Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)
Survey Track

Toward Robust Non-Transferable Learning: A Survey and Benchmark

Ziming Hong[*], **Yongli Xiang**[*] and **Tongliang Liu**[†]
Sydney AI Centre, The University of Sydney
hoongzm@gmail.com, yxia0023@uni.sydney.edu.au, tongliang.liu@sydney.edu.au

Abstract

Over the past decades, researchers have primarily focused on improving the generalization abilities of models, with limited attention given to regulating such generalization. However, the ability of models to generalize to unintended data (e.g., harmful or unauthorized data) can be exploited by malicious adversaries in unforeseen ways, potentially resulting in violations of model ethics. Non-transferable learning (NTL), a task aimed at reshaping the generalization abilities of deep learning models, was proposed to address these challenges. While numerous methods have been proposed in this field, a comprehensive review of existing progress and a thorough analysis of current limitations remain lacking. In this paper, we bridge this gap by presenting the first comprehensive survey on NTL and introducing `NTLBench`, the first benchmark to evaluate NTL performance and robustness within a unified framework. Specifically, we first introduce the task settings, general framework, and criteria of NTL, followed by a summary of NTL approaches. Furthermore, we emphasize the often-overlooked issue of robustness against various attacks that can destroy the non-transferable mechanism established by NTL. Experiments conducted via `NTLBench` verify the limitations of existing NTL methods in robustness. Finally, we discuss the practical applications of NTL, along with its future directions and associated challenges.

1 Introduction

Throughout much of deep learning (DL) history, researchers have primarily focused on improving generalization abilities [Liu *et al.*, 2021a; Zhuang *et al.*, 2020]. With advancements in novel techniques, the availability of high-quality data, and the expansion of model sizes and computational resources, DL models have demonstrated increasingly strong generalization, extending from in-distribution to out-of-distribution (OOD) scenarios [Wang *et al.*, 2022a; Radford *et al.*, 2021; Kaplan *et al.*, 2020]. This facilitates the application of DL in complex real-world scenarios. However, limited attention has been given to regulating models' generalization abilities, while strong-enough yet unconstrained generalization abilities may pose misuse risks. Specifically, the generalization of deep models to unintended data (e.g., unauthorized or harmful data) can be exploited by malicious adversaries in unexpected ways. This raises concerns regarding the regulating of powerful DL models, including issues related to model ethic [Li *et al.*, 2023], safety alignment [Ouyang *et al.*, 2022; Huang *et al.*, 2024], model privacy and intellectual property [Sun *et al.*, 2023; Jiang *et al.*, 2024], among others.

Non-transferable learning (NTL) [Wang *et al.*, 2022b], a task aimed at reshaping the generalization abilities of DL models, was proposed to address these challenges. Its goal is to prevent the model's generalization to specific target domains or tasks (such as harmful or unauthorized domains) while preserving its normal functionality on a source domain. Although numerous NTL methods have been proposed recently (e.g., Wang *et al.* [2023b], Hong *et al.* [2024b]), a comprehensive summary of existing progress in this field and an thorough analysis of current limitations is still lacking.

In this paper, we bridge this gap by presenting the first comprehensive survey of NTL. We first introduce the task settings, general framework and criteria of NTL (Section 2), followed by a summary of existing NTL approaches according to their strategies to implement non-transferability in two settings (Section 3). Then, we highlight the often-overlooked robustness against diverse attacks that can destroy the non-transferable mechanism established by NTL (Section 4).

In addition, we propose the first benchmark (`NTLBench`) to integrate 5 state-of-the-art (SOTA) and open-source NTL methods and 3 types of post-training attacks (15 attack methods) in a unified framework, as illustrated in Figure 1. Our `NTLBench` supports running NTL and attacks on 9 datasets (more than 116 domain pairs), 5 network architecture families, providing overall at least 40,000 experimental configurations for comprehensive evaluation. Main results obtained from `NTLBench` verify the unsatisfactory robustness of existing NTL methods in dealing with various post-training attacks (Section 5). Finally, we discuss applications, related work and future directions and challenges (Sections 6 to 8).

We believe that our survey and `NTLBench` can drive the development of robust NTL methods and facilitate their applications in trustworthy model deployment scenarios. Our major contributions are summarized as three folds: **(i) Comprehensive review:** We conduct a systematic review of ex-

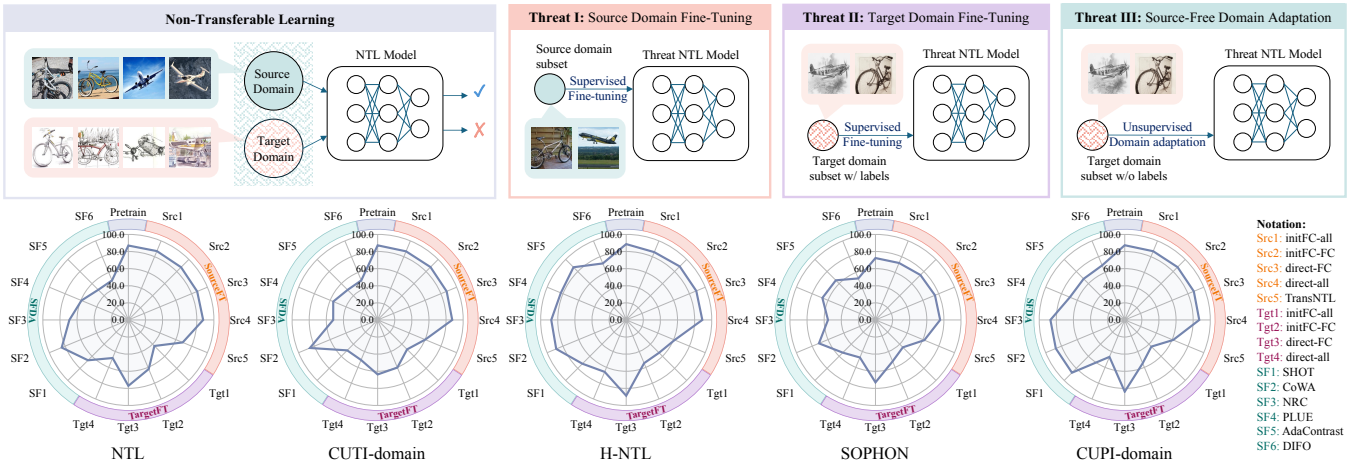

Figure 1: We systematically review non-transferable learning (NTL) and introduce `NTLBench`, an unified framework for benchmarking NTL. This figure compares 5 methods (NTL, CUTI-domain, H-NTL, SOPHON, CUPI-domain) on CIFAR & STL with VGG-13, evaluating pre-training performance and robustness against 5 source domain fine-tuning attacks, 4 target domain fine-tuning attacks, and 6 source-free domain adaptation attacks (higher value means better robustness). `NTLBench` is released at https://github.com/tmllab/NTLBench.

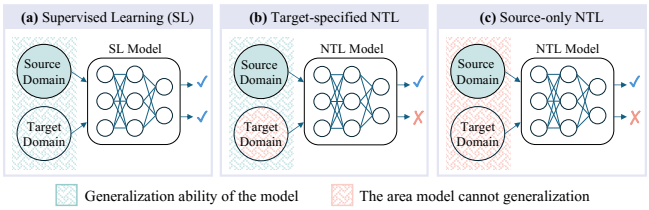

Figure 2: Comparison of (a) supervised learning (SL), (b) target-specified non-transferable learning (NTL), and (c) source-only NTL.

isting NTL works, including settings, framework, criteria, approaches, and applications. We emphasize the robustness challenges of NTL from three aspects, according to the data accessibility of different attackers. **(ii) Codebase:** We propose `NTLBench` to benchmark 5 SOTA and open-source NTL methods, covering standard assessments (5 networks and 9 datasets) and examining robustness against 15 attacks from 3 attack settings. **(iii) Evaluation and analysis:** We use `NTLBench` to fairly evaluate 5 SOTA NTL methods, covering the performance and robustness against diversity attacks. Our results identify the limitation of existing NTL methods in dealing with complex datasets and diverse attacks.

2 Preliminary

2.1 Problem Setup

In NTL, we generally consider a source domain and a target domain, where we want to keep the performance on the source domain (similar to supervised learning (SL) performance) and degrade performance on the target domain, thus implementing the resistance of generalization from the source domain to the target domain.

According to whether the target domain is known in the training stage, NTL could be divided into two settings [Wang *et al.*, 2022b]: **(i)** *target-specified NTL*, which assumes the target domain is known and aims to restrict the model generalization toward the pre-known target domain, and **(ii)** *source-only NTL*, which assumes the target domain is unknown and aims to restrict the generalization toward all other domains except the source domain. The comparison between SL and the two NTLs is shown in Figure 2.

2.2 General Framework of NTL

We use a classification task for illustration, as most existing NTL methods aim at image classification tasks. Let $\mathcal{D}_s = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N_s}$ and $\mathcal{D}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N_t}$ represent the source domain and the target domain, respectively. Note that we here assume $\mathcal{D}_s$ and $\mathcal{D}_t$ share the same label space, as considered in [Wang *et al.*, 2022b]. Considering a neural network f_θ with parameters θ, NTL aims to train the f_θ to maximize the risk on the target domain $\mathcal{D}_t$ and simultaneously minimize the risk on the source domain $\mathcal{D}_s$. To reach this goal, a basic NTL framework is to impose a regularization term on the SL to maximize the target domain error:

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

where λ is a trade-off weight, $\mathcal{L}_{\text{src}}$ and $\mathcal{L}_{\text{tgt}}$ represent the loss function (e.g., Kullback-Leibler divergence) for the source and target domain, respectively. The learning objective contains two tasks: 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.

Existing works generally can be seen as variants to Equation (1), where they may focus on different fields (modal, task), data assumptions (label space, target supervision, source data dependent), and use different approaches to conduct $\mathcal{T}_{\text{tgt}}$. The statistics of these aspects considered in existing works are shown in Table 1. More details for each NTL approach are illustrated in Section 3.

2.3 NTL Criteria

Non-transferability performance. After training, an NTL model usually be evaluated in two aspects:

Method	Venue	Field ①		Data ②			Non-Transferable Approach ③		Robustness ④	
		Modal	Task	Label Space	Target Data	Source Data	Feature Space	Output Space	Source	Target
NTL [Wang *et al.*, 2022b]	ICLR'22	CV	CLS	Close-Set	Labeled	Dependent	$\max \text{MMD}(\Phi(\boldsymbol{x}_s), \Phi(\boldsymbol{x}_t))$	$\max \text{KL}(f(\boldsymbol{x}_t), y_t)$	✔	✗
UNTL [Zeng and Lu, 2022]	EMNLP'22	NLP	CLS	Close-Set	Unlabeled	Dependent	$\max \text{MMD}(\Phi(\boldsymbol{x}_s), \Phi(\boldsymbol{x}_t)) + \min \text{CE}(\Omega_d(\Phi(\boldsymbol{x})), y_d)$	—	✗	✗
CUTI-domain [Wang *et al.*, 2023b]	CVPR'23	CV	CLS	Close-Set	Labeled	Dependent	—	$\max \text{KL}(f(\boldsymbol{x}_t), y_t)$	✔	✗
DSO [Wang *et al.*, 2023a]	ICCV'23	CV	CLS	Close-Set	Unlabeled	Dependent	—	$\min \text{KL}(f(\boldsymbol{x}_t), y_s + 1)$	✗	✗
H-NTL [Hong *et al.*, 2024b]	ICLR'24	CV	CLS	Close-Set	Labeled	Dependent	—	$\min \text{KL}(f(\boldsymbol{x}_t), f_{\text{sty}}(\boldsymbol{x}_t))$	✗	✗
ArchLock [Zhou *et al.*, 2024]	ICLR'24	CV	Cross	Open-Set	Labeled	Dependent	—	$\max \text{CE}(f(\boldsymbol{x}_t), y_t)$	✗	✔
TransNTL [Hong *et al.*, 2024a]	CVPR'24	CV	CLS	Close-Set	Labeled	Dependent	—	—	✔	✗
MAP [Peng *et al.*, 2024]	CVPR'24	CV	CLS	Close-Set	Unlabeled	Free	—	$\max \text{KL}(f(\boldsymbol{x}_t), \hat{y}_t)$	✗	✗
SOPHON [Deng *et al.*, 2024]	IEEE S&P'24	CV	CLS	Open-Set	Labeled	Dependent	—	$\min \text{CE}(f(\boldsymbol{x}_t), 1 - y_t)$ or $\min \text{KL}(f(\boldsymbol{x}_t), \mathcal{U})$	✗	✔
		CV	GEN	Open-Set	Labeled	Dependent	—	$\min \text{MSE}(f(\boldsymbol{x}_t), \mathbf{0})$		
CUPI-domain [Wang *et al.*, 2024]	TPAMI'24	CV	CLS	Close-Set	Labeled	Dependent	—	$\max \text{KL}(f(\boldsymbol{x}_t), y_t)$	✔	✗
NTP [Ding *et al.*, 2024]	ECCV'24	CV	CLS	Close-Set	Labeled	Dependent	$\min \text{FDA}(\Phi(\boldsymbol{x}_t), y_t)$	$\max \text{KL}(f(\boldsymbol{x}_t), y_t)$	✗	✔

① In **Field** column, "CV": computer vision. "NLP": natural language processing. "CLS": classification task. "GEN": generation task. "Cross": cross task.
② In **Data** column: "Close-Set": source and target domain share the same label space. "Open-Set": source and target domain have different label space. "Labeled": using labeled targeted data. "Unlabeled": do not need labeled targeted data. "Dependent": using source data. "Free": without source data.
③ In **Non-Transferable Approach**, we split the model f into a feature extractor Φ and a classifier Ω, i.e., $f(\boldsymbol{x}) = \Omega(\Phi(\boldsymbol{x}))$. Ω_d means an additional domain classifier. $\boldsymbol{x}_s$ and y_s: source-domain data and label. $\boldsymbol{x}_t$ and y_t: target-domain data and label. y_d: domain label. $\hat{y}_t$: target-domain pseudo label predicted by the model. $f_{\text{sty}}(\cdot)$: the style mapping function trained in H-NTL [Hong *et al.*, 2024b]. $\mathcal{U}$: uniform distribution. $\mathbf{0}$: zero vector. $\text{KL}(\cdot,\cdot)$: Kullback-Leibler divergence. $\text{CE}(\cdot,\cdot)$: Cross-Entropy loss. $\text{MMD}(\cdot,\cdot)$: Maximum Mean Discrepancy. $\text{MSE}(\cdot,\cdot)$: Mean Squared Error. $\text{FDA}(\cdot,\cdot)$: Fisher Discriminant Analysis (larger value indicates better feature clustering [Shao *et al.*, 2022]).
④ In **Robustness** column, ✔(or ✗) represent the robustness have (or haven't) been evaluated in their original paper.

Table 1: Summary of NTL methods according to **Field** (modal, task), **Data** (label space, target supervision, source data dependent), **Non-Transferable Approach** (feature or output space), and **Robustness** (whether source and target domain robustness have been evaluated).

- *Source domain maintenance*: Whether the NTL model is able to achieve normal performance (i.e., the same level as the SL model) on the source domain.
- *Target domain degradation*: The extent to which the NTL model can reduce performance on the target domain.

We review how existing methods achieve both the *source domain maintenance* and the *target domain degradation* in Section 3. Specifically, we focus on the setting that the target domain is known (i.e., target-specified NTL) in Section 3.1 and unknown (i.e., source-only NTL) in Section 3.2.

Post-training robustness. NTL models are expected to keep the non-transferability after malicious attacks, while not all existing works consider or evaluate the comprehensive robustness of their proposed method. We summarize the robustness considered in existing works into the following two parts, based on which domain is accessible to attackers. The statistics on which aspects have been evaluated for each NTL method are shown in Table 1 (**Robustness** column).

- *Robustness against source domain attack*: It has been verified that fine-tuning the NTL model with a small amount of source domain data is a potential risk to break non-transferability [Hong *et al.*, 2024a]. Thus, the *robustness against source domain attacks* measures how well an NTL model can resist fine-tuning attacks on the source domain.
- *Robustness against target domain attack*: If malicious attackers have access to a small amount of labeled target domain data, they can fine-tune the NTL model to re-activate target domain performance [Deng *et al.*, 2024]. The *robustness against target domain attack* evaluates how well an NTL model can defend against attack from the target domain, such as fine-tuning using target domain data.

We review robustness in existing NTL methods in Section 4.

3 Approaches for NTL

Target-specified NTL approaches contain fundamental solutions for NTL, and thus, we first review them in Section 3.1. Then, in Section 3.2, we review how existing works implement source-only NTL in the absence of a target domain.

3.1 Target-Specified NTL

Briefly, in target-specified setting, the target domain is known and we aim to restrict the generalization of a deep learning model from the source domain toward the certain target domain. Existing methods perform target-domain regularization either on the feature space or the output space, as we summarized in Table 1 (**Non-Transferable Approach** column). For more details, we introduce existing strategies as follows:

Output space regularization. Output-space regularizations directly manipulate the model logits on the target domain. More specifically, these operations can be categorized into *untargeted regularization* and *targeted regularization*. *Untargeted regularization* [Wang *et al.*, 2022b; Wang *et al.*, 2023b; Zhou *et al.*, 2024; Peng *et al.*, 2024] could usually be formalized as a maximizing optimization problem, where existing methods implement this regularization by maximizing the KL divergence between the model outputs and the real labels, thus disturbing the model predictions on the target domain. However, such untargeted regularization may face convergence issues [Deng *et al.*, 2024]. *Targeted regularization* [Wang *et al.*, 2023a; Deng *et al.*, 2024] found a proxy task on the target domain (i.e., modify the labels), thus converting the maximization objective in untargeted regularization to a minimization optimization problem. DSO [Wang *et al.*, 2023a] transforms the correct labels to error labels without overlap (e.g., $y_{\text{err}} = y + 1$) and uses er-

ror labels as the target-domain supervision. H-NTL [Hong *et al.*, 2024b] first disentangle the content and style factors via a variation inference framework [Blei *et al.*, 2017], and then, they learn the NTL model by fitting the contents of the source domain and the style of the target domain. Due to the assumption that the style is approximately to be independent to the content representations, the non-transferability could be implemented. SOPHON [Deng *et al.*, 2024] aims at both image classification and generation tasks. For classification, they propose to modify the cross-entropy (CE) loss to its inverse version (i.e., modify the label y to $1-y$) or calculate the KL divergence between the model outputs and a uniform distribution. For generation, SOPHON proposes to use a Denial of Service (DoS) loss, i.e., let the diffusion model fit a zero matrix at each step. Compared to untargeted regularizations, targeted regularizations always have better convergence.

Feature space regularization. Feature-space regularizations reduce the similarity between feature representations from different domains, thus restricting the transferability on the feature space. Feature-space regularizations can also be categorized into *untargeted* and *targeted* strategies, depending on whether they directly enlarge the distribution gap through a maximization objective or convert it to a minimization problem by finding a proxy target. For *untargeted regularization*, existing methods [Wang *et al.*, 2022b; Zeng and Lu, 2022] propose to maximize the maximum mean discrepancy (MMD) loss between the feature representations from different domains, where MMD measures the distribution discrepancy. For *targeted regularization*, UNTL [Zeng and Lu, 2022] proposes to build an auxiliary domain classifier with feature representations from different domains as inputs. By minimizing the domain-classification loss, the domain classifier could help the NTL model learn domain-distinct representations. NTP [Ding *et al.*, 2024] aims to minimize the Fisher Discriminant Analysis (FDA) term [Shao *et al.*, 2022] in the target domain. Specifically, a smaller FDA value indicates a reduced difference in class means and increased feature variance within each class, which is associated with poorer target domain performance.

3.2 Source-Only NTL

Under the assumption that only source domain data is available, existing works take various data augmentation methods to obtain auxiliary domains from the source domain and see them as the target domain. Thus, the source-only NTL problem can be solved by target-specified NTL approaches. These augmentation methods can be split into the following three categories:

Adversarial domain generation. Wang *et al.* [2022b] use generative adversarial network (GAN) [Mirza and Osindero, 2014; Chen *et al.*, 2016] to synthesize fake images from the source domain and see them as the target domain. They train the GAN by controlling the distance and direction of the synthetic distributions to the real source domain, thus enhancing the diversity of synthetic samples and improving the degradation of any distribution with shifts to the real source domain. CUTI-domain [Wang *et al.*, 2023b] and CUPI-domain [Wang *et al.*, 2024] add Gaussian noise to the GAN-based adaptive instance normalization (AdaIN) [Huang and Belongie, 2017] to obtain synthetic samples with random styles. They use both the synthetic samples from AdaIN and Wang *et al.* [2022b] as the target domain. MAP [Peng *et al.*, 2024] also follows the GAN framework. They additionally add a mutual information (MI) minimization term to enhance the variation between synthetic samples and the real source domain samples, ensuring more distinct style features.

Strong image augmentation. H-NTL [Hong *et al.*, 2024b] conducts strong image augmentation [Cubuk *et al.*, 2020] on real source domain data. Strong image augmentations (e.g., blurring, sharpness, solarize) do not influence the contents but significantly change the image styles, thus imposing interventions [Von Kügelgen *et al.*, 2021] on the style factors in images. Then, all augmented images are treated as the target domain for training source-only NTL.

Perturbation-based method. DSO [Wang *et al.*, 2023a] proposes to minimize the worst-case risk on the uncertainty set [Sagawa *et al.*, 2019] over the source domain distribution, where the risk is empirically calculated through a classification loss between the model predictions and the error label.

4 Post-Training Robustness of NTL

NTL models are expected to keep the non-transferability after malicious attacks. However, not all existing works evaluate the robustness of their method, as we listed in Table 1 (the last column). In this section, we review the robustness of the source and target domains as considered in previous works.

Robustness against source domain attack. Earlier evaluations in [Wang *et al.*, 2022b; Wang *et al.*, 2023b] show that NTL models are still resistant to SOTA watermark removal attacks when up to 30% source domain data are available for attack. Hong *et al.* [2024a] further investigate the robustness of NTL and propose TransNTL, demonstrating that non-transferability can be destroyed using less than 10% of the source domain data. Specifically, they find NTL [Wang *et al.*, 2022b] and CUTI-domain [Wang *et al.*, 2023b] inevitably result in significant generalization impairments on slightly perturbed source domains. Accordingly, they propose TransNTL to fine-tune NTL models under an impairment repair self-distillation framework, where the source-domain predictions are used to teach the model itself how to predict on perturbed source domains. As a result, the fine-tuned model is just like a SL model without the non-transferability. They also propose a defense method to fix this loophole by pre-repairing the generalization impairments in perturbed source domains. Specifically, they add a defense regularization term on existing NTL and CUT-domain training. Minimizing the defense regularization term enables NTL models to exhibit source-domain consistent behaviors on perturbed source-domain data, thus resisting TransNTL attack.

Robustness against target domain attack. If malicious attackers have access to some labeled target domain data, a more direct strategy to break the non-transferability is fine-tuning NTL models using target domain data. However, most existing methods [Wang *et al.*, 2022b; Wang *et al.*, 2023b; Zeng and Lu, 2022; Wang *et al.*, 2023a; Hong *et al.*, 2024b;

Peng *et al.*, 2024] ignore the robustness of their methods against target-domain fine-tuning attacks. SOPHON [Deng *et al.*, 2024] formally proposes the problem of non-fine-tunable learning, which aims at ensuring the target-domain performance could still be poor after being fine-tuned using target domain data. Their main idea is to involve the fine-tuning process in training stage. Specifically, they leverage model agnostic meta-learning (MAML) [Finn *et al.*, 2017] to simulate multiple-step fine-tuning for the current model on the target domain. Then, they add per-step risk of the target domain as the total target-domain risk. By maximizing the total target-domain risk, the robustness against target-domain attacks can be enhanced. ArchLock [Zhou *et al.*, 2024] aims to find the non-transferable network architectures [Liu *et al.*, 2018], where they implicitly consider the robustness on the target domain. Specifically, they maximize the *minimum risk* of an architecture on the target domain in searching the non-transferable architectures. The minimum risk is found by searching the optimal *parameters* of the *architecture* with the minimum task loss on the target domain.

However, labeled target domain data being available to malicious attackers is a strong assumption. A more realistic scenario is that attackers only has access to unlabeled target domain data. Whether NTL can resist attacks driven by unlabeled target domain data has not yet been studied.

5 Benchmarking NTL

The post-training robustness has not been well-evaluated in NTL, which motivates us to build a comprehensive benchmark. In this section, we first demonstrate the framework of our `NTLBench` (Section 5.1). Then, we present main results by conducting our `NTLBench` (Section 5.2), including pre-trained performance and robustness against different attacks.

5.1 `NTLBench`

We propose the first NTL benchmark (`NTLBench`), which contains a standard and unified training and evaluation process. `NTLBench` supports 5 SOTA NTL methods, 9 datasets (more than 116 domain pairs), 5 network architectures families, and 15 post-training attacks from 3 attack settings, providing more than 40,000 experimental configurations.

Datasets. Our `NTLBench` is compatible with: Digits (5 domains) [Deng, 2012; Hull, 1994; Netzer *et al.*, 2011; Ganin *et al.*, 2016; Roy *et al.*, 2018], RotatedMNIST (3 domains) [Ghifary *et al.*, 2015], CIFAR and STL (2 domains) [Krizhevsky and others, 2009; Coates *et al.*, 2011], VisDA (2 domains) [Peng *et al.*, 2017], Office-Home (4 domains) [Venkateswara *et al.*, 2017], DomainNet (6 domains) [Peng *et al.*, 2019], VLCS (4 domains) [Fang *et al.*, 2013], PCAS (4 domains) [Li *et al.*, 2017], and TerraInc (5 domains) [Beery *et al.*, 2018]. Different domains in any dataset share the same label space, but have distribution shifts, thus being suitable for evaluating NTL methods.

NTL baselines. `NTLBench` involves all open-source NTL methods: NTL [Wang *et al.*, 2022b], CUTI-domain [Wang *et al.*, 2023b], H-NTL [Hong *et al.*, 2024b], SOPHON [Deng *et al.*, 2024], CUPI-domain [Wang *et al.*, 2024]. Besides, we also add a vanilla supervised learning (SL) as a reference.

Network architecture. The proposed `NTLBench` is compatible with multiple network architectures, including but not limited to: VGG [Simonyan and Zisserman, 2015], ResNet [He *et al.*, 2016], WideResNet [Zagoruyko, 2016], ViT [Dosovitskiy *et al.*, 2021], SwinT [Liu *et al.*, 2021b].

Threat I: source domain fine-tuning (SourceFT). *Attacking goal*: SourceFT tries to destroy the non-transferability by fine-tuning the NTL model using a small set of source domain data. *Attacking method*: `NTLBench` involves 5 methods, including four basic fine-tuning strategies[1]: initFC-all, initFC-FC, direct-FC, direct-all [Deng *et al.*, 2024] and the special designed attack for NTL: TransNTL [Hong *et al.*, 2024a].

Threat II: target domain fine-tuning (TargetFT). *Attacking goal*: TargetFT tries to directly use labeled target domain data to fine-tune the NTL model, thus recovering target domain performance. *Attacking method*: `NTLBench` use 4 basic fine-tuning strategies[1] leveraged in [Deng *et al.*, 2024] as attack methods: initFC-all, initFC-FC, direct-FC, direct-all.

Threat III: source-free domain adaptation (SFDA). *Attacking goal*: We introduce SFDA to test whether using unlabeled target domain data poses a threat to NTL. *Attacking method*: `NTLBench` involves 6 SOTA SFDA methods: SHOT [Liang *et al.*, 2020], CoWA [Lee *et al.*, 2022], NRC [Yang *et al.*, 2021], PLUE [Litrico *et al.*, 2023], AdaContrast [Chen *et al.*, 2022], and DIFO [Tang *et al.*, 2024].

Evaluation metric. For source domain, we use source domain accuracy (**SA**) to evaluate the performance. Higher SA means lower influence of non-transferability to the source domain utility. For target domain, we use target domain accuracy (**TA**) to evaluate the performance. Lower TA means better performance of non-transferability. Besides, we calculate the overall performance (denoted as **OA**) of an NTL method as: $\text{OA} = (\text{SA} + (100\% - \text{TA}))/2$, with higher OA representing better overall performance of an NTL method. These evaluation metrics are applicable for both non-transferability performance and robustness against different attacks.

5.2 Main Results and Analysis

Due to the limited space, we present main results obtained from our `NTLBench`. We first show the key implementation details, and then we present and analyze our results.

Implementation details. Briefly, in pre-training stage, we sequentially pair i-th and (i+1)-th domains within a dataset for training. Each domain is randomly split into 8:1:1 for training, validation, and testing. The results for each dataset are averaged across domain pairs. NTL methods and the reference SL method are pretrained by up to 50 epochs. We search suitable hyper-parameters for each method by setting 5 values around their original value and choose the best value according to the best OA on validation set. All the batch size, learning rate, and optimizer are follow their original implementations. Following the original NTL paper [Wang *et al.*, 2022b], we use VGG-13 without batch-normalization. All input images are resize to 64×64. In attack stage, we use 10%

[1]initFC: re-initialize the last full-connect (FC) layer. direct: no re-initialize. all: fine-tune the whole model. FC: fine-tune last FC.

	Digits		RMNIST		CIFAR & STL		VisDA		Office-Home		DomainNet		VLCS		PCAS		TerraInc		Avg.	
	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓
SL	97.7	56.0	99.2	62.4	88.2	65.7	86.8	37.7	66.4	36.9	45.6	9.9	79.9	56.9	89.5	47.3	93.6	14.9	83.0	43.0
NTL [Wang *et al.*, 2022b]	95.6 (-2.1)	12.2 (-43.8)	98.7 (-0.5)	12.3 (-50.1)	83.9 (-4.4)	9.9 (-55.8)	82.0 (-4.8)	10.9 (-26.8)	64.8 (-1.6)	32.4 (-4.5)	7.6 (-38.0)	1.4 (-8.6)	78.0 (-1.9)	27.1 (-29.8)	85.8 (-3.7)	18.0 (-29.2)	90.0 (-3.6)	8.8 (-6.1)	76.3 (-6.7)	14.8 (-28.3)
CUTI-domain [Wang *et al.*, 2023b]	97.0 (-0.8)	9.5 (-46.5)	99.2 (-0.1)	15.5 (-46.9)	85.1 (-3.2)	10.7 (-55.0)	85.3 (-1.5)	8.9 (-28.8)	56.7 (-9.7)	17.8 (-19.1)	14.0 (-31.7)	2.0 (-7.9)	78.3 (-1.6)	26.7 (-30.1)	88.4 (-1.1)	18.3 (-28.9)	87.9 (-5.7)	0.8 (-14.1)	76.9 (-6.1)	12.2 (-30.8)
H-NTL [Hong *et al.*, 2024b]	97.5 (-0.2)	9.6 (-46.4)	99.0 (-0.2)	10.8 (-51.5)	87.2 (-1.0)	9.9 (-55.8)	86.5 (-0.3)	8.6 (-29.0)	51.1 (-15.2)	17.0 (-19.8)	33.3 (-12.3)	2.1 (-7.8)	79.2 (-0.8)	42.7 (-14.2)	89.1 (-0.3)	22.1 (-25.1)	88.4 (-5.2)	14.6 (-0.2)	79.0 (-4.0)	15.3 (-27.8)
SOPHON [Deng *et al.*, 2024]	95.2 (-2.5)	9.9 (-46.1)	96.6 (-2.6)	38.8 (-23.6)	69.5 (-18.7)	24.8 (-40.9)	77.3 (-9.5)	10.9 (-26.8)	45.9 (-20.4)	17.6 (-19.3)	30.1 (-15.6)	2.5 (-7.4)	79.4 (-0.6)	29.5 (-27.4)	86.7 (-2.8)	21.6 (-25.7)	88.8 (-4.8)	7.1 (-7.7)	74.4 (-8.6)	18.1 (-25.0)
CUPI-domain [Wang *et al.*, 2024]	96.7 (-1.0)	8.8 (-47.2)	98.8 (-0.4)	21.0 (-41.3)	86.0 (-2.3)	11.3 (-54.4)	84.6 (-2.2)	8.2 (-29.5)	11.6 (-54.7)	2.3 (-34.6)	0.8 (-44.9)	0.3 (-9.7)	77.5 (-2.5)	29.5 (-27.4)	87.8 (-1.7)	11.5 (-35.8)	82.4 (-11.1)	1.3 (-13.6)	69.6 (-13.4)	10.4 (-32.6)

Table 2: Comparison of SL and 5 NTL methods on multiple datasets. We report the source-domain accuracy (**SA**) (%) in blue and target-domain accuracy (**TA**) (%) in red. The best results of overall performance (OA) are highlighted in blue background. The accuracy drop compared to the pre-trained model is shown in brackets. The average performance of 9 datasets are shown in the last column (**Avg.**).

amount of the training set to perform attack. All attack results we reported are run on CIFAR & STL. Attack training is up to 50 epochs. We run all experiments on RTX 4090 (24G).

Non-transferability performance. The non-transferability performance are shown in Table 2, where we compare 5 NTL methods and SL on 9 datasets. From the results, all NTL methods generally effectively degrade source-to-target generalization, leading to a significant drop in TA compared to SL. However, in more complex datasets such as Office-Home and DomainNet, existing NTL methods fail to achieve a satisfactory balance between maintaining SA and degrading TA, highlighting their limitations. From the **Avg.** column, CUTI-domain reaches the overall best performance.

Post-training robustness. For **SourceFT** attack (Table 3), fine-tuning each NTL model by using basic fine-tuning strategies on 10% source domain data cannot directly recover the source-to-target generalization. However, all NTL methods are fragile when facing the TransNTL attack. For **TargetFT** attack (Table 4), all NTL methods cannot fully resist supervised fine-tuning attack by using target domain data. In particular, fine-tuning all parameters usually results in better attack effectiveness. For **SFDA** (Table 5), although the target domain data are unlabeled, advanced source-free unsupervised domain adaptation, leveraging self-supervised strategies, can still partially recover target domain performance. All these results verify the fragility of existing NTL methods.

6 Applications of NTL

NTL supports different applications, depending on which data are used as source and target domain. We introduce two applications in model intellectual property (IP) protection and then the application of harmful fine-tuning defense.

Ownership verification (OV). OV is a passive IP protection manner, which aims to verify the ownership of a deep learning model [Lederer *et al.*, 2023]. NTL solves ownership verification by triggering misclassification on data with pre-defined triggers [Wang *et al.*, 2022b]. For example, when training, we add a shallow trigger (only known by the model owner) on the original dataset data and see them as the target domain, while the original data without the trigger is regarded as the source domain. Then, target-specified NTL is used to train a model. Therefore, the ownership can be verified via observing the performance difference of a model on the original data and the data with the pre-defined trigger. For SL model, the shallow trigger has minor influence on the model performance, and thus, the model shows similar performance on original data and data with triggers. In contrast, the NTL model specific to this pre-defined trigger has high performance on the original data but random-guess-like per-

	NTL		CUTI		H-NTL		SOPHON		CUPI	
	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓
Pre-train	83.9	9.9	85.1	10.6	87.2	9.9	69.5	24.8	86.0	11.3
initFC-all	84.0 (+0.2)	9.8 (-0.1)	84.2 (-0.9)	10.6 (+0.0)	87.8 (+0.6)	16.2 (+6.3)	82.2 (+12.7)	38.1 (+13.3)	85.3 (-0.7)	11.4 (+0.1)
initFC-FC	84.2 (+0.3)	10.0 (+0.1)	85.4 (+0.3)	10.6 (+0.0)	87.2 (-0.1)	10.2 (+0.3)	71.9 (+2.4)	23.3 (-1.6)	85.9 (-0.1)	11.3 (+0.0)
direct-FC	84.0 (+0.2)	9.9 (+0.0)	85.2 (+0.2)	10.6 (+0.0)	87.3 (+0.1)	9.9 (+0.0)	74.3 (+4.8)	23.8 (-1.1)	86.1 (+0.1)	11.3 (+0.0)
direct-all	84.7 (+0.8)	9.8 (-0.1)	85.3 (+0.3)	10.9 (+0.3)	88.0 (+0.8)	10.1 (+0.2)	83.4 (+13.9)	32.2 (+7.4)	85.5 (-0.5)	11.3 (+0.0)
TransNTL	81.7 (-2.2)	44.3 (+34.4)	81.3 (-3.8)	61.0 (+50.3)	86.3 (-1.0)	63.7 (+53.8)	83.8 (+14.3)	60.1 (+35.3)	83.1 (-2.9)	60.6 (+49.3)

Table 3: NTL robustness against source domain fine-tuning (Source-FT). We show source-domain accuracy (**SA**) (%) and target-domain accuracy (**TA**) (%). The most serious threat (worst OA) to each NTL is marked as red. Accuracy drop from the pre-trained model is in (·).

	NTL		CUTI		H-NTL		SOPHON		CUPI	
	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓
Pre-train	83.9	9.9	85.1	10.7	87.2	9.9	69.5	24.8	86.0	11.3
initFC-all	23.9 (-60.0)	37.8 (+27.9)	13.3 (-71.8)	15.9 (+5.3)	19.0 (-68.3)	10.4 (+0.5)	59.0 (-10.5)	68.5 (+43.7)	41.2 (-44.8)	53.1 (+41.8)
initFC-FC	33.9 (-50.0)	9.6 (-0.4)	30.2 (-54.9)	9.7 (-1.0)	19.1 (-68.1)	9.7 (-0.2)	21.6 (-48.0)	16.8 (-8.1)	21.8 (-64.2)	12.1 (+0.8)
direct-FC	64.2 (-19.7)	10.2 (+0.3)	38.0 (-47.1)	10.6 (-0.1)	87.1 (-0.1)	10.0 (+0.1)	70.5 (+1.0)	24.5 (-0.4)	78.6 (-7.4)	11.0 (-0.4)
direct-all	13.9 (-70.0)	17.6 (+7.7)	10.1 (-75.0)	8.8 (-1.9)	84.7 (-2.5)	53.3 (+43.4)	68.0 (-1.6)	72.9 (+48.1)	51.9 (-34.1)	58.4 (+47.1)

Table 4: NTL robustness against target domain fine-tuning (Target-FT). We report source-domain accuracy (**SA**) (%) and target-domain accuracy (**TA**) (%). The most serious threat (best TA) to each NTL is marked as red. Accuracy drop from the pre-trained model is in (·).

	NTL		CUTI		H-NTL		SOPHON		CUPI	
	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓	SA ↑	TA ↓
Pre-train	83.9	9.9	85.1	10.7	87.2	9.9	69.5	24.8	85.5	11.3
SHOT	63.0 (-20.9)	29.6 (+19.7)	35.3 (-49.8)	34.7 (+24.0)	86.6 (-0.6)	41.9 (+32.0)	64.8 (-4.8)	56.7 (+31.9)	85.8 (+0.3)	11.3 (+0.0)
CoWA	81.1 (-2.8)	12.4 (+2.5)	84.0 (-1.1)	12.7 (+2.1)	87.2 (+0.0)	10.1 (+0.2)	69.2 (-0.4)	26.1 (+1.3)	85.7 (+0.2)	11.3 (+0.0)
NRC	57.7 (-26.2)	19.8 (+9.9)	39.4 (-45.7)	35.5 (+24.8)	87.3 (+0.1)	12.1 (+2.2)	66.6 (-3.0)	55.6 (+30.8)	86.0 (+0.5)	12.2 (+0.9)
PLUE	71.5 (-12.4)	52.8 (+42.9)	76.1 (-9.0)	63.8 (+53.1)	85.5 (-1.8)	20.1 (+10.2)	75.5 (+6.0)	41.1 (+16.3)	82.4 (-3.2)	43.6 (+32.3)
Ada-Contrast	9.4 (-74.5)	9.8 (-0.1)	9.3 (-75.8)	10.0 (-0.7)	86.3 (-1.0)	12.1 (+2.2)	64.5 (-5.1)	33.4 (+8.6)	47.2 (-38.3)	11.3 (+0.0)
DIFO	9.2 (-74.7)	9.2 (-0.7)	9.2 (-75.9)	9.2 (-1.5)	85.0 (-2.2)	42.1 (+32.2)	56.3 (-13.2)	51.3 (+26.5)	48.4 (-37.1)	10.4 (-1.0)

Table 5: NTL robustness against source-free domain adaptation (SFDA). We show source-domain accuracy (**SA**) (%), target-domain accuracy (**TA**) (%), and accuracy drop from the pre-trained model is in (·). The most serious threat (highest TA) to each NTL is in red.

formance on data with the trigger. This provides evidence for verifying the model's ownership.

Applicability authorization (AA). AA is an active IP protection approach that ensures models can only be effective on authorized data [Wang *et al.*, 2022b; Xu *et al.*, 2024; Si *et al.*, 2024]. NTL solves AA by degrading the model generalization outside the authorized domain. Basic solution is to add a pre-defined trigger on original data (seen as source domain), and the original data without the correct triggers is regarded as the target domain. After training by NTL, the model will only perform well on authorized data (i.e., the data with the trigger). Any unauthorized data will be randomly predicted by the NTL model. Thus, AA can be achieved.

Safety alignment and harmful fine-tuning defense. Fine-tuning large language models (LLMs) with user's own data for downstream tasks has recently become a popular online service [Huang *et al.*, 2024; OpenAI, 2024]. However, this practice raises concerns about compromising the safety alignment of LLMs [Qi *et al.*, 2023; Yang *et al.*, 2023], as harmful data may be present in users' datasets, whether intentionally or unintentionally. To address the risks of harmful fine-tuning, various defensive solutions [Huang *et al.*, 2025; Rosati *et al.*, 2024] have been proposed to ensure that fine-tuned LLMs can effectively refuse harmful queries. Specifically, these defense methods aim to limit the transferability of LLMs from harmless queries to harmful ones, which techniques are variants of the objectives of NTL. Actually, all existing NTL approaches can be applied to this task by regarding the alignment data as the source domain and the harmful data as the target domain. Then, target-specified NTL can be conducted to defend against harmful fine-tuning attacks.

7 Related Works

Machine unlearning (MU). Both MU [Xu *et al.*, 2023] and NTL serve purposes in model capacity control, albeit with differences in their applications and methodologies. MU primarily aims to forget specific data points from training datasets [Xu *et al.*, 2023] (the model behaviors are consistent to never training on the selected data points), while NTL aims at resist the generalization from the training domain to a specific target domain or task. Particularly, MU and NTL share some overlapping applications such as safety alignment of LLMs. However, MU more focus on eliminating harmful data influence (e.g., sensitive or illegal information) and the associated model capabilities [Barez *et al.*, 2025; Maini *et al.*, 2024], while NTL more focus on preventing harmful and unauthorized fine-tuning [Huang *et al.*, 2024].

Transfer learning (TL). TL [Zhuang *et al.*, 2020] aims at improving model performance on a different but related domain or task. It can be categorized into several subfields, such as domain adaptation (DA) [Venkateswara *et al.*, 2017; Liang *et al.*, 2020] and domain generalization (DG) [Wang *et al.*, 2022a]. TL is closely related to NTL, but the overall objectives is opposite to NTL. In general, TL techniques generally can be used in a reversed way to achieve NTL. Moreover, TL can also be seen as post-training attacks against NTL.

8 Future Directions and Challenges

Improving robustness. We highlight the shortcoming of NTL on post-training robustness. Existing defense attempts (e.g., SOPHON [Deng *et al.*, 2024]) require extensive resources, such as an extremely high number of training epochs, yet they may still fail to remain robust against unseen fine-tuning. This raises an open challenge: how to effectively enhance the robustness of NTL against various attacks.

Identifying more threat. There are other potential attacks that could pose risks to NTL under weaker assumptions. For example, if an attacker is unable to re-train the NTL model, can they still bypass the non-transferability constraints? In addition, attackers may have access to a large amount of data from the wild, distinct from both the source and target domains. Can they leverage these unseen domain data to break non-transferability? We believe identifying these threats can further promote the robustness of NTL.

Cross-modal non-transferability. Existing NTL works primarily focus on single modality, while the cross-modal non-transferability remains an important yet underexplored challenge. A related finding in large models suggests that the safety alignment of LLMs can be compromised through visual instruction tuning [Zong *et al.*, 2024; Liu *et al.*, 2024]. However, a deep investigation of robust cross-modal non-transferability mechanisms remains lacking.

9 Conclusion

In this paper, we conduct the first systematic review of NTL by summarizing existing approaches and highlighting the often overlooked robustness challenges. In addition, we introduce the first benchmark, `NTLBench`, which systematically evaluates five state-of-the-art NTL methods through comprehensive assessments of pretraining performance and robustness against 15 attacks across multiple datasets and network architectures. Main results from `NTLBench` verify the limitation of existing NTLs on robustness. We believe `NTLBench` can drive the development of robust NTL and facilitate their applications in practical scenarios.

Acknowledgements

*: equal contribution. †: corresponding author. TLL is partially supported by the following Australian Research Council projects: FT220100318, DP220102121, LP220100527, LP220200949, IC190100031. ZMH is supported by JD Technology Scholarship for Postgraduate Research in Artificial Intelligence No. SC4103.

References

[Barez *et al.*, 2025] Fazl Barez, Tingchen Fu, et al. Open problems in machine unlearning for ai safety. *arXiv*, 2025.

[Beery *et al.*, 2018] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018.

[Blei *et al.*, 2017] David M Blei, Alp Kucukelbir, et al. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 2017.

[Chen *et al.*, 2016] Xi Chen, Yan Duan, et al. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016.

[Chen *et al.*, 2022] Dian Chen, Dequan Wang, et al. Contrastive test-time adaptation. In *CVPR*, 2022.

[Coates *et al.*, 2011] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[Cubuk *et al.*, 2020] Ekin D Cubuk, Barret Zoph, et al. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR workshops*, 2020.

[Deng *et al.*, 2024] Jiangyi Deng, Shengyuan Pang, et al. Sophon: Non-fine-tunable learning to restrain task transferability for pre-trained models. *IEEE S&P*, 2024.

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

[Ding *et al.*, 2024] Ruyi Ding, Lili Su, Aidong Adam Ding, and Yunsi Fei. Non-transferable pruning. *arXiv*, 2024.

[Dosovitskiy *et al.*, 2021] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[Fang *et al.*, 2013] Chen Fang, Ye Xu, et al. Unbiased metric learning: on the utilization of multiple datasets and web images for softening bias. In *ICCV*, 2013.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[Ganin *et al.*, 2016] Yaroslav Ganin, Evgeniya Ustinova, et al. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 2016.

[Ghifary *et al.*, 2015] Muhammad Ghifary, W Bastiaan Kleijn, et al. Domain generalization for object recognition with multi-task autoencoders. In *ICCV*, 2015.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, et al. Deep residual learning for image recognition. In *CVPR*, 2016.

[Hong *et al.*, 2024a] Ziming Hong, Li Shen, and Tongliang Liu. Your transferability barrier is fragile: Free-lunch for transferring the non-transferable learning. In *CVPR*, 2024.

[Hong *et al.*, 2024b] Ziming Hong, Zhenyi Wang, Li Shen, et al. Improving non-transferable representation learning by harnessing content and style. In *ICLR*, 2024.

[Huang and Belongie, 2017] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[Huang *et al.*, 2024] Tiansheng Huang, Sihao Hu, Fatih Ilhan, et al. Harmful fine-tuning attacks and defenses for large language models: A survey. *arXiv*, 2024.

[Huang *et al.*, 2025] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. In *ICLR*, 2025.

[Hull, 1994] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE TPAMI*, 16(5), 1994.

[Jiang *et al.*, 2024] Yongqi Jiang, Yansong Gao, Chunyi Zhou, et al. Intellectual property protection for deep learning model and dataset intelligence. *arXiv*, 2024.

[Kaplan *et al.*, 2020] Jared Kaplan, Sam McCandlish, et al. Scaling laws for neural language models. *arXiv*, 2020.

[Krizhevsky and others, 2009] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[Lederer *et al.*, 2023] 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 TNNLS*, 2023.

[Lee *et al.*, 2022] Jonghyun Lee, Dahuin Jung, Junho Yim, and Sungroh Yoon. Confidence score for source-free unsupervised domain adaptation. In *ICML*, 2022.

[Li *et al.*, 2017] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017.

[Li *et al.*, 2023] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, et al. Trustworthy ai: From principles to practices. *ACM Computing Surveys*, 55(9):1–46, 2023.

[Liang *et al.*, 2020] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.

[Litrico *et al.*, 2023] Mattia Litrico, Alessio Del Bue, and Pietro Morerio. Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation. In *CVPR*, 2023.

[Liu *et al.*, 2018] Hanxiao Liu, Karen Simonyan, et al. Darts: Differentiable architecture search. *arXiv*, 2018.

[Liu *et al.*, 2021a] Jiashuo Liu, Zheyan Shen, et al. Towards out-of-distribution generalization: A survey. *arXiv*, 2021.

[Liu *et al.*, 2021b] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[Liu *et al.*, 2024] Haotian Liu, Chunyuan Li, et al. Visual instruction tuning. In *NeurIPS*, 2024.

[Maini *et al.*, 2024] Pratyush Maini, Zhili Feng, et al. Tofu: A task of fictitious unlearning for llms. *arXiv*, 2024.

[Mirza and Osindero, 2014] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv*, 2014.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, et al. Reading digits in natural images with unsupervised feature learning. 2011.

[OpenAI, 2024] OpenAI. Fine-tuning. https://platform.openai.com/docs/guides/fine-tuning, 2024.

[Ouyang *et al.*, 2022] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[Peng *et al.*, 2017] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, et al. Visda: The visual domain adaptation challenge. *arXiv*, 2017.

[Peng *et al.*, 2019] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

[Peng *et al.*, 2024] Boyang Peng, Sanqing Qu, Yong Wu, Tianpei Zou, et al. Map: Mask-pruning for source-free model intellectual property protection. In *CVPR*, 2024.

[Qi *et al.*, 2023] Xiangyu Qi, Yi Zeng, Tinghao Xie, et al. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv*, 2023.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[Rosati *et al.*, 2024] Domenic Rosati, Jan Wehner, Kai Williams, et al. Representation noising effectively prevents harmful fine-tuning on llms. In *NeurIPS*, 2024.

[Roy *et al.*, 2018] Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures. *arXiv*, 2018.

[Sagawa *et al.*, 2019] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, et al. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv*, 2019.

[Shao *et al.*, 2022] Wenqi Shao, Xun Zhao, Yixiao Ge, et al. Not all models are equal: Predicting model transferability in a self-challenging fisher space. In *ECCV*, 2022.

[Si *et al.*, 2024] Wai Man Si, Michael Backes, and Yang Zhang. Iclguard: Controlling in-context learning behavior for applicability authorization. *arXiv*, 2024.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[Sun *et al.*, 2023] Yuchen Sun, Tianpeng Liu, Panhe Hu, Qing Liao, Shouling Ji, Nenghai Yu, Deke Guo, and Li Liu. Deep intellectual property: A survey. *arXiv*, 2023.

[Tang *et al.*, 2024] Song Tang, Wenxin Su, Mao Ye, and Xiatian Zhu. Source-free domain adaptation with frozen multimodal foundation model. In *CVPR*, 2024.

[Venkateswara *et al.*, 2017] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, et al. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

[Von Kügelgen *et al.*, 2021] Julius Von Kügelgen, Yash Sharma, Luigi Gresele, et al. Self-supervised learning with data augmentations provably isolates content from style. In *NeurIPS*, 2021.

[Wang *et al.*, 2022a] Jindong Wang, Cuiling Lan, Chang Liu, et al. Generalizing to unseen domains: A survey on domain generalization. *IEEE TKDE*, 2022.

[Wang *et al.*, 2022b] 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 *ICLR*, 2022.

[Wang *et al.*, 2023a] Haotian Wang, Haoang Chi, Wenjing Yang, Zhipeng Lin, et al. Domain specified optimization for deployment authorization. In *ICCV*, 2023.

[Wang *et al.*, 2023b] Lianyu Wang, Meng Wang, Daoqiang Zhang, and Huazhu Fu. Model barrier: A compact untransferable isolation domain for model intellectual property protection. In *CVPR*, 2023.

[Wang *et al.*, 2024] Lianyu Wang, Meng Wang, Huazhu Fu, et al. Say no to freeloader: Protecting intellectual property of your deep model. *IEEE TPAMI*, 2024.

[Xu *et al.*, 2023] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. Machine unlearning: A survey. *ACM Computing Surveys*, 2023.

[Xu *et al.*, 2024] Chaohui Xu, Qi Cui, Jinxin Dong, Weiyang He, et al. Idea: An inverse domain expert adaptation based active dnn ip protection method. *arXiv*, 2024.

[Yang *et al.*, 2021] Shiqi Yang, Joost Van de Weijer, Luis Herranz, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *NeurIPS*, 2021.

[Yang *et al.*, 2023] Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, et al. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv*, 2023.

[Zagoruyko, 2016] Sergey Zagoruyko. Wide residual networks. *arXiv*, 2016.

[Zeng and Lu, 2022] Guangtao Zeng and Wei Lu. Unsupervised non-transferable text classification. In *EMNLP*, 2022.

[Zhou *et al.*, 2024] Tong Zhou, Shaolei Ren, and Xiaolin Xu. Archlock: Locking dnn transferability at the architecture level with a zero-cost binary predictor. In *ICLR*, 2024.

[Zhuang *et al.*, 2020] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, et al. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 2020.

[Zong *et al.*, 2024] Yongshuo Zong, Ondrej Bohdal, et al. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. *arXiv*, 2024.