Open Domain Generalization with Domain-Augmented Meta-Learning

Yang Shu*, Zhangjie Cao*, Chenyu Wang, Jianmin Wang, Mingsheng Long (✉)
School of Software, BNRist, Tsinghua University, China
{shu-y18,caozj14,cy-wang18}@mails.tsinghua.edu.cn, {jimwang,mingsheng}@tsinghua.edu.cn

Abstract

Leveraging datasets available to learn a model with high generalization ability to unseen domains is important for computer vision, especially when the unseen domain's annotated data are unavailable. We study a novel and practical problem of Open Domain Generalization (OpenDG), which learns from different source domains to achieve high performance on an unknown target domain, where the distributions and label sets of each individual source domain and the target domain can be different. The problem can be generally applied to diverse source domains and widely applicable to real-world applications. We propose a Domain-Augmented Meta-Learning framework to learn open-domain generalizable representations. We augment domains on both feature-level by a new Dirichlet mixup and label-level by distilled soft-labeling, which complements each domain with missing classes and other domain knowledge. We conduct meta-learning over domains by designing new meta-learning tasks and losses to preserve domain unique knowledge and generalize knowledge across domains simultaneously. Experiment results on various multi-domain datasets demonstrate that the proposed Domain-Augmented Meta-Learning (DAML) outperforms prior methods for unseen domain recognition.

1. Introduction

Deep convolutional neural networks have achieved state-of-the-art performance on wide ranges of computer vision applications with access to large-scale labeled data [23, 20, 39, 19]. However, for a target domain of interest, collecting enough training data is prohibitive. A practical solution is to generalize the model learned on the existing data to the unseen domain. Since the existing source datasets for training may be from different resources, they may fall into different domains and hold different label sets, e.g., ImageNet [8] and DomainNet [36]. Besides, the target domain is totally unknown, and may also have a distribution shift and a different label set from the source domains. We call the valuable and challenging problem as **Open Domain Generalization (OpenDG)**, where we need to learn generalizable representation from disparate source domains that generalizes well to any unseen target domain, as illustrated in Figure 1.

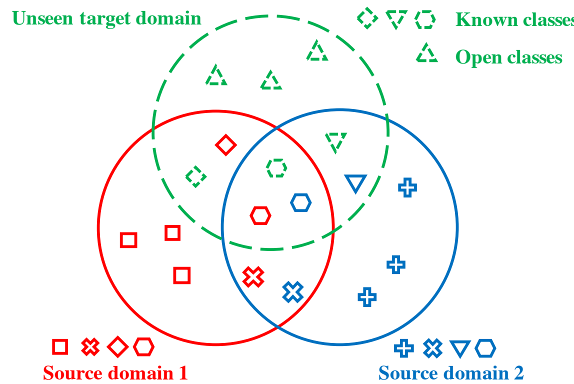


Figure 1. Open Domain Generalization (OpenDG). Different source domains hold disparate label sets. The goal is to learn generalizable representations from these source domains to help classify the known classes and detect open classes in the unseen target domain.

There are two key challenges for open domain generalization. (1) Distinct source domains and the unseen target domain are drawn from different distributions with a large distribution shift. (2) The different label sets of distinct source domains cause some classes to exist in many more domains than other classes. The data of minor classes existing in few domains are lacking in diversity. This makes the problem extremely difficult for existing methods [25, 29].

To address the first challenge, previous works minimize the distribution distance between domains by adversarial learning [34, 29], which successfully closes the domain gap when all source domains share the same label set. However, according to the second challenge, the different label sets between domains cause these distribution alignment methods to suffer from severe mismatch of classes. For the second challenge, a straightforward way is to manually sample data of minor classes existing in few domains, but the diversity in domains of the class is still limited. The generalization on the minor class is still inferior to other classes.

To generalize from *arbitrary* source domains to an unseen target domain, we propose a **Domain-Augmented Meta-**

*Equal contribution.

Table 1. Comparison of the proposed generalization setting with the previous settings related to cross-domain learning. The columns list assumptions made by the problem settings. **Note that more "✗" means the method needs less assumption and thus is more widely-applicable.** We can observe that the proposed open domain generalization problem requires no assumptions on the label set, no target data, and no post-training on target data, which is the most general problem setting. **S** means source while **T** means target. Note that "Same between **S&T** Domains" means the union of all source domain label sets equals the target label set, i.e., whether there are open classes.

Problem Setting	Label Set		Target Data for Training		Post-Training on Target Labeled Data
	Same for **S** Domains	Same between **S&T** Domains	Labeled Data	Unlabeled Data	
Domain Adaptation [31, 32]	✓	✓	✗	✓	✗
Domain Adaptation with Category Shift [35, 2, 51]	✓	✗	✗	✓	✗
Multi-Source Domain Adaptation [55]	✓	✓	✗	✓	✗
Multi-Source Domain Adaptation with Category Shift [50]	✗	✓	✗	✓	✗
Domain Generalization [34]	✓	✓	✗	✗	✗
Heterogeneous Domain Generalization [30]	✗	✗	✗	✗	✓
The Proposed Open Domain Generalization	✗	✗	✗	✗	✗

Learning (DAML) framework. To close the domain gap between disparate source domains, we avoid distribution matching but learn generalizable representations across domains by meta-learning. To overcome the disparate label sets of open domain generalization, we propose two domain augmentation methods at both feature-level and label-level. At feature-level, we design a novel Dirichlet mixup (Dir-mixup) to compensate for the missing labels. At label-level, we utilize the soft-labeling distilled from other domains' networks to transfer the knowledge of other domains to the current network. DAML learns a representation that embeds the knowledge of all source domains and is highly generalizable to the unseen target domain. We use the ensemble of all source domain network outputs as the final prediction, which naturally calibrates the predictive uncertainty. In summary:

- We propose a new and practical problem: **Open Domain Generalization (OpenDG)**, which learns from arbitrary source domains with disparate distributions and label sets to generalize to an unseen target domain.
- We propose a principled **Domain-Augmented Meta-Learning (DAML)** framework to address open domain generalization. We augment each domain with novel Dir-mixup and distilled soft-labeling to overcome the disparate label sets of source domains and conduct meta-learning across augmented domains to learn open-domain generalizable representations.
- Experiment results on several multi-domain datasets show that compared to previous generalization methods, DAML achieves higher classification accuracy on both known classes and open classes in an unseen target domain even with extremely diverse source domains.

2. Related Work

In this section, we briefly discuss works related to ours, including domain adaptation, domain generalization, and data augmentation methods. We compare our problem setting with the problem settings of previous works in Table 1.

Domain Adaptation aims to adapt the model from the source domain to the target domain, which typically mitigates the domain gap by minimizing the distribution distance [14, 32]. However, the classic domain adaptation requires the same label set between source and target domains. Recent works try to extend domain adaptation to varied source and target label sets [2, 35, 41, 51], but the solution relies on the target unlabeled data, which is not available in the open domain generalization setting.

Multi-source domain adaptation is more related to our work with more than one source domain. Most of the works assume that all the source domains share the same label set [55, 36], which can be easily violated in practice since source domains may be drawn from different resources. DCN [50] moves a step forward to remove the constraint on the source label sets but still requires the union of source label sets to be the same as the target label set. We instead require no label set constraint and no target data for training.

Domain Generalization aims to learn a generalizable model with only source data to achieve high performance in an unseen target domain [22, 34], which typically learns domain-invariant features across source domains [34, 16, 15, 28, 4, 38, 5]. When the different source domains hold different label sets, such learning causes mismatch of classes. CIDDG [29] can avoid the mismatching but still requires all the source and target domains to share the same label sets, or otherwise the low domain diversity of some classes makes it hard to learn domain-invariant features.

Meta-learning instead has the potential to learn from highly diverse domains. However, current meta-learning-based domain generalization methods still fail to consider different label sets of distinct source domains and the open classes in the target domain [25, 1, 10, 27]. Heterogeneous domain generalization [30, 49] has a similar goal of learning generalizable representations, which targets a more powerful pre-trained model by learning from heterogeneous source domains of different label sets. However, it requires additional target labeled data to induce a category model, which cannot fit into the proposed open domain generalization problem.

Augmentation The statistical learning theory [45] suggests that the generalization of the learning model can be

Algorithm 1 Training process of Domain-Augmented Meta-Learning (DAML)

Input: Source datasets $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_S$, learning rates η and β, Dir-mixup hyper-parameters α_{max} and α_{min}

Initialize $\theta_s|_{s=1}^{S}$
while Not Converged **do**
  Sample a batch of data $\mathcal{B}^{\text{tr}} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \cdots, (\mathbf{x}_S, \mathbf{y}_S)\}$ from all source domains $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_S$.
  for $s = 1, \ldots, S$ **do** ▷ Meta-training starts
    $\boldsymbol{\alpha}_s^{\text{tr}} \leftarrow \{\alpha_{\text{max}}, \alpha_{\text{min}}, s\}$ ▷ Dir-mixup parameter for meta-training
    $\mathcal{B}_s^{\text{D-mix}} = \{(\mathbf{z}_s^{\text{D-mix}}, \mathbf{y}_s^{\text{D-mix}})\} \leftarrow \text{Dir-mixup}(\{\boldsymbol{\alpha}_s^{\text{tr}}, \mathcal{B}^{\text{tr}}\})$ ▷ Obtain Dir-mixup according to Eqn. (3)
    $\mathcal{B}_s^{\text{distill}} = \{(\mathbf{x}_s, \mathbf{y}_s^{\text{distill}})\} \leftarrow \{G_j|_{j\neq s}, F_j|_{j\neq s}, \mathcal{B}^{\text{tr}}\}$ ▷ Obtain distilled soft-label according to Eqn. (4)
    $\mathcal{L}_s^{\text{tr}} \leftarrow \left\{G_s(F_s(\mathbf{x}_s)), \mathbf{y}_s, G_s(\mathbf{z}_s^{\text{D-mix}}), \mathbf{y}_s^{\text{D-mix}}, \mathbf{y}_s^{\text{distill}}\right\}$ using data in $\mathcal{B}^{\text{tr}}$, $\mathcal{B}_s^{\text{D-mix}}$, and $\mathcal{B}_s^{\text{distill}}$ ▷ According to Eqn (1)
    $\theta_{F_s', G_s'} = \theta_{F_s, G_s} - \eta \nabla_\theta \mathcal{L}_s^{\text{tr}}$
  Sample another batch of data $\mathcal{B}^{\text{obj}} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \cdots, (\mathbf{x}_S, \mathbf{y}_S)\}$ from all source domains $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_S$.
  for $s = 1, \ldots, S$ **do** ▷ Meta-objective starts
    $\boldsymbol{\alpha}_s^{\text{obj}} \leftarrow \{\alpha_{\text{min}}, \alpha_{\text{max}}, s\}$ ▷ Dir-mixup parameter for meta-objective
    $\mathcal{B}_s^{\text{D-mix}'} = \{(\mathbf{z}_s^{\text{D-mix}'}, \mathbf{y}_s^{\text{D-mix}'})\} \leftarrow \text{Dir-mixup}(\{\boldsymbol{\alpha}_s^{\text{obj}}, \mathcal{B}^{\text{obj}}\})$ ▷ Obtain Dir-mixup according to Eqn. (3)
    $\mathcal{L}_s^{\text{obj}} \leftarrow \left\{G_s'(F_s'(\mathbf{x}_j))|_{j\neq s}, \mathbf{y}_j|_{j\neq s}, G_s'(\mathbf{z}_s^{\text{D-mix}'}), \mathbf{y}_s^{\text{D-mix}'}\right\}$ using data in $\mathcal{B}^{\text{obj}}$ and $\mathcal{B}_s^{\text{D-mix}'}$ ▷ According to Eqn (2)
    $\theta_{F_s, G_s} \leftarrow \theta_{F_s, G_s} - \beta \nabla_\theta (\mathcal{L}_s^{\text{tr}} + \mathcal{L}_s^{\text{obj}})$ ▷ Update parameters with meta-learning
return $\theta_s|_{s=1}^{S}$

characterized by the model capacity and the diversity of training data. So data augmentation can improve generalization by increasing the diversity of training data. Basic augmentations including affine transformation, random cropping, and horizontal flipping are widely-used in image classification [6, 42, 24]. Recently, more advanced augmentations are proposed. Mixup [54, 44, 18] combines two samples linearly. Cutout [9] removes contiguous sections of input images. Cutmix [52] combines cutout and mixup by filling the Cutout part with sections of other image patches.

Augmentation-based generalization methods promote the generalization ability by augmenting source data, where adversarial data augmentation [47], gradient-based perturbations [43], self-supervised learning signals [3], and CutMix [33] are used as the augmentation method. Note that these augmentation methods target general situations for generalization across domains but are not designed specially for open domains with disparate label sets.

Different from all previous works, this paper studies open domain generalization, a practical but challenging problem. We develop the DAML framework to conduct meta-learning over augmented source domains. We design a novel Dir-mixup to mix samples from multiple domains instead of mixing two arbitrary samples in classic mixup. Dir-mixup bridges all the source domains and compensates each domain with missing classes from other domains, which naturally fits the disparate source label sets. We further propose a new distilled soft-labeling to transfer knowledge across domains.

3. Domain-Augmented Meta-Learning

In this section, we first introduce the open domain generalization (OpenDG) problem. Then we introduce the Domain-Augmented Meta-Learning (DAML) and describe the step-by-step algorithm and the optimization of the framework, which consists of the proposed domain augmentation and the meta-learning on the augmented domains.

3.1. Open Domain Generalization

In open domain generalization (OpenDG), we have multiple source domains $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_S$ available for training, where each source domain s consists of data-label pairs $\mathcal{D}_s = \{(\mathbf{x}_s, \mathbf{y}_s)\}$. $\mathbf{y}_s$ denotes the one-hot label of $\mathbf{x}_s$. Note that although we train the model with mini-batches in practice, here we omit the batch size of each domain to simplify the notations. We use $\mathcal{C}$ to denote the union of all the source label sets. In open domain generalization, we have no constraint on the label sets of different domains. We aim to learn open-domain generalizable representation from all the source domains, which can generalize well to an unseen target domain $\mathcal{D}_t$. Specifically, the target domain, only used for evaluation, consists of fully unlabeled data $\mathcal{D}_t = \{\mathbf{x}_t\}$ and its label set $\mathcal{C}_t$ may contain classes existing in any source label set or unknown classes not existing in the union of source label sets $\mathcal{C}$. The goal is to classify at inference each target sample with the correct class if it belongs to the source label set $\mathcal{C}$, or label it as "unknown". Note that no target data, even unlabeled, are available for training, which differs OpenDG from domain adaptation [51] or domain generalization [49].

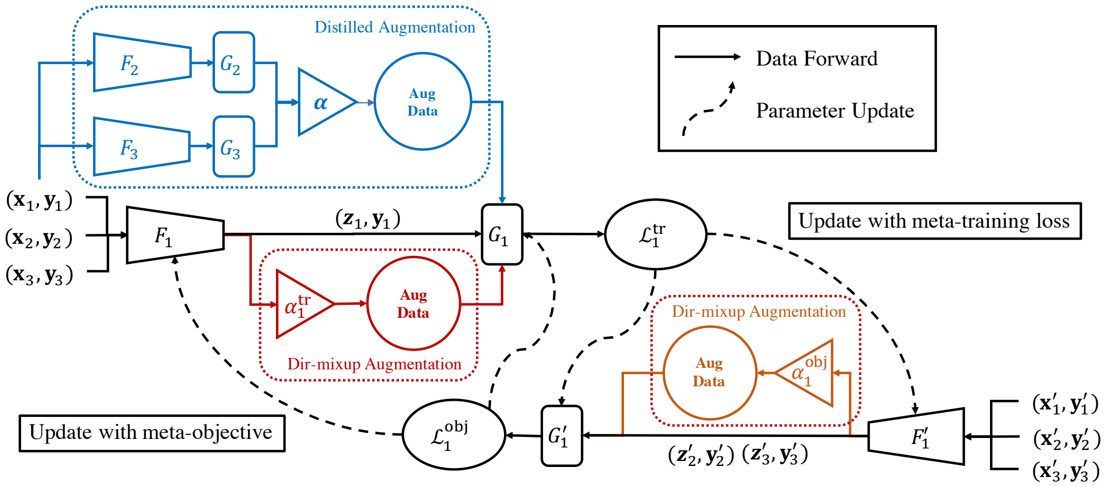


Figure 2. The architecture of the proposed DAML framework. We show the computation graph for source domain 1 as an example, and the other source domains are computed similarly. In the meta-training (up part, left to right), each source domain is augmented by Dir-mixup (red) and distilled soft-labeling (blue) to compute the $\mathcal{L}_1^{\text{tr}}$ to update the model parameters to F_1' and G_1'. In the meta-objective (down part, right to left), each source domain is augmented by Dir-mixup (red) to compute the $\mathcal{L}_1^{\text{obj}}$ to finally update the model parameters.

3.2. The DAML Framework

We propose DAML to address open domain generalization problems to mitigate the disparate label sets and distribution shifts among the diverse source domains. As shown in Algorithm 1, the idea is to learn generalizable representations by meta-learning over augmented domains.

Augmented Domains As demonstrated in [53, 17], increasing the diversity of the dataset can substantially improve the generalization of the representations. Motivated by this idea, we augment each domain to expand the diversity of the datasets. We observe that different domains have different distributions and hold different label sets, which means that each domain contains distinct knowledge but lacks domain knowledge and class knowledge of other domains. Based on the observation, we design domain augmentation to address open domain generalization. Our insight is to conduct both feature-level and label-level augmentation. For feature-level augmentation, we propose a novel Dirichlet Mixup (Dir-mixup) method, which augments each domain by the mixup with multiple domains. For label-level augmentation, we propose to augment each domain by distilling soft-labels from models of other domains. The proposed domain augmentation increases the diversity of the data and compensates each domain with missing knowledge of features and classes. The details of the proposed domain augmentation are introduced in Section 3.3.

Meta-Learning We design the learning framework to learn generalizable representations, which simultaneously preserves the unique information of each domain and aggregates the knowledge of all the domains. Thus, instead of employing a shared network for all source domains, which only embeds domain common knowledge, we build one individual classification network composed of a feature extractor F_s and a classifier G_s for each source domain s. Then we need to learn a generalizable representation aggregating the information of all the source domains. We conduct meta-learning over all the networks since meta-learning is demonstrated to be able to learn a generalizable representation from highly disparate domains. In each iteration of the parameter update, we first draw a batch of samples from each domain and compute the corresponding Dir-mixup samples and distilled soft-labels (Line 5-7 in Algorithm 1). Unlike standard meta-learning loss applied only on the raw data [12], with the augmented domains, we design a new meta-training loss as the classification loss on the original data, the domain-augmented data by Dir-mixup, and soft-labels distilled from other domain networks. For each domain s, let $\mathbf{z}_s = F_s(\mathbf{x}_s)$ be the feature of $\mathbf{x}_s$, we define the meta-training loss as

$$\begin{aligned}\mathcal{L}_s^{\text{tr}} = &\underset{(\mathbf{x}_s,\mathbf{y}_s)\sim\mathcal{D}_s}{\mathbb{E}}\left[-\sum_{k=1}^{|\mathcal{C}|}(\mathbf{y}_s)^{(k)}\log\left(G_s^{(k)}(F_s(\mathbf{x}_s))\right)\right]\\ &+\underset{(\mathbf{z}_s^{\text{D-mix}},\mathbf{y}_s^{\text{D-mix}})\sim\mathcal{D}_s^{\text{D-mix}}}{\mathbb{E}}\left[-\sum_{k=1}^{|\mathcal{C}|}(\mathbf{y}_s^{\text{D-mix}})^{(k)}\log\left(G_s^{(k)}(\mathbf{z}_s^{\text{D-mix}})\right)\right]\\ &+\underset{(\mathbf{x}_s,\mathbf{y}_s^{\text{distill}})\sim\mathcal{D}_s^{\text{distill}}}{\mathbb{E}}\left[-\sum_{k=1}^{|\mathcal{C}|}(\mathbf{y}_s^{\text{distill}})^{(k)}\log\left(G_s^{(k)}(F_s(\mathbf{x}_s))\right)\right].\end{aligned} \tag{1}$$

The superscript (k) means the probability of the k-th class. $\mathcal{D}_s^{\text{D-mix}}$ and $\mathcal{D}_s^{\text{distill}}$ are the augmented domains of Dir-mixup samples and distilled soft-label samples for meta-training on domain s. We compute one step of gradient update for each source network with respect to the meta-training loss: $\theta_{G_s',F_s'} = \theta_{G_s,F_s} - \eta\nabla_\theta\mathcal{L}_s^{\text{tr}}$ (Line 9 in Algorithm 1), where η is the step size. The design idea of meta-objective is to guide the gradient update from the meta-training loss to the desired goal. Classic meta-learning employs the losses over all sampled tasks as the meta-objective [12]. But our goal is to improve the generalization ability of the model, so different from classic meta-objective, we design the meta-objective as the classification loss on the original data and Dir-mixup data in other domains with the updated network G'_s, F'_s, which can propagate the knowledge of other domains to domain s and promote the knowledge transfer and generalization across domains. The meta-objective is defined as

$$\mathcal{L}_s^{\text{obj}} = \sum_{j \neq s} \mathop{\mathbb{E}}_{(\mathbf{x}_j, \mathbf{y}_j) \sim \mathcal{D}_j} \left[- \sum_{k=1}^{|\mathcal{C}|} (\mathbf{y}_j)^{(k)} \log \left(G_s'^{(k)}(F_s'(\mathbf{x}_j)) \right) \right] + \mathop{\mathbb{E}}_{(\mathbf{z}_s^{\text{D-mix}'}, \mathbf{y}_s^{\text{D-mix}'}) \sim \mathcal{D}_s^{\text{D-mix}'}} \left[- \sum_{k=1}^{|\mathcal{C}|} (\mathbf{y}_s^{\text{D-mix}'})^{(k)} \log \left(G_s'^{(k)}(\mathbf{z}_s^{\text{D-mix}'}) \right) \right] \tag{2}$$

$\mathcal{D}_s^{\text{D-mix}'}$ is the augmented domain of Dir-mixup samples for domain s in meta-objective. The minimization of the meta-objective finds a gradient descent update that updates the network to classify data in other domains with high accuracy, which encourages the network to learn a generalizable representation performing well across all domains. We finally update the network parameters in one iteration by $\theta_s \leftarrow \theta_s - \beta \nabla_\theta (\mathcal{L}_s^{\text{tr}} + \mathcal{L}_s^{\text{obj}})$, where β is the learning rate.

3.3. Domain Augmentation

The meta-learning framework can learn a generalizable representation aggregating information from all source domains, where the generalization power highly relies on the diversity of each source domain. To this end, we design two multiple source domain augmentation approaches: the feature-level augmentation, Dir-mixup, and the label-level augmentation, distilled augmentation. The augmentations compensate for the missing class information in each source domain and further increase domain diversity.

Dir-mixup Mixup [54] generates a new data-label by the weighted sum of the feature and one-hot label of existing samples, where the weights are sampled from a pre-defined distribution. We augment the s-th source domain by mixup of data in the s-th domain with data in other domains. Since these data may belong to the missing classes of the s-th source domain, mixup augmentation would compensate for the missing classes. Also, mixup produces inter-domain data, which further increases the diversity of data in each domain.

However, the original mixup is defined to mix two samples. When applied to open domain generalization with multiple source domains, mixup samples are only generated from pairs of domains, which, as shown in Figure 3, only generates samples between two domains (the lines between vertex) but lacks samples mixing multiple domains (the whole area). Also, to obtain all domain combinations, such pairwise mixup needs O(#domains×#domains) mixup

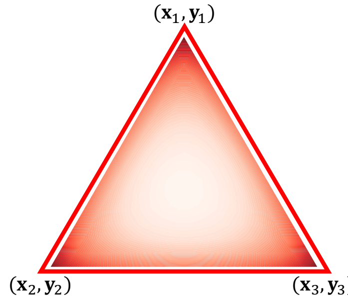


Figure 3. Comparison between Dir-mixup and classic mixup. Classic mixup only mixes two samples, so mixup samples only exist on the edge of the triangle while Dir-mixup mixes samples of multiple domains covering the whole triangle area, meaning Dir-mixup introduce mixup samples with more information and higher diversity.

samples. Therefore, to mix multiple domains, we need to sample the weight from a multi-variate distribution instead of the beta distribution used in the original mixup. We select Dirichlet distribution since it has similar properties to the beta distribution and is a multi-variate distribution. We then design a new Dir-mixup to mix samples (one for each domain) with a designed weight $\boldsymbol{\lambda}$ sampled from a Dirichlet distribution parameterized by a parameter $\boldsymbol{\alpha}$. We perform mixup at feature-level. Let $\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_S$ be the features of different domain data extracted by the network, the Dir-mixup augmented data $(\mathbf{z}^{\text{D-mix}}, \mathbf{y}^{\text{D-mix}})$ can be calculated as:

$$\boldsymbol{\lambda} \sim \text{Dirichlet}(\boldsymbol{\alpha})$$
$$(\mathbf{z}^{\text{D-mix}}, \mathbf{y}^{\text{D-mix}}) = (\sum_{s=1}^{S} \lambda^{(s)} \mathbf{z}_s, \sum_{s=1}^{S} \lambda^{(s)} \mathbf{y}_s). \tag{3}$$

Compared with recent work using mixup for domain generalization [33, 49], Dir-mixup is more efficient and effective. The parameter $\boldsymbol{\alpha}$ adjusts the distribution to generate different augmentations, better serving the meta-learning process. Consider constructing Dir-mixup for each model s. In the meta-training, we want to keep more information and focus more on domain s during mixup, so we set $\alpha^{(s)}$ larger than other components in $\boldsymbol{\alpha}$, which assigns a larger weight $\lambda^{(s)}$ to $\mathbf{z}_s$ statistically. In the meta objective, the goal is to transfer knowledge from other domains and improve the cross-domain generalization, which would be enhanced by mixup results with larger domain discrepancy. So we set $\alpha^{(s)}$ smaller than other components in $\boldsymbol{\alpha}$, which induces smaller $\lambda^{(s)}$ statistically. We employ two hyper-parameters α_{max} and α_{min} to realize this idea. For the meta-training of model s, we set $\boldsymbol{\alpha}_s^{\text{tr}}$ to be a length S vector with all entries as α_{min} but the s-th entry as α_{max}. We generate mixup data with this $\boldsymbol{\alpha}_s^{\text{tr}}$ to form the Dir-mixup augmentation set in the meta-training of model s, as $\mathcal{D}_s^{\text{D-mix}}$ in Equation 1. For the meta-objective, we set $\boldsymbol{\alpha}_s^{\text{obj}}$ to be a length S vector with all entries as α_{max} but the s-th entry as α_{min}. And the data generated from this $\boldsymbol{\alpha}_s^{\text{obj}}$ form the Dir-mixup augmentation set for model s, which is the $\mathcal{D}_s^{\text{D-mix}'}$ in Equation 2.

Distilled Augmentation For the s-th source domain, we further augment it with the soft-labeling distilled from other domains, which is the output predictions of other networks. We mix soft-labels from other domains to increase the diversity of the augmentation. We set the $\boldsymbol{\alpha}$ to be a vector of all ones with dimension $S-1$ since we do not prefer a particular other domain. The augmentation can be defined as

$$\boldsymbol{\lambda} \sim \text{Dirichlet}(\boldsymbol{\alpha})$$
$$\mathbf{y}_s^{\text{distill}} = \sum_{j=1}^{s-1} \lambda^{(j)} G_j(F_j(\mathbf{x}_s)) + \sum_{j=s+1}^{S} \lambda^{(j-1)} G_j(F_j(\mathbf{x}_s)). \tag{4}$$

The soft-label indicates the decision of the networks of other domains on the s-th domain data, which transfers the knowledge from other domains to the s-th domain. The augmentation is reflected as the third term in Equation 1, where we do not back-propagate through F_j, G_j since they are just used to generate the soft-labeling. The augmentation regularizes the s-th domain network with knowledge of other domains, which derives a more generalizable representation.

3.4. Inference

In the inference stage, we have the networks for all source domains $G_1, \cdots, G_S, F_1, \cdots, F_S$ trained by the DAML framework as shown in Algorithm 1. For a test sample $\mathbf{x}_t$ from the target domain $\mathcal{D}_t$, we compute the raw prediction of $\mathbf{x}_t$ by aggregating the predictions of all the source networks:

$$\widehat{\mathbf{y}}_t = \frac{1}{S}\sum_{s=1}^{S} G_s(F_s(\mathbf{x}_t)). \tag{5}$$

The ensemble of all source domain networks naturally calibrates the prediction confidence and enables DAML to achieve higher performance in the unseen target domain.

4. Experiments

We construct several open domain generalization scenarios with different datasets to evaluate the proposed method.

4.1. Datasets

PACS dataset [26] consists of four domains corresponding to four different image styles, including photo (**P**), art-painting (**A**), cartoon (**C**) and sketch (**S**). The four domains have the same label set of 7 classes. We use each domain as the target domain and the other three domains as source domains to form four cross-domain tasks. We evaluate the generalization performance on both the original closed-set dataset and the modified open-domain dataset.

Office-Home [46] comprises of images from four different domains: Artistic (**Ar**), Clip art (**Cl**), Product (**Pr**) and Real-world (**Rw**). It has a large domain gap and 65 classes which is much more than other DG datasets, so it is very challenging. We spread these 65 classes among the four domains to derive an open-domain dataset. We construct four open generalization tasks based on it, where each domain is used as the target domain respectively, and the other three domains serve as source domains.

Multi-Datasets scenario is constructed in this paper to consider a more realistic situation of learning generalizable representations from arbitrary source domains. We simulate the process where we obtain source domains from different resources and try to learn a generalizable model to achieve high accuracy on an unseen target domain. We leverage several public datasets including **Office-31** [40], **STL-10** [7] and **Visda2017** [37] as source domains, and evaluate the generalization performance on four domains in **DomainNet** [36]. There exist distribution discrepancy and huge label-set disparity across the four datasets, which forms a natural open domain generalization scenario. Since there are too many open classes in the DomainNet, we preserve all the classes existing in the joint label set of source domains and subsample 20 open classes.

4.2. Closed-Set Generalization

We evaluate the classification accuracy of closed-set generalization on the widely-used domain generalization dataset **PACS**. The closed-set setting exactly matches the domain generalization setting so we compare with supervised learning on the merged datasets of all source domains: AGG, domain generalization methods including domain-invariant feature learning based methods: CIDDG [29], CSD [38] and DMG [5], meta-learning based methods: MLDG [25], MetaReg [1], MASF [10] and Epi-FCR [27], and augmentation based methods: CrossGrad [43], JiGen [3] and CuMix [33]. We do not compare with domain adaptation methods since they need unlabeled target data.

As shown in Table 4, on the closed-set generalization setting, to which previous domain generalization methods are tailored, DAML still outperforms all previous methods on average and achieves at least comparable performance on all the tasks. In particular, DAML outperforms state-of-the-art meta-learning-based DG, which indicates the importance of domain augmentation to learn generalizable representations. DAML surpasses state-of-the-art augmentation-based DG, indicating that the meta-learning paradigm and the carefully designed feature-level and label-level augmentations can enable learning more generalizable representations.

4.3. Open Domain Generalization

We evaluate the generalization performance for situations where the source and target domains have different label sets and open classes exist. We conduct experiments on PACS, Office-Home, and Multi-Datasets. For PACS and Office-Home, we preserve different parts of classes in the source domains and the target domain to create disparate label sets

Table 2. Results of PACS dataset under the open-domain setting.

	Art		Sketch		Photo		Cartoon		Avg	
Method	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score
AGG	51.35	38.87	49.75	47.09	53.15	44.19	66.43	48.98	55.17 ± 0.16	44.78 ± 0.33
MLDG [25]	44.59	31.54	51.29	49.91	62.20	43.35	71.64	55.20	57.43 ± 0.14	45.00 ± 0.31
FC [30]	51.12	39.01	51.15	49.28	60.94	45.79	69.32	52.67	58.13 ± 0.20	46.69 ± 0.25
Epi-FCR [27]	**54.16**	41.16	46.35	46.14	70.03	48.38	72.00	**58.19**	60.64 ± 0.22	48.47 ± 0.29
PAR [48]	52.97	39.21	53.62	52.00	51.86	36.53	67.77	52.05	56.56 ± 0.51	44.95 ± 0.57
RSC [21]	50.47	38.43	50.17	44.59	67.53	49.82	67.51	47.35	58.92 ± 0.46	45.05 ± 0.60
CuMix [33]	53.85	38.67	37.70	28.71	65.67	49.28	**74.16**	47.53	57.85 ± 0.32	41.05 ± 0.66
DAML (ours)	54.10	**43.02**	**58.50**	**56.73**	**75.69**	**53.29**	73.65	54.47	$\mathbf{65.49} \pm 0.36$	$\mathbf{51.88} \pm 0.42$

Table 3. Results of Office-Home dataset under the open-domain setting.

	Clipart		Real-World		Product		Art		Avg	
Method	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score
AGG	42.83	**44.98**	62.40	53.67	54.27	50.11	42.22	40.87	50.43 ± 0.32	47.41 ± 0.53
MLDG [25]	41.82	41.26	62.98	55.84	56.89	52.25	42.58	40.97	51.07 ± 0.19	47.58 ± 0.42
FC [30]	41.80	41.65	63.79	55.16	54.41	52.02	44.13	43.25	51.03 ± 0.24	48.02 ± 0.57
Epi-FCR [27]	37.13	42.05	62.60	54.73	54.95	52.68	46.33	44.46	50.25 ± 0.50	48.48 ± 0.76
PAR [48]	41.27	41.77	65.98	57.60	55.37	54.13	42.40	42.62	51.26 ± 0.27	49.03 ± 0.41
RSC [21]	38.60	38.39	60.85	53.73	54.61	54.66	44.19	44.77	49.56 ± 0.44	47.89 ± 0.79
CuMix [33]	41.54	43.07	64.63	58.02	57.74	55.79	42.76	40.72	51.67 ± 0.12	49.40 ± 0.27
DAML (ours)	**45.13**	43.12	**65.99**	**60.13**	**61.54**	**59.00**	**53.13**	**51.11**	$\mathbf{56.45} \pm 0.21$	$\mathbf{53.34} \pm 0.45$

Table 4. Results on closed-set PACS dataset.

Method	**A**	**S**	**P**	**C**	**Avg**
AGG	77.6	70.3	94.4	73.9	79.1
CIDDG [29]	82.0	74.8	94.6	74.4	81.4
MLDG [25]	79.5	71.5	94.3	77.3	80.7
CrossGrad [43]	78.7	65.1	94.0	73.3	77.8
MetaReg [1]	79.5	72.2	94.3	75.4	80.4
JiGen [3]	79.4	71.4	**96.0**	75.3	80.4
MASF [10]	80.3	71.7	94.5	77.2	81.0
Epi-FCR [27]	82.1	73.0	93.9	77.0	81.5
CSD [38]	79.8	72.5	95.5	75.0	80.7
DMG [5]	76.9	**75.2**	93.4	**80.4**	81.5
CuMix [33]	82.3	72.6	95.1	76.5	81.6
DAML	**83.0**	74.1	95.6	78.1	**82.7**

among source domains and between the source and target domains. For Multi-Datasets, we preserve all the classes for all source datasets. We show the class split in each domain in the supplementary materials. We follow [51] to set a threshold on the prediction confidence and label samples with a confidence lower than the threshold as an open class: "unknown". For the evaluation metric, we report the accuracy of data from non-open classes (Acc) and also follow the state-of-the-art universal domain adaptation paper [13] to use H-score to evaluate performance over all target data.

For the open-domain classification setting, we mainly compare with previous methods that are less influenced by the different label sets of source domains. We select state-of-the-art meta-learning-based and augmentation-based DG methods [25, 27, 33], heterogeneous domain generalization methods: FC [30], recently proposed methods of learning robust and generalizable features: PAR [48] and RSC [21].

As shown in Tables 2, 3 and 5, we can observe that DAML outperforms all the compared methods with a large margin on both Acc and H-score, which indicates that DAML not only learns a generalizable representation for non-open classes but also detects open classes with higher accuracy. In particular, DAML outperforms the meta-learning-based DG methods MLDG and Epi-FCR on almost all the tasks, especially the H-score, which demonstrates that domain augmentation, compensating missing labels for each domain, is vital to addressing the different label sets across source domains. DAML outperforms CuMix, which also employs mixup for data augmentation. Note that we design the Dir-mixup to mix samples from multiple domains while CuMix mixes two arbitrary samples. So our Dir-mixup creates mixup samples with higher variations and diversity, which encourages the model to learn more generalizable representations.

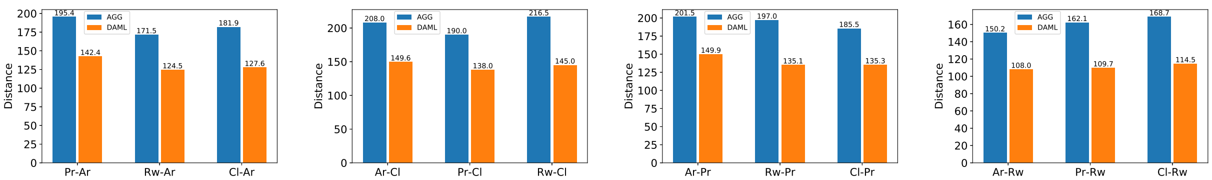
The Multi-Datasets simulates the real-world scenario where we aim to generalize from datasets available at hand to an unseen domain. The different source domains hold extremely disparate label sets. In this realistic scenario, DAML outperforms all the compared methods with a large margin, indicating that DAML can be applied to realistic generalization problems and achieve higher performance.

4.4. Analysis

Ablation Study We go deeper into the DAML framework to explore the efficacy of each module in DAML including meta-learning, Dir-mixup and distilled soft-labels. As shown in Table 6, $\mathcal{D}_s^{\text{D-mix}}$ means whether to use the Dir-mixup data in the meta-training loss, *i.e.* whether to use the second term

Table 5. Results on the Multi-Datasets scenario (naturally under the open-domain setting).

	Clipart		**Real**		**Painting**		**Sketch**		**Avg**	
Method	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score
AGG	29.78	34.06	65.33	64.72	44.30	51.04	27.59	35.41	41.75 ± 0.63	46.31 ± 0.57
MLDG [25]	29.66	35.11	65.37	54.40	44.04	50.53	26.83	34.57	41.48 ± 0.68	43.65 ± 0.71
FC [30]	29.91	35.42	64.77	63.65	44.13	50.07	28.56	34.10	41.84 ± 0.73	45.81 ± 0.69
Epi-FCR [27]	27.70	37.62	60.31	64.95	39.57	50.24	26.76	33.74	38.59 ± 1.13	46.64 ± 0.95
PAR [48]	29.29	39.99	64.09	62.59	42.36	46.37	30.21	39.96	41.49 ± 0.63	47.23 ± 0.55
RSC [21]	27.57	34.98	60.36	60.02	37.76	42.21	26.21	30.44	37.98 ± 0.77	41.91 ± 1.28
CuMix [33]	30.03	40.18	64.61	65.07	44.37	48.70	29.72	33.70	42.18 ± 0.45	46.91 ± 0.40
DAML (ours)	**37.62**	**44.27**	**66.54**	**67.80**	**47.80**	**52.93**	**34.48**	**41.82**	$\mathbf{46.61} \pm 0.59$	$\mathbf{51.71} \pm 0.52$



(a) Ar as the Target (b) Cl as the Target (c) Pr as the Target (d) Rw as the Target

Figure 4. The Fréchet distance between each source domain and the target domain for the four generalization tasks on Office-Home dataset.

Table 6. Ablation study on the open-domain Office-Home dataset.

$\mathcal{D}_s^{\text{D-mix}}$	$\mathcal{D}_s^{\text{D-mix}\prime}$	$\mathcal{D}_s^{\text{mix}}$	$\mathcal{D}_s^{\text{distill}}$	w/ Meta	**Cl**	**Rw**	**Pr**	**Ar**	**Avg**
-	-	-	-	✓	42.2	64.8	57.6	49.6	53.6
✓	-	-	-	✓	43.8	64.9	57.1	51.7	54.4
-	✓	-	-	✓	43.8	65.7	58.2	52.4	55.0
✓	✓	-	-	✓	44.8	65.9	59.7	52.9	55.9
✓	✓	-	✓	-	44.1	65.1	59.7	52.2	55.3
-	-	✓	✓	✓	44.3	65.3	59.0	51.9	55.1
✓	✓	-	✓	✓	**45.1**	**66.0**	**61.5**	**53.1**	**56.5**

in Equation 1. $\mathcal{D}_s^{\text{D-mix}\prime}$ means whether to use the Dir-mixup data in the meta-objective loss, *i.e.* whether to use the second term in Equation 2. $\mathcal{D}_s^{\text{mix}}$ means using classic mixup which mixes two arbitrary samples. $\mathcal{D}_s^{\text{distill}}$ means whether to use the distilled soft-label, *i.e.* whether to use the third term in Equation 1. w/ Meta means whether to use meta-learning or otherwise supervised learning on the augmented domains.

In Table 6, we observe that using both $\mathcal{D}_s^{\text{D-mix}}$ and $\mathcal{D}_s^{\text{D-mix}\prime}$ outperforms using only $\mathcal{D}_s^{\text{D-mix}}$ and using only $\mathcal{D}_s^{\text{D-mix}\prime}$, which indicates Dir-mixup samples are helpful in both meta-training and meta-objective losses. Changing the Dir-mixup to classic mixup drops the accuracy, which shows the importance of a built-in mixup for multiple domains. Using $\mathcal{D}_s^{\text{distill}}$ outperforms not using $\mathcal{D}_s^{\text{distill}}$ on average, indicating that transferring knowledge between domains by distilled soft-labels learns more generalizable representations. DAML outperforms meta-learning conducted on the raw domain without any domain augmentation, which indicates the importance of domain augmentation to address the different label sets of source domains. DAML also outperforms the variant that uses no meta-learning, which demonstrates that meta-learning can aggregate knowledge from augmented source domains in a more effective way.

Fréchet Distance We compare the domain gap between source and target domains on features learned by the baseline AGG model and features learned by the DAML model. We extract features of each domain and compute their mean vectors and covariance matrices. Then we evaluate the Fréchet Distance[11] between the features of each source domain and the non-open class part of the target domain. As shown in Figure 4, the domain gaps between source domains and the unseen target domain are smaller in DAML, indicating that DAML learns more generalizable representations.

5. Conclusion

In this paper, we propose a new open domain generalization problem aiming to generalize from arbitrary source domains with disparate label sets to unseen target domains, which can be widely utilized in real-world applications. We further propose a novel Domain-Augmented Meta-Learning framework (DAML) to address the problem, which conducts meta-learning over domains augmented at feature-level by specially designed Dir-mixup and at label-level by distilled soft-labels. Extensive experiments demonstrate that DAML learns more generalizable representations for classification in the target domain than the previous generalization methods.

Acknowledgments

This work was supported by National Key R&D Program of China (2020AAA0109201), NSFC grants (62022050, 62021002, 61772299, 71690231), Beijing Nova Program (Z201100006820041), and MOE Innovation Plan of China.

References

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 998–1008, 2018. 2, 6, 7

[2] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Partial transfer learning with selective adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[3] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 6, 7

[4] Fabio M Carlucci, Paolo Russo, Tatiana Tommasi, and Barbara Caputo. Agnostic domain generalization. *arXiv preprint arXiv:1808.01102*, 2018. 2

[5] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *European Conference in Computer Vision (ECCV)*, 2020. 2, 6, 7

[6] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3642–3649. IEEE, 2012. 3

[7] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 215–223, 2011. 6

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009. 1

[9] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[10] Qi Dou, Daniel C. Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2, 6, 7

[11] DC Dowson and BV Landau. The fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982. 8

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, volume 70, pages 1126–1135, 2017. 4, 5

[13] Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *European Conference on Computer Vision (ECCV)*, August 2020. 7

[14] Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning, International Conference on Machine Learning (ICML) 2015, Lille, France, 6-11 July 2015*, pages 1180–1189, 2015. 2

[15] Muhammad Ghifary, David Balduzzi, W. Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(7):1414–1430, 2017. 2

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

[17] Zhiqiang Gong, Ping Zhong, and Weidong Hu. Diversity in machine learning. *IEEE Access*, 7:64323–64350, 2019. 4

[18] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 3714–3722, 2019. 3

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988. IEEE, 2017. 1

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[21] Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision (ECCV)*, 2020. 7, 8

[22] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, pages 158–171. Springer, 2012. 2

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. 1

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 3

[25] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 1, 2, 6, 7, 8

[26] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5543–5551. IEEE, 2017. 6

[27] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. Episodic training for domain generalization. In *IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2, 6, 7, 8

[28] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[29] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *European Conference on Computer Vision (ECCV)*, pages 624–639, 2018. 1, 2, 6, 7

[30] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy M. Hospedales. Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning (ICML)*, volume 97, pages 3915–3924, 2019. 2, 7, 8

[31] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, 2015. 2

[32] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1640–1650, 2018. 2

[33] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *European Conference on Computer Vision (ECCV)*, August 2020. 3, 5, 6, 7, 8

[34] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning (ICML)*, pages 10–18, 2013. 1, 2

[35] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[36] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1406–1415, 2019. 1, 2, 6

[37] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017. 6

[38] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *International Conference on Machine Learning (ICML)*, volume 119, pages 7728–7738, 2020. 2, 6, 7

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015. 1

[40] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, 2010. 6

[41] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *European Conference on Computer Vision (ECCV)*, September 2018. 2

[42] Ikuro Sato, Hiroki Nishimura, and Kensuke Yokoi. Apac: Augmented pattern classification with neural networks. *arXiv preprint arXiv:1505.03229*, 2015. 3

[43] 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. 3, 6, 7

[44] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5486–5494, 2018. 3

[45] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013. 2

[46] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[47] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5334–5344, 2018. 3

[48] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10506–10518, 2019. 7, 8

[49] Yufei Wang, Haoliang Li, and Alex C Kot. Heterogeneous domain generalization via domain mixup. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3622–3626. IEEE, 2020. 2, 3, 5

[50] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3964–3973, 2018. 2

[51] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2720–2729, 2019. 2, 3, 7

[52] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019. 3

[53] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017. 4

[54] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 3, 5

[55] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8559–8570, 2018. 2