# Domain Decorrelation with Potential Energy Ranking

**Sen Pei[1,2], Jiaxi Sun[1,2], Richard Yi Da Xu[4], Shiming Xiang[1,2], and Gaofeng Meng[1,2,3*]**

[1] NLPR, Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] CAIR, HK Institute of Science and Innovation, Chinese Academy of Sciences
[4] Hong Kong Baptist University
{peisen,sunjiaxi}2020@ia.ac.cn, xuyida@hkbu.edu.hk, {smxiang,gfmeng}@nlpr.ia.ac.cn

## Abstract

Machine learning systems, especially the methods based on deep learning, enjoy great success in modern computer vision tasks under ideal experimental settings. Generally, these classic deep learning methods are built on the *i.i.d.* assumption, supposing the training and test data are drawn from the same distribution independently and identically. However, the aforementioned *i.i.d.* assumption is, in general, unavailable in the real-world scenarios, and as a result, leads to sharp performance decay of deep learning algorithms. Behind this, domain shift is one of the primary factors to be blamed. In order to tackle this problem, we propose using **Po**tential **E**nergy **R**anking (PoER) to decouple the object feature and the domain feature in given images, promoting the learning of label-discriminative representations while filtering out the irrelevant correlations between the objects and the background. PoER employs the ranking loss in shallow layers to make features with identical category and domain labels close to each other and vice versa. This makes the neural networks aware of both objects and background characteristics, which is vital for generating domain-invariant features. Subsequently, with the stacked convolutional blocks, PoER further uses the contrastive loss to make features within the same categories distribute densely no matter domains, filtering out the domain information progressively for feature alignment. PoER reports superior performance on domain generalization benchmarks, improving the average top-1 accuracy by at least 1.20% compared to the existing methods. Moreover, we use PoER in the ECCV 2022 NICO Challenge, achieving top place with only a vanilla ResNet-18 and winning the *jury award*. The code has been made publicly available at: https://github.com/ForeverPs/PoER.

## Introduction

Deep learning methods have been proved to be increasingly effective in many complex machine learning tasks, such as large-scale image classification, objects detection and image generation, to name a few. Generally, the human-surpassing performance that deep neural networks enjoy is greatly benefited from the *i.i.d.* assumption, supposing the training and the test data are drawn from the same distribution independently and identically. Unfortunately, in open-world scenarios, it is difficult to guarantee that the *i.i.d.* assumption always holds, and as a result, leads to sharp performance drop in the presence of inputs from unseen domains. Formally, the aforementioned problem is termed domain generalization (DG). Given several labeled domains (*a.k.a.* source domains), DG aims to train classifiers only with data from these labeled source domains that can generalize well to any unseen target domains. Different from the closely related domain adaptation (DA) task, DG has no access to the data from target domains while DA can use that for finetuning, namely the adaptation step.

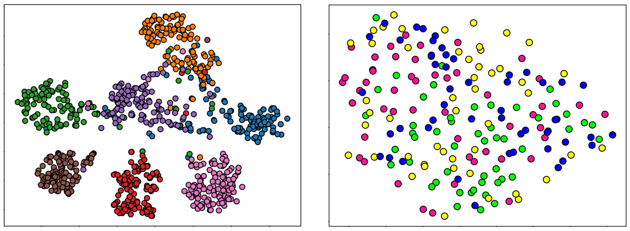
(a) category-wise (b) domain-wise

Figure 1: Feature distribution of the proposed PoER. A vanilla ResNet-18 (He et al. 2016) is trained on PACS (Li et al. 2017) dataset, and the above images show the category-wise and domain-wise feature distribution. We remove the conventional classification head and perform clustering using the outputted feature. (a): Feature distribution across different categories. It is clear that the final outputs of PoER are pure label-related representations. (b): Feature distribution across different domains. PoER makes the feature extractor aware of the characteristic of both label-related and domain-related information first and filters out the domain appearance progressively with the stacked convolutional blocks, achieving feature alignment for better generalization ability.

Commonly, a straightforward way to deal with the domain generalization problem is to collect as much data as possible from diverse domains for training. However, this solution is costly and impractical in some fields, and it is actually inevitable that the deep neural networks deployed in open-world scenarios will encounter out-of-distribution (OOD) inputs that are never exposed in the training phase no matter how much data you collect. Except for the aforementioned solution, the schemes aiming to mitigate the negative effects of domain shifts can be roughly divided into three

[*] Corresponding author.

categories, which are data augmentation schemes, representation learning techniques, and optimization methods. The data augmentation schemes such as (Zhou et al. 2020a,b) and (Zhou et al. 2021b) mainly generate auxiliary synthetic data for training, improving the robustness and generalization ability of classifiers. The representation learning represented by feature alignment (Ganin et al. 2015) enforces the neural networks to capture domain-invariant features, removing the irrelevant correlations between objects and background (*i.e.,* domain appearance). There are adequate works of literature in this line of research, such as (Shankar et al. 2018; Tzeng et al. 2014; Muandet, Balduzzi, and Schölkopf 2013) and (Zhang et al. 2021). Apart from the previously mentioned research lines, as stated in (Shen et al. 2021), optimization methods that are both model agnostic and data structure agnostic are established to guarantee the worst-case performance under domain shifts. From the overall view, our proposed PoER belongs to the representation learning methods, but it deals with the DG problem from a brand new perspective. Instead of enforcing the neural network to generate domain-invariant features directly as presented in existing methods, PoER makes the neural networks capture both label-related and domain-related information explicitly first, and then distills the domain-related features out progressively, which in turn promotes the generation of domain-invariant features.

As the saying goes, *know yourself and know your enemy, and you can fight a hundred battles with no danger of defeat*. The main drawback of existing representation learning methods is paying much attention to the generation of domain-invariant features before knowing the characteristics of the domain itself. By comparison, PoER makes the classifiers capture label-discriminative features containing domain information explicitly first in shallow layers, and with the distillation ability of the stacked convolutional blocks, filters the irrelevant correlations between objects and domains out. From the perspective of potential energy, PoER enforces the features with identical domain labels or category labels to have lower energy differences (*i.e.,* pair-potential) in shallow layers and vice versa. Further, in deeper convolutional layers, PoER penalizes the classifiers if features with identical category labels are pushed far away from each other no matter domain labels, achieving domain decorrelation for better generalization ability. The key contributions of this paper are summarized as follows:

- A plug-and-play regularization term, namely PoER, is proposed to mitigate the negative effects of domain shifts. PoER is parameters-free and training-stable, which can be effortlessly combined with the mainstream neural network architectures, boosting the generalization ability of conventional classifiers.
- PoER reports superior performance on domain generalization benchmarks, reducing the classification error by at least 1.20% compared with existing methods. PoER is scalable to the size of datasets and images.
- We tackle the domain generalization problem from a brand new perspective, *i.e.,* potential energy ranking, wishing to introduce more insights to DG task.

## Related Work

**Domain Augmentation Schemes.** This line of research argues that diverse training data is the key to more generalizable neural networks. In DDAIG (Zhou et al. 2020a), the adversarial training schemes are used for generating images from unseen domains that served as the auxiliary data, boosting the generalization ability of conventional neural networks. In MixStyle (Zhou et al. 2021b), the InstanceNorm (Ulyanov, Vedaldi, and Lempitsky 2016) and AdaIN (Huang and Belongie 2017) are used for extracting domain-related features. The mixing operation between these features results in representations from novel domains, increasing the domain diversity of training data (*i.e.,* source domains). More recently, Style Neophile (Kang et al. 2022) synthesizes novel styles constantly during training, addressing the limitation and maximizing the benefit of style augmentation. In SSAN (Wang et al. 2022), different content and style features are reassembled for a stylized feature space, expanding the diversity of labeled data. Similarly, EFDM (Zhang et al. 2022) proposes to match the empirical Cumulative Distribution Functions (eCDFs) of image features, mapping the representation from unseen domains to the specific feature space.

**Domain-invariant Representation Learning.** This is another research line for dealing with the domain generalization problem from the perspective of representation learning. In (Ganin et al. 2015), the deep features are promoted to be discriminative for the main learning task and invariant with respect to the shift across domains. In (Tzeng et al. 2014; Shankar et al. 2018), the neural networks are guided to extract domain-invariant features which can mitigate the effects of domain shifts. In (Akada et al. 2022), self-supervised manners are extended for learning domain-invariant representation in depth estimation, yielding better generalization ability. (Nguyen et al. 2021) obtains a domain-invariant representation by enforcing the representation network to be invariant under all transformation functions among domains. Also in LIRR (Li et al. 2021), the main idea is to simultaneously learn invariant representations and risks under the setting of Semi-DA. More recently, in CCT-Net (Zhou et al. 2021c), the author employs confidence weighted pooling (CWP) to obtain coarse heatmaps which help generate category-invariant characteristics, enabling transferability from the source to the target domain.

## Preliminaries

**Problem Statement.** Suppose $\mathcal{X}$, $\mathcal{D}$, and $\mathcal{Y}$ are the raw training data, domain labels, and category labels respectively. A classifier is defined as $f_\theta : \mathcal{X} \mapsto \mathcal{Y}$. Domain generalization aims to sample data only from the joint distribution $\mathcal{X} \times \mathcal{D} \times \mathcal{Y}$ for training while obtaining model $f_\theta$ which can generalize well to unseen domains. Feel free to use the domain labels or not during the training. Under the DG settings, $f_\theta$ has no access to the data from target domains, which is different from the DA task.

**Background of Potential Energy.** It is generally acknowledged that the energy is stored in objects due to their

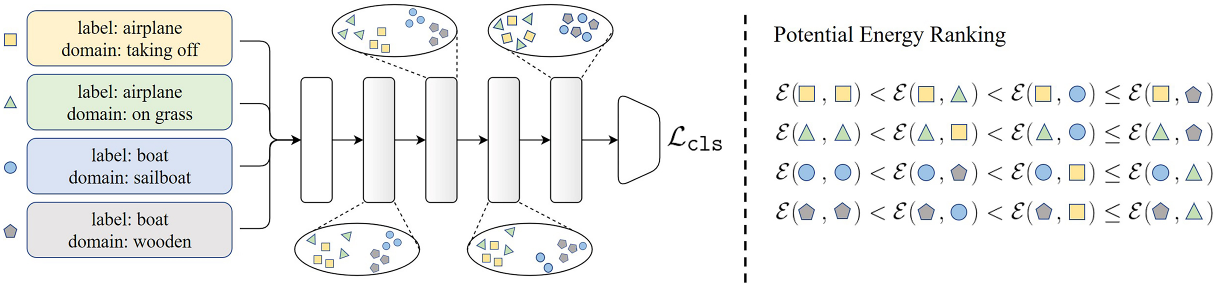
label: airplane
domain: taking off
label: airplane
domain: on grass
label: boat
domain: sailboat
label: boat
domain: wooden
$\mathcal{L}_{\texttt{cls}}$
Potential Energy Ranking
$\mathcal{E}$(□, □) < $\mathcal{E}$(□, △) < $\mathcal{E}$(□, ○) ≤ $\mathcal{E}$(□, ⬠)
$\mathcal{E}$(△, △) < $\mathcal{E}$(△, □) < $\mathcal{E}$(△, ○) ≤ $\mathcal{E}$(△, ⬠)
$\mathcal{E}$(○, ○) < $\mathcal{E}$(○, ⬠) < $\mathcal{E}$(○, □) ≤ $\mathcal{E}$(○, △)
$\mathcal{E}$(⬠, ⬠) < $\mathcal{E}$(⬠, ○) < $\mathcal{E}$(⬠, □) ≤ $\mathcal{E}$(⬠, △)

Figure 2: The proposed PoER framework. In the shallow layers, neural networks extract feature representations containing both label-related and domain-related information, and PoER pushes the features with different domains far away from each other, making the neural networks aware the characteristic across domains. Following this, with the stacked convolutional blocks, PoER enforces the features within identical category labels close to each other progressively no matter domain labels, filtering out the irrelevant correlation between objects and domains. The distilled pure label-related feature is used for classification finally. In the image above, we use data within two categories and four domains as an example for depiction. $\mathcal{E}(,)$ indicates the pair-potential which describes the difference of potential energy between any given feature pairs.

position in the field, namely potential energy. Pair potential is a function that describes the difference in potential energy between two given objects. A cluster of objects achieves stability when their pair potential between each other is fairly low. Inspired by this principle, we treat the representation and feature space of neural networks as objects and the potential field respectively. The classifier is expected to achieve stability where the energy difference (*i.e.,* pair-potential) between the same domains and the same labels is lower. Based on this fact, we enforce the neural network to capture the representation explicitly containing the domain and the label information, and with the stacked convolutional blocks, filtering out the irrelevant correlation between label-related objects and the appearance. This means the pair potential is increasing across data with different category labels while decreasing across domains with identical category labels.

## Methods

In this section, we detail our energy-based modeling first, elaborating PoER's methodology. Following that, we give the training and inference pipeline for reproducing in clarity.

### Energy-based Modeling

We treat the feature space as a potential field $\mathcal{F}$, and the feature map from different layers is described as $f$. Formally, $d(,)$ is a metric function that measures the distance (*i.e.,* potential difference) between any given feature pairs. Picking $L_2$ distance as the measurement, the potential difference is obtained as:

$$d(f^a, f^b) = \sqrt{\sum_{i=1}^{m}(f_i^a - f_i^b)^2} \tag{1}$$

where $f^a$ and $f^b$ represent the flattened feature maps extracted from the identical layers of neural networks, and $m$ is their dimensionality. With an energy kernel $\mathcal{E}(,)$, the potential difference in Eq.(1) is mapped to the pair-potential as shown below:

$$\mathcal{E}(f^a, f^b) = \exp(\beta \cdot d(f^a, f^b)) - 1 \tag{2}$$

where $\beta$ is a hyper-parameter, and it is equal to 1 by default. In the shallow layers, PoER pushes the features from identical categories and domains close to each other while keeping that from different categories or domains away, making the neural networks aware of the characteristics of domains and objects. To meet this awareness, PoER employs margin-based ranking loss to supervise the discrimination across different domains and categories. Intuitively, it is straightforward to accept that the features with identical category and domain labels should have lower pair-potential than those with different category or domain labels.

Suppose $f^{ij}$ is a feature representation from category $i$ and domain $j$, and $f^{pq}$ indicates the feature with category label $p$ and domain label $q$. Noting that $i \neq p$ and $j \neq q$. To formalize the idea of PoER, we build the ranking order across these features as follows:

$$\mathcal{E}(f^x, f^{ij}) < \mathcal{E}(f^x, f^{iq}) < \mathcal{E}(f^x, f^{pj}) < \mathcal{E}(f^x, f^{pq}) \tag{3}$$

where $f^x$ indicates another feature with the identical category and domain label as $f^{ij}$. With the margin-based ranking loss, we combine the feature pairs above to calculate the pair-wise ranking loss. Formally, we have:

$$\mathcal{L}_1 = \max(0, \mathcal{E}(f^x, f^{ij}) - \mathcal{E}(f^x, f^{iq}) + \delta) \tag{4}$$

$$\mathcal{L}_2 = \max(0, \mathcal{E}(f^x, f^{iq}) - \mathcal{E}(f^x, f^{pj}) + \delta) \tag{5}$$

$$\mathcal{L}_3 = \max(0, \mathcal{E}(f^x, f^{pj}) - \mathcal{E}(f^x, f^{pq}) + \delta) \tag{6}$$

where $\delta$ indicates a non-negative margin, and we set $\delta$ to 0 in our experiments. The complete ranking loss in shallow layers is depicted as $\mathcal{L}_{\texttt{rank}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$.

With the stacked convolutional blocks, PoER filters the domain-related information out progressively. In deeper layers, we enforce the features distribute closely intra-class while discretely inter-class, ignoring the domain labels. We use $f^{i*}$ and $f^{p*}$ to depict the data with category label $i$ and $p$ respectively. Therefore, given feature $f^x$ from category $i$, the cluster loss is formulated as follows:

$$\mathcal{L}_{\texttt{cluster}} = \exp(\mathcal{E}(f^x, f^{i*}) - \mathcal{E}(f^x, f^{p*})) \tag{7}$$

The PoER regularization is the sum of aforementioned loss functions, saying $\mathcal{L}_{\texttt{PoER}} = \mathcal{L}_{\texttt{rank}} + \mathcal{L}_{\texttt{cluster}}$. Moreover, the aforementioned regularization term can be easily calculated within each batch of the training data since all of the combinations stated above exist.

## Distance-based Classification

We classify the data in GCPL (Yang et al. 2018) manner since we regularize the features in neural networks. $f$ is the feature from the penultimate layer of the discriminative model such as conventional ResNet-18 with shape $m$, and $M$ is the learnable prototypes with shape $k \times n \times m$, where $k$ and $n$ are the numbers of classes and prototypes each class. The $L_2$ distance is calculated between $f$ and $M$ along the last dimension of prototypes. The distance matrix is obtained as $d(f, M)$ in shape $k \times n$. For calculating the category cross-entropy loss, we pick the minimal distance within prototypes of each class, saying $d$ is acquired by selecting the minimum value along the last dimension of $d(f, M)$ in shape $k$. Formally, the predicted probability that given data belongs to category $i$ is built as:

$$p_i = \frac{\exp(-d_i)}{\sum_{j=1}^{k} \exp(-d_j)} \tag{8}$$

Suppose $\mathbb{I}(y = i)$ is an indicator function that equals 1 if and only if the corresponding label of $f$ is $i$ otherwise 0, and $y$ is the category label of feature $f$. Therefore, the distance-based classification loss is then formulated as:

$$\mathcal{L}_{\texttt{cls}} = -\sum_{i=1}^{k} \mathbb{I}(y = i) \cdot \log(p_i) \tag{9}$$

The overall loss function of distance-based classification with PoER is in the form as:

$$\mathcal{L} = \mathcal{L}_{\texttt{cls}} + \alpha \mathcal{L}_{\texttt{PoER}} \tag{10}$$

where $\alpha$ is a hyper-parameter to balance the regularization term and classification error.

## Training and Inference

We detail the training and inference pipeline in this section for easy reproduction and a clear understanding. As stated in the previous section, $\mathcal{X}$, $\mathcal{D}$, and $\mathcal{Y}$ are the space of image data, domain labels, and category labels. We sample data $(x, d, y)$ from the joint distribution of $\mathcal{X} \times \mathcal{D} \times \mathcal{Y}$ for training. Suppose $h(\cdot)$ is a feature extractor that returns the flattened features of each block in neural networks. The feature extractor has no more need of a classification head since we employ the distance-based cluster manner for identification. The training and inference processes are summarized in Algorithm 1. It is worth noting that PoER performs regularization on features from all blocks while the classification is achieved only with the features from the last block of $h(\cdot)$.

**Algorithm 1:** Potential energy ranking for DG task.

```
Input: training data $\mathcal{X} \times \mathcal{D} \times \mathcal{Y}$, neural network $h(\cdot)$
Output: trained neural network $h(\cdot)$
while Training do
    Sample a batch data $\{x, d, y\}$ from $\mathcal{X} \times \mathcal{D} \times \mathcal{Y}$;
    Get the features from each block: $f_s = h(x)$;
    Calculate $\mathcal{L}_{\texttt{rank}}$ for features from the first three
      blocks in $f_s$ with pair-wise manner;
    Calculate $\mathcal{L}_{\texttt{cluster}}$ for features from the left
      blocks in $f_s$, including the last one;
    Calculate the $\mathcal{L}_{\texttt{cls}}$ with feature from the last
      block in $f_s$, summing them up as Eq.(10);
    Update the parameters of $h(\cdot)$ with gradient
      descent method.
while Inference do
    Sample $x$ from the testing set;
    Get feature $f$ from the last block of $h(\cdot)$;
    Calculate distance between $f$ and prototypes $M$
      with Eq.(1);
    Classify the given data using Eq.(8).
```

# Experiments

## Dataset

We consider four benchmarks to evaluate the performance of our proposed PoER, namely PACS (Li et al. 2017), VLCS (Ghifary et al. 2015a), Digits-DG (Zhou et al. 2021b), and Office-Home (Venkateswara et al. 2017). On the NICO (He, Shen, and Cui 2021) dataset, we only report the limited results of some new methods we collected. The datasets mentioned below can be downloaded at Dassl (Zhou et al. 2021a), which is a testing bed including many DG methods.

**PACS** contains images with shape $227 \times 227 \times 3$ in RGB channel, belonging to 7 categories within 4 domains which are **P**hoto, **A**rt, **C**artoon, and **S**ketch. Under DG settings, the model has no access to the target domain, and therefore the dataset is split into three parts used for training, validation, and test. We use the split file provided in EntropyReg (Zhao et al. 2020). The training and validation sets are data from the source domains while the test set is sampled from the target domain. We pick classifiers based on the validation metric for reporting the test results.

**Office-Home** contains images belonging to 65 categories within 4 domains which are artistic, clip art, product, and the real world. Following DDAIG (Zhou et al. 2020a), we randomly split the source domains into 90% for training and 10% for validation, reporting the metrics on the leave-one-out domain using the best-validated model.

**Digits-DG** is a mixture of 4 datasets, namely MNIST (LeCun et al. 1998), MNIST-M (Ganin et al. 2015), SVHN (Netzer et al. 2011), and SYN (Ganin et al. 2015). All images are resized into $32 \times 32 \times 3$. The reported metrics use the leave-one-domain-out manner for evaluation.

**VLCS** contains images from 5 categories within 4 domains which are Pascal **V**OC2007 (Everingham et al. 2010), **L**abelMe (Russell et al. 2008), **C**altech (Fei-Fei et al. 2004), and **S**UN09 (Choi et al. 2010). We randomly split the source domains into 70% for training and 30% for validation following (Ghifary et al. 2015a), reporting metrics on the target domain using the best-validated classifier.

**NICO** consists of natural images within 10 domains, 8 out of which are treated as the source and 2 as the target. Following (Zhang et al. 2021), we randomly split the data into 90% for training and 10% for validation, reporting metrics on the left domains with the best-validated model.

## Evaluation Protocol

We report top-1 classification accuracy on the aforementioned datasets. For avoiding occasionality, each setting is measured with 5 runs. We also give the 95% confidence intervals calculated with $\mu \pm 1.96\frac{\sigma}{\sqrt{k}}$, where $\mu$, $\sigma$, and $k$ are the mean, standard deviation, and runs of the top-1 accuracy. A part of previous methods report no 95% confidence intervals, and therefore, we give the top-1 classification accuracy.

## Experimental Setup

We use ResNet-18 (He et al. 2016) without the classification head as the feature extractor $h(\cdot)$, and the backbone is pre-trained on ImageNet (Russakovsky et al. 2015). $h(\cdot)$ has 5 blocks including the top convolutional layer. We reduce the dimension of the outputted feature of ResNet from 512 to 128 with a linear block. For summarize, our $h(\cdot)$ returns 6 flattened features in total. The learning rate starts from 1e-4 and halves every 70 epochs. The batch size is set to 128. The hyper-parameter $\alpha$ in Eq.(10) is set to 0.1 during the first 70 epochs otherwise 0.2. Only the `RandomHorizontalFlip` and `ColorJitter` are adopted as the data augmentation schemes. The AdamW optimizer is used for training. The mean-std normalization is used based on the ImageNet statistics. GCPL (Yang et al. 2018) uses the same settings as stated above, and all other methods employ the default official settings. We store the models after the first 10 epochs based on the top-1 accuracy on the validation set. The number of prototypes $n$ is set to 3.

## Comparisons with State-of-the-Arts

We report the main results of domain generalization on common benchmarks in this section. If not specified, the ResNet-18 (He et al. 2016) is adopted as the backbone across different techniques. We use **Avg.** to represent the average top-1 classification accuracy over different domains.

**Leave-one-domain-out results on PACS.** Since we only collect limited results with 95% confidence intervals, we report the mean top-1 accuracy over 5 runs. Methods shown in the upper part of Table 1 use AlexNet (Krizhevsky, Sutskever, and Hinton 2012) as the backbone while the following part in gray background uses ResNet-18 (He et al. 2016). The vanilla counterpart of PoER is GCPL (Yang et al. 2018). PoER improves the top-1 classification accuracy up to 2.32% and 0.48% compared to its vanilla counterpart and the existing *state-of-the-art* methods. Methods shown below are arranged in the decreasing order of top-1 accuracy. A, C, P, and S in Table 1 indicate Art, Cartoon, Photo, and Sketch.

| Methods | A. | C. | P. | S. | Avg. |
|---|---|---|---|---|---|
| D-MATE (Ghifary et al. 2015b) | 60.27 | 58.65 | 91.12 | 47.68 | 64.48 |
| M-ADA (Qiao et al. 2020) | 61.53 | 68.76 | 83.21 | 58.49 | 68.00 |
| DBADG (Li et al. 2017) | 62.86 | 66.97 | 89.50 | 57.51 | 69.21 |
| MLDG (Li et al. 2018a) | 66.23 | 66.88 | 88.00 | 58.96 | 70.01 |
| Feature-critic (Li et al. 2019b) | 64.89 | 71.72 | 89.94 | 61.85 | 71.20 |
| CIDDG (Li et al. 2018c) | 66.99 | 68.62 | 90.19 | 62.88 | 72.20 |
| MMLD (Matsuura et al. 2020) | 69.27 | 72.83 | 88.98 | 66.44 | 74.38 |
| MASF (Dou et al. 2019) | 70.35 | 72.46 | 90.68 | 67.33 | 75.21 |
| EntropyReg (Zhao et al. 2020) | 71.34 | 70.29 | 89.92 | 71.15 | 75.67 |
| MMD-AAE (Li et al. 2018b) | 75.20 | 72.70 | 96.00 | 64.20 | 77.03 |
| CCSA (Motiian et al. 2017) | 80.50 | 76.90 | 93.60 | 66.80 | 79.45 |
| ResNet-18 (He et al. 2016) | 77.00 | 75.90 | 96.00 | 69.20 | 79.53 |
| StableNet (Zhang et al. 2021) | 80.16 | 74.15 | 94.24 | 70.10 | 79.66 |
| JiGen (Carlucci et al. 2019) | 79.40 | 75.30 | 96.00 | 71.60 | 80.50 |
| CrossGrad (Shankar et al. 2018) | 79.80 | 76.80 | 96.00 | 70.20 | 80.70 |
| DANN (Ganin et al. 2015) | 80.20 | 77.60 | 95.40 | 70.00 | 80.80 |
| Epi-FCR (Li et al. 2019a) | 82.10 | 77.00 | 93.90 | 73.00 | 81.50 |
| MetaReg (Balaji et al. 2018) | 83.70 | 77.20 | 95.50 | 70.30 | 81.70 |
| GCPL (Yang et al. 2018) | 82.64 | 75.02 | 96.40 | 73.36 | 81.86 |
| EISNet (Wang et al. 2020) | 81.89 | 76.44 | 95.93 | 74.33 | 82.15 |
| L2A-OT (Zhou et al. 2020c) | 83.30 | 78.20 | 96.20 | 73.60 | 82.83 |
| MixStyle (Zhou et al. 2021b) | 84.10 | **78.80** | 96.10 | 75.90 | 83.70 |
| PoER (Ours) | **85.30** | 77.69 | **96.42** | **77.30** | **84.18** |

Table 1: Leave-one-domain-out results on PACS dataset without 95% confidence intervals. The methods in gray background use ResNet-18 as the backbone while other methods employ AlexNet for feature extraction.

**Leave-one-domain-out results on OfficeHome dataset.** We report the mean top-1 accuracy and 95% confidence interval results on OfficeHome. Some of the following results are from DDAIG (Zhou et al. 2020a). We use the same method as stated in DDAIG to split the source domains into 90% for training and 10% for validation. The images in OfficeHome are colorful in the RGB channel whose scale scatters from 18 × 18 pixels to 6500 × 4900 pixels. The short edge of all images is resized to 227 first, maintaining the aspect ratio, and then the training inputs are obtained through `RandomResizedCrop` with shape 224. In Table 2, CCSA, MMD-AAE, and D-SAM are from (Motiian et al. 2017), (Li et al. 2018b), and (D'Innocente et al. 2018), and other methods have been introduced before. As stated in the previous section, the vanilla counterpart of PoER is GCPL. It can be found that PoER reduces the classification error by a clear margin of 1.3% and 1.2% compared to its vanilla counterpart and the *state-of-the-art* method DDAIG.

**Domain generalization results on NICO dataset.** NICO is different from the aforementioned dataset. It consists of two super-categories, namely Animal and Vehicle, including 19 sub-classes in total. Moreover, the domains of each sub-class are different from each other. In a nutshell, NICO contains 19 classes belonging to 65 domains. For each class, we randomly select 2 domains as the target while the left 8 domains are treated as the source. Within source domains, we further split the data into 90% for training and 10% for validation. The metrics are reported with the best-validated models on target domains. RSC indicates the algorithm from (Huang et al. 2020). No pre-trained weights are used in Table 3. PoER reports the superior performance by a remarkable margin of 2.83% compared to the existing methods.

**Leave-one-domain-out results on Digits-DG dataset.** Digits-DG consists of four different datasets containing digits with different appearances. Following DDAIG (Zhou et al. 2020a), all images are resized to 32 × 32 with RGB channel. For the MNIST dataset, we replicate the gray chan-

| Method | Artistic | Clipart | Product | Real World | Avg. |
|---|---|---|---|---|---|
| ResNet-18 | 58.9±.3 | 49.4±.1 | 74.3±.1 | 76.2±.2 | 64.7 |
| CCSA | 59.9±.3 | 49.9±.4 | 74.1±.2 | 75.7±.2 | 64.9 |
| MMD-AAE | 56.5±.4 | 47.3±.3 | 72.1±.3 | 74.8±.2 | 62.7 |
| CrossGrad | 58.4±.7 | 49.4±.4 | 73.9±.2 | 75.8±.1 | 64.4 |
| D-SAM | 58.0 | 44.4 | 69.2 | 71.5 | 60.8 |
| JiGen | 53.0 | 47.5 | 71.5 | 72.8 | 61.2 |
| GCPL | 58.3±.1 | 51.9±.1 | 74.1±.2 | 76.7±.1 | 65.3 |
| DDAIG | **59.2±.1** | 52.3±.3 | 74.6±.3 | 76.0±.1 | 65.5 |
| PoER (ours) | 59.1±.2 | **53.4±.3** | **74.9±.2** | **79.1±.3** | **66.6** |

Table 2: Leave-one-domain-out results on OfficeHome dataset with 95% confidence intervals. No confidence intervals are reported in the original paper of D-SAM and JiGen.

| | M-ADA | MMLD | ResNet-18 | |
|---|---|---|---|---|
| NICO | 40.78 | 47.18 | 51.71 | PoER (ours) |
| | JiGen | RSC | StableNet | **62.62** |
| NICO | 54.42 | 57.59 | 59.76 | |

Table 3: Domain generalization results on NICO dataset.

| Method | MNIST | MNIST-M | SVHN | SYN | Avg. |
|---|---|---|---|---|---|
| ResNet-18 | 95.8±.3 | 58.8±.5 | 61.7±.5 | 78.6±.6 | 73.7 |
| CCSA | 95.2±.2 | 58.2±.6 | 65.5±.2 | 79.1±.8 | 74.5 |
| MMD-AAE | 96.5±.1 | 58.4±.1 | 65.0±.1 | 78.4±.2 | 74.6 |
| CrossGrad | 96.7±.1 | 61.1±.5 | 65.3±.5 | 80.2±.2 | 75.8 |
| GCPL | 96.3±.1 | 58.7±.5 | 70.2±.3 | 80.5±.3 | 76.4 |
| DDAIG | 96.6±.2 | **64.1±.4** | 68.6±.6 | 81.0±.5 | 77.6 |
| PoER (ours) | **97.2±.4** | 60.1±.3 | **75.6±.4** | **94.4±.3** | **81.8** |

Table 4: Leave-one-domain-out results on Digits-DG with 95% confidence intervals.

| Method | VOC | LabelMe | Caltech | SUN09 | Avg. |
|---|---|---|---|---|---|
| DBADG | 69.99 | 63.49 | 93.64 | 61.32 | 72.11 |
| ResNet-18 | 67.48 | 61.81 | 91.86 | 68.77 | 72.48 |
| JiGen | 70.62 | 60.90 | 96.93 | 64.30 | 73.19 |
| MMLD | 71.96 | 58.77 | 96.66 | 68.13 | 73.88 |
| CIDDG | 73.00 | 58.30 | 97.02 | 68.89 | 74.30 |
| EntropyReg | 73.24 | 58.26 | 96.92 | 69.10 | 74.38 |
| GCPL | 67.01 | 64.84 | 96.23 | 69.43 | 74.38 |
| RSC | **73.81** | 62.51 | 96.21 | 72.10 | 76.16 |
| StableNet | 73.59 | 65.36 | 96.67 | **74.97** | **77.65** |
| PoER (ours) | 69.96 | **66.41** | **98.11** | 72.04 | 76.63 |

Table 5: Leave-one-domain-out results on VLCS dataset. PoER gets lower metrics on Pascal VOC and SUN09 while reporting superior performance on Caltech.

nel three times to construct the color images. As stated in (Zhou et al. 2020a), we randomly pick 600 images for each class in these four datasets. Images are split into 90% for training and 10% for validation. The leave-one-domain-out protocol is used for evaluated the domain generalization performance. All images in the left domain are tested for reporting metrics. Table 4 reveals the 95% confidence intervals of PoER and its comparisons with some existing domain generalization methods. It is clear to see that PoER surpasses previous techniques on most domains by a large margin, reducing the classification error up to 4.23% and achieving newly *state-of-the-art* domain generalization performance with only a vanilla ResNet-18 backbone. All methods shown in Table 4 are presented in previous sections.

**Leave-one-domain-out results on VLCS dataset.** VLCS consists of four common datasets. All methods shown in Table 5 can be found in Table 1 in detail. VOC indicates the Pascal VOC dataset. A part of the following results is from StableNet (Zhang et al. 2021). Following (Zhao et al. 2020), the leave-one-domain-out protocol is used for evaluation. The source domains are split into 70% for training and 30% for validation. The best-validated model reports domain generalization performance on all images from the left target domain. Since the size of images in VLCS is varied from each other, we thus resize the short edge of images to 227 while keeping their aspect ratio and then randomly crop squares with shape 224 for training. Results depicted in Table 5 reveal that PoER gives a better classification accuracy surpassing other methods with a large improvement, saying 1.44% outperforming the current techniques on Caltech.

## Ablation Study

**Ablation on the weight of PoER $\alpha$.** In Eq.(10), we introduce a hyper-parameter $\alpha$ for balancing the classification loss and energy ranking loss. We set this parameter from 0.0 to 0.9 with a step 0.1 for testing its sensitivity. To clarify, $\alpha$ equals to 0 indicates the vanilla counterpart of PoER, *i.e.,* GCPL. We use the PACS benchmark and set the number of prototypes $k$ to 3. The ablation results are shown in Table 6. From Table 6, we find that the performance is better when setting $\alpha$ to 0.2, considering the training stability, we set $\alpha$ to 0.1 in the first 70 epochs and otherwise 0.2.

**Ablation on the number of prototypes $k$.** In Eq.(8), we set the learnable prototypes as a tensor with shape $k \times n \times m$ where the $k$, $n$, and $m$ are the number of classes, the number of prototypes, and the dimensionality of the outputted feature from the last block. We test the impacts of the number of prototypes with respect to the classification performance from 1 to 10 with step 1 on PACS benchmark. From Table 7, we find that more prototypes guarantee a better classification result to some extent. Considering both the performance and calculation efficiency, we set the number of prototypes $k$ to 3 default, leading to metrics which are marginally lower than the best one.

**Ablation on the proposed loss functions.** In Eq.(6) and Eq.(7), we propose calculating ranking loss in the shallow layers while performing clustering regularization in the deeper layers. Taking ResNet-18 as an example, we extract features from the first three blocks (including the first convolutional layer) for calculating the ranking loss as shown in Eq.(6) while the features extracted from the following blocks are treated as the deeper layers for getting clustering loss as shown in Eq.(7). We test the combinations of different loss functions, namely $\mathcal{L}_{\texttt{cls}}$, $\mathcal{L}_{\texttt{cls}} + \alpha\mathcal{L}_{\texttt{rank}}$, $\mathcal{L}_{\texttt{cls}} + \alpha\mathcal{L}_{\texttt{cluster}}$, and $\mathcal{L}_{\texttt{cls}} + \alpha\mathcal{L}_{\texttt{PoER}}$. Noting that $\mathcal{L}_{\texttt{PoER}}$

| $\alpha$ | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 |
|---|---|---|---|---|---|
| PACS Avg. | 81.86 | 83.92 | **84.20** | 84.16 | 84.16 |
| $\alpha$ | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| PACS Avg. | 84.13 | 84.06 | 84.12 | 83.97 | 83.60 |

Table 6: Ablation results on hyper-parameter $\alpha$.

| $k$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| PACS Avg. | 82.40 | 83.97 | 84.18 | 84.12 | 84.07 |
| $k$ | 6 | 7 | 8 | 9 | 10 |
| PACS Avg. | 84.20 | 84.18 | 84.19 | 84.19 | **84.21** |

Table 7: Ablation results on the number of prototypes $k$.

actually equals to $\mathcal{L}_{\texttt{rank}} + \mathcal{L}_{\texttt{cluster}}$. Recall that $\mathcal{L}_{\texttt{cls}}$ indicates the vanilla counterpart of PoER, *i.e.,* GCPL. We use the default settings of $\alpha$ and $k$ as stated in previous sections. The ablation results on different combinations of loss functions are presented in Table 8. Metrics are evaluated on the PACS benchmark. From Table 8, we find that both $\mathcal{L}_{\texttt{rank}}$ and $\mathcal{L}_{\texttt{cluster}}$ help to improve the domain generalization ability of conventional neural networks. Noting that $\mathcal{L}_{\texttt{cluster}}$ can be treated as the self-supervised manner for boosting domain generalization performance.

| *Loss* | $\mathcal{L}_{\texttt{cls}}$ | $\mathcal{L}_{\texttt{cls}} + \alpha\mathcal{L}_{\texttt{rank}}$ |
|---|---|---|
| **PACS Avg.** | 81.86 | 83.72 |
| *Loss* | $\mathcal{L}_{\texttt{cls}} + \alpha\mathcal{L}_{\texttt{cluster}}$ | $\mathcal{L}_{\texttt{cls}} + \alpha\mathcal{L}_{\texttt{PoER}}$ |
| **PACS Avg.** | 82.60 | **84.18** |

Table 8: Ablation results on different combinations of the proposed loss functions. Especially, as shown above, the group with $\mathcal{L}_{\texttt{rank}}$ surpasses $\mathcal{L}_{\texttt{cluster}}$ up to 1.12%, suggesting the importance of domain ranking proposed in PoER.

**Ablation on the scalability of PoER.** We investigate the scalability of PoER among different model architectures. To verify this, we employ both the convolution- and attention-based models to perform experiments on NICO (He, Shen, and Cui 2021) dataset. Since the attention-based models are hard to converge, we resort to the pre-trained weights on ImageNet (Russakovsky et al. 2015). The convolution-based models are trained from scratch. Noting that we focus more on the improvement that PoER introduces to the corresponding vanilla counterparts. ResNet (He et al. 2016) and DenseNet (Huang et al. 2017) are the representatives of convolutionl models while Swin-Transformer (Tiny) (Liu et al. 2021) is that of attention-based models. Results in Table 9 demonstrate that PoER improves the top-1 classification accuracy over 3.55% consistently, evidencing its scalability.

## Feature Visualization Results

We provide the feature distribution within the identical category from the shallow layers to the deeper. Recall that in the shallow layers, PoER employs the ranking loss to make the model aware of the difference across different domains, thus we expect the distinction among varied domains within the identical category to become greater progressively.

| Model | RN-18 | RN-50 | DN-121 | Swin-T |
|---|---|---|---|---|
| Avg. (✗) | 51.71 | 55.43 | 65.19 | 73.43 |
| Avg. (✓) | 62.62 | 64.10 | 69.75 | 76.98 |

Table 9: Ablation results of different model architectures on NICO dataset. RN, DN, and Swin-T indicate ResNet, DenseNet, and Swin-Transformer. ✗ is the vanilla classifier while ✓ is the corresponding PoER version.

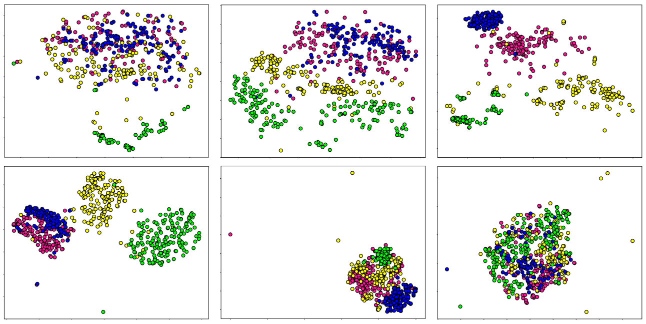

Figure 3: Visualization results of the feature distribution in each layer. Different color indicates different domains. Images above are feature distributions with identical category label from block 1 to block 6 (from left to right and from top to bottom) successively on PACS.

From the feature distribution of each block shown in Figure 3, it can be concluded that in the first three blocks (the first row), PoER employs domain ranking loss to make the neural network aware of the differences across domains, separating the features from different domains. In the following blocks (the second row), PoER aims to filter the domain-related information out for clustering, making the features with identical category labels close together no matter the domains. As stated in the beginning, PoER learns the characters of each domain and category before generating domain-invariant features, laying the foundation for distilling pure label-related features.

## Conclusion

This paper proposes using PoER to make the classifier aware of the characters of different domains and categories before generating domain-invariant features. We find and verify that PoER is vital and helpful for improving the generalization ability of models across domains. PoER reports superior results on sufficient domain generalization benchmarks compared to existing techniques, achieving *state-of-the-art* performance. Insights of the proposed idea are given both statistically and visually. We hope the mentioned energy perspective can inspire the following works.

## Acknowledgments

This research was supported by the National Key Research and Development Program of China under Grant No.2020AAA0109702, the National Natural Science Foundation of China under Grants 61976208, 62076242, and the InnoHK project.

## References

Akada, H.; Bhat, S. F.; Alhashim, I.; and Wonka, P. 2022. Self-Supervised Learning of Domain Invariant Features for Depth Estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 3377–3387.

Balaji, Y.; Sankaranarayanan, S.; Chellappa, R.; and Chellappa, R. 2018. MetaReg: Towards Domain Generalization using Meta-Regularization. In *Advances in Neural Information Processing Systems 31: NeurIPS 2018*.

Carlucci, F. M.; D'Innocente, A.; Bucci, S.; Caputo, B.; and Tommasi, T. 2019. Domain Generalization by Solving Jigsaw Puzzles. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.

Choi, M. J.; Lim, J. J.; Torralba, A.; and Willsky, A. S. 2010. Exploiting hierarchical context on a large database of object categories. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010*.

D'Innocente, A.; Caputo, B.; Caputo, B.; and D'Innocente, A. 2018. Domain Generalization with Domain-Specific Aggregation Modules. In *Pattern Recognition - 40th German Conference, GCPR 2018*.

Dou, Q.; de Castro, D. C.; Kamnitsas, K.; and Glocker, B. 2019. Domain Generalization via Model-Agnostic Learning of Semantic Features. In *Advances in Neural Information Processing Systems 32: NeurIPS 2019*.

Everingham, M.; Gool, L. V.; Williams, C. K. I.; Winn, J. M.; and Zisserman, A. 2010. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.*

Fei-Fei, L.; Fergus, R.; Perona, P.; and Perona, P. 2004. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2004*.

Ganin, Y.; Lempitsky, V. S.; Ganin, Y.; and Lempitsky, V. S. 2015. Unsupervised Domain Adaptation by Backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*.

Ghifary, M.; Kleijn, W. B.; Zhang, M.; and Balduzzi, D. 2015a. Domain Generalization for Object Recognition with Multi-task Autoencoders. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society.

Ghifary, M.; Kleijn, W. B.; Zhang, M.; and Balduzzi, D. 2015b. Domain Generalization for Object Recognition with Multi-task Autoencoders. In *2015 IEEE International Conference on Computer Vision, ICCV 2015*, 2551–2559. IEEE Computer Society.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, Y.; Shen, Z.; and Cui, P. 2021. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, 110: 107383.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Huang, X.; and Belongie, S. J. 2017. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In *IEEE International Conference on Computer Vision, ICCV 2017*, 1510–1519. IEEE Computer Society.

Huang, Z.; Wang, H.; Xing, E. P.; and Huang, D. 2020. Self-challenging Improves Cross-Domain Generalization. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*.

Kang, J.; Lee, S.; Kim, N.; and Kwak, S. 2022. Style Neophile: Constantly Seeking Novel Styles for Domain Generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7130–7140.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE.*

Li, B.; Wang, Y.; Zhang, S.; Li, D.; Keutzer, K.; Darrell, T.; and Zhao, H. 2021. Learning Invariant Representations and Risks for Semi-Supervised Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1104–1113.

Li, D.; Yang, Y.; Song, Y.; and Hospedales, T. M. 2017. Deeper, Broader and Artier Domain Generalization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 5543–5551. IEEE Computer Society.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. 2018a. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Li, D.; Zhang, J.; Yang, Y.; Liu, C.; Song, Y.; and Hospedales, T. M. 2019a. Episodic Training for Domain Generalization. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV*.

Li, H.; Pan, S. J.; Wang, S.; and Kot, A. C. 2018b. Domain Generalization With Adversarial Feature Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*.

Li, Y.; Tian, X.; Gong, M.; Liu, Y.; Liu, T.; Zhang, K.; and Tao, D. 2018c. Deep Domain Generalization via Conditional Invariant Adversarial Networks. In *Computer Vision - ECCV 2018 - 15th European Conference*.

Li, Y.; Yang, Y.; Zhou, W.; and Hospedales, T. 2019b. Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning*, 3915–3924. PMLR.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *Proceedings* *of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Matsuura, T.; Harada, T.; Matsuura, T.; and Harada, T. 2020. Domain Generalization Using a Mixture of Multiple Latent Domains. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press.

Motiian, S.; Piccirilli, M.; Adjeroh, D. A.; and Doretto, G. 2017. Unified Deep Supervised Domain Adaptation and Generalization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society.

Muandet, K.; Balduzzi, D.; and Schölkopf, B. 2013. Domain Generalization via Invariant Feature Representation. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, 10–18. JMLR.org.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Nguyen, A. T.; Tran, T.; Gal, Y.; and Baydin, A. G. 2021. Domain invariant representation learning with domain density transformations. *Advances in Neural Information Processing Systems*, 34: 5264–5275.

Qiao, F.; Zhao, L.; Peng, X.; and Peng, X. 2020. Learning to Learn Single Domain Generalization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 12553–12562. Computer Vision Foundation / IEEE.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*.

Russell, B. C.; Torralba, A.; Murphy, K. P.; and Freeman, W. T. 2008. LabelMe: A Database and Web-Based Tool for Image Annotation. *Int. J. Comput. Vis.*

Shankar, S.; Piratla, V.; Chakrabarti, S.; Chaudhuri, S.; Jyothi, P.; and Sarawagi, S. 2018. Generalizing Across Domains via Cross-Gradient Training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Shen, Z.; Liu, J.; He, Y.; Zhang, X.; Xu, R.; Yu, H.; and Cui, P. 2021. Towards Out-Of-Distribution Generalization: A Survey. *CoRR*, abs/2108.13624.

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2014. Deep Domain Confusion: Maximizing for Domain Invariance. *CoRR*, abs/1412.3474.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. S. 2016. Instance Normalization: The Missing Ingredient for Fast Stylization. *CoRR*, abs/1607.08022.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep Hashing Network for Unsupervised Domain Adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society.

Wang, S.; Yu, L.; Li, C.; Fu, C.; and Heng, P. 2020. Learning from Extrinsic and Intrinsic Supervisions for Domain Generalization. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IX*.

Wang, Z.; Wang, Z.; Yu, Z.; Deng, W.; Li, J.; Gao, T.; and Wang, Z. 2022. Domain Generalization via Shuffled Style Assembly for Face Anti-Spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4123–4133.

Yang, H.; Zhang, X.; Yin, F.; and Liu, C. 2018. Robust Classification With Convolutional Prototype Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*.

Zhang, X.; Cui, P.; Xu, R.; Zhou, L.; He, Y.; and Shen, Z. 2021. Deep Stable Learning for Out-of-Distribution Generalization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE.

Zhang, Y.; Li, M.; Li, R.; Jia, K.; and Zhang, L. 2022. Exact Feature Distribution Matching for Arbitrary Style Transfer and Domain Generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8035–8045.

Zhao, S.; Gong, M.; Liu, T.; Fu, H.; and Tao, D. 2020. Domain Generalization via Entropy Regularization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhou, K.; Yang, Y.; Hospedales, T. M.; and Xiang, T. 2020a. Deep Domain-Adversarial Image Generation for Domain Generalisation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*.

Zhou, K.; Yang, Y.; Hospedales, T. M.; and Xiang, T. 2020b. Learning to Generate Novel Domains for Domain Generalization. In *Computer Vision - ECCV 2020 - 16th European Conference*.

Zhou, K.; Yang, Y.; Hospedales, T. M.; and Xiang, T. 2020c. Learning to Generate Novel Domains for Domain Generalization. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVI*.

Zhou, K.; Yang, Y.; Qiao, Y.; and Xiang, T. 2021a. Domain Adaptive Ensemble Learning. *IEEE Transactions on Image Processing (TIP)*.

Zhou, K.; Yang, Y.; Qiao, Y.; and Xiang, T. 2021b. Domain Generalization with MixStyle. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Zhou, Y.; Huang, L.; Zhou, T.; and Shao, L. 2021c. CCT-Net: Category-Invariant Cross-Domain Transfer for Medical Single-to-Multiple Disease Diagnosis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 8260–8270.