Domain Generalization in Vision: A Survey

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy

Abstract—Generalization to out-of-distribution (OOD) data is a capability natural to humans yet challenging for machines to reproduce. This is because most learning algorithms strongly rely on the i.i.d. assumption on source/target data, which is often violated in practice due to domain shift. Domain generalization (DG) aims to achieve OOD generalization by using only source data for model learning. Since first introduced in 2011, research in DG has made great progresses. In particular, intensive research in this topic has led to a broad spectrum of methodologies, e.g., those based on domain alignment, meta-learning, data augmentation, or ensemble learning, just to name a few; and has covered various vision applications such as object recognition, segmentation, action recognition, and person re-identification. In this paper, for the first time a comprehensive literature review is provided to summarize the developments in DG for computer vision over the past decade. Specifically, we first cover the background by formally defining DG and relating it to other research fields like domain adaptation and transfer learning. Second, we conduct a thorough review into existing methods and present a categorization based on their methodologies and motivations. Finally, we conclude this survey with insights and discussions on future research directions.

Index Terms—Out-of-Distribution Generalization, Domain Shift, Model Robustness, Machine Learning, Computer Vision

1 Introduction

If an image classifier was trained on photo images, would it work on sketch images? What if a car detector trained using urban images is tested in rural environments? Is it possible to deploy a semantic segmentation model trained using sunny images under rainy or snowy weather conditions? Can a health status classifier trained using one patient's electrocardiogram data be used to diagnose another patient's health status? Answers to all these questions depend on how well the machine learning models can deal with one common problem, namely the *domain shift* problem. Such a problem refers to the distribution shift between a set of training (source) data and a set of test (target) data [1], [2], [3], [4].

Most statistical learning algorithms strongly rely on an over-simplified assumption, that is, the source and target data are independent and identically distributed (i.i.d.), while ignoring out-of-distribution (OOD) scenarios commonly encountered in practice. This means that they are not designed with the domain shift problem in mind. As a consequence, a learning agent trained only with source data will typically suffer significant performance drops on an OOD target domain. The domain shift problem has seriously impeded large-scale deployments of machine learning models. One might be curious if recent advances in deep neural networks [5], [6], known as deep learning [7], can mitigate this problem. Studies in [2], [8] suggest that deep learning models' performance degrades significantly on OOD datasets, even with just small variations in the data generating process. This highlights the fact that the successes achieved by deep learning so far have been largely driven by supervised learning with large-scale annotated datasets like ImageNet [9]—again, relying on the i.i.d. assumption.

Research on how to deal with domain shift has been extensively conducted in the literature. A straightforward solution to bypass the OOD data issue is to collect some data from the target domain to adapt a source-domain-trained model. Indeed, this domain adaptation (DA) problem has received much attention recently [10], [11], [12], [13], [14], [15], [16]. However, DA relies on a strong assumption that target data are accessible for model adaptation, which does not always hold in practice. In many applications, target data are difficult to obtain or even unknown before deploying the model. For example, in biomedical applications where domain shift occurs between different patients' data, it is impractical to collect each new patient's data in advance [17]; in traffic scene semantic segmentation it is infeasible to collect data capturing all different scenes and under all possible weather conditions [18].

To overcome the domain shift problem, as well as the absence of target data, the problem of *domain generalization* (DG) is introduced [19]. Specifically, the goal in DG is to learn a model using data from a single or multiple related but distinct source domains in such a way that the model can generalize well to any OOD target domain. In recent years, DG has received increasing attention from the research community due to its importance to practical applications [20], [21], [22], [23], [24], [25], [26].

Since the first introduction in 2011 by Blanchard et al. [19], a plethora of methods have been developed to tackle the OOD generalization issue. This includes methods based on aligning source domain distributions for domain-invariant representation learning [27], [28], exposing the model to domain shift during training via meta-learning [29], [30], or augmenting data with image synthesis [31], [32], just to name a few. From the application point of view, existing DG methods have been applied to handwritten digit recognition [31], [32], object recognition [33], [34], semantic segmentation [18], [35], person re-

- *K. Zhou, Z. Liu and C.C. Loy are with the S-Lab, Nanyang Technological University, Singapore. E-mail: {kaiyang.zhou, ziwei.liu, ccloy}@ntu.edu.sg.*
- *Y. Qiao is with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. E-mail: yu.qiao@siat.ac.cn.*
- *T. Xiang is with the Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, UK. E-mail: t.xiang@surrey.ac.uk.*

identification [20], [31], face recognition [36], action recognition [27], [37], and many more. Despite these efforts, it is commonly acknowledged that DG remains an open challenge. Indeed, without access to target domain data, training a generalizable model that can work effectively in any unseen target domain is arguably one of the hardest problems in machine learning.

In this survey paper, we aim to provide a timely and comprehensive literature review. Major methods and applications introduced in the past decade are summarized, with a focus in the computer vision area. Potential future directions are also discussed. The rest of the paper is organized as follows. In § 2, we cover the background knowledge, giving the problem definitions and comparing DG with several related research areas like domain adaptation and transfer learning. Commonly used datasets for benchmarking DG algorithms are also discussed. In § 3, we review the existing DG methodologies proposed in the last decade and present a categorization. In § 4, we conclude this paper with insights and discussions on potential research directions for future work. As the first survey paper on this topic, we hope this timely survey can provide the research community with clarity and motivations for further advances.

2 Background

2.1 A Brief History of Domain Generalization

The DG problem was first introduced by Blanchard et al. [19] as a machine learning problem, while the term 'domain generalization' was later coined by Muandet et al. [17]. Unlike other related learning problems such as domain adaptation and transfer learning, DG considers scenarios where the target data are *inaccessible* during model learning. In [19], the motivation behind DG originates from a medical application called automatic gating of flow cytometry data. The objective for the gating problem is to design algorithms to automate the process of classifying cells in patients' blood samples based on different properties, e.g., to distinguish between lymphocytes and non-lymphocytes. Such a technology is crucial in facilitating the diagnosis of the health of patients since manual gating is extremely time-consuming and requires domain-specific expertise. However, due to distribution shift between different patients' data, a classifier learned using data from historic patients does not generalize to new patients, and meanwhile, collecting new data for model fine-tuning is impractical, thus motivating research on the DG problem.

In computer vision, a seminal work done by Torralba and Efros [38] raised attention on the cross-domain generalization issue. They performed a thorough investigation into the cross-dataset generalization performance of object recognition models using six popular benchmark datasets. Their findings suggested that dataset biases, which are difficult to avoid, can lead to poor generalization performance. For example, as shown in [38], a person classifier trained on Caltech101 [39] obtained a very low accuracy (11.8%) on LabelMe [40], though its same-dataset performance was near-perfect (99.6%). Following [38], Khosla et al. [41] targeted the cross-dataset generalization problem in classification and detection tasks, and proposed to learn domain-specific bias vectors and domain-agnostic weight vectors based on support vector machine (SVM) classifiers. In recent years, the DG problem has also been studied for various computer vision applications, such as instance retrieval [31], [42], image segmentation [18], [43], [44], face recognition [36], and face anti-spoofing [45], [46].

2.2 Problem Definition

Notations We first introduce some notations and definitions that will be used in this survey. Let $\mathcal{X}$ be the input (feature) space and $\mathcal{Y}$ the target (label) space, a *domain* is defined as a joint distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$.[1] For a specific P_{XY}, we refer to P_X as the marginal distribution on X, $P_{Y|X}$ the posterior distribution of Y given X, and $P_{X|Y}$ the class-conditional distribution of X given Y. A learning function or model is defined as $f : \mathcal{X} \rightarrow \mathcal{Y}$. A loss function is defined as $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \infty)$.

In the context of DG, we assume to have access to K similar but distinct source domains, $\mathcal{S} = \{S_k\}_{k=1}^{K}$, each associated with a joint distribution $P_{XY}^{(k)}$. In general, $P_{XY}^{(k)} \neq P_{XY}^{(k')}$ with $k \neq k'$ and $k, k' \in \{1, ..., K\}$. Each source domain S_k contains i.i.d. data-label pairs sampled from $P_{XY}^{(k)}$, namely $S_k = \{(x_i^{(k)}, y_i^{(k)})\}_{i=1}^{N_k}$ with $(x_i^{(k)}, y_i^{(k)}) \sim P_{XY}^{(k)}$. We use $P_{XY}^{\mathcal{S}}$ to denote the overall source joint distribution. The target domain is denoted by $\mathcal{T} = \{x_i^{\mathcal{T}}\}_{i=1}^{N_\mathcal{T}}$ where the data are sampled from the marginal $P_X^{\mathcal{T}}$. Note that the labels in $\mathcal{T}$ are unavailable and need to be predicted. The corresponding joint distribution of $\mathcal{T}$ is denoted by $P_{XY}^{\mathcal{T}}$. Also, $P_{XY}^{\mathcal{T}} \neq P_{XY}^{(k)}, \forall k \in \{1, ..., K\}$.

Definition Given labeled source domains $\mathcal{S}$, the goal of DG is to learn a model f using data from $\mathcal{S}$ such that the model can generalize well to an unseen domain $\mathcal{T}$.

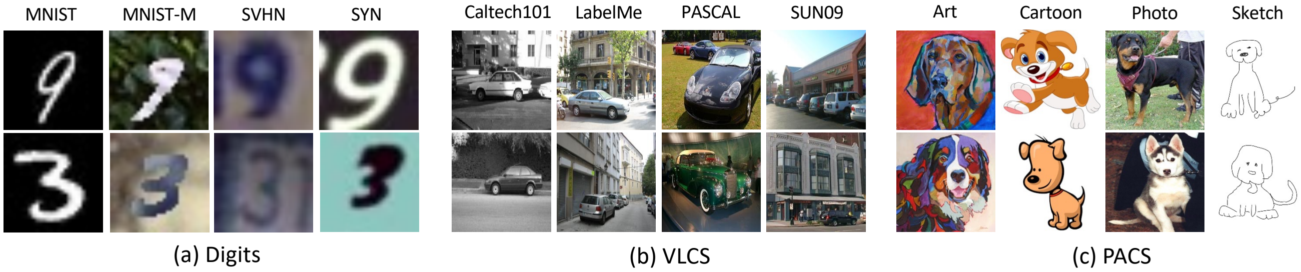
Settings DG has typically been studied under two different settings, namely *multi-source DG* and *single-source DG*. The majority of research has been dedicated to the multi-source setting, which assumes multiple distinct but relevant domains are available (i.e. $K > 1$). As stated in [19], the original motivation for studying DG is to leverage multi-source data to learn representations that are invariant to different marginal distributions. This makes sense because without having access to the target data, it is challenging for a source-learned model to generalize well. As such, using multiple domains allows a model to discover stable patterns across source domains, which generalize better to unseen domains.

In contrast, the single-source setting assumes training data are homogeneous, i.e. they are sampled from a single domain ($K = 1$). This problem is closely related to the topic of OOD robustness [8], [47], [48], which investigates model robustness under image corruptions. Essentially, single-source DG methods do not require domain labels for learning and thus they are applicable to multi-source scenarios as well. In fact, most existing methods able to solve single-source DG do not distinguish themselves as a single- or a multi-source approach, but rather a more generic solution to OOD generalization, with experiments covering both single- and multi-source datasets [49], [50], [51], [52].

[1] In this paper, we use P_{XY} and $P(X, Y)$ interchangeably.

TABLE 1
Comparison between domain generalization and its related topics. K: number of source domains/tasks. $P_{XY}^{S/T}$: source/target joint distribution. $\mathcal{Y}_{S/T}$: source/target label space. $P_X^{\mathcal{T}}$: target marginal.

	K		$P^{\mathcal{S}}_{XY}$ vs. $P^{\mathcal{T}}_{XY}$		$\mathcal{Y}_S$ vs. $\mathcal{Y}_T$		Access to $P_X^{\mathcal{T}}$?
	$=1$	>1	$=$	$\neq$	$=$	$\neq$	
Supervised Learning	✓		✓		✓		
Multi-Task Learning		✓	✓		✓		
Transfer Learning	✓	✓		✓		✓	✓
Zero-Shot Learning	✓			✓		✓	
Domain Adaptation	✓	✓		✓	✓	✓	✓
Domain Generalization	✓	✓		✓	✓	✓	



(a) Digits (b) VLCS (c) PACS

Fig. 1. Example images from three domain generalization benchmarks manifesting different types of domain shift. In (a), the domain shift mainly corresponds to changes in font style, color and background. In (b), dataset-specific biases are clear, which are caused by changes in environment/scene and viewpoint. In (c), image style changes are the main reason for domain shift.

2.3 Evaluation and Datasets

In this section, we first introduce how a model's generalization performance is evaluated in DG, and then summarize the most commonly used datasets. The evaluation protocol is straightforward, which follows the *leave-one-domain-out* idea [33]: given a dataset containing at least two distinct domains, one or multiple domains are used for model training, depending on whether the setting is single- or multi-source DG; then the model is evaluated on the remaining unseen domain(s).

In the literature, many datasets have been proposed for benchmarking DG approaches. Notably, due to DA's relatedness to DG, several DG datasets are derived from DA datasets, such as OfficeHome [59] and DomainNet [60], both containing multiple domains. Besides the specifically designed multi-domain datasets, some DG datasets are the combination of several related datasets addressing the same problem, e.g., VLCS [56] is a combination of Caltech101 [39], LabelMe [40], PASCAL [57], and SUN09 [58], all targeting the object recognition problem but encoding dataset-specific biases [38].

A few approaches [31], [34], [37] have also been evaluated on datasets where label space changes between source and target (termed heterogeneous DG [34]), such as Visual Decathlon [63] for image classification and image matching datasets like those designed for person re-identification [78]. For image classification, evaluation on the heterogeneous setting requires learning a new linear classifier using target data. For image matching, the source-learned representation is directly used for comparison.

Table 2 summarizes existing DG datasets, each with basic statistics and a short description. Below we provide more detailed discussions and categorize them based on different tasks/application areas.

Handwritten Digit Recognition Several handwritten digit datasets have been widely used as DG testbed, e.g., MNIST [54], MNIST-M [13], SVHN [55], and SYN [13]. MNIST contains images of handwritten digits. Extended from MNIST, MNIST-M mixes the images with random color patches. SVHN comprises images of street view house numbers. SYN is a synthetic dataset containing digit images with variations in font, background and stroke color. Example images of these four datasets can be found in Figure 1(a). Digits-DG [31] combines the four aforementioned datasets, aiming to evaluate a model's robustness to variations in font style, color and background. Furthermore, rotation has also been studied as a domain shift variable, e.g., in Rotated MNIST [53] a rotation degree is regarded as a domain.

Object Recognition is generally the most studied task in DG, as can be seen in Table 2. Below we summarize those datasets based on their domain shift types. 1) The domain shift in VLCS [56] and Office-31 [10] is mainly caused by changes in environment and viewpoint. For example, in VLCS the captured scenes vary from urban to rural, and the viewpoints are biased toward either side-views or non-canonical views (see Figure 1(b)).

2) Several datasets have been focused on image style changes, including OfficeHome [59], PACS [33], DomainNet [60], and ImageNet-Sketch [51]. Example images manifesting the image style changes are shown in Figure 1(c). Based on studies in these datasets [37], [61], [79], [80], it is generally acknowledged that when the source image

TABLE 2
Commonly used domain generalization datasets.

Benchmark	# samples	# domains	Task	Description
Rotated MNIST [53]	70,000	6	Handwritten digit recognition	Rotation degree $\in \{0, 15, 30, 45, 60, 75\}$
Digits-DG [31]	24,000	4	Handwritten digit recognition	Combination of MNIST [54], MNIST-M [13], SVHN [55] and SYN [13]
VLCS [56]	10,729	4	Object recognition	Combination of Caltech101 [39], LabelMe [40], PASCAL [57], and SUN09 [58]
Office-31 [10]	4,652	3	Object recognition	Domain ∈ {amazon, webcam, dslr}
OfficeHome [59]	15,588	4	Object recognition	Domain ∈ {art, clipart, product, real}
PACS [33]	9,991	4	Object recognition	Domain ∈ {photo, art, cartoon, sketch}
DomainNet [60]	586,575	6	Object recognition	Domain ∈ {clipart, infograph, painting, quickdraw, real, sketch}
miniDomainNet [61]	140,006	4	Object recognition	A smaller and less noisy version of DomainNet; domain ∈ {clipart, painting, real, sketch}
ImageNet-Sketch [51]	50,000	2	Object recognition	Domain shift between real and sketch images
VisDA-17 [62]	280,157	3	Object recognition	Synthetic-to-real generalization
CIFAR-10-C [8]	60,000	-	Object recognition	The test data are damaged by 15 corruptions (each with 5 intensity levels) drawn from 4 categories (noise, blur, weather, and digital)
CIFAR-100-C [8]	60,000	-	Object recognition	
ImageNet-C [8]	≈1.3M	-	Object recognition	
Visual Decathlon [63]	1,659,142	10	Object/action/handwritten digit recognition	Combination of 10 datasets
IXMAS [64]	1,650	5	Action recognition	5 camera views; 10 subjects; 5 actions (see [27])
UCF-HMDB [65], [66]	3,809	2	Action recognition	12 overlapping actions (see [67])
SYNTHIA [68]	2,700	15	Semantic segmentation	4 locations; 5 weather conditions (see [43])
GTA5-Cityscapes [69], [70]	29,966	2	Semantic segmentation	Synthetic-to-real generalization
TerraInc [71]	24,788	4	Animal classification	Captured at different geographical locations
Market-Duke [72], [73]	69,079	2	Person re-identification	Cross-dataset re-ID; heterogeneous DG
Face [36]	>5M	9	Face recognition	Combination of 9 face datasets
COMI [74], [75], [76], [77]	≈8,500	4	Face anti-spoofing	Combination of 4 face anti-spoofing datasets

style is close to the target image style (both sharing the same visual cues), the performance would be higher (e.g., photo→painting, both relying on colors and textures); otherwise, if the source image style is drastically different from the target image style, the performance would be poor (e.g., photo→quickdraw, with the latter strongly relying on shape information while requiring no color information at all). This observation also applies to unsupervised domain adaptation. For instance, the performance on the quickdraw domain of DomainNet is usually the lowest among all target domains [61], [81], [82].

3) A couple of recent DG studies [83], [84] have investigated, from a transfer learning perspective, how to preserve the knowledge learned via large-scale pre-training when training on abundant labeled synthetic data for synthetic-to-real applications. The experiments were carried out on VisDA-17 [62]. This is an important yet under-studied topic in DG: when only given sufficient synthetic data, how can we avoid over-fitting in synthetic images by leveraging the initialization weights learned on real images? Such a setting is particularly useful to problems where manual labels are difficult/expensive to obtain.

4) Synthetic image corruptions like Gaussian noise and motion blur have also been used to simulate domain shift by Hendrycks and Dietterich [8]. In their proposed datasets, i.e. CIFAR-10-C, CIFAR-100-C and ImageNet-C, a model is learned using the original images but tested on the corrupted images. This research is largely motivated by adversarial attacks [85], and aims to evaluate model robustness under common image perturbations for safety applications.

5) Lastly, a hybrid dataset initially proposed for multi-domain/task learning, i.e. Visual Decathlon [63], has also been employed, for evaluating heterogeneous DG [34], [37]. However, due to both the changes in label space and the use of target data for training SVM classifiers, this setup overlaps with transfer learning [86].

Action Recognition Learning generalizable models is critical for action recognition. This is because the test data typically contain actions performed by new subjects in new environments. IXMAS [64] has been widely used as a cross-view action recognition benchmark [27], [37], which contains action videos collected from five different views. The common practice is to use four views for training and the remaining view for test. In addition to view changes, different subjects and environments might also cause failure. Intuitively, different persons can perform the same action in (dramatically) different ways, so it is common that a model might not be able to recognize actions performed by new subjects not seen during training. Also, we cannot expect a model trained using indoor data to work well in outdoors. In the future, it would be interesting to investigate more domain shift variables, such as subject and environment.

Semantic Segmentation is important for autonomous driving. Though this task has been greatly advanced by deep neural networks, the performance is still far from being satisfactory when deploying trained deep models in novel scenarios, such as new cities and unseen weather conditions [87]. Since it is generally impractical to collect data that cover all possible scenarios, DG is pivotal in facilitating large-scale deployment of semantic segmentation systems. To evaluate the DG performance in cross-city scenarios, one can use the SYNTHIA dataset [68], which contains synthetic images of different locations in different weather conditions. As a dense prediction task, collecting annotations for training semantic segmentation models is very costly. To address this issue, one can study how to generalize a model from synthetic data like GTA5 [69] to real image data like Cityscapes [70], without using any real images.

Person Re-Identification is an important surveillance and security application. It is essentially an instance retrieval task, with a goal to match people across disjoint camera views. Most existing person re-ID methods [78], [88], [89], [90], [91] have been focused on the same-dataset setting, i.e. training and evaluation are performed under the same set of camera views, with performance almost reaching saturation. Recently, cross-dataset re-ID [42], [92], [93] has gained significant interests. The objective is to generalize a re-ID model learned from source camera views to target camera views installed in a different environment. In particular, images captured by different camera views across different environments can exhibit drastically different characteristics, reflected in image resolution, viewpoint, lighting condition, background, etc., thus making cross-dataset re-ID a challenging problem. Moreover, unlike image classification tasks where the label space remains the same between source and target, person re-ID mainly targets the heterogeneous setting since training and test identities are completely different, which further exacerbate the DG problem. The existing DG works [20], [31], [32] have investigated cross-dataset re-ID between Market1501 [78] and DukeMTMC-reID [73]. A few works [92], [93] have attempted the multiple-to-one setting, e.g., using Market1501, CUHK03 [94] and MSMT17 [95] for training and DukeMTMC-reID for testing.

Face Recognition has seen significant advances in recent years, mainly attributed to deep learning technologies [96], [97], [98]. However, studies have shown that deep face recognition models trained even on large-scale datasets, such as MS-Celeb-1M [99], still suffer substantial performance drops on unseen datasets with OOD data. For example, the face images in a new dataset might have a lower resolution [100], [101], [102], large variations in illumination/occlusion/head pose [103], [104], [105], or drastically different viewpoints [106]. This has motivated research on learning universal face representations [36].

Face Anti-Spoofing aims to prevent face recognition systems from being attacked by using fake faces [107], such as printed photos, videos and 3D masks. Conventional face anti-spoofing methods do not take into account distribution shift, which is often caused by different attack types (e.g., photo vs. video) or different display devices. Therefore, their performance usually plunges when encountered with novel attacks [45]. To make face anti-spoofing systems more robust and secure, researchers have been working on designing effective DG algorithms [45], [108]. Currently, there are no specifically designed DG benchmarks for face anti-spoofing. A commonly used setting is to combine several face anti-spoofing datasets for training and test the model on an unseen dataset, e.g., using CASIA-MFSD [74], Oulu-NPU [75] and MSU-MFSD [76] as the sources and Idiap Replay-Attack [77] as the target.

Very recently, Koh et al. [109] introduced the WILDS benchmark, which consists of eight datasets with diverse applications (e.g., animal recognition, cancer detection, molecule classification, satellite imaging, etc.) and contains distribution shift originated in the wild (shift in cameras, hospitals, geographical regions, users, etc.). This could also be of interest to readers.

2.4 Related Topics

In this section, we discuss the relations between DG and its related topics, and clarify their differences. See Table 1 for an overview.

Supervised Learning generally aims to learn an input-output mapping by minimizing the following risk: $\mathbb{E}_{(x,y)\sim\hat{P}_{XY}}\ell(f(x),y)$, where $\hat{P}_{XY}$ denotes the empirical distribution rather than the real data distribution P_{XY}, which is inaccessible. The hope is that once the loss is minimized, the learned model can work well on data generated from P_{XY}, which heavily relies on the i.i.d. assumption. The crucial difference between SL and DG is that in the latter training and test data are drawn from different distributions, thus violating the i.i.d. assumption. DG is arguably a more practical setting in real-world applications [38].

Multi-Task Learning (MTL) The goal of MTL is to simultaneously learn multiple related tasks ($K > 1$) using a single model [110], [111], [112], [113], [114]. As shown in Table 1, MTL aims to make a model perform well on the same set of tasks that the model was trained on ($\mathcal{Y}_S = \mathcal{Y}_T$), whereas DG aims to generalize a model to unseen data distributions ($P_{XY}^S \neq P_{XY}^T$). Though being different in terms of the problem setup, the MTL paradigm has been exploited in some DG methods, notably for those based on self-supervised learning [49], [81], [115]. Intuitively, MTL benefits from the effect of regularization brought by parameter sharing [110], which may in part explain why the MTL paradigm works for DG.

Transfer Learning (TL) aims to transfer the knowledge learned from one (or multiple) problem/domain/task to a different but related one [86]. A well-known TL example in contemporary deep learning is fine-tuning: first pretrain deep neural networks on large-scale datasets, such as ImageNet [9] for vision models or BooksCorpus [116] for language models; then fine-tune them on downstream tasks [117]. Given that pre-trained deep features are highly transferable, as shown in several studies [118], [119], a couple of recent DG works [83], [84] have researched how to preserve the transferable features learned via large-scale pre-training when learning new knowledge from source synthetic data for synthetic-to-real applications. As shown in Table 1, a key difference between TL and DG lies in whether the target data are used. In TL, the target data are required for model fine-tuning for new downstream tasks, whereas in DG we assume to have no access to the target data, thus focusing more on model generalization. Nonetheless, TL and DG share some similarities: the target distribution in both TL and DG is different from the source distribution; in terms of label space, TL mainly concerns disjoint label space, whereas DG considers both cases, i.e. same label space for homogeneous DG and disjoint label space for heterogeneous DG.

Zero-Shot Learning (ZSL) is related to DG in the sense that the goal in both problems is to deal with unseen distributions. Differently, distribution shift in ZSL is mainly caused by label space changes [120], i.e. $P_Y^{\mathcal{T}} \neq P_Y^{\mathcal{S}}$, since the task is to recognize new classes, except for generalized ZSL [121] which considers both new and old classes at test time; while in DG, domain shift mostly results from covariate shift [17], i.e. only the marginal distribution changes ($P_X^{\mathcal{T}} \neq P_X^{\mathcal{S}}$).[2] To recognize unseen classes in ZSL, a common practice is to learn a mapping between the input image space and the attribute space [123] since the label space is disjoint between training and test data. Interestingly, attributes have also been exploited in DG for learning domain-generalizable representations [124].

Domain Adaptation (DA) is the closest topic to DG and has been extensively researched in the literature [11], [12], [13], [14], [15], [87], [125], [126], [127], [128]. Both DA and DG aim to tackle the domain shift problem ($P_{XY}^{\mathcal{S}} \neq P_{XY}^{\mathcal{T}}$) encountered in new test environments. Differently, DA assumes the availability of sparsely labeled [129] or unlabeled [125] target data for model adaptation, hence having access to the marginal $P_X^{\mathcal{T}}$. Though there exist different variants of DA where some methods do not explicitly use target data during training, such as zero-shot DA [130] that exploits task-irrelevant but target domain-relevant data (equivalent to accessing the marginal), their main idea remains unchanged, i.e. to leverage additional data that expose information related to the target domain. As shown in Table 1, studies in DA share some commonalities with DG, such as single- [125] and multi-source [60] DA, and heterogeneous DA [15], [131], [132], [133].

3 Methodologies: A Survey

Numerous domain generalization (DG) methods have been proposed in the past ten years, and the majority of them are designed for multi-source DG, despite some methods not explicitly requiring domain labels for learning and thus suitable for single-source DG as well. In this section, we categorize existing DG methods into seven groups based on their methodologies and motivations behind their design. Within each group, we further discuss different variants and indicate whether domain labels are required for learning to differentiate their uses—those requiring domain labels can only be applied to multi-source DG while those not requiring domain labels are applicable to both single- and multi-source DG. See Table 3 for an overview.

3.1 Domain Alignment

Most existing DG approaches belong to the category of domain alignment [17], [27], [28], [45], [139], [142], where the central idea is to minimize the difference between source domains for learning domain-invariant representations. The motivation is straightforward: features that are invariant to the source domain shift should also be robust to any unseen target domain shift. Domain alignment has been applied in many DG applications, e.g., object recognition [28], [53], action recognition [27], face anti-spoofing [45], [108], and medical imaging analysis [143], [149]. Domain labels are required for domain alignment methods.

To measure the distance between distributions and thereby achieve alignment, there are a wide variety of statistical distance metrics for us to borrow, such as the simple ℓ_2 distance, f-divergences, or the more sophisticated Wasserstein distance [180]. However, designing an effective domain alignment method is a non-trivial task because one needs to consider *what to align* and *how to align*. In the following sections, we analyze the existing alignment-based DG methods from these two aspects.

3.1.1 What to Align

Recall that a domain is modeled by a joint distribution $P(X, Y)$ (see § 2.2 for the background), we can decompose it into

$$\begin{aligned} P(X, Y) &= P(Y|X)P(X), && (1) \\ &= P(X|Y)P(Y). && (2) \end{aligned}$$

A common assumption in DG is that distribution shift only occurs in the marginal $P(X)$ while the posterior $P(Y|X)$ remains relatively stable [17] (see Eq. (1)). Therefore, numerous domain alignment methods have been focused on aligning the marginal distributions of source domains [17], [27], [134], [135].

From a causal learning perspective [181], it is valid to align $P(X)$ only when X is the cause of Y. In this case, $P(Y|X)$ is not coupled with $P(X)$ and thus remains stable when $P(X)$ varies. However, it is also possible that Y is the cause of X, and as a result, shift in $P(X)$ will also affect $P(Y|X)$. Therefore, some domain alignment methods [28], [136], [138] proposed to instead align the class-conditional $P(X|Y)$, assuming that $P(Y)$ does not change (see Eq. (2)). For example, Li et al. [136] learned a feature transformation by minimizing for all classes the variance of class-conditional distributions across source domains. To allow $P(Y)$ to change along with $P(X|Y)$, i.e. heterogeneous DG, Hu et al. [138] relaxed the assumption made in [136] by removing the minimization constraint on marginal distributions and proposed several discrepancy measures to learn generalizable features.

[2] It is worth mentioning that a recent ZSL work [122] has studied ZSL+DG, i.e. distribution shift occurs in both P_Y and P_X, which is analogous to heterogeneous DG.

TABLE 3
Categorization of domain generalization methods. Note that methods requiring domain labels can only be applied to multi-source DG while those not requiring domain labels are applicable to both multi- and single-source DG.

Category	Domain labels	Methods
Domain Alignment (§ 3.1)		
- Minimizing Moments	✓	[17], [134], [135], [136], [137], [138]
- Minimizing Contrastive Loss	✓	[139], [140], [141]
- Minimizing the KL Divergence	✓	[142], [143]
- Minimizing Maximum Mean Discrepancy	✓	[27]
- Domain-Adversarial Learning	✓	[28], [45], [108], [144], [145], [146], [147], [148], [149], [150]
Meta-Learning (§ 3.2)	✓	[29], [30], [34], [37], [92], [93], [151], [152], [153], [154], [155]
Data Augmentation (§ 3.3)		
- Image Transformations	✗	[35], [36], [156], [157], [158]
- Task-Adversarial Gradients	✗	[43], [159], [160]
- Domain-Adversarial Gradients	✓	[161]
- Random Augmentation Networks	✗	[162]
- Off-the-Shelf Style Transfer Models	✓	[18], [26], [79], [163]
- Learnable Augmentation Networks	✓	[31], [32], [164]
- Feature-Based Augmentation	✓	[20], [25], [122]
Ensemble Learning (§ 3.4)		
- Exemplar-SVMs	✗	[165], [166], [167]
- Domain-Specific Neural Networks	✓	[61], [80], [168], [169], [170]
- Domain-Specific Batch Normalization	✓	[171], [172], [173], [174]
- Weight Averaging	✗	[175]
Self-Supervised Learning (§ 3.5)		
- Single Pretext Task	✗	[49], [53], [81], [176]
- Multiple Pretext Tasks	✗	[50], [115]
Learning Disentangled Representations (§ 3.6)		
- Decomposition	✓	[33], [41], [177], [178]
- Generative Modeling	✓	[46], [179]
Regularization Strategies (§ 3.7)	✗	[51], [52]

Since the posterior $P(Y|X)$ is what we need at test time, Wang et al. [142] introduced hypothesis invariant representations, which are obtained by directly aligning the posteriors within each class regardless of domains via the Kullback–Leibler (KL) divergence.

3.1.2 How to Align

Having discussed what to align in the previous section, here we turn to the exact techniques used in the DG literature for distribution alignment.

Minimizing Moments Moments are parameters used to measure a distribution, such as mean (1st-order moment) and variance (2nd-order moment) calculated over a population. Therefore, to achieve invariance between source domains, one can learn a mapping function (e.g., a simple projection matrix [135] or a complex non-linear function modeled by deep neural networks [137]) with an objective of minimizing the moments of the transformed features between source domains, in terms of variance [17], [134] or both mean and variance [135], [136], [137], [138].

Minimizing Contrastive Loss is another option for reducing distribution mismatch [139], [140], [141], which takes into account the semantic labels. There are two key design principles. The first is about how to construct the anchor group, the positive group (same class as the anchor but from different domains) and the negative group (different class than the anchor). The second is about the formulation of the distance function (e.g., using ℓ_2 [139] or softmax [141]). The objective is to pull together the anchor and the positive groups, while push away the anchor and the negative groups.

Minimizing the KL Divergence As a commonly used distribution divergence measure, the KL divergence has also been employed for domain alignment [142], [143]. In [142], domain-agnostic posteriors within each class are aligned via the KL divergence. In [143], the KL divergence is used to force all source domain features to be aligned with a Gaussian distribution.

Minimizing Maximum Mean Discrepancy (MMD) The MMD distance [182] measures the divergence between two probability distributions by first mapping instances to a reproducing kernel Hilbert space (RKHS) and then computing the distance based on their mean. Using the autoencoder architecture, Li et al. [27] minimized the MMD distance between source domain distributions on the hidden-layer features, and meanwhile, forced the feature distributions to be similar to a prior distribution via adversarial learning [183].

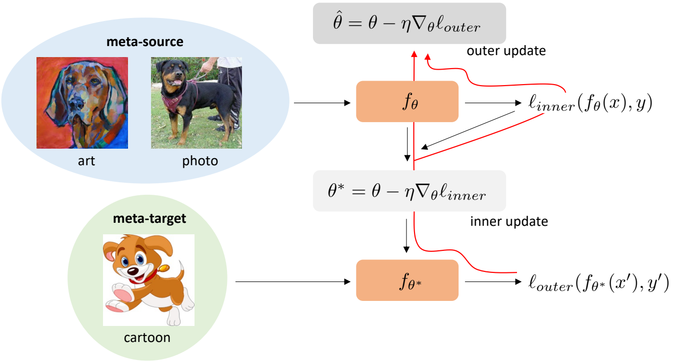
Domain-Adversarial Learning Different from explicit distance measures like the MMD, adversarial learning [183] formulates the distribution minimization problem through a minimax two-player game. Initially proposed by Goodfellow et al. [183], adversarial learning was used to train a generative model, which takes as input random noises and generates photorealistic images. This is achieved by learning a discriminator to distinguish between real and the generated fake images (i.e. minimizing the binary classification loss), while encouraging the generator to fool the discriminator (i.e. maximizing the binary classification loss). In particular, the authors in [183] theoretically justified that generative adversarial learning is equivalent to minimizing the Jensen-Shannon divergence between the real distribution and the generated distribution. Therefore, it is natural to use adversarial learning for distribution alignment, which has already been extensively studied in the domain adaptation area for aligning the source-target distributions [13], [184], [185], [186].

In DG, adversarial learning is performed between source domains to learn source domain-agnostic features that are expected to work in novel domains [28], [45], [144], [145], [146]. Simply speaking, the learning objective is to make features confuse a domain discriminator, which can be implemented as a multi-class domain discriminator [147], [148], [149], or a binary domain discriminator in a per-domain basis [28], [45]. Typically, the learning steps alternate between the feature generator and the domain discriminator(s) [28]. However, one can simplify the process to achieve single-step update by using the gradient-reversal layer [13] to flip the sign of the gradients back-propagated from the domain discriminator(s) [108].

To enhance domain alignment, researchers have also combined domain-adversarial learning with explicit distance measures like moments minimization [144], or with some regularization constraints such as entropy [150].

Multi-Task Learning has also been explored for domain alignment [53], [176]. Different from directly minimizing the distribution divergence, MTL facilitates the learning of generic features by parameter sharing [110]. This is easy to understand: in order to simultaneously deal with different tasks the features have to be generic enough. In [53], the authors proposed a denoising autoencoder architecture (later employed in [176]) where the encoder is shared but the decoder is split into domain-specific branches, each connected to a reconstruction task. The model was trained with two objectives, one being self-domain reconstruction while the other being cross-domain reconstruction, which aim to force the hidden representations to be as generic as possible.

Domain alignment is still a popular research direction in DG. This idea has also been extensively studied in the domain adaptation (DA) literature [13], [14], [126], [187], [188], but with a rigorous theoretical support [3]. In particular, the DA theory introduced in [3] suggested that minimizing the distribution divergence between source and target has a huge impact on lowering the upper-bound of the target error. However, in DG we cannot access the target data and therefore, the alignment is performed only among source domains. This inevitably raises a question of whether a representation learned to be invariant to source domain shift is guaranteed to generalize to an unseen domain shift in the target data. To solve this concern, one can focus on developing novel theories to explain how alignment in source domains improves generalization in unseen domains.



meta-source
art
photo
meta-target
cartoon
$\hat{\theta} = \theta - \eta\nabla_\theta \ell_{outer}$
outer update
f_θ
$\ell_{inner}(f_\theta(x), y)$
$\theta^* = \theta - \eta\nabla_\theta \ell_{inner}$
inner update
f_{θ^*}
$\ell_{outer}(f_{\theta^*}(x'), y')$

Fig. 2. A commonly used meta-learning paradigm [29] in domain generalization. The source domains (i.e. art, photo and cartoon from PACS [33]) are divided into disjoint meta-source and meta-target domains. The outer learning, which simulates domain shift using the meta-target data, back-propagates the gradients all the way back to the base parameters such that the model learned by the inner algorithm with the meta-source data improves the outer objective. The red arrows in this figure denote the gradient flow through the second-order differentiation.

3.2 Meta-Learning

Meta-learning has been a fast growing area with applications to many machine learning and computer vision problems [29], [37], [92], [153], [189]. Also known as learning-to-learning, meta-learning aims to learn from episodes sampled from related tasks to benefit future learning (see [190] for a comprehensive survey on meta-learning). The meta-learning paper most related to DG is MAML [189], which divides training data into meta-train and meta-test sets, and trains a model using the meta-train set in such a way to improve the performance on the meta-test set. The MAML-style training usually involves a second-order differentiation through the update of the base model, thus posing issues on efficiency and memory consumption for large neural network models [190]. In [189], MAML was used for parameter initialization, i.e. to learn an initialization state that is only a few gradient steps away from the solution to the target task.

The motivation behind applying meta-learning to DG is to expose a model to domain shift during training with a hope that the model can better deal with domain shift in unseen domains. Existing meta-learning DG methods can only be applied to multi-source DG where domain labels are provided.

There are two components that need to be carefully designed, namely *episodes* and *meta-representation*. Specifically, episodes construction concerns how each episode should be constructed using available samples, while meta-representation answers the question of what to meta-learn.

Episodes Construction Most existing meta-learning-based DG methods [30], [34], [92], [93], [151], [152], [153], [154], [155] followed the learning paradigm proposed in [29]—which is the first method applying meta-learning to DG. Specifically, source domains are divided into non-overlapping *meta-source* and *meta-target* domains to simulate domain shift. The learning objective is to update a model using the meta-source domain(s) in such a way that the test error on the meta-target domain can be reduced, which is often achieved by bi-level optimization. See Figure 2 for a graphical representation.

Meta-Representation is a term defined in [190] to represent the model parameters that are meta-learned. Most deep learning methods meta-learned the entire neural network models [29], [151], [154]. Balaji et al. [30] instead proposed to meta-learn the regularization parameters. In [152], a stochastic neural network is meta-learned to handle uncertainty. In [153], an MRI segmentation model is meta-learned, along with two shape-aware losses to ensure compactness and smoothness in the segmentation results. Batch normalization layers are meta-learned in [92], [93], [155] to cope with the training-test discrepancy in CNN feature statistics.

Overall, meta-learning is a promising direction to work on given its effectiveness in not only DG but also a wide range of applications like few-shot classification [189], object detection [191] and image generation [192]. However, meta-learning in DG still suffers the same issue with that in domain alignment—a representation is only learned to be robust under source domain shift (simulated by meta-source and meta-target domains). Such an issue could be aggravated if the source domains are limited in terms of diversity. As observed from recent work [31], [175], both meta-learning and domain alignment methods are underperformed by methods based on directly augmenting the source training data—a topic that will be visited later. One might alleviate the generalization issue in meta-learning, as well as in domain alignment, by combining them with data augmentation. Moreover, advances may also be achieved by designing novel meta-learning algorithms in terms of meta-representation, meta-optimizer, and/or meta-objective.[3]

3.3 Data Augmentation

Data augmentation has been a common practice to regularize the training of machine learning models to avoid overfitting and improve generalization [193], which is particularly important for over-parameterized deep neural networks. The basic idea in data augmentation is to augment the original (x, y) pairs with new $(A(x), y)$ pairs where $A(\cdot)$ denotes a transformation, which is typically label-preserving. Not surprisingly, given the advantages of data augmentation, it has been extensively studied in DG where $A(\cdot)$ is usually seen as a way of simulating domain shift and the design of $A(\cdot)$ is key to performance.

Based on how $A(\cdot)$ is formulated, data augmentation methods generally fall into four groups. See Figure 3 for an overview. Below we provide more detailed reviews, with a more fine-grained categorization where adversarial gradients are divided into task-adversarial gradients and domain-adversarial gradients; and model-based augmentation is further split into three sub-groups: random augmentation networks, off-the-shelf style transfer models, and learnable augmentation networks.

Image Transformations This type of approach exploits traditional image transformations, such as random flip, rotation and color augmentation. Figure 4 visualizes some effects of transformations. Though using image transformations does not require domain labels during learning, the selection of transformations is usually problem-specific. For example, for object recognition where image style changes are the main domain shift, one can choose transformations that are more related to color intensity changes, such as `brightness`, `contrast` and `solarize` in Figure 4. To avoid manual picking, one can design a searching mechanism to search for the optimal set of transformations that best fit the target problem. An example is [35] where the authors proposed an evolution-based searching algorithm and used a worst-case formulation to make the transformed images deviate as much as possible from the original image distribution. One can also select transformations according to the specific downstream task. For instance, [36] addressed the universal feature learning problem in face recognition by synthesizing meaningful variations such as lowering image resolution, adding occlusions and changing head poses.

Traditional image transformations have been shown very effective in dealing with domain shift in medical images [156], [157], [158]. This makes sense because image transformations can well simulate changes in color and geometry caused by device-related domain shift, such as using different types of scanners in different medical centers. However, image transformations can be limited in some applications as they might cause label shift, such as digit recognition or optical character recognition where the horizontal/vertical flip operation is infeasible. Therefore, transformations should be carefully chosen to not conflict with the downstream task.

Task-Adversarial Gradients Inspired by adversarial attacks [85], [194], several data augmentation methods are based on using adversarial gradients obtained from the task classifier to perturb the input images [43], [159], [160]. In doing so, the original data distribution is expanded, allowing the model to learn more generalizable features. Though this type of approach is often developed for tackling single-source DG, the idea can also be directly applied to multi-source scenarios.

Domain-Adversarial Gradients When it comes to multi-source DG where domain labels are provided, one can exploit domain-adversarial gradients to synthesize domain-agnostic images. For instance, [161] trained a domain classifier and used its adversarial gradients to perturb the input images. Intuitively, by learning with domain-agnostic images, the task model is allowed to learn more domain-invariant patterns.

Since adversarial gradients-based perturbation is purposefully designed to be visually imperceptible [85], methods based on adversarial gradients are often criticized for not being able to simulate real-world domain shift, which is much more complicated than salt-and-pepper noise [32]. Furthermore, the computational cost is often doubled in these methods because the forward and backward passes need to be computed twice, which could pose serious efficiency issues for large neural networks. Below we discuss model-based methods that formulate the transformation $A(\cdot)$ using neural networks and can produce more diverse visual effects.

Random Augmentation Networks RandConv [162] is based on the idea of using randomly initialized, single-layer convolutioinal neural network to transform the input images to "novel domains". Since the weights are randomly sampled from a Gaussian distribution at each iteration and

[3] These terms are defined in [190].

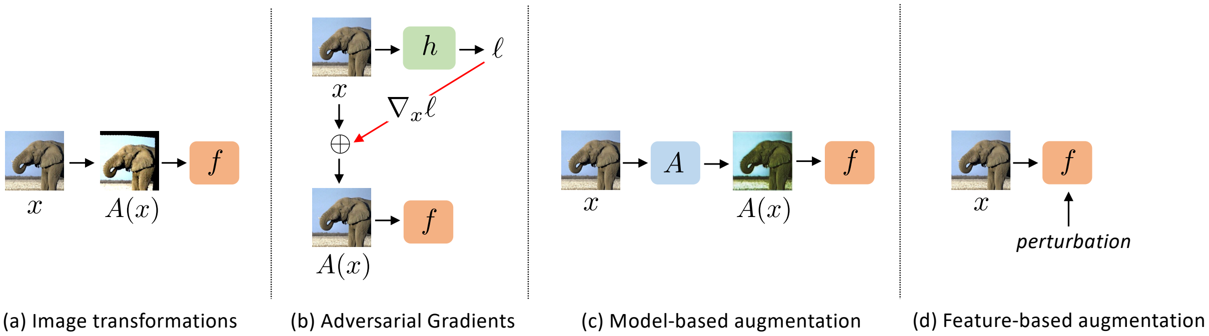

x
$A(x)$
f
h
ℓ
$\nabla_x \ell$
$\oplus$
A
perturbation
(a) Image transformations
(b) Adversarial Gradients
(c) Model-based augmentation
(d) Feature-based augmentation

Fig. 3. Based on the formulation of the transformation $A(\cdot)$, existing data augmentation methods can be categorized into four groups. **a)** The first group enhances the generalization of the classifier f by applying hand-engineered image transformations like random crop or color augmentation to simulating domain shift. **b)** The second group is based on adversarial gradients obtained from either a category classifier ($h = f$) or a domain classifier. **c)** The third group models $A(\cdot)$ using neural networks, such as random CNNs [162], an off-the-shelf style transfer model [26], or a learnable image generator [31]. **d)** The final group injects perturbation into intermediate features in the task model.

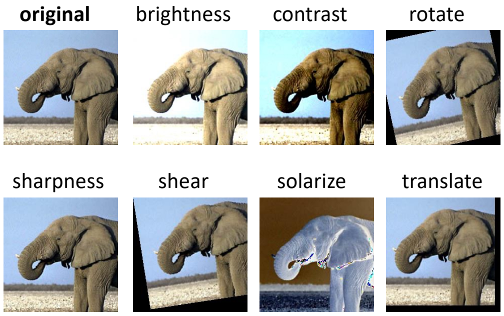


Fig. 4. Common image transformations used as data augmentation in domain generalization [35], [156], [157], [158].

no learning is performed, the transformed images mainly contain random color distortions, which do not contain meaningful variations and are best to be mixed with the original images before passing to the task network.

Off-the-Shelf Style Transfer Models Taking advantage of the advances in style transfer [195], several DG methods [26], [79], [163] use off-the-shelf style transfer models like AdaIN [195] to represent $A(\cdot)$, which essentially maps images from one source domain to another for data augmentation. Instead of transferring image styles between source domains, one can exploit external styles to further diversify the source training data [18]. Though these methods do not need to train the style transfer component, they still need domain labels for domain translation.

Learnable Augmentation Networks This group of methods aims to learn augmentation neural networks to synthesize new domains [31], [32], [164]. In [32], [164], domain-agnostic images are generated by maximizing the domain classification loss with respect to the image generator. In [31], pseudo-novel domains are synthesized by maximizing for each source domain the domain distance measured by optimal transport [180] between the original and synthetic images.

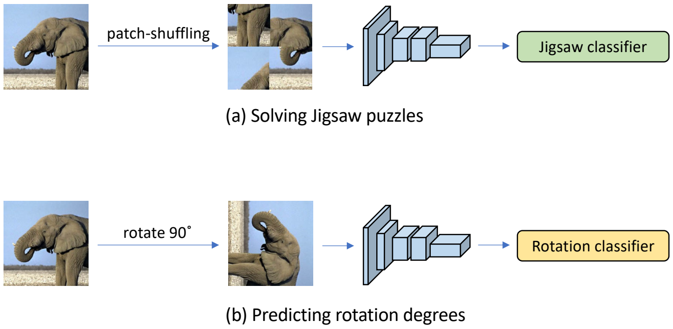
Feature-Based Augmentation Though the above learnable augmentation models have shown promising results, their efficiency is a main concern as they need to train heavy image-to-image translation models. Another line of research focuses on feature-level augmentation [20], [25], [122]. Motivated by the observation that image styles are captured in CNN feature statistics, MixStyle [20], [25] achieves style augmentation by mixing CNN feature statistics between instances of different domains. In [122], Mixup [196] is applied to mixing instances of different domains in both pixel and feature space.

3.4 Ensemble Learning

As an extensively studied topic in machine learning research, ensemble learning [197] typically learns multiple copies of the same model with different initialization weights or using different splits of training data, and uses their ensemble for prediction. Such a simple technique has been shown very effective in boosting the performance of a single model across a wide range of applications [5], [6], [198]. In DG, ensemble learning has also been explored, with examples including using traditional ensemble methods like exemplar-SVMs [165] and training domain-specific models [61].

Exemplar-SVMs are a collection of SVM classifiers, each learned using one positive instance and all negative instances [199]. As the ensemble of such exemplar SVMs have shown excellent generalization performance on the object detection task in [199], Xu et al. [165] have extended exemplar-SVMs to DG. In particular, given a test sample the top-K exemplar classifiers that give the highest prediction scores (hence more confident) are selected for ensemble prediction. Such an idea of learning exemplar classifiers was also investigated in [166], [167] for DG.

Domain-Specific Neural Networks Since CNNs excel at discriminative feature learning [6], it is natural to replace hand-engineered SVM classifiers with CNN-based models for ensemble learning. A common practice is to learn domain-specific neural networks, each specializing in a source domain [61], [168]. Rather than learning an independent CNN for each source domain [168], it is more efficient, and makes more sense as well, to share between source domains some shallow layers [61], which capture generic features [118]. Another question is how to compute the prediction. One can simply use the ensemble prediction averaged over all individuals with equal weights (e.g., [61],



(a) Solving Jigsaw puzzles

(b) Predicting rotation degrees

Fig. 5. Common pretext tasks used for self-supervised learning in domain generalization. One can use a single pretext task, like solving Jigsaw puzzles [202] or predicting rotations [203], or combine multiple pretext tasks in a multi-task learning fashion.

[169]). Alternatively, weighted averaging can be adopted where the weights are estimated by, for example, a source domain classifier aiming to measure the similarity of the target sample to each source domain [170]. Also, the weights can be used to determine the most confident candidate whose output will serve for final prediction [80].

Domain-Specific Batch Normalization In batch normalization (BN) [200], the statistics are computed on-the-fly during training and their moving averages are stored in buffers for inference. Since the statistics typically vary in different source domains, one could argue that mixing statistics of multiple source domains is detrimental to learning generalizable representations. One solution is to use domain-specific BNs [171], [174], one for each source domain for collecting domain-specific statistics. This is equivalent to constructing domain-specific classifiers but with parameter sharing for most parts of a model except the normalization layers. Such a design was later adopted in [172] for dealing with MRI segmentation. In [173], the domain-specific predictions are aggregated using as weights the distance between a test data's instance-level feature statistics and the source domain BN statistics.

Weight Averaging aggregates model weights at different time steps during training to form a single model at test time [201]. Unlike explicit ensemble learning where multiple models (or model parts) need to be trained, weight averaging is a more efficient solution as the model only needs to be trained once. In [175], the authors have demonstrated that weight averaging can greatly improve model robustness under domain shift. In fact, such a technique is orthogonal to many other DG approaches and can be applied as a post-processing method to further boost the DG performance.

3.5 Self-Supervised Learning

Self-supervised learning is often referred to as learning with free labels generated from data itself (see [204] for a comprehensive survey on self-supervised learning). In computer vision, this can be achieved by teaching a model to predict the transformations applied to the image data, such as the shuffling order of patch-shuffled images [202] or rotation degrees [203]. See Figure 5 for illustrations.

So why can self-supervised learning improve DG? An intuitive explanation is that solving pretext tasks allows a model to learn generic features regardless of the target task, and hence less over-fitting to domain-specific biases [49]. An obvious advantage of self-supervised learning is that it can be applied to both single- and multi-source scenarios without requiring any domain labels.

Single Pretext Task In addition to using the standard classification loss, Carlucci et al. [49] taught a neural network to solve the Jigsaw puzzles problem [202], hoping that the network can learn regularities that are more generalizable across domains. Similarly, Wang et al. [81] used the Jigsaw solving task as an intrinsic supervision, together with an extrinsic supervision implemented using metric learning. Reconstruction has also been investigated for DG, such as learning an autoencoder to reconstruct image pixels/features [53], [176].

Multiple Pretext Tasks It is also possible to combine multiple pretext tasks. In [50], the authors combined two pretext tasks, namely solving Jigsaw puzzles and predicting rotations. In [115], three pretext tasks are combined, namely reconstructing the Gabor filter's response, predicting rotations, and predicting feature cluster assignments [205]. Overall, using multiple pretext tasks gives a better performance than using a single pretext task, as shown in [50].

Currently, these self-supervised learning-based DG methods have only been evaluated on the object recognition task. It is still unclear whether they will work on a wider range of OOD generalization tasks, which would be interesting to investigate in future work. Another concerns are that in general none of the existing pretext tasks is universal, and that the selection of pretext tasks is problem-specific. For instance, when the target domain shift is related to rotations, the model learned with the rotation prediction task will capture rotation-sensitive information, which is harmful to generalization.

Recent state-of-the-art self-supervised learning methods [206], [207] are mostly based on combining contrastive learning with data augmentation. The key idea is to pull together the same instance (image) undergone different transformations (e.g., random flip and color distortion) while push away different instances to learn instance-aware representations. Different from predicting transformations such as rotation, contrastive learning aims to learn transformation-invariant representations. Future work can explore whether invariances learned via contrastive learning can better adapt to OOD data.

3.6 Learning Disentangled Representations

Instead of forcing the entire model or features to be domain-invariant, which is challenging, one can relax this constraint by allowing some parts to be domain-specific, essentially learning disentangled representations. The existing approaches falling into this group are either based on decomposition [33], [41], [177], [178] or generative modeling [46], [179], both requiring domain labels for feature disentanglement.

Decomposition An intuitive way to achieve disentangled representation learning is to decompose a model into two parts, one being domain-specific while the other being domain-agnostic. Based on SVMs, Khosla et al. [41] decomposed a classifier into domain-specific biases and domain-agnostic weights, and only kept the latter when dealing with unseen domains. This approach was later extended to neural networks in [33]. One can also design domain-specific modules such as in [177] where domain-specific binary masks are imposed on the final feature vector to distinguish between domain-specific and domain-invariant components. Another solution is to apply low-rank decomposition to a model's weight matrices in order to identify common features that are more generalizable [178].

Generative Modeling has been a powerful tool for learning disentangled representations [208]. In [179], a variational autoencoder (VAE) is utilized to learn three independent latent subspaces for class, domain and object, respectively. In [46], two separate encoders are learned in an adversarial way to capture identity and domain information respectively for cross-domain face anti-spoofing.

3.7 Regularization Strategies

Some approaches focus on regularization strategies designed based on some heuristics. Wang et al. [51] argued that generalizable features should capture the global structure/shape of objects rather than relying on local patches/textures, and therefore proposed to suppress the predictive power of auxiliary patch-wise CNNs (maximizing their classification errors), implemented as a stack of 1×1 convolution layers. With a similar motivation, Huang et al. [52] iteratively masked out over-dominant features with large gradients, thus forcing the model to rely more on the remaining features. These methods do not require domain labels for learning, and are orthogonal to other DG methods like those based on domain alignment [45], [139] and data augmentation [20], [31], [35]. Therefore, one could potentially combine them to improve the performance in practice.

4 Future Research Directions

So far we have covered the background on domain generalization (DG) in § 2—knowing what DG is about and how DG is typically evaluated under different settings/datasets—as well as gone though the existing methodologies developed over the last decade in § 3. The following questions would naturally arise: 1) Has DG been solved? 2) If not, how far are we from solving DG?

The answer is of course not—DG is a very challenging problem and is far from being solved. In this section, we aim to share some insights on future research directions, pointing out what have been missed in the current research and discussing what are worth exploring to further this field. Specifically, we discuss potential directions from three perspectives: *model* (§ 4.1), *learning* (§ 4.2), and *benchmarks* (§ 4.3).

4.1 Model Architecture

Dynamic Architectures The weights in a convolutional neural network (CNN), which serve as feature detectors, are normally fixed once learned from source domains. This may result in the representational power of a CNN model restricted to the seen domains while generalizing poorly when the image statistics in an unseen domain are significantly different. One potential solution is to develop *dynamic* architectures [209], e.g., with weights conditioned on the input [210]. The key is to make neural networks' parameters (either partly or entirely) dependent on the input while ensuring that the model size is not too large to harm the efficiency. Dynamic architectures such as dynamic filter networks [210] and conditional convolutions [211] have been shown effective on generic visual recognition tasks like classification and segmentation. It would be interesting to see whether such a flexible architecture can be used to cope with domain shift in DG.

Adaptive Normalization Layers Normalization layers [200], [212], [213] have been a core building block in contemporary neural networks. Following [214], a general formulation for different normalization layers can be written as $\gamma\frac{x-\mu}{\sigma}+\beta$, where μ and σ denote mean and variance respectively; γ and β are learnable scaling and shift parameters respectively. Typically, (μ, σ) are computed on-the-fly during training but are saved in buffers using their moving averages for inference. Regardless of whether they are computed within each instance or based on a mini-batch, they can only represent the distribution of training data. The affine transformation parameters, i.e. γ and β, are also learned for source data only. Therefore, a normalization layer's parameters are not guaranteed to work well under domain shift in unseen test data. It would be a promising direction to investigate how to make these parameters adaptive to unseen domains [215].

4.2 Learning

Learning without Domain Labels Most existing methods leveraged domain labels in their models. However, in real-world applications it is possible that domain labels are difficult to obtain, e.g., web images crawled from the Internet are taken by arbitrary users with arbitrary domain characteristics and thus the domain labels are extremely difficult to define [167]. In such scenarios where domain labels are missing, many top-performing DG approaches are not viable any more. Though this topic has been studied in the past (e.g., [20], [147], [216]), methods that can deal with the absence of domain labels are still scarce and noncompetitive with methods that utilize domain labels. Considering that learning without domain labels is much more efficient and scalable, we encourage more future work to tackle this topic.

Learning to Synthesize Novel Domains The DG performance can greatly benefit from increasing the diversity of source domains. This is also confirmed in a recent work [217] where the authors emphasized the importance of having diverse training distributions to out-of-distribution (OOD) generalization. However, in practice it is impossible to collect training data that cover all possible domains. As such, learning to synthesize novel domains can be a potential solution. Though this idea has been roughly explored in a couple of recent DG works [20], [31], the results still have much room for improvements.

Avoiding Learning Shortcut Shortcut learning can be interpreted as a problem of learning 'easy' representations that can perform well on training data but are irrelevant to the task [218]. For example, given the task of distinguishing between digits blended with different colors, a neural network might be biased toward recognizing colors rather than the digit shapes during training, thus leading to poor generalization on unseen data [219]. Such a problem can be intensified on multi-source data in DG as each source domain typically contains its own domain-specific bias. As a consequence, a DG model might simply learn to memorize the domain-specific biases, such as image styles [33], when tasked to differentiate between instances from different domains. The shortcut learning problem has been overlooked in DG.

Causal Representation Learning Currently, the common pipeline used in DG, as well as in many other fields, for representation learning is to learn a mapping $P(Y|X)$ by sampling data from the marginal distribution $P(X)$ with an objective to match the joint distribution $P(X,Y) = P(Y|X)P(X)$ (typically via maximum likelihood optimization). However, the learned representations have turned out to be lacking in the ability to adapt to OOD data [220]. A potential solution is to model the underlying causal variables (e.g., by autoencoder [220]) which cannot be directly observed but are much more stable and robust under distribution shift. This is closely related to the topic of causal representation learning, a recent trend in the machine learning community [221].

Exploiting Side Information Side information has been commonly used to boost the performance of a pattern recognition system. For example, depth information obtained from RGB-D sensors can be used alongside RGB images to improve the performance of, e.g., generic object detection [222] or human detection [223]. In DG, there exist a few works that utilize side information, such as attribute labels [124] or object segmentation masks [224]. In terms of attributes, they could be more generalizable because they capture mid- to low-level visual cues like colors, shapes and stripes, which are shared among different objects and less sensitive to domain biases [124]. Notably, attributes have been widely used in zero-shot learning to recognize unseen classes [120], [123]. In contrast, features learned for discrimination are usually too specific to objects, such as dog ears and human faces as found in top-layer CNN features [225], which are more likely to capture domain biases and hence less transferable between tasks [118].

Transfer Learning A couple of recent works [83], [84] have focused on the transfer learning perspective when designing DG methods for synthetic-to-real applications. Given a model pre-trained on large real datasets like ImageNet [9], the main goal is to learn new knowledge that is useful to the downstream task from synthetic data, and in the meantime, to maintain the knowledge on real images that was acquired from pre-training. Such a setting is closely related to learning-without-forgetting (LwF) [226]. In particular, a technique used in [83] was borrowed from LwF [226], i.e. minimizing the divergence between the new model's output and the old model's output to avoid erasing the pre-trained knowledge. Synthetic-to-real transfer learning is a realistic and practical setting but research in this direction has been less explored for DG.

Semi-Supervised Domain Generalization Most existing DG research assumes data collected from each source domain are fully annotated so the proposed methods are purely based on supervised learning, which are unable to cope with unlabeled data. However, in practice the size of labeled data could well be limited due to high annotation cost, but collecting abundant unlabeled data is much easier and cheaper. This leads to a more realistic and practical setting termed semi-supervised domain generalization [25], [26], [227], which has recently picked up attention from the DG community. In [26], pseudo-labels are assigned to unlabeled source data and an off-the-shelf style transfer model is used to augment the domain space. In [25], feature statistics are mixed between labeled and pseudo-labeled source data for data augmentation. Since designing data-efficient, and yet generalizable learning systems is essential for practical applications, we believe semi-supervised domain generalizable is worth investigating for future work.

Open Domain Generalization is a recently introduced problem setting [24] where a model is learned from heterogeneous source domains with different label sets (with overlaps) and deployed in unseen domains for recognizing known classes while being able to reject unknown classes. This problem setting is related to existing heterogeneous DG [31], [37] but focuses on classification applications and emphasizes the ability to detect (reject) unknown classes, which is often studied in open-set recognition [228]. In [24], a variant of Mixup [196] is proposed for data augmentation at both feature and label level, and a confidence threshold is used to reject test samples that likely belong to unknown classes.

4.3 Benchmarks

Incremental Learning + DG Most existing research on DG implicitly assumes that source domains are fixed and a model needs to be learned only once. However, in practice, it might well be the case that source domains are incrementally introduced, thus requiring incremental learning [229]. For instance, in cross-dataset person re-identification we might well have access to, say only two datasets at the beginning for model learning, e.g., Market1501 [72] and DukeMTMC-reID [73], but later another dataset comes in, e.g., CUHK03 [94], which increases the number of source datasets from two to three. In this case, several problems need to be addressed, such as 1) how to efficiently fine-tune the model on the new dataset without training from scratch using all available datasets, 2) how to make sure the model does not over-fit the new dataset and forget the previously learned knowledge, and 3) will the new dataset be beneficial or detrimental to the DG performance on the target domain.

Heterogeneous Domain Shift The current DG datasets mainly contain homogeneous domain shift, which means the source-source and source-target domain shifts are highly correlated with each other. For example, on PACS [33] the source-source domain shift and the source-target domain shift are both related to image style changes; on Rotated MNIST [53] rotation is the only cause of domain shift. However, in real-world scenarios the target domain shift is unpredictable and less likely to be correlated with the source domain shift, e.g., the source domains might be photo, art and sketch but the target domain might be images of novel viewpoints; or the source domains contain digit images with different rotations but the target domain images might be in a different font style or background. Such a setting, which we call heterogeneous domain shift, has never been brought up but is critical to practical applications.

5 Conclusion

Domain generalization (DG) has been a fast growing area, with plenty of methodologies proposed each year and various datasets curated for benchmarking. As the first survey paper in this topic, we have introduced the background covering the problem definitions and the commonly used datasets, as well as comparisons with related topics; and have summarized the ten-year development in DG methodologies with a clear categorization. Potential research directions based on three perspectives (model, learning and benchmarks) have also been discussed. We hope this timely and up-to-date survey can offer a clear overview of the DG research and inspire more future work to advance this field.

Acknowledgement. This research was conducted in collaboration with SenseTime. This work is supported by A*STAR through the Industry Alignment Fund - Industry Collaboration Projects Grant; National Natural Science Foundation of China (61876176, U1713208); the National Key Research and Development Program of China (No. 2020YFC2004800); Science and Technology Service Network Initiative of Chinese Academy of Sciences (KFJ-STS-QYZX-092); the Shanghai Committee of Science and Technology, China (Grant No. 20DZ1100800).

References

[1] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *PR*, 2012.

[2] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do imagenet classifiers generalize to imagenet?" in *ICML*, 2019.

[3] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *ML*, 2010.

[4] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, "Measuring robustness to natural distribution shifts in image classification," in *NeurIPS*, 2020.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, 2015.

[8] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *ICLR*, 2019.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[10] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *ECCV*, 2010.

[11] Z. Lu, Y. Yang, X. Zhu, C. Liu, Y.-Z. Song, and T. Xiang, "Stochastic classifiers for unsupervised domain adaptation," in *CVPR*, 2020.

[12] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *CVPR*, 2018.

[13] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, 2015.

[14] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *ICML*, 2015.

[15] Z. Liu, Z. Miao, X. Pan, X. Zhan, D. Lin, S. X. Yu, and B. Gong, "Open compound domain adaptation," in *CVPR*, 2020.

[16] B. Li, Y. Wang, S. Zhang, D. Li, T. Darrell, K. Keutzer, and H. Zhao, "Learning invariant representations and risks for semi-supervised domain adaptation," in *CVPR*, 2021.

[17] K. Muandet, D. Balduzzi, and B. Scholkopf, "Domain generalization via invariant feature representation," in *ICML*, 2013.

[18] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, "Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data," in *ICCV*, 2019.

[19] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," in *NeurIPS*, 2011.

[20] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," in *ICLR*, 2021.

[21] X. Fan, Q. Wang, J. Ke, F. Yang, B. Gong, and M. Zhou, "Adversarially adaptive normalization for single domain generalization," in *CVPR*, 2021.

[22] X. Zhang, P. Cui, R. Xu, L. Zhou, Y. He, and Z. Shen, "Deep stable learning for out-of-distribution generalization," in *CVPR*, 2021.

[23] P. Pandey, M. Raman, S. Varambally, and P. AP, "Domain generalization via inference-time label-preserving target projections," in *CVPR*, 2021.

[24] Y. Shu, Z. Cao, C. Wang, J. Wang, and M. Long, "Open domain generalization with domain-augmented meta-learning," in *CVPR*, 2021.

[25] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Mixstyle neural networks for domain generalization and adaptation," *arXiv:2107.02053*, 2021.

[26] K. Zhou, C. C. Loy, and Z. Liu, "Semi-supervised domain generalization with stochastic stylematch," *arXiv preprint arXiv:2106.00592*, 2021.

[27] H. Li, S. Jialin Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *CVPR*, 2018.

[28] Y. Li, X. Tiana, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao, "Deep domain generalization via conditional invariant adversarial networks," in *ECCV*, 2018.

[29] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *AAAI*, 2018.

[30] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "Metareg: Towards domain generalization using meta-regularization," in *NeurIPS*, 2018.

[31] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Learning to generate novel domains for domain generalization," in *ECCV*, 2020.

[32] K. Zhou, Y. Yang, T. M. Hospedales, and T. Xiang, "Deep domain-adversarial image generation for domain generalisation." in *AAAI*, 2020.

[33] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *ICCV*, 2017.

[34] Y. Li, Y. Yang, W. Zhou, and T. Hospedales, "Feature-critic networks for heterogeneous domain generalization," in *ICML*, 2019.

[35] R. Volpi and V. Murino, "Addressing model vulnerability to distributional shifts over image transformation sets," in *ICCV*, 2019.

[36] Y. Shi, X. Yu, K. Sohn, M. Chandraker, and A. K. Jain, "Towards universal representation learning for deep face recognition," in *CVPR*, 2020.

[37] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," in *ICCV*, 2019.

[38] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR*, 2011.

[39] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *CVPR-W*, 2004.

[40] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *IJCV*, 2008.

[41] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *ECCV*, 2012.

[42] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Learning generalisable omni-scale representations for person re-identification," *TPAMI*, 2021.

[43] R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," in *NeurIPS*, 2018.

[44] J. Huang, D. Guan, A. Xiao, and S. Lu, "Fsdr: Frequency space domain randomization for domain generalization," in *CVPR*, 2021.

[45] R. Shao, X. Lan, J. Li, and P. C. Yuen, "Multi-adversarial discriminative deep domain generalization for face presentation attack detection," in *CVPR*, 2019.

[46] G. Wang, H. Han, S. Shan, and X. Chen, "Cross-domain face presentation attack detection via multi-domain disentangled representation learning," in *CVPR*, 2020.

[47] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," in *ICLR*, 2020.

[48] Z. Tang, Y. Gao, Y. Zhu, Z. Zhang, M. Li, and D. Metaxas, "Selfnorm and crossnorm for out-of-distribution robustness," *arXiv preprint arXiv:2102.02811*, 2021.

[49] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *CVPR*, 2019.

[50] S. Bucci, A. D'Innocente, Y. Liao, F. M. Carlucci, B. Caputo, and T. Tommasi, "Self-supervised learning across domains," *arXiv preprint arXiv:2007.12368*, 2020.

[51] H. Wang, Z. He, Z. C. Lipton, and E. P. Xing, "Learning robust representations by projecting superficial statistics out," in *ICLR*, 2019.

[52] Z. Huang, H. Wang, E. P. Xing, and D. Huang, "Self-challenging improves cross-domain generalization," in *ECCV*, 2020.

[53] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *ICCV*, 2015.

[54] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *IEEE*, 1998.

[55] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NeurIPS-W*, 2011.

[56] C. Fang, Y. Xu, and D. N. Rockmore, "Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias," in *ICCV*, 2013.

[57] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, 2010.

[58] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010.

[59] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *CVPR*, 2017.

[60] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *ICCV*, 2019.

[61] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *arXiv preprint arXiv:2003.07325*, 2020.

[62] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "Visda: The visual domain adaptation challenge," *arXiv preprint arXiv:1710.06924*, 2017.

[63] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *NeurIPS*, 2017.

[64] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *CVIU*, 2006.

[65] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[66] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *ICCV*, 2011.

[67] Z. Yao, Y. Wang, X. Du, M. Long, and J. Wang, "Adversarial pyramid network for video domain generalization," *arXiv preprint arXiv:1912.03716*, 2019.

[68] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *CVPR*, 2016.

[69] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *ECCV*, 2016.

[70] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[71] S. Beery, G. Van Horn, and P. Perona, "Recognition in terra incognita," in *ECCV*, 2018.

[72] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *ICCV*, 2015.

[73] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *ECCV*, 2016.

[74] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in *ICB*, 2012.

[75] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, "Oulu-npu: A mobile face presentation attack database with real-world variations," in *FG*, 2017.

[76] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *TIFS*, 2015.

[77] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *BIOSIG*, 2012.

[78] W. Li, X. Zhu, and S. Gong, "Scalable person re-identification by harmonious attention," *IJCV*, 2019.

[79] N. Somavarapu, C.-Y. Ma, and Z. Kira, "Frustratingly simple domain generalization via image stylization," *arXiv preprint arXiv:2006.11207*, 2020.

[80] M. Mancini, S. R. Bulò, B. Caputo, and E. Ricci, "Best sources forward: domain generalization through source-specific nets," in *ICIP*, 2018.

[81] S. Wang, L. Yu, C. Li, C.-W. Fu, and P.-A. Heng, "Learning from extrinsic and intrinsic supervisions for domain generalization," in *ECCV*, 2020.

[82] L. Yang, Y. Balaji, S.-N. Lim, and A. Shrivastava, "Curriculum manager for source selection in multi-source domain adaptation," in *ECCV*, 2020.

[83] W. Chen, Z. Yu, Z. Wang, and A. Anandkumar, "Automated synthetic-to-real generalization," in *ICML*, 2020.

[84] W. Chen, Z. Yu, S. D. Mello, S. Liu, J. M. Alvarez, Z. Wang, and A. Anandkumar, "Contrastive syn-to-real generalization," in *ICLR*, 2021.

[85] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[86] S. J. Pan and Q. Yang, "A survey on transfer learning," *TKDE*, 2009.

[87] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *ICML*, 2018.

[88] Y. Sun, L. Zheng, Y. Li, Y. Yang, Q. Tian, and S. Wang, "Learning part-based convolutional features for person re-identification," *TPAMI*, 2019.

[89] T. Chen, S. Ding, J. Xie, Y. Yuan, W. Chen, Y. Yang, Z. Ren, and Z. Wang, "Abd-net: Attentive but diverse person re-identification," in *ICCV*, 2019.

[90] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *ICCV*, 2019.

[91] X. Chang, T. M. Hospedales, and T. Xiang, "Multi-level factorisation net for person re-identification," in *CVPR*, 2018.

[92] Y. Zhao, Z. Zhong, F. Yang, Z. Luo, Y. Lin, S. Li, and N. Sebe, "Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification," in *CVPR*, 2021.

[93] S. Choi, T. Kim, M. Jeong, H. Park, and C. Kim, "Meta batch-instance normalization for generalizable person re-identification," *arXiv preprint arXiv:2011.14670*, 2020.

[94] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *CVPR*, 2014.

[95] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer gan to bridge domain gap for person re-identification," in *CVPR*, 2018.

[96] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *CVPR*, 2014.

[97] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *CVPR*, 2014.

[98] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*, 2016.

[99] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *ECCV*, 2016.

[100] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *CVPR*, 2011.

[101] N. D. Kalka, B. Maze, J. A. Duncan, K. O'Connor, S. Elliott, K. Hebert, J. Bryan, and A. K. Jain, "Ijb–s: Iarpa janus surveillance video benchmark," in *BTAS*, 2018.

[102] Z. Cheng, X. Zhu, and S. Gong, "Low-resolution face recognition," in *ACCV*, 2018.

[103] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a," in *CVPR*, 2015.

[104] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney *et al.*, "Iarpa janus benchmark-c: Face dataset and protocol," in *ICB*, 2018.

[105] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs, "Frontal to profile face verification in the wild," in *WACV*, 2016.

[106] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *CVPR*, 2016.

[107] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *arXiv preprint arXiv:1408.5601*, 2014.

[108] Y. Jia, J. Zhang, S. Shan, and X. Chen, "Single-side domain generalization for face anti-spoofing," in *CVPR*, 2020.

[109] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang, "Wilds: A benchmark of in-the-wild distribution shifts," *arXiv preprint arXiv:2012.07421*, 2020.

[110] Y. Yang and T. Hospedales, "Deep multi-task representation learning: A tensor factorisation approach," in *ICLR*, 2017.

[111] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *ECCV*, 2018.

[112] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *CVPR*, 2019.

[113] P. Guo, C.-Y. Lee, and D. Ulbricht, "Learning to branch for multi-task learning," in *ICML*, 2020.

[114] X. Sun, R. Panda, R. Feris, and K. Saenko, "Adashare: Learning what to share for efficient deep multi-task learning," in *NeurIPS*, 2020.

[115] I. Albuquerque, N. Naik, J. Li, N. Keskar, and R. Socher, "Improving out-of-distribution generalization via multi-task self-supervised pretraining," *arXiv preprint arXiv:2003.13525*, 2020.

[116] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *ICCV*, 2015.

[117] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[118] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *NeurIPS*, 2014.

[119] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *ICML*, 2014.

[120] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *TPAMI*, 2014.

[121] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," in *ECCV*, 2016.

[122] M. Mancini, Z. Akata, E. Ricci, and B. Caputo, "Towards recognizing unseen categories in unseen domains," in *ECCV*, 2020.

[123] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly," *TPAMI*, 2018.

[124] C. Gan, T. Yang, and B. Gong, "Learning attributes equals multi-source domain generalization," in *CVPR*, 2016.

[125] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *CVPR*, 2012.

[126] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *NeurIPS*, 2016.

[127] Y. Balaji, R. Chellappa, and S. Feizi, "Normalized wasserstein for mixture distributions with applications in adversarial learning and domain adaptation," in *ICCV*, 2019.

[128] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *CVPR*, 2019.

[129] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *CVPR*, 2011.

[130] K.-C. Peng, Z. Wu, and J. Ernst, "Zero-shot deep domain adaptation," in *ECCV*, 2018.

[131] P. Panareda Busto and J. Gall, "Open set domain adaptation," in *ICCV*, 2017.

[132] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *ECCV*, 2018.

[133] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Universal domain adaptation," in *CVPR*, 2019.

[134] S. Erfani, M. Baktashmotlagh, M. Moshtaghi, X. Nguyen, C. Leckie, J. Bailey, and R. Kotagiri, "Robust domain generalisation by enforcing distribution invariance," in *IJCAI*, 2016.

[135] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *TPAMI*, 2017.

[136] Y. Li, M. Gong, X. Tian, T. Liu, and D. Tao, "Domain generalization via conditional invariant representations," in *AAAI*, 2018.

[137] X. Jin, C. Lan, W. Zeng, and Z. Chen, "Feature alignment and restoration for domain generalization and adaptation," *arXiv preprint arXiv:2006.12009*, 2020.

[138] S. Hu, K. Zhang, Z. Chen, and L. Chan, "Domain generalization via multidomain discriminant analysis," in *UAI*, 2020.

[139] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *ICCV*, 2017.

[140] C. Yoon, G. Hamarneh, and R. Garbi, "Generalizable feature learning in the presence of data bias and domain class imbalance with application to skin lesion classification," in *MICCAI*, 2019.

[141] D. Mahajan, S. Tople, and A. Sharma, "Domain generalization using causal matching," *arXiv preprint arXiv:2006.07500*, 2020.

[142] Z. Wang, M. Loog, and J. van Gemert, "Respecting domain relations: Hypothesis invariance for domain generalization," *arXiv preprint arXiv:2010.07591*, 2020.

[143] H. Li, Y. Wang, R. Wan, S. Wang, T.-Q. Li, and A. C. Kot, "Domain generalization for medical imaging classification with linear-dependency regularization," in *NeurIPS*, 2020.

[144] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "Correlation-aware adversarial domain adaptation and generalization," *PR*, 2020.

[145] I. Albuquerque, J. Monteiro, M. Darvishi, T. H. Falk, and I. Mitliagkas, "Generalizing to unseen domains via distribution matching," *arXiv preprint arXiv:1911.00804*, 2019.

[146] Z. Deng, F. Ding, C. Dwork, R. Hong, G. Parmigiani, P. Patil, and P. Sur, "Representation via representations: Domain generalization via adversarially learned invariant representations," *arXiv preprint arXiv:2006.11478*, 2020.

[147] T. Matsuura and T. Harada, "Domain generalization using a mixture of multiple latent domains," in *AAAI*, 2020.

[148] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Adversarial invariant feature learning with accuracy constraint for domain generalization," in *ECMLPKDD*, 2019.

[149] S. Aslani, V. Murino, M. Dayan, R. Tam, D. Sona, and G. Hamarneh, "Scanner invariant multiple sclerosis lesion segmentation from mri," in *ISBI*, 2020.

[150] S. Zhao, M. Gong, T. Liu, H. Fu, and D. Tao, "Domain generalization via entropy regularization," in *NeurIPS*, 2020.

[151] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, "Sequential learning for domain generalization," in *ECCV-W*, 2020.

[152] Y. Du, J. Xu, H. Xiong, Q. Qiu, X. Zhen, C. G. Snoek, and L. Shao, "Learning to learn with variational information bottleneck for domain generalization," in *ECCV*, 2020.

[153] Q. Liu, Q. Dou, and P.-A. Heng, "Shape-aware meta-learning for generalizing prostate mri segmentation to unseen domains," in *MICCAI*, 2020.

[154] Q. Dou, D. C. Castro, K. Kamnitsas, and B. Glocker, "Domain generalization via model-agnostic learning of semantic features," in *NeurIPS*, 2019.

[155] Y. Du, X. Zhen, L. Shao, and C. G. M. Snoek, "Metanorm: Learning to normalize few-shot batches across domains," in *ICLR*, 2021.

[156] S. Otálora, M. Atzori, V. Andrearczyk, A. Khan, and H. Müller, "Staining invariant features for improving generalization of deep convolutional neural networks in computational pathology," *Frontiers in bioengineering and biotechnology*, 2019.

[157] C. Chen, W. Bai, R. H. Davies, A. N. Bhuva, C. H. Manisty, J. B. Augusto, J. C. Moon, N. Aung, A. M. Lee, M. M. Sanghvi *et al.*, "Improving the generalizability of convolutional neural network-based segmentation on cmr images," *Frontiers in cardiovascular medicine*, 2020.

[158] L. Zhang, X. Wang, D. Yang, T. Sanford, S. Harmon, B. Turkbey, B. J. Wood, H. Roth, A. Myronenko, D. Xu *et al.*, "Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation," *TMI*, 2020.

[159] F. Qiao, L. Zhao, and X. Peng, "Learning to learn single domain generalization," in *CVPR*, 2020.

[160] A. Sinha, H. Namkoong, R. Volpi, and J. Duchi, "Certifying some distributional robustness with principled adversarial training," *arXiv preprint arXiv:1710.10571*, 2017.

[161] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi, "Generalizing across domains via cross-gradient training," in *ICLR*, 2018.

[162] Z. Xu, D. Liu, J. Yang, C. Raffel, and M. Niethammer, "Robust and generalizable visual representation learning via random convolutions," in *ICLR*, 2021.

[163] F. C. Borlino, A. D'Innocente, and T. Tommasi, "Rethinking domain generalization baselines," *arXiv preprint arXiv:2101.09060*, 2021.

[164] F. M. Carlucci, P. Russo, T. Tommasi, and B. Caputo, "Hallucinating agnostic images to generalize across domains." in *ICCV-W*, 2019.

[165] Z. Xu, W. Li, L. Niu, and D. Xu, "Exploiting low-rank structure from latent domains for domain generalization," in *ECCV*, 2014.

[166] L. Niu, W. Li, and D. Xu, "Multi-view domain generalization for visual recognition," in *ICCV*, 2015.

[167] ——, "Visual recognition by learning from web data: A weakly supervised domain generalization approach," in *CVPR*, 2015.

[168] Z. Ding and Y. Fu, "Deep domain generalization with structured low-rank constraint," *TIP*, 2017.

[169] A. D'Innocente and B. Caputo, "Domain generalization with domain-specific aggregation modules," in *GCPR*, 2018.

[170] S. Wang, L. Yu, K. Li, X. Yang, C.-W. Fu, and P.-A. Heng, "Dofe: Domain-oriented feature embedding for generalizable fundus image segmentation on unseen datasets," *TMI*, 2020.

[171] S. Seo, Y. Suh, D. Kim, J. Han, and B. Han, "Learning to optimize domain specific normalization for domain generalization," in *ECCV*, 2020.

[172] Q. Liu, Q. Dou, L. Yu, and P. A. Heng, "Ms-net: Multi-site network for improving prostate segmentation with heterogeneous mri data," *TMI*, 2020.

[173] M. Segù, A. Tonioni, and F. Tombari, "Batch normalization embeddings for deep domain generalization," *arXiv preprint arXiv:2011.12672*, 2020.

[174] M. Mancini, S. R. Bulo, B. Caputo, and E. Ricci, "Robust place categorization with deep domain generalization," *RA-L*, 2018.

[175] J. Cha, H. Cho, K. Lee, S. Park, Y. Lee, and S. Park, "Domain generalization needs stochastic weight averaging for robustness on domain shifts," *arXiv preprint arXiv:2102.08604*, 2021.

[176] U. Maniyar, A. A. Deshmukh, U. Dogan, and V. N. Balasubramanian, "Zero shot domain generalization," in *BMVC*, 2020.

[177] P. Chattopadhyay, Y. Balaji, and J. Hoffman, "Learning to balance specificity and invariance for in and out of domain generalization," in *ECCV*, 2020.

[178] V. Piratla, P. Netrapalli, and S. Sarawagi, "Efficient domain generalization via common-specific low-rank decomposition," in *ICML*, 2020.

[179] M. Ilse, J. M. Tomczak, C. Louizos, and M. Welling, "Diva: Domain invariant variational autoencoder," in *ICLR-W*, 2019.

[180] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008.

[181] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij, "On causal and anticausal learning," in *ICML*, 2012.

[182] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *JMLR*, 2012.

[183] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[184] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *CVPR*, 2017.

[185] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *CVPR*, 2018.

[186] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *NeurIPS*, 2018.

[187] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *CVPR*, 2019.

[188] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *JMLR*, 2016.

[189] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017.

[190] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *arXiv preprint arXiv:2004.05439*, 2020.

[191] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, and T. Xiang, "Incremental few-shot object detection," in *CVPR*, 2020.

[192] J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, and R. E. Turner, "Meta-learning probabilistic inference for prediction," in *ICLR*, 2019.

[193] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[194] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[195] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *ICCV*, 2017.

[196] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[197] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.

[198] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[199] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *ICCV*, 2011.

[200] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[201] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *UAI*, 2018.

[202] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *ECCV*, 2016.

[203] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *ICLR*, 2018.

[204] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *TPAMI*, 2020.

[205] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *ECCV*, 2018.

[206] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020.

[207] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," in *NeurIPS*, 2020.

[208] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *NeurIPS*, 2016.

[209] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *arXiv preprint arXiv:2102.04906*, 2021.

[210] X. Jia, B. De Brabandere, T. Tuytelaars, and L. Van Gool, "Dynamic filter networks," in *NeurIPS*, 2016.

[211] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "Condconv: Conditionally parameterized convolutions for efficient inference," in *NeurIPS*, 2019.

[212] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv:1607.08022*, 2016.

[213] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[214] P. Luo, J. Ren, Z. Peng, R. Zhang, and J. Li, “Differentiable learning-to-normalize via switchable normalization,” in *ICLR*, 2019.

[215] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, “Semantic image synthesis with spatially-adaptive normalization,” in *CVPR*, 2019.

[216] L. Deecke, T. Hospedales, and H. Bilen, “Latent domain learning with dynamic residual adapters,” *arXiv preprint arXiv:2006.00996*, 2020.

[217] K. Xu, M. Zhang, J. Li, S. S. Du, K.-I. Kawarabayashi, and S. Jegelka, “How neural networks extrapolate: From feedforward to graph neural networks,” in *ICLR*, 2021.

[218] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, “Shortcut learning in deep neural networks,” *Nature Machine Intelligence*, 2020.

[219] B. Kim, H. Kim, K. Kim, S. Kim, and J. Kim, “Learning not to learn: Training deep neural networks with biased data,” in *CVPR*, 2019.

[220] Y. Bengio, T. Deleu, N. Rahaman, R. Ke, S. Lachapelle, O. Bilaniuk, A. Goyal, and C. Pal, “A meta-transfer objective for learning to disentangle causal mechanisms,” *arXiv preprint arXiv:1901.10912*, 2019.

[221] B. Scholkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, “Towards causal representation learning,” *arXiv preprint arXiv:2102.11107*, 2021.

[222] J. Hoffman, S. Gupta, and T. Darrell, “Learning with side information through modality hallucination,” in *CVPR*, 2016.

[223] K. Zhou, A. Paiement, and M. Mirmehdi, “Detecting humans in rgb-d data with cnns,” in *MVA*, 2017.

[224] A. Zunino, S. A. Bargal, R. Volpi, M. Sameki, J. Zhang, S. Sclaroff, V. Murino, and K. Saenko, “Explainable deep classification models for domain generalization,” *arXiv preprint arXiv:2003.06498*, 2020.

[225] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *ECCV*, 2014.

[226] Z. Li and D. Hoiem, “Learning without forgetting,” *TPAMI*, 2017.

[227] H. Sharifi-Noghabi, H. Asghari, N. Mehrasa, and M. Ester, “Domain generalization via semi-supervised meta learning,” *arXiv preprint arXiv:2009.12658*, 2020.

[228] C. Geng, S.-j. Huang, and S. Chen, “Recent advances in open set recognition: A survey,” *TPAMI*, 2020.

[229] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, “Large scale incremental learning,” in *CVPR*, 2019.