Meta-causal Learning for Single Domain Generalization

Jin Chen[1*], Zhi Gao[1*], Xinxiao Wu[1,2†], Jiebo Luo[3]
[1]Beijing Key Laboratory of Intelligent Information Technology,
School of Computer Science & Technology, Beijing Institute of Technology, China
[2]Guangdong Laboratory of Machine Perception and Intelligent Computing,
Shenzhen MSU-BIT University, China
[3]Department of Computer Science, University of Rochester, Rochester NY 14627, USA
{chen_jin,gaozhi_2017,wuxinxiao}@bit.edu.cn, jluo@cs.rochester.edu

Abstract

Single domain generalization aims to learn a model from a single training domain (source domain) and apply it to multiple unseen test domains (target domains). Existing methods focus on expanding the distribution of the training domain to cover the target domains, but without estimating the domain shift between the source and target domains. In this paper, we propose a new learning paradigm, namely simulate-analyze-reduce*, which first simulates the domain shift by building an auxiliary domain as the target domain, then learns to analyze the causes of domain shift, and finally learns to reduce the domain shift for model adaptation. Under this paradigm, we propose a meta-causal learning method to learn meta-knowledge, that is, how to infer the causes of domain shift between the auxiliary and source domains during training. We use the meta-knowledge to analyze the shift between the target and source domains during testing. Specifically, we perform multiple transformations on source data to generate the auxiliary domain, perform counterfactual inference to learn to discover the causal factors of the shift between the auxiliary and source domains, and incorporate the inferred causality into factor-aware domain alignments. Extensive experiments on several benchmarks of image classification show the effectiveness of our method.*

1. Introduction

Single domain generalization [28] aims to generalize a model trained using one training domain (source domain) into multiple unseen test domains (target domains). Since only one source domain is given and the target domains are out-of-distribution and unavailable during training, single domain generalization is a challenging task and attracts increasing interests. Existing works have made considerable successes through expanding the distribution of the source domain by data augmentation [19, 28, 34] or learning adaptive data normalization [8] typically. However, such successes have been achieved without explicitly considering the domain shift between the source and target domains, which limits the generalization performance of model in real-world scenarios.

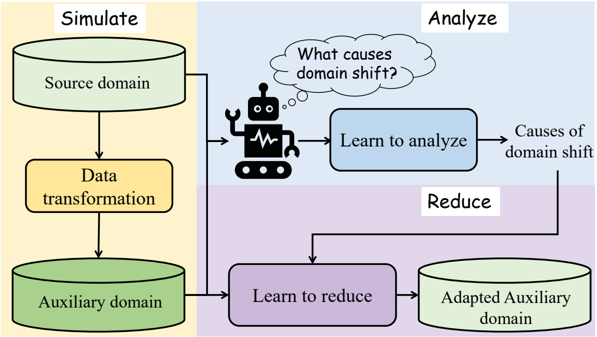

Figure 1. Illustration of the *simulate-analyze-reduce* paradigm. In this paradigm, we first simulate the domain shift by constructing an auxiliary domain as the unseen target domain, then learn to analyze the domain shift, and finally learn to reduce the domain shift based on inferred causes.

In this paper, we propose a new learning paradigm, namely *simulate-analyze-reduce*, to address single domain generalization by enabling the model to analyze the real domain shift between the source domain and unseen target domain. This new paradigm is shown in Figure 1. We first build an auxiliary domain as the target domain to simulate the real domain shift between the source and target domains, since the target data is unavailable during training. We then learn to analyze the intrinsic causal factors of the domain shift to facilitate the subsequent model adapta-

[*] Jin Chen and Zhi Gao are co-first authors.
[†] Corresponding author: Xinxiao Wu.

tion. Finally, we learn to reduce the domain shift with its inferred causes.

Under this paradigm, we propose a meta-causal learning method to learn the meta-knowledge about how to infer the causes of the simulated domain shift between the auxiliary and source domains via causal inference in training, and then apply the meta-knowledge to analyze the real domain shift between the target and source domains during testing, through which the source and given target domains are adaptively aligned. Specifically, we perform multiple transformations on source data to generate an auxiliary domain with great diversity. Then we build a causal graph to represent the dependency among data, variant factors, semantic concepts and category labels, and conduct counterfactual inference over the causal graph to exploit the intrinsic causality of the simulated domain shift between the auxiliary and source domains. For each sample in the auxiliary domain, we construct counterfactual scenes by intervening variant factors to infer their causal effects on the category prediction, and these inferred causal effects of variant factors can be regarded as the causes of domain shift. To reduce the domain shift, we propose a factor-aware domain alignment by learning and integrating multiple feature mappings, where an effect-to-weight network is designed to convert the causal effects of variant factors into the weights of feature mappings.

During testing, the distribution discrepancy between the input target sample and the source domain is analyzed and reduced by applying the learnt meta-knowledge, *i.e.,* inferring the causal effects of variant factors and incorporating them into the factor-aware domain alignment. In summary, the main contributions of this paper are as follows:

- We propose a novel learning paradigm, *simulate-analyze-reduce*, for single domain generalization. This paradigm empowers the model with the ability to estimate the domain shift between the source domain and unseen target domains, thus boosting the model adaptation across different domains.
- We propose a meta-causal learning method based on counterfactual inference to learn the meta-knowledge about analyzing the intrinsic causality of domain shift, thus facilitating the reduction of domain shift.
- Our method achieves the state-of-the-art results on several benchmarks of image classification, especially on the more challenging tasks with a large domain shift, clearly demonstrating the effectiveness of our method.

2. Related Work

2.1. Domain Generalization

Domain generalization focuses on generalizing a model learned from multiple source domains to the unseen target domain. The key difference between domain adaptation and domain generalization is that during training, domain adaptation leverages unlabelled target data while domain generalization has no access to the target domain. Existing domain generalization methods can be roughly divided into two categories: learning domain-invariant feature representation from multiple source domains [7, 10, 24, 25, 32] and generating diverse more samples via data augmentation [2, 29, 31, 39].

Recently, single domain generalization [28] has attracted growing attention, where only one source domain is available during training and the model is evaluated on multiple unseen target domains. A rich line of works employ data augmentation for generating out-of-domain samples to expand the distribution of the source domain [19, 28, 34]. Qiao *et al.* [28] propose meta-learning based adversarial domain augmentation to generate samples. Li *et al.* [19] propose a progressive domain expansion network to generate multiple domains progressively via simulating various photometric and geometric transforms by style transfer based generators. Wang *et al.* [34] propose a style-complement module to generate diverse images with different styles. Fan *et al.* [8] use data normalization for single domain generalization, where an adaptive normalization scheme is learned to be incorporated with adversarial domain augmentation to enhance the generalization of the model.

The aforementioned methods aim to generate the source data distribution as diverse as possible to cover unseen target domains. When the expanded source distribution does not approximate the target distribution, the performance may significantly degrade, since there still exists a domain gap between the source and target domains. To address this problem, our method learns to analyze and reduce the domain shift by building an auxiliary domain during training.

2.2. Causality for Domain Generalization

Several recent methods exploit causality to learn domain-invariant semantic representation for domain generalization [20–22]. Considering the cross-domain invariance of the causality between semantic factors and predictions, Liu *et al.* [20] propose a causal semantic generative model to remove the domain-specific correlation between semantic factors and variant factors, and thus make the prediction affected only by the semantic factors. Assuming that images of the same object across domains should have the same representation, Mahajan *et al.* [22] use the cross-domain invariance of the causality between objects and feature representations to capture the within-class variation for domain generalization. Lv *et al.* [21] introduce causal inference to extract causal factors that are invariant across domains in order to learn invariant feature representation.

From a different perspective, our method focuses on uti-

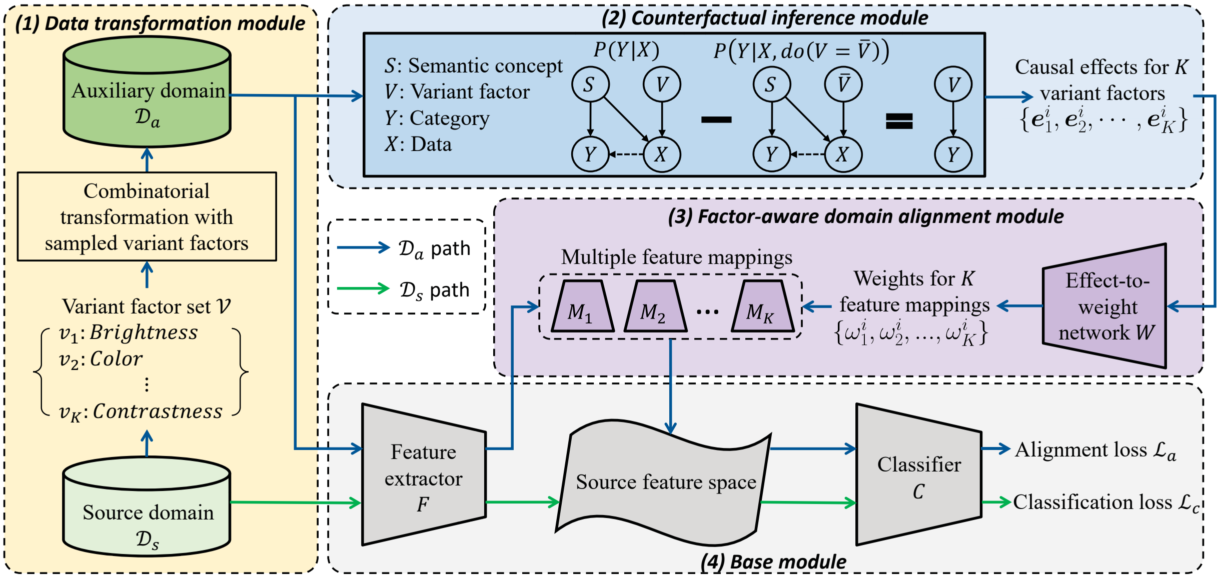
(1) Data transformation module
Auxiliary domain $\mathcal{D}_a$
Combinatorial transformation with sampled variant factors
Variant factor set $\mathcal{V}$
v_1: *Brightness*
v_2: *Color*
⋮
v_K: *Contrastness*
Source domain $\mathcal{D}_s$
(2) Counterfactual inference module
S: Semantic concept
V: Variant factor
Y: Category
X: Data
$P(Y|X)$
$P(Y|X, do(V = \bar{V}))$
S
V
Y
X
$\bar{V}$
−
=
Causal effects for K variant factors $\{\boldsymbol{e}_1^i, \boldsymbol{e}_2^i, \cdots, \boldsymbol{e}_K^i\}$
$\mathcal{D}_a$ path
$\mathcal{D}_s$ path
(3) Factor-aware domain alignment module
Multiple feature mappings
M_1
M_2
⋯
M_K
Weights for K feature mappings $\{\omega_1^i, \omega_2^i, ..., \omega_K^i\}$
Effect-to-weight network W
Feature extractor F
Source feature space
Classifier C
Alignment loss $\mathcal{L}_a$
Classification loss $\mathcal{L}_c$
(4) Base module

Figure 2. Overview of the proposed meta-causal learning method. (1) A data transformation module simulates the domain shift via generating an auxiliary domain $\mathcal{D}_a$. (2) A counterfactual inference module analyzes the domain shift by inferring the causal effects of V on Y. "–" denotes comparing the values of Y before and after do-operation. The edge $X \rightarrow Y$ is a dashed arrow to represent the classification model $P(Y|X)$. (3) A factor-aware domain alignment module reduces the domain shift via multiple feature mappings according to weights learned by an effect-to-weight network. (4) A base module is used for feature extraction and classification.

lizing causal inference to discover the intrinsic causes of domain shift by constructing counterfactual scenes over the learnt causal graph, which facilitates the reduction of domain shift and in turn benefits cross-domain adaptation.

3. Method

3.1. Problem Definition

For single domain generalization, during training, we are given a labeled source domain $\mathcal{D}_s = \{(\boldsymbol{x}_i^s, y_i^s)|_{i=1}^{N_s}\}$ drawn from the distribution P_s, where $\boldsymbol{x}_i^s$ is the i-th source sample with its category label $y_i^s \in \mathcal{Y}$. During testing, the learnt model is applied to multiple unseen target domains $\mathcal{D}_t = \{\mathcal{D}_t^j\}_{j=1}^J$, where $\mathcal{D}_t^j$ is the j-th target domain drawn from the distribution P_t^j, and $P_t^j \neq P_s$.

To bridge the domain gap between the source domain and multiple unseen target domains, we propose a new learning paradigm, called *simulate-analyze-reduce*, which starts with simulating the domain shift between the source and target domains, then learns to analyze the domain shift, and finally learns to reduce the domain shift. Under this paradigm, we propose a meta-causal learning method that has four components: a data transformation module to generate auxiliary domains as target domains, a counterfactual inference module to discover the causes of the domain shift, a factor-aware domain alignment module to reduce the domain shift, and a base module for feature extraction and classification, as illustrated in Figure 2.

3.2. Data Transformation for Domain Shift Simulation

To simulate the real domain shift between the source and target domains, we generate an auxiliary domain $\mathcal{D}_a$ as the unseen target domain by performing transformations on source data. The domain shift is actually the data distribution discrepancy between the source and auxiliary domains, and is usually caused by the variations of extrinsic attributes of data, independent of intrinsic semantics. Taking the image data for example, the domain shift is mainly caused by the variations of visual attributes, such as the variations of brightness, viewpoint, and color. Therefore, to make the simulated domain shift as realistic as possible, we formulate the extrinsic attributes as variant factors, and design data transformations according to the variant factors.

We define a set of variant factors, denoted as $\mathcal{V} = \{v_1, v_2, ..., v_K\}$, where v_k denotes the k-th variant factor. Each variant factor corresponds to a data transformation function that aims to generate new data by making changes on the corresponding extrinsic attribute, denoted as $G_{v_k}(\boldsymbol{x}; \theta_{v_k})$, where $\boldsymbol{x}$ is an input sample, and $\theta_{v_k} \in [g_{min}^k, g_{max}^k]$ represents the degree parameter to control the magnitude of transformation with the scale range $[g_{min}^k, g_{max}^k]$.

In real-world scenarios, the complex domain shift is often caused by combinatorial multiple variant factors and accordingly we design a sampling strategy to enable the combinatorial data transformation. Given a source sample $\boldsymbol{x}_i^s$, we randomly sample N_v^i variant factors from $\mathcal{V}$ to form a factor subset $\mathcal{V}_i = \{v_1^i, v_2^i, \cdots, v_{N_v^i}^i\}$. Then, a corresponding auxiliary sample $\boldsymbol{x}_i^a$ is generated by

$$\boldsymbol{x}_i^a = G_{v^i_{N_v^i}}\Big(\cdots G_{v_2^i}\big(G_{v_1^i}(\boldsymbol{x}_i^s;\theta_{v_1^i});\theta_{v_2^i}\big)\cdots;\theta_{v^i_{N_v^i}}\Big), \tag{1}$$

where the transformation degree parameter $\theta_{v_k^i}$ is randomly selected from its range scale. In this way, the domain shift is generated as diverse as possible to approximate the real domain shift.

3.3. Counterfactual Inference for Domain Shift Analysis

After simulating the domain shift, we introduce counterfactual inference to analyze the domain shift. During training, we learn the meta-knowledge about how to infer the causes of data discrepancy between one auxiliary sample and the source domain, and during testing, we apply the learnt meta-knowledge to unseen target samples.

We build a causal graph to model the causal dependency among the input sample (node X), the variant factors (node V), the semantic concepts (node S), and the output category (node Y), as shown in Figure 2 (2). The semantic concepts denote the intrinsic attributes of data that are related to the category, and the variant factors denote the extrinsic attributes of data that are domain specific, independent of the intrinsic semantics. For example, when the input sample is an image of "zebra", the semantic concepts are like "four legs" and "black-white stripes", and the variant factors include brightness, viewpoint, and so on. The edge $S \rightarrow Y$ represents that the category of the input data is determined by the semantic concepts. The edges $S \rightarrow X$ and $V \rightarrow X$ represent that the semantic concepts and the variant factors together determine what the input sample looks like. Since the semantic concepts represent the intrinsic semantics and are invariant across different domains, ideally the cross-domain classification is only determined by the semantic concepts, which is implemented by learning the edge $S \rightarrow Y$, *i.e.,* estimating the conditional probability $P(Y|S)$. However, in reality, the semantic concepts are unobserved from the input samples, so the classification should be implemented by learning the edge $X \rightarrow Y$, *i.e.,* estimating the conditional probability $P(Y|X)$. The edge $X \rightarrow Y$ is a dashed arrow to represent the classification model $P(Y|X)$. Since the input sample node X is affected by the semantic concept node S and the variant factor node V together, the category node Y is also affected by the variant factor node V through edges $V \rightarrow X \rightarrow Y$. As the variant factors are domain-specific, their causal effects on the category lead to the domain shift. Hence, we infer the causal effects of the variant factors on the category prediction to discover the causes of the domain shift. That is to say, we infer the causal effects of the node V on the node Y as causes of domain shift, and then learn to reduce the domain shift based on its causes.

To infer the causal effects of the node V on the node Y, we first learn the edge $X \rightarrow Y$ using the source data $\mathcal{D}_s$ by a classification loss:

$$\mathcal{L}_c = \mathbb{E}_{(\boldsymbol{x}_i^s, y_i^s)\sim P_s}\left[-\sum_u \mathbb{I}_{u=y_i^s}\log C(F(\boldsymbol{x}_i^s))_u\right], \tag{2}$$

where $(\boldsymbol{x}_i^s, y_i^s) \in \mathcal{D}_s$, F is a feature extractor, C is the classifier to output $|\mathcal{Y}|$ category probabilities, $C(\cdot)_u$ is the u-th element of $|\mathcal{Y}|$ category probabilities, and $\mathbb{I}_{u=y_i^s}$ is an indicator function, meaning that if $u = y_i^s$, the value of $\mathbb{I}_{u=y_i^s}$ is 1 and 0 otherwise. Then, for each sample from the auxiliary domain, we construct a factual scene and multiple counterfactual scenes over the causal graph, so as to infer the causal effects of variant factors on the category prediction. Given an auxiliary sample $\boldsymbol{x}_i^a \in \mathcal{D}_a$, its factual category is predicted by

$$\boldsymbol{y}_i^a = P(Y|X) = C(F(\boldsymbol{x}_i^a)), \tag{3}$$

where $\boldsymbol{y}_i^a \in \mathbb{R}^{|\mathcal{Y}|}$ is a category probability vector, and represents the value of node Y. For each variant factor, we construct a counterfactual scene to infer its causal effect by doing intervention on the variant factor node V. Let $do(V = v_k)$ denote the intervention on the node V, which is implemented by changing extrinsic attributes of data through the transformation G_{v_k} with multiple degree parameters. Accordingly, the counterfactual category of $\boldsymbol{x}_i^a$ is predicted by

$$\begin{aligned}\boldsymbol{y}_{i,v_k}^a &= P\big(Y|X, do(V = v_k)\big)\\ &= \frac{1}{|\mathcal{M}|}\sum_{\theta_{v_k}\in\mathcal{M}} C\Big(F\big(G_{v_k}(\boldsymbol{x}_i^a;\theta_{v_k})\big)\Big),\end{aligned} \tag{4}$$

where $\mathcal{M}$ is a set of degree parameters, obtained by uniformly sampling magnitudes of transformation G_{v_k} from the scale range. By comparing the factual and counterfactual category probabilities of $\boldsymbol{x}_i^a$, the causal effect of the k-th variant factor is calculated by

$$\boldsymbol{e}_k^i = P(Y|X) - P\big(Y|X, do(V = v_k)\big) = \boldsymbol{y}_i^a - \boldsymbol{y}_{i,v_k}^a. \tag{5}$$

The inferred causal effect represents how much contribution the corresponding variant factor makes to the domain shift, and the larger the causal effect is, the more seriously the domain shift is caused by the variant factor.

3.4. Factor-aware Domain Alignment for Domain Shift Reduction

After analyzing the causes of domain shift, we propose a factor-aware domain alignment to reduce the domain shift by learning multiple feature mappings, with guidance of the

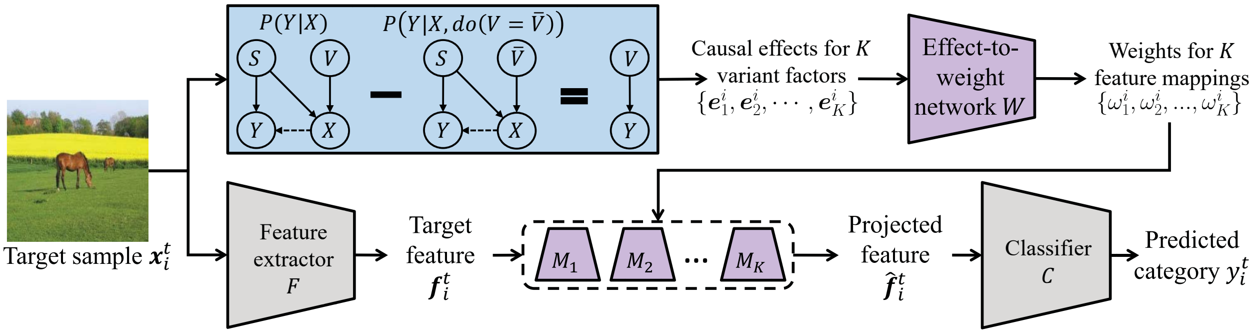
$P(Y|X)$
$P(Y|X, do(V=\bar{V}))$
S
V
Y
X
S
$\bar{V}$
Y
X
V
Y
Causal effects for K variant factors $\{e_1^i, e_2^i, \cdots, e_K^i\}$
Effect-to-weight network W
Weights for K feature mappings $\{\omega_1^i, \omega_2^i, ..., \omega_K^i\}$
Target sample x_i^t
Feature extractor F
Target feature f_i^t
M_1
M_2
M_K
Projected feature $\hat{f}_i^t$
Classifier C
Predicted category y_i^t

Figure 3. Inference process of meta-causal learning for a given target sample.

inferred causal effects of variant factors. Each feature mapping addresses a specific domain shift caused by one variant factor. We construct K feature mappings for K variant factors, and the k-th feature mapping aims to address the domain shift caused by the k-th variant factor. In order to incorporate the causal effects of variant factors into the learning of mappings, we build an effect-to-weight network that converts the causal effect of each variant factor into the weight of the corresponding feature mapping.

For the auxiliary sample $\boldsymbol{x}_i^a$ and its inferred causal effects of all variant factors $\{\boldsymbol{e}_1^i, \boldsymbol{e}_2^i, \cdots, \boldsymbol{e}_K^i\}$, the weights of feature mappings are calculated by

$$\boldsymbol{\omega}^i = \mathrm{softmax}\big(W(\boldsymbol{e}_1^i), W(\boldsymbol{e}_2^i), \cdots, W(\boldsymbol{e}_K^i)\big), \quad (6)$$

where $\boldsymbol{\omega}^i \in \mathbb{R}^K$ and its k-th element ω_k^i denotes the weight of the k-th factor-aware feature mapping. $W(\cdot)$ denotes the effect-to-weight network.

According to the weights of feature mappings, we integrate the K feature mappings to project the auxiliary samples into the source feature space. The alignment of the source and auxiliary domains is implemented by minimizing the feature distance between the source and auxiliary samples in the source feature space. Given a source sample $\boldsymbol{x}_i^s$ with the corresponding category label y_i^s, generated auxiliary sample $\boldsymbol{x}_i^a$ and inferred mapping weights $\boldsymbol{\omega}^i$, the alignment loss incorporated with inferred causal effects is defined as

$$\begin{aligned}\mathcal{L}_a^c = &\frac{1}{N_s}\sum_i ||F(\boldsymbol{x}_i^s) - \sum_k \omega_k^i M_k\big(F(\boldsymbol{x}_i^a)\big)||_2 \\ &+ \frac{1}{N_s}\sum_i \mathcal{H}\Big(C\Big(\sum_k \omega_k^i M_k\big(F(\boldsymbol{x}_i^a)\big), y_i^s\Big)\Big).\end{aligned} \quad (7)$$

where F is the feature extractor, M_k is the k-th feature mapping, C is the classifier, ω_k^i is the k-th element of $\boldsymbol{\omega}^i$, $\sum_k \omega_k^i M_k\big(F(\boldsymbol{x}_i^a)\big)$ represents the projected feature of $\boldsymbol{x}_i^a$, and N_s is the number of source samples. $||\cdot||_2$ is the L2-loss, and $\mathcal{H}(\cdot)$ is the cross-entropy loss. The first term measures the feature distance between the source and auxiliary samples after projection, *i.e.,* the data distribution discrepancy between the source and auxiliary domains. The second term encourages the auxiliary samples to belong to the same categories as the source samples.

Moreover, to enable each feature mapping to address the specific domain shift caused by the corresponding variant factor, we introduce another alignment loss $\mathcal{L}_a^m$:

$$\begin{aligned}\mathcal{L}_a^m = &\frac{1}{N_s}\frac{1}{K}\sum_i\sum_k ||F(\boldsymbol{x}_i^s) - M_k\big(F(\boldsymbol{x}_i^k)\big)||_2 \\ &+ \frac{1}{N_s}\frac{1}{K}\sum_i\sum_k \mathcal{H}\Big(C\Big(M_k\big(F(\boldsymbol{x}_i^k)\big), y_i^s\Big)\Big),\end{aligned} \quad (8)$$

where $\boldsymbol{x}_i^k$ is an sample generated by conducting the data transformation G_{v_k} on the source sample $\boldsymbol{x}_i^s$, denoted by $\boldsymbol{x}_i^k = G_{v_k}(\boldsymbol{x}_i^s; \theta_{v_k})$. The transformation degree parameter θ_{v_k} is randomly selected from its range scale.

Then the overall loss function is defined as

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_a^c + \mathcal{L}_a^m. \quad (9)$$

The whole training process is summarized in Algorithm 1.

3.5. Inference

During testing, given a target sample $\boldsymbol{x}_i^t$, firstly, the causal effects of variant factors are inferred by counterfactual inference in Eq. (5), and the weights $\boldsymbol{\omega^i}$ of feature mappings are calculated by the effect-to-weight network in Eq. (6). Then the target feature $\boldsymbol{f}_i^t = F(\boldsymbol{x}_i^t)$ is projected into the source feature space by integrating K feature mappings according to their weights $\boldsymbol{\omega^i}$ to obtain the projected feature $\hat{\boldsymbol{f}}_i^t = \sum_k \omega_k^i M_k\big(\boldsymbol{f}_i^t\big)$. Finally, the category of $\boldsymbol{x}_i^t$ is predicted by $C(\hat{\boldsymbol{f}}_i^t)$. The whole inference process is shown in Figure 3.

4. Experiments

4.1. Datasets

Digits. The Digits dataset consists of five datasets: MNIST [16], MNIST-M [9], SVHN [26], USPS [13], and

Algorithm 1 Training process

```
Input: The source domain $\mathcal{D}_s$, the variant factor set $\mathcal{V}$.
Output: The feature extractor $F$, the classifier $C$, $K$ feature mappings $\{M_i\}_{i=1}^K$, the effect-to-weight network $W$.
Initialize $F$, $C$, $\{M_i\}_{i=1}^K$, $W$;
while not converge do
    Sample an image $\boldsymbol{x}_i^s$ from $\mathcal{D}_s$;
    Construct a factor subset $\mathcal{V}_i$ by sampling $N_v^i$ variant factors from $\mathcal{V}$;
    Generate an auxiliary sample $\boldsymbol{x}_i^a$ by Eq.(1);
    for $v_k$ in $\mathcal{V}$ do
        Calculate the factual category $\boldsymbol{y}_i^a$ of auxiliary sample $\boldsymbol{x}_i^a$ by Eq.(3);
        Calculate the counterfactual category $\boldsymbol{y}_{i,v_k}^a$ of auxiliary sample $\boldsymbol{x}_i^a$ by Eq.(4);
        Infer the causal effect $\boldsymbol{e}_k^i$ of variant factor $v_k$ via comparing $\boldsymbol{y}_i^a$ and $\boldsymbol{y}_{i,v_k}^a$ by Eq.(5);
    end for
    Calculate the weights of $K$ feature mappings $\{M_i\}_{i=1}^K$ by Eq.(6);
    Update $F$, $C$, $\{M_i\}_{i=1}^K$, $W$ by Eq.(9).
end while
```

SYN [9], with 10 categories. We use MNIST as the source domain, and the other four datasets as the target domains. The first $10,000$ images in the training set of MNIST are used for training.

CIFAR10-C. The CIFAR10-C dataset [11] is proposed to evaluate the robustness of classification model. The images are corrupted from the test set of the CIFAR10 dataset [15] by 19 corruption types with five levels of severity. A higher level means the more serious corruption. There are 10 categories. We use CIFAR10 as the source domain, and CIFAR10-C as the target domains where images of one severity level form one target domain.

PACS. The PACS dataset [17] is a benchmark for domain generalization, and consists of four domains: art painting, cartoon, photo, and sketch. There are $9,991$ images of seven categories. We use one domain as the source domain, and the rest three domains as the target domains. So there are four tasks with different domains as the source domains.

4.2. Implementation Details

Auxiliary Domain. We define 16 variant factors to generate the images in the auxiliary domain, including 12 photometric factors (*Brightness, Contrast, Color, Sharpness, Auto-Contrast, Invert, Equalize, Solarize, SolarizeAdd, Posterize, NoiseSalt, NoiseGaussian*) and 4 geometric factors (*Shear-X, Shear-Y, Rotate, Flip*). Since the *Rotate* and *Flip* variant factors will affect the semantic information of digit images, we use the other 14 variant factors for the Digits dataset. For the CIFAR10-C and PACS datasets, all 16 variant factors are used. To make the auxiliary domain as diverse as possible, for each source image at each iteration, we randomly sample several variant factors and use them to generate a new auxiliary image.

Table 1. Single domain generalization results (%) on Digits with ConvNet as backbone. The model is trained on MNIST, and evaluated on SVHN, SYN, MNIST-M, and USPS.

Method	SVHN	SYN	MNIST-M	USPS	Avg
ERM [14]	27.83	39.65	52.72	76.94	49.29
CCSA [24]	25.89	37.31	49.29	83.72	49.05
d-SNE [36]	26.22	37.83	50.98	**93.16**	52.05
JiGen [2]	33.80	43.79	57.80	77.15	53.14
GUD [31]	35.51	45.32	60.41	77.26	54.62
M-ADA [28]	42.55	48.95	67.94	78.53	59.49
ME-ADA [37]	42.56	50.39	63.27	81.04	59.32
PDEN [19]	62.21	69.39	82.20	85.26	74.77
L2D [34]	62.86	63.72	**87.30**	83.97	74.46
AA [4]	45.23	64.52	60.53	80.62	62.72
RA [5]	54.77	59.60	74.05	77.33	66.44
RSDA [30]	47.40	62.00	81.50	83.10	68.50
RSDA+ASR [8]	52.80	64.50	80.80	82.40	70.10
Ours	**69.94**	**78.47**	78.34	88.54	**78.82**

Table 2. Single domain generalization results (%) on CIFAR10-C with WRN as backbone. Each level is viewed as a target domain, a higher level denotes the more serious corruption and the domain discrepancy between the source and target domains is larger.

Method	level1	level2	level3	level4	level5	Avg
ERM [14]	87.80	81.50	75.50	68.20	56.10	73.82
GUD [31]	88.30	83.50	77.60	70.60	58.30	75.66
M-ADA [28]	90.50	86.80	82.50	76.40	65.60	80.36
PDEN [19]	90.62	88.91	87.03	83.71	77.47	85.55
AA [4]	91.42	87.88	84.10	78.46	71.13	82.60
RA [5]	91.74	88.89	85.82	81.03	74.93	84.48
Ours	**92.38**	**91.22**	**89.88**	**87.73**	**84.52**	**89.15**

4.3. Results on Single Domain Generalization

We compare our method with several state-of-the-art methods, including the baseline method (ERM [14]), the methods of learning domain-invariant features (CCSA [24], d-SNE [36], JiGen [2]), and the methods of making data augmentation (GUD [31], M-ADA [28], ME-ADA [37], PDEN [19], L2D [34], AA [4], RA [5], RSDA [30], RSC [12], ASR [8]).

Table 1, Table 2, and Table 3 show the comparison results on Digits, CIFAR10-C, and PACS, respectively. From the results, there are several interesting observations as follows. First, our method generally outperforms the compared methods on all datasets, which clearly shows the effectiveness of the proposed simulate-analyze-reduce learning paradigm for single domain generalization. Second, compared with the data augmentation methods (*e.g.*,

Table 3. Single domain generalization results (%) on PACS with ResNet-18 as backbone. One domain (name in column) is used as the source domain and the other three domains are used as the target domains.

Method	Artpaint	Cartoon	Sketch	Photo	Avg
ERM [14]	70.90	76.50	53.10	42.20	60.70
RSC [12]	73.40	75.90	56.20	41.60	61.80
RSC+ASR [8]	76.70	79.30	61.60	54.60	68.10
Ours	**77.13**	**80.14**	**62.55**	**59.60**	**69.86**

Table 4. Leave-one-domain-out results (%) on PACS with ResNet-18 as backbone. One domain (name in column) is used as the target domain and the other three domains are used as source domains.

Method	Artpaint	Cartoon	Photo	Sketch	Avg
MetaReg [1]	83.70	77.20	95.50	70.30	81.70
GUD [31]	78.32	77.65	95.61	74.21	81.44
Epi-FCR [18]	82.10	77.00	93.90	73.00	81.50
MASF [6]	80.29	77.17	94.99	71.68	81.03
JiGen [2]	79.42	75.25	96.03	71.35	80.51
DMG [3]	76.90	80.38	93.55	75.21	81.46
DDAIG [38]	84.20	78.10	95.30	74.70	83.10
CSD [27]	78.90	75.80	94.10	76.70	81.40
L2A-OT [39]	83.30	78.20	96.20	73.60	82.80
EISNet [33]	81.89	76.44	95.93	74.33	82.15
RSC [12]	83.43	80.31	95.99	80.85	85.15
ME-ADA [37]	78.61	78.65	95.57	75.59	82.10
MMLD [23]	81.28	77.16	96.09	72.29	81.83
L2D [34]	81.44	79.56	95.51	80.58	84.27
FACT [35]	85.37	78.38	95.15	79.15	84.51
MatchDG [22]	81.32	80.70	**96.53**	79.72	84.57
CIRL [21]	**86.08**	80.59	95.93	82.67	86.32
Ours	85.30	**80.93**	**96.53**	**85.24**	**87.00**

Table 5. Ablation study (%) on PACS with ResNet-18 as backbone. One domain (name in column) is used as the source domain and the other three domains are used as target domains. "T", "A", "C" denote Domain Transformation, Domain Alignment, and Counterfactual Inference, respectively.

Method	T	A	C	Artpaint	Cartoon	Sketch	Photo	Avg
Base				71.26	67.64	43.97	36.99	54.97
DT	✓			75.28	78.46	59.45	56.09	67.32
DTA	✓	✓		71.64	72.78	57.11	52.02	63.39
Ours	✓	✓	✓	**77.13**	**80.14**	**62.55**	**59.60**	**69.86**

AA [4], RA [5] and RSDA [30]) that are more related to our method, our method achieves much better results, and especially yields a 10.32% gain over RSDA on Digits, strongly suggesting that it is beneficial to empower the model with the ability of analyzing the causes of domain shift by counterfactual inference. Third, on more difficult tasks with larger domain shift (*e.g.,* SVHN on Digits, level5 on CIFAR10-C, and Photo on PACS), our method significantly improves the performance, further demonstrating the superiority of our method on handling more challenging situations. Forth, our method performs little worse on MNIST-M and USPS of Digits. The possible reason is that other compared methods use more well-designed data augmentation and network regularization, such as AdaIN based generators in PDEN [19] and stochastic neighborhood embedding techniques in d-SNE [36].

4.4. Results on Multiple Domain Generalization

We extend the proposed method to multi-source domain setting by regarding the multiple source domains as one source domain without using domain labels. We employ the leave-one-domain-out protocol following existing multi-source domain generalization [21, 35]. We compare our method with most related methods that introduces causal inference into domain generalization (MatchDG [22], CIRL [21]), and existing popular domain generalization methods (MetaReg [1], GUD [31], Epi-FCR [18], MASF [6], JiGen [2], DMG [3], DDAIG [38], CSD [27], L2A-OT [39], EISNet [33], RSC [12], ME-ADA [37], MMLD [23], L2D [34], FACT [35]).

Table 4 shows the leave-one-domain-out results on the PACS dataset with ResNet-18 as backbone. From the results, we make several observations. First, it is noteworthy that our method achieves the state-of-the-art overall performance ("Avg") although our method is not designed for multi-source domain generalization, which further demonstrates that the proposed simulate-analyze-reduce learning paradigm not only benefits single domain generalization but also boosts multi-source domain generalization. Second, compared with the methods of introducing causal inference to learn domain-invariant features (MatchDG [22], CIRL [21]), our method achieves better results on the overall metric "Avg", especially making 5.52% and 2.57% gains over MatchDG [22] and CIRL [21], respectively, on the more challenging task (Sketch→Others). Such improvements are attributed to the ability of analyzing and reducing the domain shift, further demonstrating the advantages of causal inference in analyzing the causes of the domain shift.

4.5. Ablation Studies

To evaluate each component of our method, we conduct ablation experiments on the PACS dataset. We design several degraded variants of our method for comparison: (1) "Base", where only the base module is utilized and optimized by Eq. (2) using the source domain; (2) "DT", where the data transformation module is added into the base module and the model is trained using both source and auxiliary domains; (3) "DTA", where the simulated domain shift between auxiliary domain and source domain is directly reduced without analyzing the causes of the domain shift via the proposed counterfactual module.

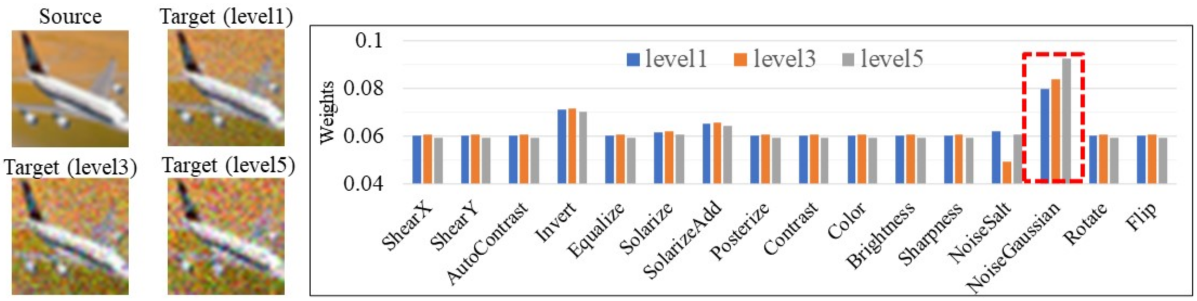

Figure 4. Examples of the inferred causal effects (represented as weights) of variant factors. The left part shows a source image from the CIFAR10 dataset and three target images from the CIFAR10-C dataset with Gaussian noise corruption. As the corruption severity increases from level 1 to level 5, the inferred weights of the *NoiseGaussian* variant factor become larger accordingly.

Table 6. Factor analysis (%) on PACS with ResNet-18 as backbone. One domain (name in column) is used as the source domain and the other three domains are used as target domains.

Method	Artpaint	Cartoon	Sketch	Photo	Avg
Ours_GT	74.62	69.78	52.74	43.34	60.12
Ours_PT	74.47	73.48	50.35	51.67	62.49
Ours	**77.13**	**80.14**	**62.55**	**59.60**	**69.86**

The results are shown in Table 5. From the results, we make several observations. First, our method achieves better performance than "DT", which validates the superiority of our method on analyzing and reducing the domain shift, rather than directly enlarging the source data distribution. Second, "DTA" performs worse than "DT", probably because brute-force domain alignment without analyzing causes of the domain shift leads to negative transfer and thus hurts the performance. Third, our method achieves the best performance thanks to the proposed simulate-analyze-reduce paradigm.

4.6. Factor Analysis

In order to further analyze the effect of variant factors, we design several variants of our method by using different variant factors, including only using geometric factors ("Ours_GT"), and only using photometric factors ("Ours_PT"). Since the factor number of the two type factors are different, we randomly select 4 photometric factors to keep the same number as geometric factors and repeat experiments 10 times to avoid the effect of sampling. The results are shown in Table 6. From the results, it is noteworthy that "Ours_PT" performs better than "Ours_GT", especially with the 8.33% gain when using Photo as the source domain. The reason may be that the domain shift between Photo and the other target domains is mainly caused by photometric factors. Moreover, our method outperforms both "Ours_PT" and "Ours_GT", showing that both photometric factors and geometric factors are required to simulate the domain shift as diverse as possible.

4.7. Causality Visualization

In Figure 4, we visualize the inferred causal effects (represented by the weights) of variant factors for the unseen target images on the CIFAR10-C dataset during testing. The target images are actually corrupted from the source images of CIFAR10 by Gaussian noise of five levels. It is interesting to observe that the inferred weights of the *NoiseGaussian* variant factor are larger than that of the other variant factors, indicating that the counterfactual inference succeeds in discovering the real cause of the domain shift. We can also observe that when the corruption severity of Gaussian noise increases from level 1 to level 5 (*i.e.,* the domain shift is more serious), the inferred weights of the *NoiseGaussian* variant factor become larger, which further demonstrates that our method measures the magnitude of the domain shift correctly.

5. Conclusion

We have presented a new paradigm, *simulate-analyze-reduce*, for single domain generalization. Our paradigm empowers the model with the ability to analyze the domain shift, instead of directly expanding the distribution of the source domain to cover unseen target domains. Under this paradigm, we have presented a meta-causal learning method that can learn meta-knowledge about inferring the causes of domain shift during training, and apply such meta-knowledge to reduce the domain shift for boosting adaptation during testing. Extensive experiments on several benchmark datasets have validated the effectiveness of the new learning paradigm and the advantage of meta-causal learning on analyzing the domain shift for domain generalization.

Acknowledgments. This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant No 62072041.

References

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. volume 31, pages 1006–1016, 2018. 7

[2] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2229–2238, 2019. 2, 6, 7

[3] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *European Conference on Computer Vision (ECCV)*, pages 301–318. Springer, 2020. 7

[4] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019. 6, 7

[5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 702–703, 2020. 6, 7

[6] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. volume 32, pages 6447–6458, 2019. 7

[7] Yingjun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees GM Snoek, and Ling Shao. Learning to learn with variational information bottleneck for domain generalization. In *European Conference on Computer Vision (ECCV)*, pages 200–216. Springer, 2020. 2

[8] Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. Adversarially adaptive normalization for single domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8208–8217, 2021. 1, 2, 6, 7

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, pages 1180–1189. PMLR, 2015. 5, 6

[10] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2551–2559, 2015. 2

[11] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019. 6

[12] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision (ECCV)*, pages 124–140. Springer, 2020. 6, 7

[13] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 16(5):550–554, 1994. 5

[14] Vladimir Koltchinskii. *Oracle inequalities in empirical risk minimization and sparse recovery problems: École D'Été de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Science & Business Media, 2011. 6, 7

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

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[17] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5542–5550, 2017. 6

[18] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1446–1455, 2019. 7

[19] Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 224–233, 2021. 1, 2, 6, 7

[20] Chang Liu, Xinwei Sun, Jindong Wang, Haoyue Tang, Tao Li, Tao Qin, Wei Chen, and Tie-Yan Liu. Learning causal semantic representation for out-of-distribution prediction. volume 34, pages 6155–6170, 2021. 2

[21] Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. Causality inspired representation learning for domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8046–8056, 2022. 2, 7

[22] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning (ICML)*, pages 7313–7324. PMLR, 2021. 2, 7

[23] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 11749–11756, 2020. 7

[24] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5715–5725, 2017. 2, 6

[25] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning (ICML)*, pages 10–18. PMLR, 2013. 2

[26] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5

[27] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *International Conference on Machine Learning (ICML)*, pages 7728–7738. PMLR, 2020. 7

[28] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12556–12565, 2020. 1, 2, 6

[29] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[30] Riccardo Volpi and Vittorio Murino. Addressing model vulnerability to distributional shifts over image transformation sets. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7980–7989, 2019. 6, 7

[31] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. volume 31, pages 5339–5349, 2018. 2, 6, 7

[32] Haohan Wang, Zexue He, Zachary C. Lipton, and Eric P. Xing. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[33] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. In *European Conference on Computer Vision (ECCV)*, pages 159–176. Springer, 2020. 7

[34] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 834–843, 2021. 1, 2, 6, 7

[35] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14383–14392, 2021. 7

[36] Xiang Xu, Xiong Zhou, Ragav Venkatesan, Gurumurthy Swaminathan, and Orchid Majumder. d-sne: Domain adaptation using stochastic neighborhood embedding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2497–2506, 2019. 6, 7

[37] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 14435–14447, 2020. 6, 7

[38] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 13025–13032, 2020. 7

[39] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision (ECCV)*, pages 561–578. Springer, 2020. 2, 7