Domain Specified Optimization for Deployment Authorization

Haotian Wang*, Haoang Chi*, Wenjing Yang, Zhipeng Lin, Mingyang Geng, Long Lan
National University of Defense Technology
wanghaotian13@nudt.edu.cn, haoangchi618@gmail.com, {wenjing.yang,long.lan}@nudt.edu.cn

Jing Zhang, Dacheng Tao
The University of Sydney
jing.zhang1@sydney.edu.au, dacheng.tao@gmail.com

Abstract

This paper explores Deployment Authorization (DPA) as a means of restricting the generalization capabilities of vision models on certain domains to protect intellectual property. Nevertheless, the current advancements in DPA are predominantly confined to fully supervised settings. Such settings require the accessibility of annotated images from any unauthorized domain, rendering the DPA approaches impractical for real-world applications due to its exorbitant costs.

To address this issue, we propose Source-Only Deployment Authorization (SDPA), which assumes that only authorized domains are accessible during training phases, and the model's performance on unauthorized domains must be suppressed in inference stages. Drawing inspiration from distributional robust statistics, we present a lightweight method called Domain-Specified Optimization (DSO) for SDPA that degrades the model's generalization over a divergence ball. DSO comes with theoretical guarantees on the convergence property and its authorization performance. As a complementary of SDPA, we also propose Target-Combined Deployment Authorization (TPDA), where unauthorized domains are partially accessible, and simplify the DSO method to a perturbation operation on the pseudo predictions, referred to as Target-Dependent Domain-Specified Optimization (TDSO). We demonstrate the effectiveness of our proposed DSO and TDSO methods through extensive experiments on six image benchmarks, achieving dominant performance on both SDPA and TDPA settings.

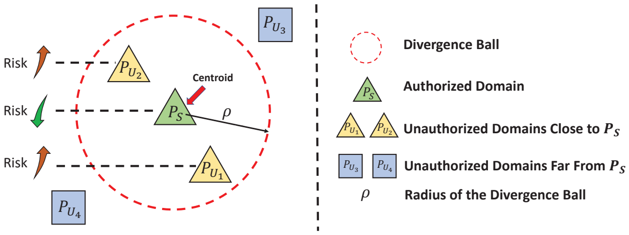

P_{U_3}
Risk
Risk
Risk
P_{U_2}
Centroid
P_S
ρ
P_{U_1}
P_{U_4}
Divergence Ball
Authorized Domain
Unauthorized Domains Close to P_S
Unauthorized Domains Far From P_S
ρ Radius of the Divergence Ball

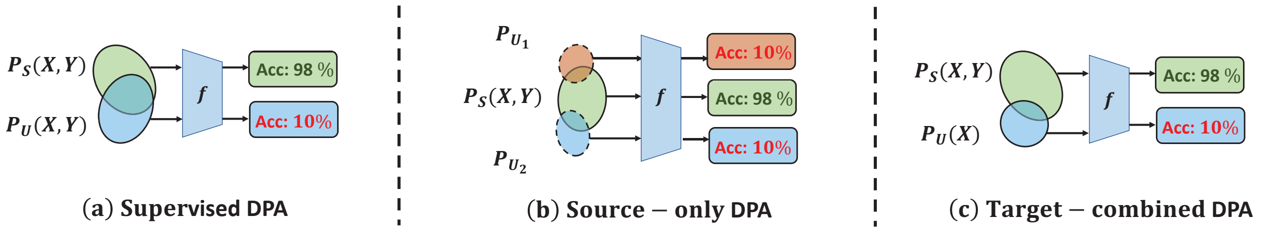
Figure 1: Our domain specified optimization is designed for maximizing the risk over unauthorized domains (e.g., P_{U_1} and P_{U_2} as framed yellow triangles) close to the authorized domain P_s, and minimizing the risk on P_S as the centroid (risk quantifies the consistency between the learning model and the domain). Besides, unauthorized domains far away from the authorized domain (e.g., P_{U_3} and P_{U_4} as framed blue squares) are not considered, as the model trained on P_S naturally generalizes poorly on them.

1. Introduction

Deep learning has achieved remarkable progress in a variety of vision applications [4, 7, 47, 48, 51, 38] with tremendous commercial values [22, 41, 20, 40]. However, training a deep model from scratch is non-trivial, as it contains abundant knowledge including a professional tuning process [7], large-scale image datasets with exhaustive annotations [9], and expensive computational resources [2]. Therefore, the intellectual property (IP) protection/authorization of the knowledge embodied in a pre-trained deep model has been raised as a matter of increasing concern [46, 43].

Previous studies on IP protection/authorization consist of two branches, including authorizations of the model owner and the model user [43], respectively. In addition to addressing who can own or use the model, a data-centric authorization problem known as deployment authorization (DPA) has been proposed [43]. DPA aims to address which data can

[*] Indicates equal contribution.



(a) Supervised DPA (b) Source − only DPA (c) Target − combined DPA

Figure 2: Overview of supervised DPA (SUDPA), Source-only DPA (SDPA), and Target-combined DPA (TDPA), where the subscripts S, U, and f refer to the authorized data, the unauthorized domains, and the learning model, respectively. SUDPA allows the model authorizer to access both the images $\mathbf{X}$ and labels $\mathbf{Y}$ on P_U in the training stage. Without labels, TPDA allows the authorizer only to access the images $\mathbf{X}$ on P_U. Both $\mathbf{X}$ and $\mathbf{Y}$ are inaccessible for the authorizer in SPDA.

be deployed on the model. *More specifically, DPA seeks to restrict the model's generalization capability on certain domains so that its performance is severely degraded when deployed on unauthorized data, while it maintains its performance when deployed on authorized data.*

The practical significance of DPA can be attributed to two main factors: (a) its commercial aspect, which safeguards the IP of commercial models from competitors, and (b) its ethical aspect, which restricts unauthorized users from exploiting models to target vulnerable populations. Concerning the commercial aspect, DPA is essential for applications such as online interactions between users and models via interfaces, where well-trained models on specific data cannot be shared and require payment to deploy, such as Amazon's Machine-Learning-as-a-Service. In terms of the ethical aspect, a typical example would be a company, like Meta, that trains a recommendation system (RS) using adult data and applies DPA to prevent teenagers from abusing the system [43]. Without DPA, if the RS generalizes well to teenagers' data and is deployed without authorization, it could potentially recommend inappropriate items such as alcohol to teenagers. Due to the growing importance of data-centric IP protection, our paper focuses on exploring DPA-related issues.

Concerning the current advances in DPA, a promising solution called non-transferable learning (NTL) has been proposed to address the supervised deployment authorization (SUDPA) problem [43], as illustrated in Figure 2(a). In particular, SUDPA assumes that the model authorizer can fully access both authorized and unauthorized domains [43]. As highlighted in our teenager-adult example, implementing SUDPA requires annotated data (e.g., labeled images of items) from both the teenager and adult domains. Unfortunately, collecting and labeling images from unauthorized domains is too expensive, which is a significant limitation for practical applications. Furthermore, this cost can escalate infinitely since the authorizer might specify an arbitrarily large range of unauthorized domains. To address this issue, we propose two more practical yet more challenging DPA problems, as illustrated in Figure 2(b) and (c).

- **Source-only deployment authorization (SDPA)**: In the teenager-adult example, the authorizer may **only** want to deploy the recognition model trained on adult data on the authorized domain. As illustrated in Figure 2 (b), our proposed SDPA approach addresses this requirement by assuming that the authorizer: (a) has access to only the authorized domain during the training phase, and (b) intends to suppress the model's performance on all other domains.
- **Target-combined authorization problem (TDPA)**: When additional unauthorized domains are specified, the authorizer only has to collect them directly without annotations. Figure 2 (c) illustrates this scenario, where the TDPA problem assumes that the authorizer has access to both the entire authorized domain (i.e., labeled adult data) and the partially labeled unauthorized domain (i.e., unlabeled teenager data).

In this paper, we propose a lightweight method named "Domain Specified Optimization" (DSO) to address the SDPA problem. DSO defines a divergence ball centered around the training distribution, covering each neighboring distribution close to the training domain [11]. Fig 1 illustrates that DSO simultaneously minimizes the model risk on the training domain and maximizes the model risk on each domain in the divergence ball except for the training domain. Theoretical results ensure that: (a) DSO possesses well-established convergence properties, and (b) DSO achieves reliable authorization on any unauthorized domain. To solve the TDPA problem, we adapt our DSO method into the Target-specified DSO (TDSO) method, which perturbs the pseudo predictions on the unauthorized domain. In summary, we list our main contributions as follows:

Problem Contribution In addition to the vanilla DPA problem, we introduce two more practical and challenging problems: the Source-only DPA (SDPA) and the Target-combined DPA (TDPA). The SDPA problem aims to achieve uniform deployment authorization of vision models by degrading the model's performance on any other domains, while the TDPA problem responds to the requirement that the model authorizer specifies unauthorized domains without incurring the cost of annotation.

Method Contribution To solve SDPA, we contribute a novel method named "Domain Specified Optimization" (DSO), which achieves reliable authorization on any unauthorized domain close to the training distribution. Furthermore, we solve the TDPA problem by proposing a pseudo-prediction perturbation strategy named "TDAO".

Experimental Results We conduct extensive experiments on six image classification benchmarks, including digit datasets (MNIST, USPS, SVHN, SYN_D and MNIST_M), Cifar10 & STL10, Office-31, Visda 2017, PACS and VLCS, to verify the effectiveness of our DSO and TDSO on SDPA and TDPA problems.

2. Related Work

2.1. IP Attack and Protection on Learning Models

Due to the huge business value, deep models are prone to be attacked by a variety of techniques including stealing private training data and deploying the model for private usage without verification [17, 5, 1, 34, 49]. To protect the IP of deep models against attacks, methods of authorizing model ownership and model usage have emerged in recent years [43]. The typical framework for ownership authorization is to compare model behaviors when encountering samples with and without pre-embedded watermarks to verify the model owner in the testing phase [46, 50, 23]. On the other hand, usage authorization directly locks the deep models via some pre-defined protocols in the encryption and decryption process [2]. To restrict the generalization capability of models on certain domains, the deployment authorization (DPA) has recently been proposed from the data-centric IP protection viewpoint, which authorizes the deployment of pre-trained models across different domains [43]. Following the concept proposed by [43] research on the protection of domain knowledge has emerged recently [42, 45]. For example, [42] has proposed the CUTI method by highlighting the private style features on the training domains such that unauthorized domains with irrelevant private styles will fail to deploy the trained models. Meanwhile, [45] extended the notion of intelligent protection from image processing tasks to NLP tasks.

2.2. Out-of-distribution Generalization/Detection

To enhance the generalization capability of learning models on both seen and unseen domains, researchers have explored various approaches ranging from domain adaptation (DA) to Out-of-distribution (OOD) generalization [24, 14, 3, 33, 39]. Specifically, domain adaptation transfers the knowledge learned from the source domain to a partially accessible target domain [14], while OOD generalization focuses on developing models that can generalize well on any completely invisible target domain [3]. *An analogy of DPA, TDPA, and SDPA is supervised fine-tuning, unsupervised domain adaptation, and domain generalization.*

In the context of OOD generalization, the framework of distributionally robust optimization (DRO)[35, 11] has played a significant role, which is closely related to our work. In particular, DRO seeks to achieve robustness by identifying models that perform well on distributions that are sufficiently close to the training distribution, but it faces challenges when modeling the overall distributional shift. In contrast to promoting model generalization, characterizing these close domains and degrading model performance on them is enough for our SPDA problem (as shown in Figure1). This observation naturally links the formulation of DRO to our problems.

Additionally, Out-of-distribution (OOD) detection, particularly the Near-distribution OOD detection (ND-OOD), has relevance to our problem [13, 25, 44][1]. Generally, OOD detection aims to identify whether the testing data is distributed differently from the training data[44]. However, conventional OOD detection methods often fail to recognize "near" testing data with small distributional shifts [25]. To overcome this challenge, ND-OOD has been proposed with a first-generate-then-detect approach using deep generative models [25]. Interestingly, we have found that such an approach is *almost identical to* the DPA baseline GNTL [43].

3. Problem Overview

This section overviews three deployment authorization problem settings including SUDPA, SDPA, and TDPA. Throughout this paper, we focus on the image classification as the learning task with features $\mathbf{X} \in \mathcal{X}$ and labels $\mathbf{Y} \in \mathcal{Y}$, where the authorized training domain and unauthorized domains share the same label space $\mathcal{Y}$ with C classes. In addition, we denote the classifier as f parameterized by $\theta \in \Theta$ throughout this paper. The authorizer aims for a model $f(\cdot|\theta)$ with parameters θ that satisfies the following conditions: (a) f preserves high accuracy on the authorized (training) domains, (b) f achieves poor performance on unauthorized domains. For ease to understand, one can compare the SUDPA to supervised learning, SDPA to domain generalization, and TDPA to domain adaptation, respectively.

Supervised Deployment Authorization (SUDPA). As shown in Fig. 2 (a), SUDPA (first proposed in [43]) assumes that the model authorizer possesses the full information on both training and unauthorized domains, as $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n_S} \sim P_S(\mathbf{X}, \mathbf{Y})$ and $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n_U} \sim P_U(\mathbf{X}, \mathbf{Y})$ (n_U and n_S are samples sizes). To solve SUDPA, the Non-Transferable Learning (NTL) [43] framework simultaneously decreases

[1] One can employ an ND-OOD detection model and assign an incorrect label to OOD samples.

the fitting loss on $P_S(\mathbf{X}, \mathbf{Y})$ and increases the fitting loss on $P_U(\mathbf{X}, \mathbf{Y})$, as follows:

$$\begin{aligned}\mathcal{L}_{NTL} =& \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim P_S(\mathbf{X},\mathbf{Y})}\left[D_{\mathrm{KL}}(f(\mathbf{x}|\theta)\|\mathbf{y})\right] \\ &- \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim P_U(\mathbf{X},\mathbf{Y})}\left[D_{\mathrm{KL}}(f(\mathbf{x}|\theta)\|\mathbf{y})\right],\end{aligned} \tag{1}$$

where D_{KL} refers to the KL-divergence loss [43].

Source-only Deployment Authorization with auxiliary patches. [43] proposes another problem that the (authorized) training domain has been embedded with pre-known patches but the (unseen) unauthorized domains are free of patch embedding. As the extension of NTL, a first-generate-then-authorize approach named GNTL is proposed [43] based on generative adversarial network (GAN). However, the such problem relies on the effectiveness of the reliability of the watermark [46]. In other words, if some model attackers intimate the patch embedded in the training data and embed such patches into the unauthorized images, then DPA fails. Regardless of ownership verification, we only consider the DPA problem in this paper.

Source-only Deployment Authorization (SDPA). To promote the practicality of SUDPA without any auxiliary information, we focus on the generic problem named the source-only deployment authorization (SDPA) in this paper, as shown in Fig. 2 (c). SDPA assumes that the model authorizer only owns the training data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n_S} \sim P_S(\mathbf{X}, \mathbf{Y})$ without access to any information on unauthorized data. The authorizer aims to develop a well pre-trained model $f(\cdot|\theta)$ that generalizes poorly on any unseen unauthorized P_U.

Target-combined Deployment Authorization (TDPA). Under certain conditions, we claim that the model authorizer can provide the features of unauthorized data as a trade-off between the SUDPA and SDPA. More formally, the model authorizer currently owns the training data as $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n_S} \sim P_S(\mathbf{X}, \mathbf{Y})$, together with the unauthorized features $\{\mathbf{x}_i\}_{i=1}^{n_U} \sim P_U(\mathbf{X})$, as shown in Fig. 2 (b).

4. Domain Specified Optimization

For SDPA, searching a model f with degraded performance over the total distributions on $\mathcal{X} \times \mathcal{Y}$ is exhaustive and difficult. Recalling the goal of deployment authorization, the model authorizer only concerns with suppressing the performance of f on the neighboring distributions P_U close to the training domain. Thus, we formulate the above intuition by bounding the distributional shift between P_U and P_S using some statistical divergence D (e.g., F-divergence [11]) as $D(P_U(\mathbf{X}, \mathbf{Y})\|P_S(\mathbf{X}, \mathbf{Y})) \leq \rho$, where ρ is a small value. Due to the difficulty encountered by existing methods when modeling a general distribution shift on $\mathcal{X} \times \mathcal{Y}$ since $P_U(\mathbf{X}, \mathbf{Y})$ is visible [37], we simplify the problem via the adversarial example shift assumption as in [11, 10, 27, 35] such that $P_U(\mathbf{X}) \neq P_S(\mathbf{X})$ and $P_U(\mathbf{Y} \mid \mathbf{X}) = P_S(\mathbf{Y} \mid \mathbf{X})$. Consequently, an ideal model $f(\cdot|\theta)$ for SDPA should fit the conditional distribution $P_S(\mathbf{Y} \mid \mathbf{X})$ when $P_U(\mathbf{X}) = P_S(\mathbf{X})$, while deviating from $P_U(\mathbf{Y} \mid \mathbf{X})$ when $P_U(\mathbf{X}) \neq P_S(\mathbf{X})$.

4.1. Restricting the model generalization over the uncertainty set

To move towards the above goal, we first have to model the unauthorized distributions surrounding closely to $P_S(\mathbf{X})$. Following [11], we first choose F-divergence [27] which is convex and solvable in polynomial time as the distribution divergence D; here, the former characteristic provides a well-established convergence property, while the latter one facilitates our implementation. Based on the adversarial shift assumption and the basic property of F-divergence [10], we have $D(P_U(\mathbf{X}, \mathbf{Y})\|P_S(\mathbf{X}, \mathbf{Y})) = D(P_U(\mathbf{X})\|P_S(\mathbf{X}))$. Finally, we define the F-divergence ball around $P_S(\mathbf{X})$ with the Cressie-Read divergence D_{ϕ_k} [29]: $\mathcal{U}_{P_S} = \{P_U(\mathbf{X}) : \mathbb{E}_{P_S}\left[\phi_k\left(\frac{dP_U(\mathbf{X})}{dP_S(\mathbf{X})}\right)\right] \leq \rho\}$, where $\phi_k := \frac{t^k - kt + k - 1}{k(k-1)}$ parameterized by k is convex and ρ is the radius parameter [11]. More formally, we refer to the ball $\mathcal{U}_{P_S} = B(P_S(\mathbf{X}), \rho)$ as the uncertainty set over $P_S(\mathbf{X})$ [11].

Subsequently, we propose the objective of Domain Specified optimization (DSO) via a min-max framework:

$$\mathcal{L}_{DSO} = \sup_{P_U}\left\{\mathbb{E}_{P_U}[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})] : P_U \in U_{\setminus P_s}\right\}, \tag{2}$$

where $\mathbf{Y}^{err}$ plays as the "error" labels to corrupt the training of f on $P_U(\mathbf{X}) \in U_{\setminus P_s}$. More formally, we construct the corrupted labels $\mathbf{Y}^{err}$ deviating from the underlying labels $\mathbf{Y}^U$ and force f to fit $\mathbf{Y}^{err}$. As $\mathbf{Y}^U$ is also invisible, we approximate $\mathbf{Y}^U \approx \mathbf{Y}^S$ and let $\mathbf{Y}^{err} = \mathbf{Y}^S + 1$ ($\mathbf{Y}^{err}$ is obtained as the remainder of C), such that the overlap between $\mathbf{Y}^{err}$ and $\mathbf{Y}^U$ reduces to zero. The reasons behind our approximation contain two critical aspects: (a) the adversarial shift assumption allows us to approximate $P_U(\mathbf{Y} \mid \mathbf{X})$ using the posterior of f as $P_U(\mathbf{Y} \mid \mathbf{X}) = P_S(\mathbf{Y} \mid \mathbf{X})$. Thus, it is reasonable for us to approximate $\mathbf{Y}^U$ sampled from $P_U(\mathbf{Y} \mid \mathbf{X})$ as $\mathbf{Y}^S$ sampled from $P_S(\mathbf{Y} \mid \mathbf{X})$. (b) previous DRO studies [10, 11, 35] have proved that (2) can be equivalently regarded as the optimization on some "hard" examples sampled from P_S. Hence, the approximation $\mathbf{Y}^U \approx \mathbf{Y}^S$ performed in these methods is natural, as $\mathbb{E}_{P_U}[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})]$ and $\mathbb{E}_{P_S}[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})]$ are both calculated on visible $P_S(\mathbf{X})$ via different weights.

4.2. Facilitating the min-max optimization

To optimize the min-max game in (2), it must be guaranteed that the Lagrangian function of (2) is a convex optimization problem [35]. Unfortunately, it is difficult to guarantee the convexity of inequality constraint $0 < \mathbb{E}_{P_S}\left[\phi_k\left(\frac{dP_U}{dP_S}\right)\right] \leq \rho$ in the Lagrangian function of (2) even with the convex ϕ_k and the linear expectation operator [35], which renders the non-convexity of the objective (2). Thus, we instead assume that the supremum in (2) is achieved in $\mathcal{U}_{P_S}$ as P_0 and modify (2) as follows:

$$\mathcal{L}_{DSO} = K(D_{\phi_k}(P_0 \parallel P_S)) \sup_{P_U \in \mathcal{U}_{P_S}} \mathbb{E}_{P_U}[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})], \tag{3}$$

where $K(t)$ is a scaling function such that $K(t) = t$ when $t \leq 0.2$ and $K(t) = 1.0$ otherwise. Note that the new objective in (3) vanishes when the supremum distribution P_0 reaches P_S, which is in fact equivalent to the formulation in (2) to some extent. Moreover, as the scaling term $K(D_{\phi_k}(P_0 \parallel P_S))$ is a constant with respect to the supremum over $P_U \in \mathcal{U}_{P_S}$, the formulation in (3) is convex based on previous conclusions [35]. Finally, combined with the standard supervised training loss on P_S, we present the batch-level formulation as the empirical version of (3) as follows:

$$\begin{aligned}\mathcal{L}_{Total} &= \sum_{i=1}^{n_b} \ell(f(\mathbf{X}_i^S \mid \theta), \mathbf{Y}_i^S) \\ &+ K(D_{\phi_k}(P_0 \parallel \mathbb{1}/n_b)) \sup_{U \in \mathcal{U}_{\mathbb{1}/n_b}} \sum_{i=1}^{n_b} U_i \ell(f(\mathbf{X}_i^S \mid \theta), \mathbf{Y}_i^{err}),\end{aligned} \tag{4}$$

where n_b is batch sample size and $\mathcal{U}_{\mathbb{1}/n_b} := \left\{U \in \mathbb{R}^{n_b} : U^\top \mathbb{1} = 1, U \geq 0, D_f(U \| \mathbb{1}/n_b) \leq \frac{\rho}{n_b}\right\}$ [27]. To facilitate computation, we set the parameter $k = 2$ such that the F-divergence D_f is reduced to the χ^2-divergence. As the projection of the loss vector $\{\ell_{i=1}^{n_b}\}$ on the uncertainty ball $\mathcal{U}_{\mathbb{1}/n_b}$, the supremum vector $P_0 \in \mathbb{R}^{n_b}$ is supported by the fast bisection method provided by [27].

When the unauthorized features $\mathbf{X}^U \sim P_U(\mathbf{X})$ are provided, the uncertainty set $U_{\backslash P_s}$ is reduced to a single point as P_U, such that the formulation of DSO in (4) is simplified. Therefore, we propose the Target-DSO (TDSO) for target-combined deployment authorization with the following empirical version:

$$\mathcal{L}_{Total}^T = \sum_{i=1}^{n_b} \ell(f(\mathbf{X}_i^S \mid \theta), \mathbf{Y}_i) + \sum_{i=1}^{n_b} \ell(f(\mathbf{X}_i^U \mid \theta), \mathbf{Y}_i^{err}). \tag{5}$$

For saving space, we leave the algorithms of detailed training process of DSO and TDSO in the appendix.

4.3. Discussion

Issues on F-divergence Although other alternatives (e.g., Wasserstein distance [26]) might allow worst-case distributions with different support from the training one, their tractable reformulations are only available under restrictive scenarios [11], which remains computationally challenging. By contrast, F-divergence is computationally efficient with the fast bisection method developed by [27], which is solvable in polynomial time.

Issues on the scaling function $K(t)$ can be considered as a hyper-parameter to achieve the Pareto optimality between minimizing the source risk and maximizing the target risk. In detail, $K(t) = t$ ensures the DSO loss will vanish to zero if the worst-case distribution is close to the training one. Second, the constant 1.0 constrains the weight of DSO loss to be less or equal to that of the training loss. We find such a truncation function achieves good empirical performance.

Issues on Adversarial Shift Unlike methods aimed at improving generalization in domain adaptation and out-of-distribution (OOD) generalization, such as those proposed in [37, 24], the adversarial shift assumption does not compromise the authorization capability of the function f. If the function f does not generalize well under an adversarial shift, it will perform even more poorly when confronted with a general shift. In other words, when every target domain is subject to an adversarial shift, the general shift yields distributions that are further away from the training distribution.

Extension to open-set recognition We note that the extension of our methodology to handle open-set scenarios by authorizing images with unseen styles or classes is intuitive. This framework treats open-set tasks equally to closed-set tasks, as explained in our paper. Pre-trained models naturally generalize poorly on testing-only classes, and considering training-only classes during testing is unnecessary.

The approximations of $Y_u \approx Y_s$ The presented approximation captures the notion that the SDPA task primarily focuses on P_U that are close to P_S, and that P_U with $Y_u \approx Y_s$ are closer to P_U where Y_U differs significantly from Y_s. In this scenario, testing distributions P_U differ conceptually from P_S, causing poor generalization of models trained on P_S. Hence, considering P_U with significantly different Y_U from Y_S is unnecessary. We highlight that such label approximation in marginal DRO methods [11] is very common.

5. Theoretical Analysis

In this section, we provide a deeper insight into the behavior of the proposed DSO method from three aspects: (a) its generalization bound in the finite sample case; (b) its convergence property with increasing sample number (c) its authorization performance on any testing distribution. Following previous protocols [11], we assume the bounded losses as $\left|\ell(f(\mathbf{X}^S \mid \theta), \mathbf{Y}^{err})\right| \leq M_1$ and $\left|\ell(f(\mathbf{X}^S \mid \theta), \mathbf{Y}^S)\right| \leq M_2$. Moreover, we denote the empirical distribution of P_S by $\widehat{P}_N^S$. Detailed proof of our theory is provided in the appendix.

5.1. Generalization Bound of DSO

Following [27], we denote the empirical and expected worst-subgroup risk in (3) as $\mathcal{R}_k(\ell(f(\mathbf{X}^S \mid \theta), \mathbf{Y}^{err}); P^S)$

and $\widehat{\mathcal{R}}_k(\ell(f(\mathbf{X}^S \mid \theta), \mathbf{Y}^{err}); \widehat{P}^S)$, respectively. Subsequently, we establish the generalization bound for each model parameter θ in the following theorem:

Theorem 5.1. *For a fixed θ and $u \geq 0$, the following inequality holds with probability $1 - 2e^{-u}$:*

$$\|\mathcal{R}_k(\ell(\theta); \widehat{P}_N^S) - \mathcal{R}_k(\ell(\theta); P^S)\|$$
$$\leq 5 s_k \rho_k^2 {n_s}^{-\frac{1}{\max(k_*,2)}} (\frac{1}{k} + \sqrt{u + \log n_s})(\sqrt{v(\rho_k, M_1)} + \frac{1}{2})^2,$$

where $v(\rho_k, M_1) = \max\left(\frac{\rho_k}{\rho_k - 1}, 2\right) M_1$ and $k^* = \frac{k}{k-1}$.

Remark Intuitively, Theorem 5.1 characterizes the fact that the gap between the empirical and expected versions of the objective function of our DSO vanishes within increasing training samples.

5.2. Convergence Property of DSO

We then present the convergence analysis of our DSO method over the total parameter space $\theta \in \Theta$. With the covering number technique on the model class space $\mathcal{F} = \{f(\cdot \mid \theta)\}$ equipped with the norm $\|f\|_{\mathcal{L}_\infty} = \sup_x f(x \mid \theta)$, we define the ϵ-covering number of $\mathcal{F}$ as $N(\mathcal{F}, \xi)$, where $V(\mathcal{F}, \xi)$ denotes the corresponding covering set. More specifically, we let $N(\mathcal{F}, \frac{\xi(u,v,n_s)}{3})$ with $\xi(u, v, n_s) = 5 \frac{\rho_k}{C} {n_s}^{-\frac{1}{\max(k_*,2)}} (\frac{1}{k} + \sqrt{u + \log n_s})(\sqrt{v(\rho_k, M_1)} + \frac{1}{2})^2$ be a $\xi(u, v, n_s)$-finite cover of $\mathcal{F}$. With the empirical risk minimizer $\hat{\theta}_{n_s}$ of $\mathcal{R}_k$ under $\widehat{P}_N^S$, we denote the expected and empirical versions of the total loss in (4) as by $\widehat{\mathcal{R}}_{Total}(\ell(\hat{\theta}_{n_s}); P^S)$ and $\widehat{\mathcal{R}}_{Total}^N(\ell(\theta); P^S)$, respectively. Finally, we characterize the convergence behaviour of the total objective function in (4).

Theorem 5.2. *For $u \geq 0$, the loss function $\ell(f(\cdot \mid \theta), \cdot)$ that is C-Lipschitz with respect to its first parameter $f(\cdot \mid \theta)$, and under the condition that $\left|\ell(f(\mathbf{X}^S \mid \theta), \mathbf{Y}^{err})\right| \leq M_1$ and $\left|\ell(f(\mathbf{X}^S \mid \theta), \mathbf{Y}^S)\right| \leq M_2$, the following inequality holds with probability $1 - 2\left(N(\mathcal{F}, \frac{\xi(u,v,n_s)}{3}) + N(\mathcal{F}, \sqrt{\frac{2u}{n_s}} \frac{M_2}{4C})\right) e^{-u}$:*

$$\widehat{\mathcal{R}}_{Total}(\ell(\hat{\theta}_{n_s}); P^S) - \inf_{\theta \in \Theta} \widehat{\mathcal{R}}_{Total}^N(\ell(\theta); P^S)$$
$$\leq 2\sqrt{\frac{2u}{n_s}} M_2 + 30 s_k \rho_k^2 {n_s}^{-\frac{1}{\max(k_*,2)}} C(k, u, n_s, v, \rho_k, M_1),$$

$C(k, u, n_s, v, \rho_k, M_1) = (\frac{1}{k} + \sqrt{u + \log n_s})(\sqrt{v(\rho_k, M_1)} + \frac{1}{2})^2$ is a constant.

Remark Intuitively, Theorem 5.2 entails that our DSO method has well-established convergence property in the statistical sense within increasing training samples.

5.3. Authorization Analysis of DSO

In this section, we provide an analysis of the authorization guarantee of DSO in the SDPA scenario. For any unauthorized distribution P_U, if $P_U \notin \mathcal{U}_{P_S} \backslash P_S$, we demonstrate that the expected risk of f with respect to $\mathbf{Y}^{err}$ on P_U, namely $\mathbb{E}_{P_U}\left[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})\right]$, is bounded by the corresponding expected risk on any distribution $P_1 \in \mathcal{U}_{P_S} \backslash P_S$.

Lemma 5.3. *Suppose that $P_1 \in \mathcal{U}_{P_S}$ and $P_U \notin \mathcal{U}_{P_S}$ have the same support and $P_1(Y \mid \mathbf{X}) = P_U(Y \mid \mathbf{X})$ and $|\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})| \leq M_1$, then $\mathbb{E}_{P_U}\left[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})\right]$ is bounded by $\mathbb{E}_{P_1}\left[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})\right]$ with the F-divergence $D_{\phi_k}(P \| P_1)$ as follows:*

$$\mathbb{E}_{P_U}\left[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})\right]$$
$$\leq k(k-1) D_{\phi_k}(P_U \| P_1)^{\frac{1}{k}} {M_1}^{\frac{1}{k}} \mathbb{E}_{P_1}\left[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})\right]^{1-\frac{1}{k}}.$$

Remark Lemma 5.3 entails that when the divergence D_{ϕ_k} is bounded as $\sup_{P_1 \in \mathcal{U}_{P_S}} D_{\phi_k}(P_U \| P_1) \leq W$, the expected risk of P_U that distributes out of the divergence ball is bounded by $k(k-1) W^{\frac{1}{k}} {M_1}^{\frac{1}{k}} \mathbb{E}_{P_1}\left[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})\right]^{1-\frac{1}{k}}$. Since the convergence property of DSO has been proven in the previous subsection, the well-optimized $\mathbb{E}_{P_1}\left[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})\right]$ will reach a small value, which ensures a small value of $\mathbb{E}_{P_U}\left[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})\right]$ with poor authorization performance.

6. Experiments

Datasets and Implementation In this section, we choose the main body of the "DomainBed" test suite [15] to conduct experiments, which includes four domain adaptation benchmarks—Digits (MNIST (N) [8], USPS (U) [18], SVHN (H) [28], SYN_D (S) [31] and MNIST_M (M) [14]), Cifar10 [6]&STL10 [6], Office-31 [32], and Visda 2017 [30]—together with two OOD generalization benchmarks, PACS [21] and VLCS [12]. Different from DomainBed, we do not consider Terra Incognita. Office-Home and DomainNet due to their poor cross-domain generalization results under the supervised setting [15]. For PACS and VLCS, we additionally train the model on each of the combined three domains, with the remaining one as the unauthorized domain [15]. Our experiments are performed on Python 3.6.9, PyTorch 1.9.0+cu111, CUDA 11.4, and NVIDIA GeForce RTX 3090 GPUs. Details on network structures of solutions are introduced in the appendix.

Baselines We mainly compare our DSO and TDSO with a well-trained source model and an existing DPA baseline GNTL [43]. Notably, as GNTL follows a first-generate-then-classify pipeline by tuning a GAN model, it can be equivalently regarded as an ND-OOD detection baseline.

Metrics To facilitate clear comparisons, following [43], we provide the relative performance drop of DSO, TDSO, and GNTL with respect to a well-trained source model. Notably, in Table 1, we compute the average authorization performance by first training each method on a single domain (Amazon in Office-31), and subsequently averaging the testing drop in the remaining domains (Webcam and Dslr in Office-31). To mitigate the impact of random variations, we report all metrics based on five independent experimental repetitions.

Parameters settings Following [43], we apply VGG-11 [36] for digits recognition and ResNet-50 [16] for the remaining benchmarks, where all networks are initialized as the pre-trained version of ImageNet. Moreover, we adopt the Adam [19] optimizer with the learning rate initialized as 0.00005. For the DSO and TDSO methods, we set $\rho = 50$ as in [27] throughout the experiments. For the GAN-augmented NTL (GNTL) method, we conduct each experiment using the original implementation developed in [43]. Note that, we test the effectiveness of GNTL without any watermarks (patches), as we focus on the pure deployment authorization in this paper and wish to facilitate a fair comparison.

Questions Throughout the experiment, we validate our proposed methods by answering the following two questions:

(a) Does our methods hurt the performance on the source (authorized) distribution?

(b) Does our methods effectively degrade the performance on the target (unauthorized) domains?

6.1. Results on average authorization performance

As shown in Tab. 1, we report the average authorization performance for each solution under both the source-only and target-combined authorization. In the interest of saving space, detailed results are provided in the appendix. Regarding the first question, our DSO and TDSO have a smaller performance drop (often less than 3%) on the authorized domain, which entails that our methods will not hurt the authorization performance. Meanwhile, we observe that while the GNTL method [43] achieves source-only authorization in some cases (e.g., Digit), in most cases, its authorization performance cannot be guaranteed with nearly no performance degradation observed. In contrast, our DSO achieves dominant authorization results with an acceptable decline in its pre-training performance (usually less than 3%), while requiring no prior network component or augmentation operation. This comparison answers our second problem, which indicates the superiority of distributional performance degradation via the uncertainty set formulation in our DSO. Meanwhile, when the unauthorized features are available in the target-combined scenario, our TDSO nearly achieves near-perfect authorization by dramatically degrading the performance of the unauthorized domains.

6.2. Study on Uniform deployment authorization

Beyond the average performance over multiple unauthorized domains in Tab. 1, the model authorizer instead prefers to authorize the deployment of the learning model on unauthorized domains much closer to the training data, which we call as the uniform authorization. For example, the supervised model trained on the VOC domain in the VLCS benchmark generalizes well on the Caltech domain with 91% accuracy, while naturally performing poorly on the Labelme domain with only 51% accuracy. Thus, a reliable authorization solution for realistic cases only needs to degrade the model performance on the Caltech domain, while the accuracy decrease on Labelme is not important, as it is not necessary to authorize the Labelme domain. Based on the intuition in above, we perform a case study on the six benchmarks by simulating a realistic scenario in which the model authorizer aims to (a) preserve a pre-training accuracy over 95% on the training domain, and (b) degrade the model performance to be lower than 70% on domains with supervised generalization over 80%. We obtain the results in Tab. 2, selected from the total authorization results (shown in appendix). We can observe that the proposed DSO achieves near-perfect authorization (except for $\overline{\mathrm{P}} \rightarrow \mathrm{P}$) by successfully degrading the performance on unauthorized domains with a large margin on 15 out of the total 16 tasks. By contrast, GNTL evidently fails on these challenging but realistic authorization tasks with a success ratio of 4/16, which further claims the superiority of our DSO.

6.3. Studies on the Label Corruption Protocol

In Section 4.1, we propose the label corruption protocol $\mathbf{Y}^{err} = \mathbf{Y}^{S} + 1$. One might question whether an attacker could easily recover the model's utility on OOD data by exploiting this protocol. However, we contend that this concern is unwarranted for two reasons: (a) the corruption protocols are not disclosed to model users, and (b) $\mathbf{Y}^{err} = \mathbf{Y}^{S} + 1$ is merely one intuitive implementation of our DSO framework, which also supports irreversible protocols such as random corruption (i.e., uniformly selecting labels from the remaining incorrect classes). As illustrated in Figure 3, we evaluate the effectiveness of random corruption on the Office-31 and Visda-2017 benchmarks. Results demonstrate that both the transition ($\mathbf{Y}^{err} = \mathbf{Y}^{S} + 1$) and random corruption approaches achieve promising authorization performance. Besides, we note that using random error labels without our DSO framework is impractical because maintaining the model's performance on data originating from the same distribution as the training domain is necessary.

6.4. Convergence and Parameter Sensitivity

Besides, we show that both the DSO loss and the supervised loss convergence well in Fig. 4a and Fig. 4b, which is coherent with our Theorem 5.2. We also claim that the

Table 1: Average performance drop (%) of each solution on SDPA and TDPA, where bold value represents the highest accuracy in each row. Specifically, S-Drop denotes the accuracy drop on the source (authorized) domain, while T-Drop refers to the **average** accuracy drop on multiple target (unauthorized) domains in the same benchmark. Note that both S-Drop and T-Drop are obtained by comparing each solution with the performance of the purely supervised model. In addition, the combined results on PACS and VLCS indicate the average performance on the four combined training tasks.

Benchmark	Task	Source-Only				Target-Combined	
	Methods	GNTL		DSO		TDSO	
	Training Domain	S-Drop	T-Drop	S-Drop	T-Drop	S-Drop	T-Drop
Office-31	Amazon	0.0%↓	9.0%↓	1.0%↓	**30.0%**↓	0.0%↓	**67.5%**↓
	Webcam	0.0%↓	21.5%↓	3.0%↓	**64.5%**↓	0.0%↓	**94.0%**↓
	Dslr	3.0%↓	27.5%↓	3.0%↓	**63.5%**↓	0.0%↓	**87.0%**↓
Digit	MNIST	0.0%↓	15.5%↓	1.0%↓	**19.5%**↓	0.0%↓	**45.5%**↓
	USPS	3.0%↓	22.0%↓	2.0%↓	**25.0%**↓	0.1%↓	**40.5%**↓
	SVHN	5.0%↓	6.8%↓	2.0%↓	**17.8%**↓	0.0%↓	**57.3%**↓
	SYN_D	0.0%↓	3.8%↓	1.0%↓	**21.0%**↓	1.0%↓	**74.0%**↓
	MNIST_M	1.0%↓	**35.7%**↓	3.0%↓	35.0%↓	0.0%↓	**61.0%**↓
Cifar10 & STL10	Cifar10	0.0%↓	6.0%↓	3.0%↓	**12.0%**↓	0.0%↓	**32.0%**↓
	STL10	0.0%↓	11.0%↓	1.0%↓	**25.0%**↓	2.0%↓	**51.0%**↓
Visda2017	Real	5.0%↓	6.0%↓	0.0%↓	**28.0%**↓	0.0%↓	**72.0%**↓
	Synthetic	0.0%↓	1.0%↓	1.0%↓	**16.0%**↓	0.0%↓	**28.0%**↓
PACS	Art_painting	0.0%↓	2.0%↓	1.0%↓	**13.0%**↓	0.0%↓	**51.0%**↓
	Cartoon	3.0%↓	0.3%↓	1.0%↓	**8.0%**↓	1.0%↓	**47.3%**↓
	Photo	1.0%↓	9.7%↓	1.0%↓	**20.0%**↓	1.0%↓	**40.0%**↓
	Sketch	0.0%↓	2.0%↓	1.0%↓	**12.0%**↓	0.0%↓	**15.0%**↓
	Combined	0.0%↓	11.8%↓	0.0%↓	**25.8%**↓	0.7%↓	**58.0%**↓
VLCS	Caltech101	3.0%↓	0.7%↓	0.0%↓	**6.7%**↓	3.0%↓	**10.7%**↓
	Labelme	0.0%↓	2.3%↓	0.0%↓	**18.3%**↓	1.0%↓	**32.3%**↓
	VOC2007	0.0%↓	0.0%↓	3.0%↓	**26.3%**↓	0.0%↓	**46.7%**↓
	Sun09	0.0%↓	0.0%↓	1.0%↓	**13.6%**↓	2.0%↓	**32.3%**↓
	Combined	0.0%↓	0.0%↓	2.0%↓	**23.0%**↓	0.7%↓	**47.8%**↓

Table 2: Uniform authorization results of case study, where $\overline{\text{S}}$ → S, $\overline{\text{A}}$ → A and $\overline{\text{P}}$ → P refers to P+C+A → S, P+C+S → A and A+C+S → P. Result in red and green represents the successful authorization and the failure.

Tasks	Office-31		PACS			VLCS		
	D→W	W→D	$\overline{\text{S}}$ →S	$\overline{\text{A}}$ →A	$\overline{\text{P}}$ →P	L→C	V→C	$\overline{\text{C}}$ →C
Sup	96.2%	96.1%	80.1%	81.9%	94.4%	81.3%	92.0%	92.3%
G_NTL	89.1%	93.6%	64.2%	66.6%	83.8%	77.5%	92.7%	92.1%
DSO	40.1%	46.2%	57.5%	47.5%	72.1%	33.1%	43.5%	43.5%

Tasks	Digit							Visda
	N→U	U→N	S→N	S→U	S→H	M→N	M→U	R→S
Sup	86.2%	89.5%	88.0%	83.1%	85.6%	96.4%	83.2%	80.1%
G_NTL	71.1%	72.9%	84.4%	81.0%	86.8%	53.1%	13.4%	74.5%
DSO	68.5%	53.2%	68.1%	69.5%	60.6%	46.2%	19.1%	52.1%

proposed DSO is not sensitive to the selection of the radius parameter ρ, as already explained in [27]. The corresponding results are shown in Fig. 5 by varying ρ on three tasks including MNIST → USPS in Digit, Real → Synthetic in Visda2017 and VOC2007 → Caltech101 in VLCS.

Finally, we investigate the performance of TDSO with varying sample sizes in the unauthorized (target) domain. By tuning the ratio of N_u (number of the used unlabelled data) to N_t (the number of the whole target samples) ranging from 0.2 to 1.0, we find our TDSO method achieves good authorization performance at all the time in Table 3.

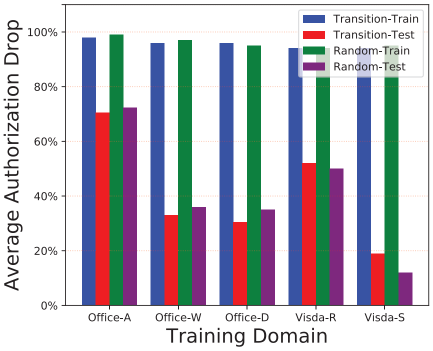
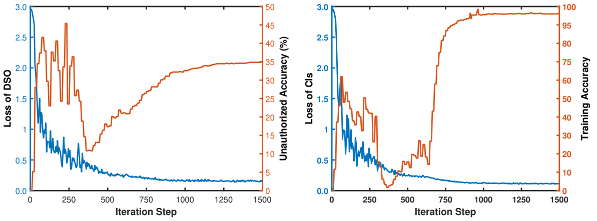


Figure 3: Authorization with different label corruption protocols, where the x-axis denote training domains. For the random corruption protocal, Random-train refer to the model performance on the training (authorized) domain, while Random-test refer to the average performance on the rest testing (unauthorized) domain.



(a) DSO Loss

(b) Supervised Loss

Figure 4: Convergence results of DSO on the task Dslr → Amazon in Office-31 benchmark, where the left side shows the loss curve and the right side shows the accuracy curve. In the training phase, we record the first 1500 steps to report the convergence situation.

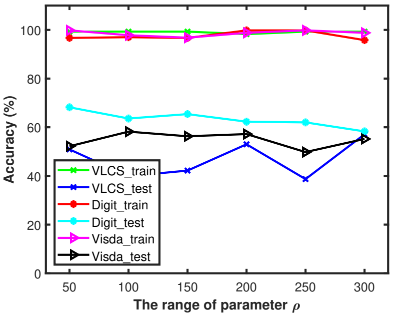


Figure 5: Parameter Sensitivity of DSO on VLCS, Digit and Visda2017 with $\rho \in \{50, 100, 150, 200, 250, 300\}$.

Comparison with recent Non-transferable baselines Besides, we have also conducted a comprehensive comparison with the recently proposed CUTI model [42] on Visda2017 (Synthetic to Real) and Office-31 (Dslr to Webcam, and Webcam to Dslr) benchmarks in Figure 6. Corresponding results demonstrates the superiority of our DSO.

Table 3: Authorization results of TDSO with varying unlabelled sample size, which is reported on the DSLR → Webcam task in the Office-31 benchmark.

Ratio	0.2	0.4	0.6	0.8	1.0
Authorized	97.6%	96.8%	97.0%	96.2%	98.2%
Unauthorized	36.9%	23.6%	10.3%	5.0%	4.8%

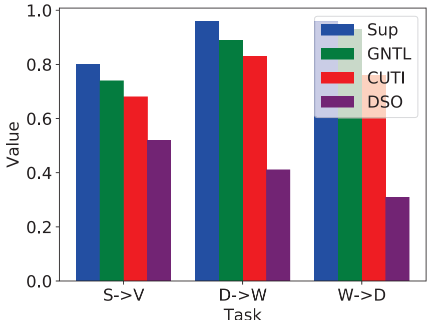


Figure 6: Comparison with CUTI on Visda and Office-31.

7. Conclusion and Discussion

This paper contributes the Domain Specified Optimization (DSO) method to achieve the newly proposed deployment authorization for the intelligent protection of pretrained models. By distributionally degrading the model performance over the uncertainty set surrounding the training domain, our DSO can successfully restrict the generalization capability of the pre-trained models on unauthorized domains. Extensive experiments conducted on six benchmarks confirm the effectiveness of our methods.

However, we assume there is only a single authorized domain in this paper, while a more complicated scene exists. For example, someone wants the model trained on MNIST to act poorly on USPS but generalize well on SVHN. Achieving heterogeneous control of the generalization capability will be considered in our future work. Besides, considering inducing the spurious correlation between labels and training-specific styles are also interesting.

Acknowledgement

This work was supported in part by National Natural Science Foundation of China (No.91948303-1, 62006207) and National Key RD Program of China (2021ZD0140301). We would also like to thank Dr. Hao Zou and Dr. Adi Lin for productive discussions.

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.

[2] Manaar Alam, Sayandeep Saha, Debdeep Mukhopadhyay, and Sandip Kundu. Deep-lock: Secure authorization for deep neural networks. *arXiv preprint arXiv:2008.05966*, 2020.

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

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[5] Xinyun Chen, Wenxiao Wang, Chris Bender, Yiming Ding, Ruoxi Jia, Bo Li, and Dawn Song. Refit: a unified watermark removal framework for deep learning systems with limited data. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 321–335, 2021.

[6] 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.

[7] Robert Dale. Gpt-3: What's it good for? *Natural Language Engineering*, 27(1):113–118, 2021.

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

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[10] John C Duchi, Peter W Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 46(3):946–969, 2021.

[11] John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

[12] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.

[13] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34:7068–7081, 2021.

[14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(1):2096–2030, 2016.

[15] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 148–162, 2019.

[18] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Long Lan, Xiao Teng, Jing Zhang, Xiang Zhang, and Dacheng Tao. Learning to purification for unsupervised person reidentification. *IEEE Transactions on Image Processing*, 2023.

[21] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[22] Haoxuan Li, Chunyuan Zheng, and Peng Wu. Stabledr: Stabilized doubly robust learning for recommendation on data missing not at random. In *The Eleventh International Conference on Learning Representations*, 2022.

[23] Zheng Li, Chengyu Hu, Yang Zhang, and Shanqing Guo. How to prove your model belongs to you: A blind-watermark based framework to protect intellectual property of dnn. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 126–137, 2019.

[24] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

[25] Hossein Mirzaei, Mohammadreza Salehi, Sajjad Shahabi, Efstratios Gavves, Cees GM Snoek, Mohammad Sabokrou, and Mohammad Hossein Rohban. Fake it till you make it: Near-distribution novelty detection by score-based generative models. *arXiv preprint arXiv:2205.14297*, 2022.

[26] Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

[27] Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Advances in neural information processing systems*, 29, 2016.

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

[29] Hiroaki Ogata and Masanobu Taniguchi. Cressie–read power-divergence statistics for non-gaussian vector stationary processes. *Scandinavian journal of statistics*, 36(1):141–156, 2009.

[30] 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.

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

[32] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of the European Conference on Computer Vision*, pages 213–226. Springer, 2010.

[33] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

[34] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. $updates-leak$: Data set inference and reconstruction attacks in online learning. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1291–1308, 2020.

[35] Alexander Shapiro. Distributionally robust stochastic programming. *SIAM Journal on Optimization*, 27(4):2258–2275, 2017.

[36] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*. Computational and Biological Learning Society, 2015.

[37] Tobias Sutter, Andreas Krause, and Daniel Kuhn. Robust generalization despite distribution shift via minimum discriminating information. *Advances in Neural Information Processing Systems*, 34, 2021.

[38] Hao Wang, Jianxun Lian, Mingqi Wu, Haoxuan Li, Jiajun Fan, Wanyue Xu, Chaozhuo Li, and Xing Xie. Convformer: Revisiting transformer for sequential user modeling. *arXiv preprint arXiv:2308.02925*, 2023.

[39] Haotian Wang, Wenjing Yang, Ji Wang, Ruxin Wang, Long Lan, and Mingyang Geng. Pairwise similarity regularization for adversarial domain adaptation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2409–2418, 2020.

[40] Haotian Wang, Wenjing Yang, Longqi Yang, Anpeng Wu, Liyang Xu, Jing Ren, Fei Wu, and Kun Kuang. Estimating individualized causal effect with confounded instruments. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1857–1867, 2022.

[41] Haotian Wang, Wenjing Yang, Zhenyu Zhao, Tingjin Luo, Ji Wang, and Yuhua Tang. Rademacher dropout: An adaptive dropout for deep neural network via optimizing generalization gap. *Neurocomputing*, 357:177–187, 2019.

[42] 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.

[43] Lixu Wang, Shichao Xu, Ruiqi Xu, Xiao Wang, and Qi Zhu. Non-transferable learning: A new approach for model ownership verification and applicability authorization. In *International Conference on Learning Representations*, 2021.

[44] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

[45] Guangtao Zeng and Wei Lu. Unsupervised non-transferable text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10071–10084, 2022.

[46] 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.

[47] Jing Zhang and Dacheng Tao. Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10):7789–7817, 2020.

[48] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *International Journal of Computer Vision*, pages 1–22, 2023.

[49] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 253–261, 2020.

[50] Jingjing Zhao, Qingyue Hu, Gaoyang Liu, Xiaoqiang Ma, Fei Chen, and Mohammad Mehedi Hassan. Afa: Adversarial fingerprinting authentication for deep neural networks. *Computer Communications*, 150:488–497, 2020.

[51] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003.