# Domain Generalization: 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 statistical learning algorithms strongly rely on the i.i.d. assumption while in practice the target data often come from a different distribution than the source data, known as domain shift. Domain generalization (DG) aims to achieve OOD generalization by only using source domain data for model learning. Since first introduced in 2011, research in DG has undergone a decade progress. Ten years of research in this topic have led to a broad spectrum of methodologies, e.g., based on domain alignment, meta-learning, data augmentation, or ensemble learning, just to name a few; and have covered various 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 ten-year development in DG. First, we cover the background by giving the problem definitions and discussing how DG is related to other fields like domain adaptation and transfer learning. Second, we conduct a thorough review into existing methods and present a taxonomy based on their methodologies and motivations. Finally, we conclude this survey with potential research directions.

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

arXiv:2103.02503v1 [cs.LG] 3 Mar 2021

## 1 Introduction

If an image classifier was trained on photo images, can it be applied to 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 in 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? All these questions are linked to the same problem that most machine learning systems have been struggling with—the *domain shift* problem, specifically referring to distribution shift between training (source) and test (target) data [1], [2], [3].

Most statistical learning algorithms strongly rely on an over-simplified assumption that source and target data are independent and identically distributed (i.i.d.) [4], while ignoring the out-of-distribution (OOD) scenarios commonly encountered in practice—just like those examples given at the beginning. 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 deployment of machine learning applications. One might be curious if recent advances in deep neural networks [5], [6], known as deep learning [7], can mitigate this problem. The reality is that successes achieved by deep learning so far have been largely driven by supervised learning with large-scale annotated datasets like ImageNet [8]—again, relying on the i.i.d. assumption. Studies [2], [9] have revealed the weak performance of deep learning models in OOD datasets, even with small variations in the data generating process.


- *K. Zhou, Z. Liu and C.C. Loy are with 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; and Shanghai AI Lab, Shanghai, 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.*


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 fine-tune the model. This has been extensively studied under the topic of domain adaptation (DA) [10], [11], [12], [13], [14], [15], [16]. However, DA relies on a strong assumption that target data are accessible for model adaptation, which is often invalid. 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 semantic segmentation it is nearly impossible to collect data from all cities capturing all different scenes and under all possible weather conditions [18]. In these scenarios, the model is required to be intrinsically domain-generalizable once trained on source data.

To overcome the domain shift problem, as well as the absence of target data, *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], research in DG has undergone roughly a decade, with a plethora of methods 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 [21], [22], or augmenting data with image synthesis [24], [25], just to name a few. From the application point of view, existing research in DG has covered handwritten digit recognition [24], [25], object recognition [20], [29], semantic segmentation [18], [30], person re-identification [25], [26], face recognition [31], action recognition [27], [32], etc. Yet, DG is still an open challenge and requires more efforts from the research community to make progress.

In this survey paper, we aim to provide a comprehensive literature review, summarizing what we have learned from the ten years of research on DG. 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 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 taxonomy to categorize them. In § 4, we conclude this paper by discussing potential research directions for future work. It is worth mentioning that this is the first survey paper in this topic. We hope this timely survey can provide the research community with a clear picture on the up-to-date development in DG.

## 2 Background

In this section, we give the background knowledge about domain generalization (DG). In § 2.1, we briefly talk about the history of DG, covering 1) the motivation behind DG in both machine learning and computer vision, and 2) how the DG methodologies have developed. In § 2.2, we first define the DG problem and then present a taxonomy of DG settings. In § 2.3, related topics to DG are discussed and compared. In § 2.4, we talk about the evaluation process in DG and review the commonly used DG datasets.

### 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 the 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, which is also the main problem tackled in its follow-up [17]. 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 well to new patients and meanwhile, collecting new data for model fine-tuning is impractical, thus motivating the research on the DG problem.

In computer vision, a seminal work done by Torralba and Efros [33] 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 bias, which is difficult to avoid, can lead to poor generalization performance. For example, as shown in [33], a person classifier trained on Caltech101 [34] obtained a very low accuracy (11.8%) on LabelMe [35], though its same-dataset performance was near-perfect (99.6%). Following [33], Khosla et al. [36] 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 [25], [37], image segmentation [18], [38], face recognition [31], and face anti-spoofing [39], [40].

DG was initially studied under the *multi-source* setting [17], [19], i.e., the source data are collected from multiple related but distinct domains. The goal is simple—to learn a model by exploiting the multi-source data such that the model can generalize well to unseen domains. Moreover, the label space is assumed to be consistent across source and target domains, which is also known as *homogeneous* DG [20]. Recently, researchers have proposed a more challenging yet practical setting, named *heterogeneous* DG [24], [25], [29], [32], where the label space is different between source and target domains. Essentially, this setting requires the feature representations learned from source data to be domain-generalizable so that they can be directly deployed in the target domain to recognize novel categories. A representative application is cross-dataset person re-identification (re-ID) [41], [42] where a re-ID model learned from source datasets needs to produce discriminative features for unseen person identities captured in novel camera views. In contrast to multi-source DG, *single-source* DG has also been investigated [38], [43], [44].

In terms of the development of methodologies in DG, early works have been focused on using traditional models such as SVMs [17], [19], [36], [45]. For instance, to overcome the generalization issue in the aforementioned gating problem, Blanchard et al. [19] incorporated an $\ell_2$ regularization term into a kernel SVM formulation, hoping that the learned classifier could generalize to unseen patients' data. In the follow-up work in [17], Muandet et al. focused on domain-invariant subspace learning through minimizing the discrepancy between source domains, which laid the foundation for domain alignment-based methods [27], [28], [46], [47]. Ensemble methods based on exemplar SVMs have also been explored for DG in visual recognition problems [45], [48]. In [49], Gan et al. argued that attributes are more domain-generalizable features for image classification and thus studied attribute detection based on SVMs.

Similar to the development trends in many other machine learning and computer vision problems [6], [50], [51], [52], deep learning-based methods have received increasing attention in DG, and has been dominant in recent DG approaches [20], [22], [24], [26], [53]. The most popular line of work is based on domain alignment, which has led to a plethora of methods for learning deep domain-invariant representations, e.g., multi-task autoencoders [46], distance minimization [47], [54], and adversarial learning [27], [28]. Besides domain alignment, many other directions have also been explored to improve the generalization of deep models, such as meta-learning [21], [22], data augmentation [23],

[24], network architecture design [55], [56], [57], and self-supervised learning [58], [59], [60], just to name a few. We will talk about in more detail the existing DG methodologies in § 3.

### 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 $P_{XY}^{S}$ to denote the overall source joint distribution. The target domain is denoted by $\mathcal{T} = \{x_i^T\}_{i=1}^{N_T}$ where the data are sampled from the marginal $P_X^T$. Note that the labels in $\mathcal{T}$ are unavailable and need to be predicted. The corresponding joint distribution of $\mathcal{T}$ is denoted by $P_{XY}^T$. Also, $P_{XY}^T \neq P_{XY}^{(k)}, \forall k \in \{1, ..., K\}$.

**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 been studied under various settings in the literature. Here we summarize them and present a clear categorization based on 1) the number of source domains $K$, and 2) whether or not the label space is consistent between source and target domains (i.e. the relations between $\mathcal{Y}_{k/k'}$ and $\mathcal{Y}_T$). An overview is given in Table 1. See below for detailed discussions.

1) **Multi-source DG**: DG was initially studied under the multi-source setting in [19], i.e. $K > 1$. As stated in [19], the motivation was to leverage multi-source data to learn representations that are invariant to different marginal distributions. This makes sense because without accessing the target data, it is very challenging for a source-learned model to generalize well; therefore, using multiple domains offers more opportunities for a model to discover stable patterns across source domains that are also useful to novel domains. A recent work by Xu et al. [61] also confirmed the importance of having diverse training distributions for out-of-distribution (OOD) generalization. In addition to the normal setup where domain labels are known, researchers have also studied the scenarios where domain labels are hidden (e.g., latent domain learning [62]).
2) **Single-source DG**: This setting corresponds to $K = 1$, and has been mostly studied under the topic of OOD generalization [9], [63] where the goal is to assess a model's robustness to corruptions and perturbations. Motivated by the benefits of having diverse training distributions, most existing single-source DG methods have been focused on data augmentation [18], [38], [43]. This can be achieved by perturbing the original images with adversarial gradients produced by a classifier [30] or through style transfer with an external style bank for generating novel-style images [18]. Methods like domain alignment [17] or meta-learning [21] cannot be directly applied to this setting.
3) **Homogeneous DG**: This is a commonly adopted setting, especially for the applications of image classification [20] and segmentation [18] where the task remains the same from source to target—namely predicting the same set of classes. This setting can be formally defined as $\mathcal{Y}_k = \mathcal{Y}_{k'} = \mathcal{Y}_T$.
4) **Heterogeneous DG**: In this setting, the label space differs between different domains (including both source and target), i.e. $\mathcal{Y}_k \neq \mathcal{Y}_{k'} \neq \mathcal{Y}_T$. The term "heterogeneous DG" first appeared in [29] and the concurrent work [32]. In general, the goal is to learn from source domains a generalizable *feature representation* that can be directly deployed in the target domain to recognize novel classes. Both [29] and [32] conducted experiments using the Visual Decathlon (VD) dataset [64], which contains ten visual recognition tasks, such as CIFAR100 [65] (for object recognition), Omniglot [66] (for handwritten digit recognition) and UCF101 [67] (for action recognition). However, this setup could cause confusion with transfer learning [68] because the test tasks have been changed and moreover, the test data are used to train an SVM (or KNN) classifier based on a fixed representation (learned from the source tasks), violating the basic assumption of the absence of target data in DG. Some recent works [24], [25], [26] evaluated heterogeneous DG on an instance retrieval task, i.e. cross-dataset person re-ID [69], [70], which better fits the heterogeneous DG requirements.[2]

### 2.3 Related Topics

In this section, we discuss the relations between DG and its related topics, and clarify their differences (see Table 2 for an overview).

**Supervised Learning** generally aims to learn an input-output mapping by minimizing the following risk [4]: $\mathbb{E}_{(x,y)\sim\hat{P}_{XY}} \ell(f(x), y)$, 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 [33].

**Multi-Task Learning (MTL)** The goal of MTL is to simultaneously learn multiple related tasks using a single model [71], [72], [73], [74], [75]. As shown in Table 2, MTL aims to make a model perform well on the same set of

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

2. The re-ID process is essentially feature matching, hence the source-learned model can be directly deployed in the target domain for feature extraction and comparison without further learning a classifier.

TABLE 1
Summary of the existing domain generalization (DG) settings. $K$: number of source domains. $\mathcal{Y}_{k/k'}$: label space of source domains ($k \neq k'$). $\mathcal{Y}_T$: label space of the target domain.

| | Homogeneous DG | Heterogeneous DG |
|---|---|---|
| Single-source DG | $K=1, \mathcal{Y}_k = \mathcal{Y}_{k'} = \mathcal{Y}_T$ | $K=1, \mathcal{Y}_k \neq \mathcal{Y}_{k'} \neq \mathcal{Y}_T$ |
| Multi-source DG | $K>1, \mathcal{Y}_k = \mathcal{Y}_{k'} = \mathcal{Y}_T$ | $K>1, \mathcal{Y}_k \neq \mathcal{Y}_{k'} \neq \mathcal{Y}_T$ |

TABLE 2
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. $P_X^T$: target marginal.

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

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 [58], [60], [76]. Intuitively, MTL benefits from the effect of regularization brought by parameter sharing [71], 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 [68]. A well-known TL example in contemporary deep learning is fine-tuning: first pretrain deep neural networks on large-scale datasets, such as ImageNet [8] for vision models or BooksCorpus [77] for language models; then fine-tune the models on downstream tasks [50]. Given that pre-trained deep features are highly transferable, as shown in several studies [78], [79], a couple of recent DG works [44], [80] have researched how to preserve the transferable features learned via large-scale pretraining when learning new knowledge from source synthetic data for synthetic-to-real applications. As shown in Table 2, 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 [81], i.e. $P_Y^T \neq P_Y^S$, since the task is to recognize new classes, except for generalized ZSL [82] 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 ($P_X^T \neq P_X^S$). To recognize unseen classes in ZSL, a common practice is to learn a mapping between the input image space and the attribute space [83] since the label space is disjoint between training and test data. Interestingly, attributes have also been exploited in DG for learning domain-generalizable representations [49].

**Domain Adaptation (DA)** As the closest topic to DG, DA has been extensively researched over the literature [11], [12], [13], [14], [15], [84], [85], [86], [87], [88]. Similar to DG, DA also aims to tackle the domain shift problem, but differently, leverages sparse [89] or unlabeled [84] target data along with abundant labeled source data for model adaptation. In terms of pros and cons, DA lacks efficiency because adaptation has to be repeated whenever a new domain arrives, while DG trains a model only once and applies it everywhere; DA's performance is typically higher than DG's due to the use of target data; DA relies on a strong assumption that target data are always available, which may not be realistic in some applications, such as medical imaging analysis [90], [91] where data collection is often expensive and mostly requires domain-specific expertise. Though the focus is different in DA and DG—DA cares more about how to exploit unlabeled target data while generalization is the main focus in DG—many DG methods (e.g., [27], [46], [47], [54]) are inspired by the paradigm of domain alignment widely adopted in DA [3], i.e. to minimize the domain discrepancy between source domains for leaning domain-invariant models.

### 2.4 Evaluation and Datasets

In this section, we first talk about how a model's generalization performance is evaluated in DG, and then summarize the most commonly used datasets (with a focus on computer vision) for benchmarking DG algorithms. As discussed in § 2.2, the goal of DG is to improve a model's generalization ability in unseen domains. Therefore, the evaluation in DG is quite straightforward, which typically follows the *leave-one-domain-out* protocol [20]: given a dataset containing at least two distinct domains, one or multiple domains will

TABLE 3
Commonly used domain generalization benchmarks.

| Benchmark | # samples | # domains | Task | Description |
|---|---|---|---|---|
| Rotated MNIST [46] | 70,000 | 6 | Handwritten digit recognition | Extended from MNIST [92]; 10 classes; rotation degree $\in \{0, 15, 30, 45, 60, 75\}$ |
| Digits-DG [25] | 24,000 | 4 | Handwritten digit recognition | Combination of MNIST [92], MNIST-M [13], SVHN [93] and SYN [13]; 10 classes; domain shift mainly in font style and background |
| VLCS [94] | 10,729 | 4 | Object recognition | Combination of Caltech101 [34], LabelMe [35], PASCAL [95], and SUN09 [96]; 5 classes (bird, car, chair, dog, and person) |
| Office-31 [10] | 4,652 | 3 | Object recognition | Domain $\in$ {amazon, webcam, dslr}; 31 classes |
| OfficeHome [97] | 15,588 | 4 | Object recognition | Domain $\in$ {art, clipart, product, real}; 65 classes (related to objects in office and home environments) |
| PACS [20] | 9,991 | 4 | Object recognition | Domain $\in$ {photo, art, cartoon, sketch}; 7 classes (dog, elephant, giraffe, guitar, horse, house, and person); domain shift mainly in image style |
| DomainNet [98] | 586,575 | 6 | Object recognition | Domain $\in$ {clipart, infograph, painting, quick-draw, real, sketch}; 345 classes |
| miniDomainNet [99] | 140,006 | 4 | Object recognition | Extended from DomainNet; smaller and less noisy; domain $\in$ {clipart, painting, real, sketch}; 126 classes |
| ImageNet-Sketch [53] | 50,000 | 2 | Object recognition | Domain shift between real and sketch images; 1,000 classes |
| VisDA-17 [100] | 280,157 | 3 | Object recognition | 12 classes; synthetic images from renderings of 3D models (w/ varying angles and lighting conditions); real images from MS COCO [101] and the YouTube BBox dataset [102] |
| CIFAR-10-C [9] | 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 [9] | 60,000 | - | Object recognition | |
| ImageNet-C [9] | ≈1.3M | - | Object recognition | |
| Visual Decathlon [64] | 1,659,142 | 10 | Object/action/handwritten digit recognition | Initially designed as a multi-domain learning benchmark; consisting of ten datasets: Aircraft [103], CIFAR100 [65], DPed [104], DTD [105], GTSR [106], Flowers102 [107], ImageNet [8], Omniglot [66], SVHN [93], and UCF101 [67] |
| IXMAS [108] | 1,650 | 5 | Action recognition | 5 camera views; 10 subjects; 5 actions (following [27]) |
| UCF-HMDB | 3,809 | 2 | Action recognition | 12 overlapping actions shared between UCF101 [67] and HMDB51 [109] are selected by [110] |
| SYNTHIA [111] | 2,700 | 15 | Semantic segmentation | Synthetic dataset of urban scenarios; 13 classes; four locations (highway, New York-like city, and old European town) each with five different weather conditions (dawn, fog, night, spring, and winter), selected by [38] |
| GTA5-Cityscapes | 29,966 | 2 | Semantic segmentation | GTA5 [112] is a synthetic dataset built on a computer game, while Cityscapes [113] contains real-world street scenes from 50 different cities |
| TerraInc [114] | 24,788 | 4 | Animal classification | Extracted by [115] from a subset of the Caltech Camera Traps-20 dataset [114]; containing wild animals captured by fixed camera traps at different locations in the American Southwest; 10 classes |
| Market-Duke | 69,079 | 14 | Person re-identification | Cross-dataset re-ID between Market1501 [116] (w/ 1,501 identities) and DukeMTMC-reID [117], [118] (w/ 1,812 identities); each domain corresponds to a camera view; heterogeneous DG |
| Face | >5M | 9 | Face recognition | Setting used by [31]; MS-Celeb-1M [119] for training; evaluation on LFW [120], CFP [121], YTF [122], MegaFace [123], IJB-A [124], IJB-C [125], IJB-S [126], and TinyFace [127]; heterogeneous DG |
| COMI | ≈8,500 | 4 | Face anti-spoofing | Including CASIA-MFSD [128], Oulu-NPU [129], MSU-MFSD [130], and Idiap Replay-Attack [131]; domain shift in materials, illumination, background, resolution, etc. |


Fig. 1. Example images from handwritten digit datasets. The domain shift between datasets mainly corresponds to changes in font style, color and background.

be used for model training depending on whether the setting is single- or multi-source DG; then the model will be evaluated on the remaining unseen domain(s). In terms of performance metric, it depends on the specific dataset and task. For instance, for image classification, the evaluation criterion is typically top-1 accuracy; while for semantic segmentation, the mean Intersection-over-Union (mIoU) metric is adopted.

Over 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 [97] and DomainNet [98], 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 [94] is a combination of Caltech101 [34], LabelMe [35], PASCAL [95], and SUN09 [96], all targeting the object recognition problem but encoding dataset-specific biases [33]. We list all datasets in Table 3 with basic statistics and short descriptions, and briefly review their challenges below.

**Handwritten Digit Recognition** Several handwritten digit datasets have been widely used as DG testbed, e.g., MNIST [92], MNIST-M [13], SVHN [93], and SYN [13]. MNIST contains images of handwritten digits. Extended from MNIST, MNIST-M contains MNIST images mixed 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. Digits-DG [25] 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—in Rotated MNIST [46] a rotation degree is regarded as a domain.

**Object Recognition** is generally the most studied task in DG, as can be seen in Table 3. Below we summarize those datasets based on their domain shift types. 1) The domain shift in VLCS [94] 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 2(a)). 2) Several datasets have been focused on image style changes, including OfficeHome [97], PACS [20], DomainNet [98], and ImageNet-Sketch [53]. Example images manifesting the image style changes are shown in Figure 2(b). Based on studies in these datasets [32], [99], [132], [133], it is generally acknowledged that when the source image 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 [60], [99], [134]. 3) A couple of recent DG studies [44], [80] 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 [100] (see Figure 2(c)). This is an important yet understudied 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 [9]. 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. The research is largely motivated by adversarial attacks [135], 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 [64], has also been employed, for evaluating heterogeneous DG [29], [32]. However, due to both the changes in label space and the use of target data for training SVM classifiers, this setup is closer to transfer learning [68].

**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 [108] has been widely used as a cross-view action recognition benchmark [27], [32], 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 [88]. Since it is generally impractical to collect

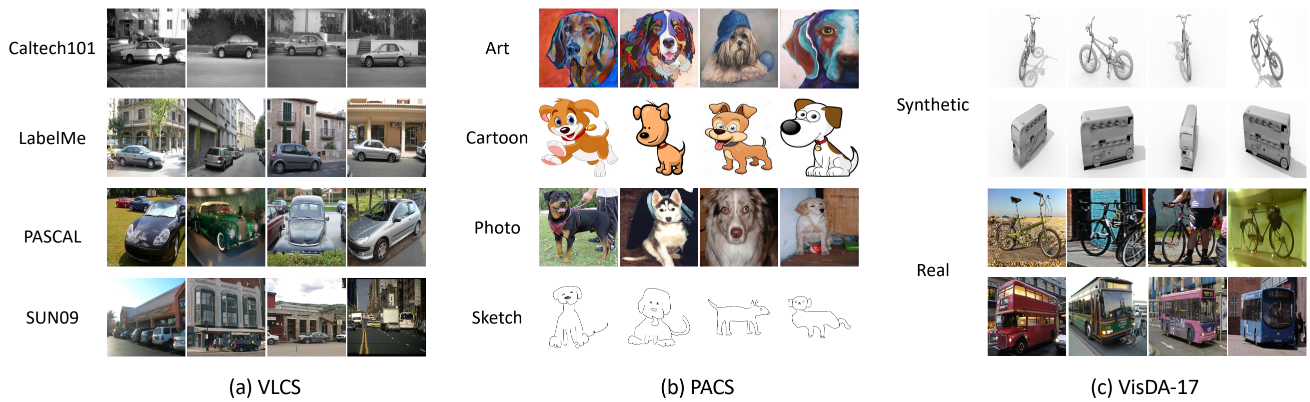



Fig. 2. Example images from three DG benchmarks for object recognition. In (a), dataset-specific biases are very clear, which are mainly caused by changes in environment/scene and viewpoint [33]. In (b), image style changes are the main cause of domain shift. In (c), the domain shift between synthetic 3D models and real-world objects mainly takes place in color, texture, background and occlusion.

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 [111], 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 [112] to real image data like Cityscapes [113], 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 [57], [136], [137], [138], [139] 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 [37], [69], [70] 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 [24], [25], [26] have investigated cross-dataset re-ID between Market1501 [137] and DukeMTMC-reID [117], [118]. A few works [69], [70] have attempted the multiple-to-one setting, e.g., using Market1501, CUHK03 [140] and MSMT17 [141] for training and DukeMTMC-reID for testing.

**Face Recognition** has seen significant advances in recent years, mainly attributed to deep learning technologies [52], [142], [143]. However, studies have shown that deep face recognition models trained even on large-scale datasets, such as MS-Celeb-1M [119], 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 [122], [126], [127], large variations in illumination/occlusion/head pose [121], [124], [125], or drastically different viewpoints [123]. This has motivated the research on learning universal face representations [31].

**Face Anti-Spoofing** aims to prevent face recognition systems from being attacked by using fake faces [144], 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 [39]. To make face anti-spoofing systems more robust and secure, researchers have been working on designing effective DG algorithms [39], [145]. 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 [128], Oulu-NPU [129] and MSU-MFSD [130] as the sources and Idiap Replay-Attack [131] as the target.

## 3 Methodologies: A Survey

In this section, we conduct a comprehensive review into existing domain generalization (DG) methods. Specifically, we categorize them into 11 groups (from § 3.1 to § 3.11), and discuss in detail the methodologies and motivations behind for each group, aiming to sketch a clear picture for readers to understand how the DG problem has been tackled in the last decade since 2011. See Table 4 for an overview.

### 3.1 Domain Alignment

Most existing DG approaches belong to the category of domain alignment [17], [27], [28], [39], [47], [152], 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], [46],

TABLE 4
Categorization of domain generalization methods.

| | |
|---|---|
| **Domain Alignment** (§ 3.1) | |
| - Minimizing Moments | [17], [54], [146], [147], [148], [149] |
| - Minimizing Contrastive Loss | [47], [150], [151] |
| - Minimizing the KL Divergence | [152], [153] |
| - Minimizing Maximum Mean Discrepancy | [27] |
| - Domain-Adversarial Learning | [28], [39], [59], [62], [145], [154], [155], [156], [157], [158] |
| - Multi-Task Learning | [46] |
| **Meta-Learning** (§ 3.2) | [21], [22], [29], [32], [69], [70], [90], [91], [159], [160], [161] |
| **Data Augmentation** (§ 3.3) | |
| - Hand-Engineered Transformations | [30], [31], [63], [162], [163], [164], [165] |
| - Gradient-Based Augmentation | [23], [38], [43], [166] |
| - Model-Based Augmentation | [18], [24], [25], [132], [167], [168], [169], [170] |
| - Feature-Based Augmentation | [26], [171] |
| **Ensemble Learning** (§ 3.4) | |
| - Exemplar-SVMs | [45], [48], [172] |
| - Domain-Specific Neural Networks | [99], [133], [173], [174], [175] |
| - Domain-Specific Batch Normalization | [176], [177], [178] |
| - Weight Averaging | [179] |
| **Network Architecture Design** (§ 3.5) | |
| - Exploiting Instance Normalization | [37], [55], [56], [57], [180] |
| - Problem-Specific Modules | [53], [110], [181], [182] |
| **Self-Supervised Learning** (§ 3.6) | |
| - Single Pretext Task | [58], [60] |
| - Multiple Pretext Tasks | [76], [183] |
| **Learning Disentangled Representations** (§ 3.7) | |
| - Decomposition | [20], [36], [184], [185] |
| - Generative Modeling | [40], [186] |
| **Invariant Risk Minimization** (§ 3.8) | [187], [188], [189] |
| **Training Heuristics** (§ 3.9) | [53], [171] |
| **Side Information** (§ 3.10) | |
| - Attributes | [49] |
| - Semantic Segmentation Masks | [190] |
| **Transfer Learning** (§ 3.11) | [44], [80] |

action recognition [27], face anti-spoofing [39], [145], and medical imaging analysis [153], [157].

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 $\ell_2$ distance, $f$-divergences, or the more sophisticated Wasserstein distance [191]. 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

$$P(X, Y) = P(Y|X)P(X), \quad (1)$$
$$= P(X|Y)P(Y). \quad (2)$$

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], [54], [146]. This is usually achieved by imposing a distance minimization loss. In traditional subspace learning methods [54], the loss is applied to the features after some transformation functions; whereas in deep convolutional neural network (CNN) models [148], the loss is computed using the top-layer features. Such a way of only aligning marginal distributions is also called marginal transfer learning [192].

From a causal learning perspective [193], 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], [147], [149] 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. [147] 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. [149] relaxed the assumption made in [147] by removing the minimization constraint on marginal distributions and proposed several discrepancy measures to learn generalizable features.

Since the posterior $P(Y|X)$ is what we need at test time, Wang et al. [152] 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 [54] or a complex non-linear function modeled by deep neural networks [148]) with an objective of minimizing the moments of the transformed features between source domains, in terms of variance [17], [146] or both mean and variance [54], [147], [148], [149].

**Minimizing Contrastive Loss** is another option for reducing distribution mismatch [47], [150], [151], 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 $\ell_2$ [47] or softmax [151]). 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 [152], [153]. In [152], domain-agnostic posteriors within each class are aligned via the KL divergence. In [153], 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 [194] 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 [195].

**Domain-Adversarial Learning** Different from explicit distance measures like the MMD, adversarial learning [195] formulates the distribution minimization problem through a minimax two-player game. Initially proposed by Goodfellow et al. [195], 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 [195] 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], [196], [197], [198].

In DG, adversarial learning is performed between source domains to learn source domain-agnostic features that are expected to work in novel domains [28], [39], [59], [154], [155]. Simply speaking, the learning objective is to make features confuse a domain discriminator, which can be implemented as a multi-class domain discriminator [62], [156], [157], or a binary domain discriminator in a per-domain basis [28], [39]. 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) [145].

To enhance domain alignment, researchers have also combined domain-adversarial learning with explicit distance measures like moments minimization [154], or with some regularization constraints such as entropy [158].

**Multi-Task Learning** has also been explored for domain alignment. Different from directly minimizing the distribution divergence, MTL facilitates the learning of generic features by parameter sharing [71]. This is easy to understand: in order to simultaneously deal with different tasks the features have to be generic enough. In [46], the authors proposed a denoising autoencoder architecture 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], [85], [199], [200], 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.

## 3.2 Meta-Learning

Meta-learning has been a fast growing area with applications to many machine learning and computer vision problems [21], [32], [69], [91], [201]. Also known as learning-to-learning, meta-learning aims to learn from episodes sampled from related tasks to benefit future learning (see [202] for a comprehensive survey on meta-learning). The meta-learning paper most related to DG is MAML [201], 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

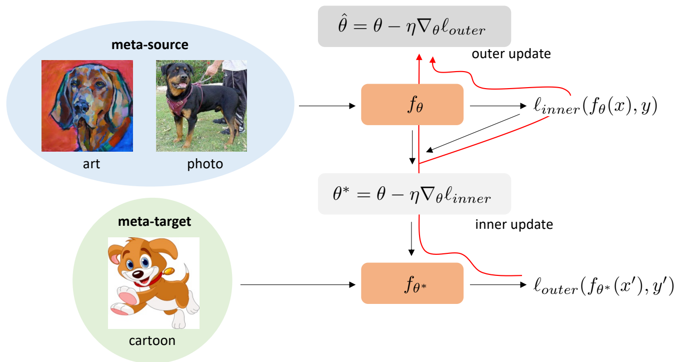



Fig. 3. A commonly used meta-learning paradigm [21] in domain generalization. The source domains (i.e. art, photo and cartoon from PACS [20]) are divided into disjoint meta-source and meta-target domains. The outer learning, which simulates domain shift using the meta-target data, back-propagates the 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.

through the update of the base model, thus posing issues on efficiency and memory consumption for large neural network models [202]. In [201], 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. 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 [22], [29], [69], [70], [90], [91], [159], [160], [161] followed the learning paradigm proposed in [21]—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 3 for a graphical representation.

**Meta-Representation** is a term defined in [202] to represent the model parameters that are meta-learned. Most deep learning methods meta-learned the entire neural network models [21], [90], [159]. Balaji et al. [22] instead proposed to meta-learn the regularization parameters. In [160], a stochastic neural network is meta-learned to handle uncertainty. In [91], 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 [69], [70], [161] 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 [201], object detection [203] and image generation [204]. 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 [25], [179], both meta-learning and domain alignment methods are under-performed 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 over-fitting and improve generalization [205], 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.

Various ways of designing $A(\cdot)$ have been proposed in the literature. For instance, $A(\cdot)$ can be hand-engineered, such as using traditional image transformations [30]; or be modeled with deep neural networks [25] and end-to-end learned. Below, based on the design of $A(\cdot)$, we classify the existing data augmentation methods into four groups and discuss each group in more detail. See Figure 4 for an overview.

**Hand-Engineered Transformations** The most common way is to use a single or stack multiple traditional image transformations in $A(\cdot)$, e.g., random flip, rotation and color augmentation. See Figure 5 for some examples. However, 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 5. 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 [30] 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 example, in [31] the authors addressed the universal feature learning problem in face recognition by synthesizing meaningful variations such as lowering image resolution, adding occlusions and changing head poses.

It is worth noting that traditional image transformations have been shown very effective in dealing with domain

3. These terms are defined in [202].

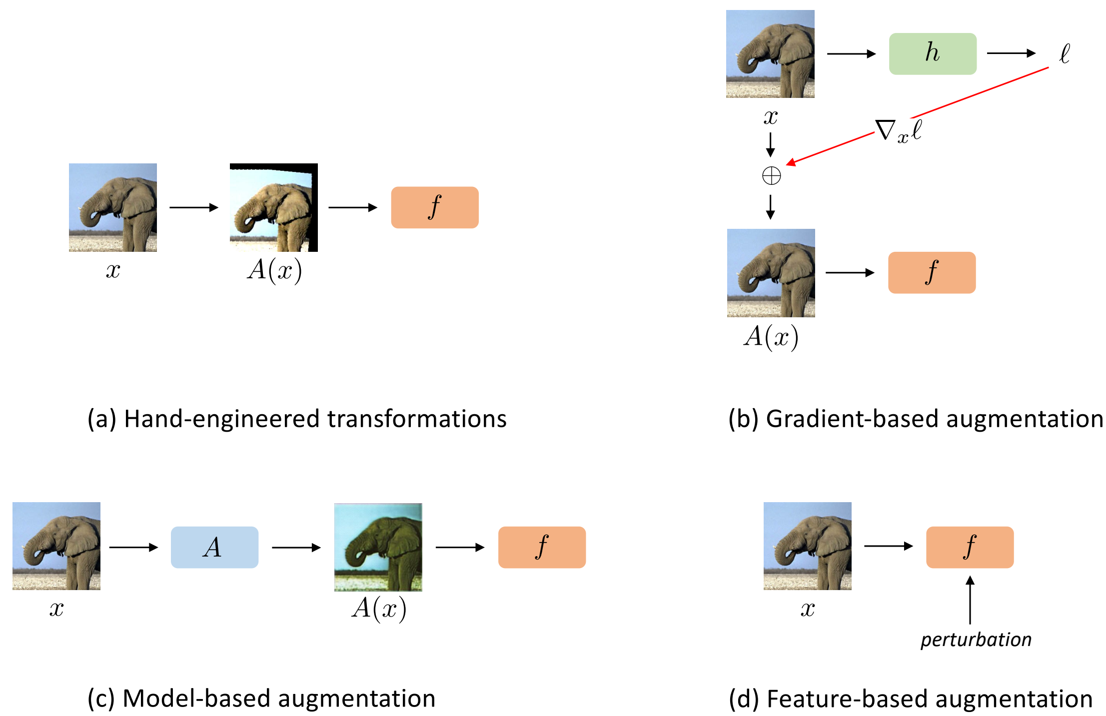



Fig. 4. Four groups of data augmentation methods based on the design of the transformation function $A(\cdot)$. In general, (a) mainly relies on traditional image transformations like translation and color augmentation; (b) perturbs the input using adversarial gradients (with hard-to-perceive visual effect [135]), which have mostly been generated by maximizing a domain/label classification loss with respect to the input (i.e. $h$ can either be a domain classifier or the label classifier $f$); (c) models $A(\cdot)$ using CNNs (the image $A(x)$ is from [25]); (d) injects perturbation into the intermediate features in the task model $f$.

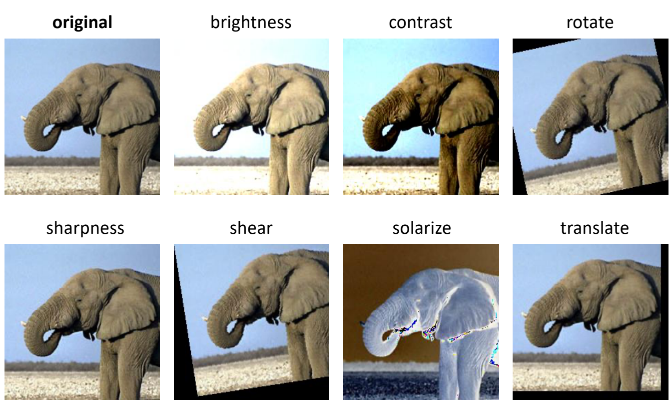



Fig. 5. Common image transformations used as data augmentation in domain generalization [30], [162], [163], [164].

shift in medical images [162], [163], [164]. 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 the label to be changed, 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.

In addition to traditional image transformations, mixing instances at the image level has also been explored, which is largely inspired by Mixup [206], a popular regularization method that mixes two instances at both the image and label space with convex combination. In [165], Mixup is applied to samples from different source domains. In [63], AugMix is introduced to combine a raw image with its different augmented versions.

**Gradient-Based Augmentation** mostly uses adversarial gradients, which are largely inspired by research on adversarial attacks [135], [207]. The key idea is to perturb images using sign-flipped gradients back-propagated from a domain classifier [23] or a label classifier [38], [43], [166]. In the former, the gradients aim to make the images indistinguishable to the domain classifier, i.e. forcing them to shift away from the original source distribution. While the latter is usually used in single-source DG for generating "challenging" images that make the label classifier difficult to judge (this is often formulated as a worst-case problem [207]). Since adversarial gradient-based perturbation is designed to be visually imperceptible on purpose [135], these methods are often criticized for not being able to simulate real-world domain shift, which is much more complicated than salt-and-pepper noise [24]. Moreover, the computational cost is typically doubled in these methods because the forward and backward passes need to be performed twice (one for computing the adversarial gradients and the other for updating the model), which will pose a serious issue for large neural network models.

**Model-Based Augmentation** This family of methods models $A(\cdot)$ using CNNs, which is in general more powerful than gradient-based augmentation because CNNs can model more sophisticated domain shift like image style changes [25]. Based on the learning objectives, we can further classify model-based augmentation methods into the following three categories. 1) *Non-learning*: Currently, only RandConv [167] falls into this group. RandConv is based on a single convolution layer with weights randomly sampled from a Gaussian distribution at each iteration. Since no learning is performed, the transformed images do not contain meaningful variations but random color distortions, and are best to be mixed with the original images before feeding to a neural network. 2) *Learning-based within-source image synthesis*: The main idea is to use a style transfer model (essentially performing domain translation), such as CycleGAN [208] used in [168] or AdaIN [209] used in [132], [169], to map images from one source domain to the other. In this way, the diversity of source data is increased as each instance now can appear in different styles (domains), but the number of source domains remains the same. 3) *Learning-based outside-source image synthesis*: This group of methods aims to synthesize new domains to diversify source domain distributions [18], [24], [25], [170]. Also based on style transfer, Yue et al. [18] relied on external styles to transform source images onto novel styles. In [24], [170], domain-agnostic images are generated by maximizing the domain classification loss with respect to the image generator. In [25], new domains are synthesized by maximizing between $A(x)$ and $x$ the domain difference measured by optimal transport [191].

A drawback in these learnable image synthesis methods is that the new images do not look significantly different from other source domain images [25]. This is because the source-source domain shift is highly correlated between different source domains. For instance, on PACS [20] image style changes dominate in domain shift; on Rotated MNIST [46] rotation is the only cause of domain shift. As such, a generation model learned on PACS cannot simulate novel domain shift like viewpoint changes; likewise, a model learned on Rotated MNIST will be unable to synthesize new image styles.

In particular, the aforementioned issue is caused by datasets themselves—the domain shift is too homogeneous, either between source domains or from source to target domains. Such a setup cannot fully reflect the real-world scenarios where the target domain shift can be significantly different from the source domain shift, e.g., image style changes for the source domain shift while viewpoint changes for the target domain shift. We call this setup *heterogeneous domain shift* and will discuss it as a potential direction for future work in § 4.3.

**Feature-Based Augmentation** Most model-based augmentation methods need to train heavy image translation models (except RandConv [167]) to augment data at the image level. However, studies in style transfer [209], [210] have shown that CNN feature statistics essentially capture image style. Therefore, style augmentation can be achieved by mixing CNN feature statistics between instances from different domains [26], [171], which is much more efficient in both implementation and training than image-level style augmentation. In the future, it would be interesting to combine image- and feature-level data augmentation.

### 3.4 Ensemble Learning

As an extensively studied topic in machine learning research, ensemble learning [211] 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], [212]. In DG, ensemble learning has also been explored, with examples including using traditional ensemble methods like exemplar-SVMs [45] and training domain-specific models [99]. Below we talk about in detail how the idea of ensemble learning has been extended in DG.

**Exemplar-SVMs** are a collection of SVM classifiers, each learned using one positive instance and all negative instances [213]. As the ensemble of such exemplar SVMs have shown excellent generalization performance on the object detection task in [213], Xu et al. [45] 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 [48], [172] 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 [99], [173]. Rather than learning an independent CNN for each source domain [173], it is more efficient, and makes more sense as well, to share between source domains some shallow layers [99], which capture generic features [78]. Another question is how to compute the prediction. One can simply use the ensemble prediction averaged over all individuals with equal weights (e.g., [99], [174]). 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 [175]. Also, the weights can be used to determine the most confident candidate whose output will serve for final prediction [133].

**Domain-Specific Batch Normalization** In batch normalization (BN) [214], 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. Seo et al. [176] proposed domain-specific BNs, one for each source domain to collect 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 [177] for dealing with MRI segmentation. In [178], 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 [215]. 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 [179], 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 Network Architecture Design

Besides developing effective learning algorithms, researchers have also devoted efforts to designing novel neural network architectures for generalizable feature learning.

**Exploiting Instance Normalization** Instance normalization (IN) [216] has been shown effective in removing instance-specific styles in style transfer [209], [210]. Different from BN [214] that computes batch-level statistics, IN computes feature statistics for each instance independently. Since domains are closely related to image style [26], it makes sense to apply IN to removing domain-related information in feature learning with CNNs.

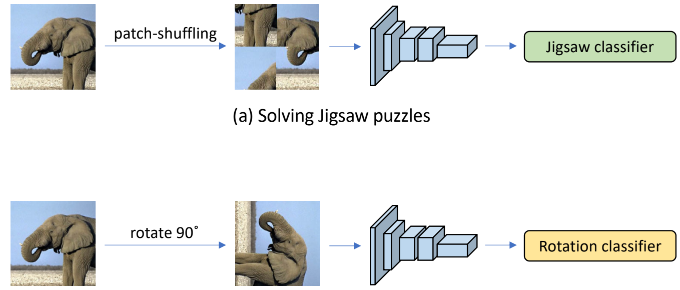
Pan et al. [55] suggested that the divergence between images of different domains mainly occurs in shallow CNN layers where IN should be inserted. They justified their design in the cross-dataset semantic segmentation task (Cityscapes [113]⇔GTA5 [112]). Similarly, Nam and Kim [56] combined IN's output with BN's output in a convex way. In person re-identification (re-ID), domain shift mainly affects image contrast and style, and therefore, can be mitigated by using IN-equipped CNNs [57], [180].

A challenge with integrating IN with CNNs is that there is no design principle to determine, for a specific problem, where to insert IN to a CNN architecture and how many IN layers should be used. Only knowing to insert IN to shallow CNN layers is too vague as there is no clear definition as to what shallow layers are for a CNN. To circumvent this problem, one can resort to automatic neural architecture search [217], [218], as attempted in [37].

**Problem-Specific Modules** It is beneficial to customize neural network modules for specific problems. In [110], video recognition is tackled by a pyramid CNN model which applies attention to multiple different time scales. In [181], person re-ID is addressed by a hypernetwork-based CNN, which synthesizes classifier weights for each new incoming instance. In [53], a novel block based on gray-level co-occurrence matrix is designed to detect superficial features related to textures. In [182], dynamic residual adapters based on a mixture-of-expert gating mechanism are proposed for latent domain learning.

### 3.6 Self-Supervised Learning

Self-supervised learning is often referred to as learning with free labels generated from data itself (see [221] 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 [219] or rotation degrees [220]. See Figure 6 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 [58].




(a) Solving Jigsaw puzzles

(b) Predicting rotation degrees

Fig. 6. Common pretext tasks used for self-supervised learning in domain generalization. One can use a single pretext task, like solving Jigsaw puzzles [219] or predicting rotations [220], or combine multiple pretext tasks in a multi-task learning fashion.

**Single Pretext Task** In addition to using the standard classification loss, Carlucci et al. [58] taught a neural network to solve the Jigsaw puzzles problem [219], hoping that the network can learn regularities that are more generalizable across domains. Similarly, Wang et al. [60] used the Jigsaw solving task as an intrinsic supervision, together with an extrinsic supervision implemented using metric learning.

**Multiple Pretext Tasks** It is also possible to combine multiple pretext tasks. In [183], the authors combined two pretext tasks, namely solving Jigsaw puzzles and predicting rotations. In [76], three pretext tasks are combined, namely reconstructing the Gabor filter's response, predicting rotations, and predicting feature cluster assignments [222]. Overall, using multiple pretext tasks gives a better performance than using a single pretext task, as shown in [183].

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 [223], [224] 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.

### 3.7 Learning Disentangled Representations

Instead of forcing the entire model or features to be domain-invariant, which is of course very 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 [20], [36], [184], [185] or generative modeling [40], [186].

**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. [36] 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 [20]. One can also design domain-specific modules such as in [184] 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 [185].

**Generative Modeling** has been a powerful tool for learning disentangled representations [225]. In [186], a variational autoencoder (VAE) is utilized to learn three independent latent subspaces for class, domain and object, respectively. In [40], two separate encoders are learned in an adversarial way to capture identity and domain information respectively for cross-domain face anti-spoofing.

### 3.8 Invariant Risk Minimization

The motivation behind this category of methods is to remove spurious correlations (i.e. dataset-specific biases [33]) in a representation [187]. For example, we do not want to learn a cow classifier that can only recognize cows in pastures but fail when cows are in novel backgrounds like beaches. The pioneer work done by Arjovsky et al. [187] introduced invariant risk minimization (IRM), a learning paradigm that combines the standard empirical risk minimization (ERM) [4] with a gradient norm penalty over a dummy classifier, both computed within each source domain. IRM can be interpreted as imposing a regularization term on the representation learning model, to ensure that the learned representation can lead to a minimal classification error over all source domains. The follow-up works [188], [189] are basically different variants of IRM. In [188], a strict equality between the risks in different domains is enforced. In [189], the variance of the gradient norm across source domains is minimized.

However, these methods have mainly been justified on toy datasets. On larger DG datasets like PACS [20], they still cannot compete with specially designed DG approaches, as shown in [115], [188]. Since IRM is quite scalable (it is not limited to any architectures), it is worth extending the research in IRM by further improving the formulation and conducting larger-scale experiments.

### 3.9 Training Heuristics

Some approaches are based on training heuristics. Wang et al. [53] 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. [171] iteratively maksed out over-dominant features with large gradients, thus forcing the model to rely more on the remaining features. These methods are orthogonal to other DG approaches like domain alignment [39], [47] and data augmentation [25], [26], [30]. Therefore, one could potentially combine them to achieve better DG performance.

### 3.10 Side Information

Side information has been commonly used to boost the performance of a pattern recognition system. For example, in computer vision, depth information obtained from RGB-D sensors can be used alongside RGB images to improve the performance of, e.g., generic object detection [226] or human detection [227]. In DG, attribute labels [49] and object segmentation masks [190] have been explored to facilitate the learning of domain-generalizable knowledge.

**Attributes** Why use attributes? Intuitively, attributes are more generalizable because they capture mid- to low-level visual cues like colors, shapes and stripes, which are commonly shared among different objects and generally less sensitive to domain biases [49]. Notably, attributes have been widely used in zero-shot learning to recognize unseen classes [81], [83]. 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 [228], which are more likely to capture domain biases and hence less transferable between tasks [78].

However, attribute learning has so far been under-studied in DG. Only an early work done by Gan et al. [49] attempted this topic, where visual feature learning was formulated as an attribute detection problem using kernel methods. None of the existing work has fused attribute learning with modern deep feature learning for DG, which could be an interesting topic for future research.

**Semantic Segmentation Masks** Focusing on explainability, Zunino et al. [190] identified that CNNs trained on synthetic images often attend to class-irrelevant backgrounds in real images. Such an observation has a connection to the aforementioned spurious correlations [187] caused by dataset biases. As a countermeasure, semantic segmentation masks were used in [190] as an auxiliary supervision to force CNNs to focus on object-related pixels. This paper could inspire more future work solving DG from the perspective of explainable AI models.

### 3.11 Transfer Learning

A couple of recent works [44], [80] have been focused on the transfer learning perspective when designing DG methods for synthetic-to-real applications (see examples from VisDA-17 [100] in Figure 2(c)). Given a model pre-trained on large real datasets like ImageNet [8], 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) [229]. In particular, a technique used in [44] was also borrowed from LwF [229], i.e. minimizing the divergence between the new model's output and the old model's output to avoid erasing the pre-trained knowledge.

As shown in [44], learning rate is a key design choice when fine-tuning a model on synthetic data. A large learning rate can lead to the pre-trained weights completely destroyed while a small learning rate might hamper the learning of task-related features. Chen et al. [44] overcame this issue by automating the learning rate selection process using reinforcement learning. Moreover, increasing the feature diversity (e.g., using contrastive learning [80]) has been shown helpful for preventing features from over-fitting synthetic images.

## 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 talk about potential directions from three perspectives: *model* (§ 4.1), *learning* (§ 4.2), and *benchmarks* (§ 4.3).

### 4.1 Model

Model architectures matter as they directly impact on how well the learned representations will be. However, most existing DG methods have been focused on the learning part (see Table 4) while paying less attention to designing effective model architectures for DG. Indeed, the ImageNet-winning architectures like ResNet [5] have been widely adopted as a backbone model on several DG datasets, such as PACS [20] and DomainNet [98]. This is probably due to the stereotype, i.e. a model architecture that can achieve high accuracy on ImageNet [8] should also be able to produce domain-generalizable features. In fact, such an over-simplified assumption does not hold—many studies have shown that specially designed model architectures work better than the ImageNet-winning architectures in DG [37], [53], [57], [110], [181], [182].

Also, by looking at the leaderboard of commonly used DG benchmarks, we can observe that the performance improvements obtained by recently published methods are diminishing, e.g., on PACS the results based on ResNet18 are saturating at around 85% accuracy [115]. Therefore, to further push the limit on the DG performance, we call for more future work to contribute to novel model architecture designs.

Undoubtedly, when it comes to representation learning, neural networks will still be the mainstream architecture in the foreseeable future. Therefore, in this section we focus on potential design ideas for building neural network-based model architectures.

**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 [230], e.g., with weights conditioned on the input [231]. 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 [231] and conditional convolutions [232] 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 [214], [216], [233] have been a core building block in contemporary neural networks. Following [234], a general formulation for different normalization layers can be written as $\gamma\frac{x-\mu}{\sigma}+\beta$, where $\mu$ and $\sigma$ denote mean and variance respectively; $\gamma$ and $\beta$ are learnable scaling and shift parameters respectively. Typically, $(\mu, \sigma)$ 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. $\gamma$ and $\beta$, 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 [235].

**Transformer** Initially developed as a language model, the Transformer architecture [236] has recently gained enormous interests in computer vision [237], [238], [239], [240]. From a technical point of view, Transformer departs from the widely used recurrent and convolutional architectures and is purely based on a feed-forward self-attention mechanism, which basically processes a sequence of feature vectors by comparing each vector with all vectors. When applied to the image domain, Transformer benefits from modeling long-range dependencies. Future work can explore the possibility of applying Transformer to DG tasks to see whether such a self-attention model offers any advantages over the mainstream CNNs.

### 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 [172]. 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., [26], [62], [182]), 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 [61] 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 [25], [26], the results still have much room for improvements.

**Learning Not To Learn 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 [241]. 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 [242]. 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 [20], 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 [243]. A potential solution is to model the underlying causal variables (e.g., by autoencoder [243]) 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 [244].

### 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 [245]. 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 [116] and DukeMTMC-reID [117], [118], but later another dataset comes in, e.g., CUHK03 [140], 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 [20] the source-source domain shift and the source-target domain shift are both related to image style changes; on Rotated MNIST [46] 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 taxonomy. 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] V. Vapnik, "Principles of risk minimization for learning theory," in *NeurIPS*, 1992.
[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] 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.

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

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

[21] ——, "Learning to generalize: Meta-learning for domain generalization," in *AAAI*, 2018.

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

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

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

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

[26] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," in *ICLR*, 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] Y. Li, Y. Yang, W. Zhou, and T. Hospedales, "Feature-critic networks for heterogeneous domain generalization," in *ICML*, 2019.

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

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

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

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

[34] 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.

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

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

[37] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Learning generalisable omni-scale representations for person re-identification," *arXiv preprint arXiv:1910.06827*, 2019.

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

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

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

[41] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *CVPR*, 2018.

[42] J. Wang, X. Zhu, S. Gong, and W. Li, "Transferable joint attribute-identity deep learning for unsupervised person re-identification," in *CVPR*, 2018.

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

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

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

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

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

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

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

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

[51] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

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

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

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

[55] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via ibn-net," in *ECCV*, 2018.

[56] H. Nam and H.-E. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," in *NeurIPS*, 2018.

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

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

[59] 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.

[60] 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.

[61] 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.

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

[63] 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.

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

[65] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[66] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, 2015.

[67] 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.

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

[69] 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," *arXiv preprint arXiv:2012.00417*, 2020.

[70] 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.

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

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

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

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

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

[76] 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.

[77] 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.

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

[79] 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.

[80] 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.

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

[82] 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.

[83] 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.

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

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

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

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

[88] 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.

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

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

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

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

[93] 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.

[94] 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.

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

[96] 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.

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

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

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

[100] 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.

[101] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[102] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video," in *CVPR*, 2017.

[103] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," *arXiv preprint arXiv:1306.5151*, 2013.

[104] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *TPAMI*, 2006.

[105] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *CVPR*, 2014.

[106] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *NN*, 2012.

[107] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *ICCVGIP*, 2008.

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

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

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

[111] 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.

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

[113] 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.

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

[115] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," in *ICLR*, 2021.

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

[117] 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.

[118] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *ICCV*, 2017.

[119] 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.

[120] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep., 2007.

[121] 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.

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

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

[124] 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.

[125] 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.

[126] 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.

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

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

[129] 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.

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

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

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

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

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

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

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

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

[138] 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.

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

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

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

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

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

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

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

[146] 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.

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

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

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

[150] 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.

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

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

[153] 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.

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

[155] 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.

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

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

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

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

[160] 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.

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

[162] 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.

[163] 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.

[164] 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.

[165] Y. Wang, H. Li, and A. C. Kot, "Heterogeneous domain generalization via domain mixup," in *ICASSP*, 2020.

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

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

[168] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "Multi-component image translation for deep domain generalization," in *WACV*, 2019.

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

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

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

[172] L. Niu, W. Li, and D. Xu, "Visual recognition by learning from web data: A weakly supervised domain generalization approach," in *CVPR*, 2015.

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

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

[175] 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.

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

[177] 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.

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

[179] 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.

[180] X. Jin, C. Lan, W. Zeng, Z. Chen, and L. Zhang, "Style normalization and restitution for generalizable person re-identification," in *CVPR*, 2020.

[181] J. Song, Y. Yang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Generalizable person re-identification by domain-invariant mapping network," in *CVPR*, 2019.

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

[183] 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.

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

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

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

[187] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.

[188] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, R. L. Priol, and A. Courville, "Out-of-distribution generalization via risk extrapolation (rex)," *arXiv preprint arXiv:2003.00688*, 2020.

[189] A. Bellot and M. van der Schaar, "Accounting for unobserved confounding in domain generalization," *arXiv preprint arXiv:2007.10653*, 2021.

[190] 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.

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

[192] G. Blanchard, A. A. Deshmukh, U. Dogan, G. Lee, and C. Scott, "Domain generalization by marginal transfer learning," *JMLR*, 2021.

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

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

[195] 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.

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

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

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

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

[200] 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.

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

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

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

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

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

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

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

[208] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.

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

[210] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *ICLR*, 2017.

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

[212] 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.

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

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

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

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

[217] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *ICLR*, 2019.

[218] S. Xie, H. Zheng, C. Liu, and L. Lin, "Snas: stochastic neural architecture search," in *ICLR*, 2019.

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

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

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

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

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

[224] 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.

[225] 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.

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

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

[228] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.

[229] Z. Li and D. Hoiem, "Learning without forgetting," *TPAMI*, 2017.

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

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

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

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

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

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

[236] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NeurIPS*, 2017.

[237] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *ECCV*, 2020.

[238] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *ICLR*, 2021.

[239] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” in *ICLR*, 2021.

[240] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, “Bottleneck transformers for visual recognition,” *arXiv preprint arXiv:2101.11605*, 2021.

[241] 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.

[242] 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.

[243] 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.

[244] 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.

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