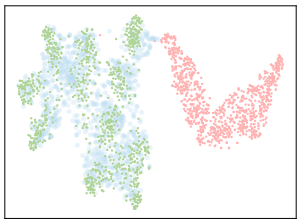
(b) VisDA-T → VisDA-V

Figure 11. Influence of λ_{cf} and λ_{ba}

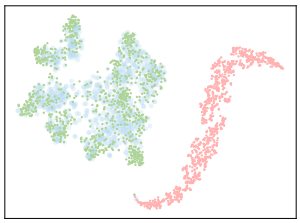
C.4. More Visualization Analysis

In this section, we present extended visualizations to illustrate further the effects of JailNTL on the NTL model's attention and feature space representation on different domains. We employ Gradient-weighted Class Activation Mapping (GradCAM [44]) to visualize the attention and t-distributed Stochastic Neighbor Embedding (t-SNE [49]) to represent the NTL feature space. These visualizations are extended to encompass various domains for both NTL and CUTI methods, providing a more comprehensive analysis of our approach's performance.

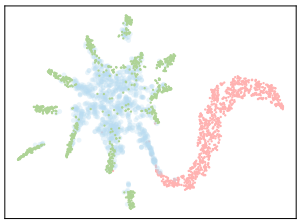
t-SNE Feature Visualization. As shown in Fig. 12, we observe a clear separation between the authorized (green) and unauthorized (red) domains, indicating a significant domain gap that typically hinders knowledge transfer. Notably, the disguised domain samples (blue) consistently



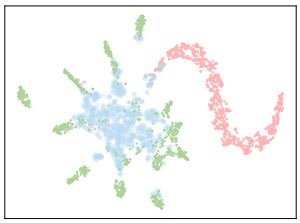
(a) CIFAR10 → STL10 NTL



(b) CIFAR10 → STL10 CUTI



(c) VisDA-T → VisDA-V NTL



(d) VisDA-T → VisDA-V CUTI

Figure 12. t-SNE visualization in different tasks. We present data from the authorized domain as green, data from the unauthorized domain as red, and data from the disguised domain as blue.

cluster closely with the authorized domain samples while remaining distinctly separate from the unauthorized domain. This visualization provides compelling evidence for the effectiveness of JailNTL. By generating the disguised domains that closely align with the authorized domain's distribution, JailNTL successfully jailbreaks the non-transferability barrier.

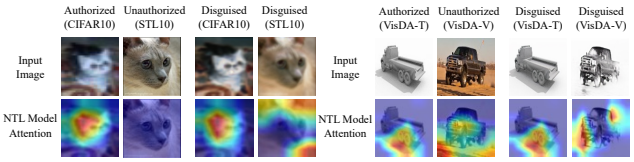
GradCAM Attention Visualization. We visualize the effect of JailNTL on the NTL model's attention using GradCAM [44]. As shown in Fig. 13, The first row of the subfigure presents the input images, comprising samples from the original authorized, unauthorized, and disguised domains. The second row of the subfigure depicts the model's attention using GradCAM, where cooler colors (blue) denote areas of low attention, while warmer colors (red) highlight regions of high attention. Through effective disguising, we successfully altered the model's attention in the disguised unauthorized domain. The Grad-CAM visualizations reveal that the attention map for the disguised image closely resembles that of the original authorized image, exhibiting high attention to the object. This contrasts sharply with the low attention observed on the object in the unauthorized image. These findings demonstrate that the JailNTL method successfully disguised the domain, manipulated the model's attention, and achieved an effective NTL attack.

C.5. Effectiveness of JailNTL with Fewer Authorized Domain Data

In this section, we analyze the performance of JailNTL compared to other attack methods when less (0.5%) authorized domain data are available. As shown in Tab. 5,

Table 5. Attack the NTL by using RTAL, FTAL, TransNTL, and JailNTL with **0.5%** of the authorized domain data. We represent authorized domain accuracy(%) in black and the unauthorized domain accuracy (%) in red. The change in accuracy compared to the pre-trained model is indicated in brackets. We evaluate *both the accuracy increase in unauthorized domain and the performance drop in uthorized domain*. Best results are highlighted in red background and second-best in yellow . * denotes white-box attacks and † indicates black-box attacks.

Domain	NTL method	Pre-trained	RTAL*	FTAL*	TransNTL*	JailNTL†
CIFAR10 → STL10	NTL	85.6 9.8	61.3 (-24.3) 9.7 (-0.1)	85.9 (+0.3) 9.8 (+0.0)	74.6 (-11.0) 22.5 (+12.7)	80.1 (-5.5) 54.6 (+44.8)
	CUTI domain	85.8 9.0	66.9 (-18.9) 9.1 (+0.1)	86.7 (+0.9) 9.0 (+0.0)	76.4 (-9.4) 60.6 (+51.6)	80.9 (-4.9) 63.0 (+54.0)
STL10 → CIFAR10	NTL	84.5 11.0	67.6 (-16.9) 10.9 (-0.1)	84.9 (+0.4) 11.0 (+0.0)	66.2 (-18.3) 29.1 (+18.1)	83.0 (-1.5) 38.8 (+27.8)
	CUTI domain	88.3 9.9	79.0 (-9.3) 10.7 (+0.8)	88.2 (-0.1) 9.9 (+0.0)	76.1 (-12.2) 57.0 (+47.1)	86.4 (-1.9) 44.9 (+35.0)
VisDA-T → VisDA-V	NTL	93.0 6.7	85.1 (-7.9) 7.0 (+0.3)	93.0 (+0.0) 6.7 (+0.0)	65.6 (-27.4) 10.8 (+4.1)	90.9 (-2.1) 20.9 (+14.2)
	CUTI domain	94.7 10.0	93.6 (-1.1) 11.3 (+1.3)	95.2 (+0.5) 10.4 (+0.4)	84.6 (-10.1) 29.2 (+19.2)	93.8 (-0.9) 20.7 (+10.7)



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

Figure 13. Visualization of JailNTL's effect on model attention using GradCAM.

JailNTL effectively recovers performance in the unauthorized domain for all tasks, achieving an increase of up to 44.8% in NTL and up to 54.0% in CUTI. Meanwhile, it successfully maintains performance in the authorized domain, with minimal decreases of only 1.5% in NTL and 0.9% in CUTI. In contrast, existing fine-tuning methods (RTAL and FTAL [1]) fail to recover performance in unauthorized domains for both NTL and CUTI. The SOTA white-box attack TransNTL [17] can partially recover the performance of unauthorized domains, while presents a significant decrease in the performance of the authorized domain. Overall, our black-box attack JailNTL still outperforms existing white-box attack baselines with access to only 0.5% of authorized domain data.

C.6. Effectiveness of JailNTL Across Backbones

In this section, we present the performance of JailNTL on various backbone architectures (VGGbn [47], ResNet34, and WRN502 [13]), extending beyond the VGG results presented in the main paper. As shown in Tab. 6, JailNTL maintains stable performance across different backbone networks. Specifically, JailNTL effectively improves performance in the unauthorized domain across various NTL backbones while maintaining performance in the authorized domain, thereby demonstrating its effectiveness to diverse NTL network architectures.

Table 6. Effectiveness of JailNTL on Various Backbones. We present authorized domain accuracy (%) in black, and unauthorized domain accuracy (%) in red. Change *vs* pre-trained NTL models are shown in brackets.

Domain	NTL	VGG	VGGbn	ResNet34	WRN502
CIFAR10 → STL10	NTL	81.2 (-4.4) 61.4 (+51.6)	76.4 (-6.5) 49.2 (+39.8)	81.9 (-3.8) 61.9 (+52.0)	85.2 (-3.2) 68.2 (+58.1)
	CUTI domain	82.5 (-3.3) 64.7 (+55.7)	82.6 (-6.3) 61.9 (+41.5)	80.0 (-2.4) 60.1 (+55.7)	84.0 (-2.3) 64.0 (+50.3)
VisDA-T → VisDA-V	NTL	91.5 (-1.5) 21.7 (+15.0)	97.2 (-0.1) 21.6 (+13.2)	94.5 (-0.2) 14.3 (+5.7)	95.4 (-1.4) 19.0 (+12.5)
	CUTI domain	93.6 (-1.1) 25.4 (+15.4)	96.5 (-0.3) 19.1 (+8.7)	87.6 (-1.0) 17.7 (+14.3)	90.0 (-1.4) 17.9 (+10.9)

D. Discussion of the Data Accessibility

When attack, we follow [17] to assume that attackers can access a small part of authorized data. We argue this assumption is true and practical in black-box scenario.

As illustrated in Sec. 1 and Fig. 1(a), NTL aims to establish a "*non-transferable barrier*" [17, 19, 53] to restrict the model's generalization from an *authorized domain* to an *unauthorized domain*. In this way, NTL can protect model IP by preventing unauthorized usage, such as applications on illegal data or in unapproved environments.

Usually, in black-box scenario (e.g., online APIs [21]), only the **authorized users** can (i) *access some authorized data* and (ii) *have the access to use the black-box NTL model* **at the same time**. However, the following situations may still pose potential risks:

- **Access stolen**. Both (i) accesses to *authorized data* and (ii) accesses to *use the black-box NTL model* can either *be intentionally leaked by authorized users* or *stolen by thieves*. In such situations, the unauthorized users who obtain both the data and model access may try to crack the

authorization limitations of NTL models for any unauthorized data.

- **Malicious authorized users**. Even if we exclude the situation of access stolen, there still remains a risk that authorized users try to crack the authorization limitations to apply the NTL model to unauthorized data. That is, authorized users act as attackers and try to jailbreak the non-transferable barrier.

In above situations, the attackers (unauthorized users or authorized users) can access a small part of authorized data.

E. Limitations

In this paper, we adopt the settings used in previous studies on black-box attacks [4], which allow attackers to obtain logits from the NTL model. When attackers can only access prediction labels, removing the *confidence loss* still yields good performance (see in Sec. 4.3). Additionally, the *class balance loss* in model-guided disguising is designed for scenarios with class-balanced authorized and unauthorized domains. For unbalanced domain distributions, users can omit this component without significantly compromising the model's performance (as demonstrated in Sec. 4.3).

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. 3, 6, 7, 1

[2] Olaoluwa Adigun and Bart Kosko. Training generative adversarial networks with bidirectional backpropagation. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 1178–1185. IEEE, 2018. 4

[3] Massimo Bertolini, Davide Mezzogori, Mattia Neroni, and Francesco Zammori. Machine learning for industrial applications: A comprehensive literature review. *Expert Systems with Applications*, 175:114820, 2021. 1

[4] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Proceedings of the European conference on computer vision (ECCV)*, pages 154–169, 2018. 2, 3, 7

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1

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

[7] Shuhuang Chen, Dingjie Fu, Shiming Chen, Shuo Ye, Wenjin Hou, and Xinge You. Causal visual-semantic correlation for zero-shot learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 4246–4255, 2024. 3

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 6, 2

[9] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. *Advances in neural information processing systems*, 32, 2019. 5

[10] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 5

[11] Ida Merete Enholm, Emmanouil Papagiannidis, Patrick Mikalef, and John Krogstie. Artificial intelligence and business value: A literature review. *Information Systems Frontiers*, 24(5):1709–1734, 2022. 1

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2, 3

[13] 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*, pages 770–778, 2016. 2, 6

[14] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 3

[15] Timothy O Hodson. Root mean square error (rmse) or mean absolute error (mae): When to use them or not. *Geoscientific Model Development Discussions*, 2022:1–10, 2022. 5

[16] 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 *International Joint Conferences on Artificial Intelligence Organization*, pages 956–963, 2022. 4

[17] Ziming Hong, Li Shen, and Tongliang Liu. Your transferability barrier is fragile: Free-lunch for transferring the non-transferable learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28805–28815, 2024. 1, 2, 3, 5, 6, 7, 8

[18] 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*, 2024. 1, 3

[19] Ziming Hong, Yongli Xiang, and Tongliang Liu. Toward robust non-transferable learning: A survey and benchmark. *arXiv preprint arXiv:2502.13593*, 2025. 1, 2, 6

[20] Wenjin Hou, Shiming Chen, Shuhuang Chen, Ziming Hong, Yan Wang, Xuetao Feng, Salman Khan, Fahad Shahbaz Khan, and Xinge You. Visual-augmented dynamic semantic prototype for generative zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23627–23637, 2024. 3

[21] Zixuan Hu, Li Shen, Zhenyi Wang, Baoyuan Wu, Chun Yuan, and Dacheng Tao. Learning to learn from apis: black-box data-free meta-learning. In *International Conference on Machine Learning*, pages 13610–13627. PMLR, 2023. 1, 6

[22] 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*, pages 1501–1510, 2017. 3

[23] 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. *arXiv preprint arXiv:2207.03162*, 2022. 1

[24] 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*, pages 16175–16185, 2023. 3

[25] Zhuo Huang, Muyang Li, Li Shen, Jun Yu, Chen Gong, Bo Han, and Tongliang Liu. Winning prize comes from losing tickets: Improve invariant learning by exploring variant parameters for out-of-distribution generalization. *International Journal of Computer Vision*, 133(1):456–474, 2025. 1

[26] Minyoung Huh, Shao-Hua Sun, and Ning Zhang. Feedback adversarial learning: Spatial feedback for improving generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1476–1485, 2019. 2, 4

[27] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[28] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021. 3

[29] Minguk Jang, Sae-Young Chung, and Hye Won Chung. Test-time adaptation via self-training with nearest neighbor information. *arXiv preprint arXiv:2207.10792*, 2022. 3

[30] Sanjay Kariyappa, Atul Prakash, and Moinuddin K Qureshi. Maze: Data-free model stealing attack using zeroth-order gradient estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13814–13823, 2021. 5, 2

[31] Konstantina Kourou, Themis P Exarchos, Konstantinos P Exarchos, Michalis V Karamouzis, and Dimitrios I Fotiadis. Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal*, 13:8–17, 2015. 1

[32] Alex Krizhevsky, Geoffrey Hinton, and et al. Learning multiple layers of features from tiny images, 2009. 6, 2

[33] Runqi Lin, Bo Han, Fengwang Li, and Tongliang Liu. Understanding and enhancing the transferability of jailbreaking attacks. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[34] Yexiong Lin, Yu Yao, Xiaolong Shi, Mingming Gong, Xu Shen, Dong Xu, and Tongliang Liu. Cs-isolate: Extracting hard confident examples by content and style isolation. *Advances in Neural Information Processing Systems*, 36: 58556–58576, 2023. 3

[35] Abhishek Mishra. *Machine learning in the AWS cloud: Add intelligence to applications with Amazon Sagemaker and Amazon Rekognition*. John Wiley & Sons, 2019. 1

[36] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018. 3

[37] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees GM Snoek, and Ling Shao. Latent embedding feedback and discriminative features for zero-shot classification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 479–495. Springer, 2020. 2, 4

[38] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022. 3

[39] John D Owens, Mike Houston, David Luebke, Simon Green, John E Stone, and James C Phillips. Gpu computing. *Proceedings of the IEEE*, 96(5):879–899, 2008. 1

[40] Boyang Peng, Sanqing Qu, Yong Wu, Tianpei Zou, Lianghua He, Alois Knoll, Guang Chen, and Changjun Jiang. Map: Mask-pruning for source-free model intellectual property protection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23585–23594, 2024. 1, 3

[41] 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. 1, 6, 2

[42] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. Mlaas: Machine learning as a service. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pages 896–902. IEEE, 2015. 1

[43] Iqbal H Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2 (3):160, 2021. 1

[44] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 8, 5

[45] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948. 5, 2

[46] Abhinav Sharma, Arpit Jain, Prateek Gupta, and Vinay Chowdary. Machine learning applications for precision agriculture: A comprehensive review. *IEEe Access*, 9:4843–4873, 2020. 1

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 6

[48] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020. 3

[49] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 8, 5

[50] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020. 3

[51] Haotian Wang, Haoang Chi, Wenjing Yang, Zhipeng Lin, Mingyang Geng, Long Lan, Jing Zhang, and Dacheng Tao. Domain specified optimization for deployment authorization. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5072–5082. IEEE, 2023. 1

[52] Lixu Wang, Shichao Xu, Ruiqi Xu, Xiao Wang, and Qi Zhu. Non-transferable learning: A new approach for model ownership verification and applicability authorization. *arXiv preprint arXiv:2106.06916*, 2021. 1, 2, 3, 6, 7

[53] 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*, pages 20475–20484, 2023. 1, 3, 6, 7, 2

[54] Enneng Yang, Zhenyi Wang, Li Shen, Nan Yin, Tongliang Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Continual learning from a stream of apis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[55] Yu Yao, Tongliang Liu, Mingming Gong, Bo Han, Gang Niu, and Kun Zhang. Instance-dependent label-noise learning under a structural causal model. *Advances in Neural Information Processing Systems*, 34:4409–4420, 2021. 3

[56] Shuo Ye, Shujian Yu, Wenjin Hou, Yu Wang, and Xinge You. Coping with change: Learning invariant and minimum sufficient representations for fine-grained visual categorization. *Computer Vision and Image Understanding*, 237: 103837, 2023. 3

[57] Guangtao Zeng and Wei Lu. Unsupervised non-transferable text classification. *arXiv preprint arXiv:2210.12651*, 2022. 1, 3

[58] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia conference on computer and communications security*, pages 159–172, 2018. 1

[59] 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

[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*, 45(4):4396–4415, 2022. 1, 3

[61] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2, 4