

Compound Domain Generalization via Meta-Knowledge Encoding

Chaoqi Chen¹, Jiongcheng Li^{2,4}, Xiaoguang Han³, Xiaoqing Liu², Yizhou Yu^{1*}

¹ The University of Hong Kong ² Deepwise AI Lab

³ The Chinese University of Hong Kong (Shenzhen) ⁴ Xiamen University

cqchen1994@gmail.com, jiongchengli@stu.xmu.edu.cn

hanxiaoguang@cuhk.edu.cn, Liuxiaoqing@deepwise.com, yizhouy@acm.org

Abstract

Domain generalization (DG) aims to improve the generalization performance for an unseen target domain by using the knowledge of multiple seen source domains. Mainstream DG methods typically assume that the domain label of each source sample is known a priori, which is challenged to be satisfied in many real-world applications. In this paper, we study a practical problem of compound DG, which relaxes the discrete domain assumption to the mixed source domains setting. On the other hand, current DG algorithms prioritize the focus on semantic invariance across domains (one-vs-one), while paying less attention to the holistic semantic structure (many-vs-many). Such holistic semantic structure, referred to as meta-knowledge here, is crucial for learning generalizable representations. To this end, we present **COM**ound domain generalization via **ME**ta-knowledge **EN**coding (COMEN), a general approach to automatically discover and model latent domains in two steps. Firstly, we introduce *Style-induced Domain-specific Normalization (SDNorm)* to re-normalize the multi-modal underlying distributions, thereby dividing the mixture of source domains into latent clusters. Secondly, we harness the prototype representations, the centroids of classes, to perform relational modeling in the embedding space with two parallel and complementary modules, which explicitly encode the semantic structure for the out-of-distribution generalization. Experiments on four standard DG benchmarks reveal that COMEN exceeds the state-of-the-art performance without the need of domain supervision.

1. Introduction

The success of many computer vision algorithms hinges on a strong presumption that the training and test data are independent and identically distributed (i.i.d.). In practice, however, this hypothesis is prone to be violated due to the

*Corresponding author

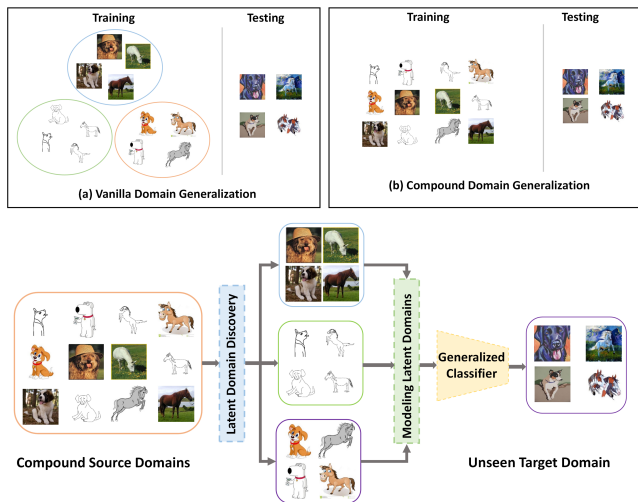


Figure 1. **Top:** (a) Vanilla domain generalization. (b) Compound domain generalization. **Bottom:** The workflow of our approach.

change of environments, imaging devices, selection bias, to name a few. How to generalize a well-trained model to out-of-distribution domains has motivated a body of research on Domain Adaptation (DA) [15, 37, 48, 52, 54, 68, 70] and Domain Generalization (DG) [2, 8, 9, 31, 34, 41]. In contrast to DA problem where the unlabeled or partially labeled target data is available, DG considers a harder problem setting where a model trained on a set of source domain data should directly generalize to an unseen target domain with different data statistics, without the need of accessing to target domain data for retraining or fine-tuning (cf. Fig. 1 (a)).

Mainstream DG literature typically assumes that the domain label of each source sample is known a priori, which is too restrictive to be satisfied in many practical applications. For instance, with the explosive increase of the multi-source data from Internet, a mixture of multiple source data can be easily collected. However, manually dividing the crawled images into multiple domains requires tedious labor-intensive work given a large amount of training data. Meanwhile, the latent domains may blend and interact in

complex ways. Thus, it is non-trivial and significant to automatically discover and model distinct underlying domains.

In this paper, we investigate a practical problem of *compound domain generalization*, where the domain label is unavailable for training (cf. Fig. 1 (b)). An intuitive approach is to use the embedded features along with clustering strategies to conduct domain separation in advance. However, the source samples are naturally divided into different clusters according to their semantic categories rather than the domain-wise distinctions. When the source data contains heterogeneous but unknown latent domains, most conventional DG methods will not be applicable. Although some recent works [2, 23, 33, 43] do not require domain label information during training, the informative latent domain structure contained in the source data is neglected.

On the other hand, existing DG methods, which mainly span domain invariant feature learning [16, 34, 35, 41, 42, 69], gradient based meta-learning [9, 30, 31], and augmentation based generalization [50, 58], are devoted to learn semantic representations in virtue of one-vs-one consistency constraints. Unfortunately, the learned representations may be insufficient to precisely encode the semantic information. By prioritizing the focus on semantic invariance across domains, the holistic semantic structure, which contains rich many-vs-many information regarding the inter-class relations and interactions, is yet to be thoroughly investigated. In particular, we refer such holistic semantic structure as *meta-knowledge*, namely, knowledge that is domain-agnostic and generalizable for unseen target domains.

Grounded on these findings, we present a novel DG framework called **CO**mpound domain generalization via **ME**ta-knowledge **EN**coding (COMEN). The basic idea of the COMEN is to identify latent domain structure and model the semantic correlations among different categories across domains. To achieve this goal, we establish the COMEN in two stages. During the first stage, we introduce Style-induced Domain-specific Normalization (SDNorm) to reveal the complex combinations of latent domains and learn a set of domain assignment probabilities for each source sample. SDNorm is end-to-end trainable and can be easily plugged into modern deep neural networks. In the second stage, based on the discovered latent domains, we resort to prototypes (the feature centroids of categories) to model the relations among different semantic categories via two parallel modules, *i.e.*, Prototypical Graph Reasoning (ProtoGR) and Prototypical Category-aware Contrastive Learning (ProtoCCL). ProtoGR helps each categorical prototype attend and reason over its neighborhoods’ prototypical features instead of using pairwise alignment, thereby capturing the topological structure of semantic space. ProtoCCL compensates for the shortage of samples in the prototypical feature space by contrastively learning the relations of different categories while preserving their discriminability.

These two modules work in a synergistic manner towards encoding semantic structures into the embedding space.

Our contributions can be summarized as follows:

- We introduce a more practical compound DG setting that imposes no prior knowledge on the domain label of each source sample. Then, a unified learning framework, called COMEN, is introduced to jointly discover and model the latent domains.
- In terms of latent domain discovery, we uncover and re-normalize the multimodal underlying distributions with the proposed SDNorm.
- In terms of encoding the semantic structure, we propose two complementary modules, ProtoGR and ProtoCCL, to explicitly explore the relations and interactions of prototype in the common feature space.
- Experiments on four standard DG benchmarks (PACS, Digits-DG, VLCS, and Office-Home) demonstrate that COMEN outperforms the state-of-the-art methods by large margins without the need of domain supervision.

2. Related Work

2.1. Domain Generalization

Domain generalization [59], which aims at generalizing model learned from multiple seen source domains to an unseen target domain, have received a great surge of interest in the deep learning era. In practice, numerous approaches have been developed to learn a good predictive model for the unseen test domain with generalizable representations.

Domain Alignment. An intuitive solution for domain generalization is to extract features that are domain-invariant but preserve discriminability for the task prediction. For example, Ghifary *et al.* [16] propose to extract invariant features with a multi-task auto-encoder. Motivated by the success of mainstream domain adaptation methods, domain adversarial training [15] are introduced to adversarially learn generalizable representations across multiple source domains via a two-player min-max game [34, 35, 40].

Augmentation. Volpi *et al.* [58] augment the dataset with a new adversarial data augmentation procedure by synthesizing the “hard” samples for the training model, and then provide an iterative updating procedure. Xu *et al.* [66] resort to randomized convolutions by leveraging the outputs of random convolutions as new images or mixing them with the original inputs during training. By contrast, feature augmentation, such as mixstyle [73], curriculum mixup [38], and stochastic feature augmentation [33], is newly investigated to augment the latent feature space without explicitly changing the original inputs. In addition, Carlucci *et al.* [2] use a self-supervision loss, called solving

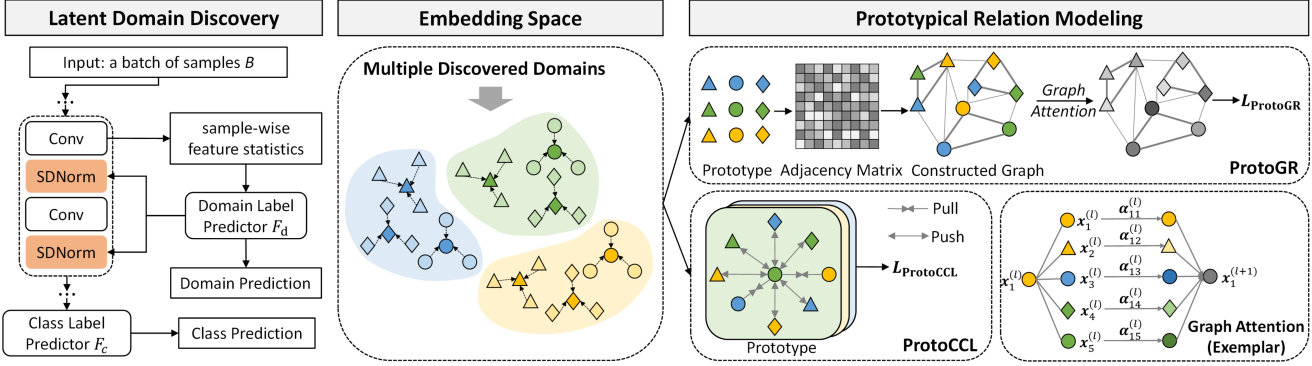


Figure 2. The overall structure of COMEN. Given a mixture of source domains, it discovers the latent domain structure via SDNorm in the first stage. During the second stage, it first converts all samples into the embedding space with pseudo domain labels (based on the domain prediction). Then, a set of prototypes are computed and fed into two parallel modules, ProtoGR and ProtoCCL.

jigsaw puzzles, to enhance the generalization ability of deep model by distinguishing different jigsaw puzzles.

Meta-Learning. A category of methods resort to the meta-learning [13] by constructing meta-train and meta-test domains during training time for mimicking the train and test distributional disparity. Li *et al.* [30] and Balaji *et al.* [1] propose to update the network parameters by minimizing the meta-optimization objective. Li *et al.* [31] first develop a strong baseline by aggregating data from all source domains and training a deep model on all the data, and then a episodic training strategy is proposed to mimic the domain shift. Dou *et al.* [9] propose to explicitly enforce the cross-domain semantic coherence via metric learning losses along with the episode training strategy to enhance the semantic robustness. Despite their general efficacy, meta-learning methods are notoriously slow to train.

Feature Decomposition. Khosla *et al.* [25] propose to a decomposition-based approaches that decompose network parameters to domain-invariant and domain-specific low-rank components. This idea was further extended by [29] to the deep learning model. A low common-specific low-rank decomposition algorithm [46] is proposed to achieve efficient domain generalization by only modifying the final linear classification layer of deep networks.

2.2. Multiple Latent Domains

Learning from multiple latent domains is a long-standing and challenging problem in machine learning and computer vision community. Some previous works have explored the latent domain discovery problem, but most in the regime of domain adaptation. In particular, Hoffman *et al.* [21] develop a hierarchical clustering technique to achieve domain separation for the following adaptation procedure. Gong *et al.* [17] aim to reshape the visual datasets by imposing two properties (distinctiveness and learnability) on domains. Xiong *et al.* [65] draw motivation from multi-

ple manifold learning to infer the domain assignment by a new local subspace representation. Mancini *et al.* [39] propose to automatically discover latent domains by using a set of mDA-layers to estimate the domain assignment of each source sample. In contrast, how to discover and model latent domains are less investigated in DG especially under the presence of unseen target domains. More importantly, exploring the topological relations among multiple latent domains remains out of reach for current DG approaches.

3. Methodology

The overall architecture of COMEN is depicted in Fig. 2, which consists of three components, SDNorm, ProtoGR, and ProtoCCL. SDNorm statistically estimates the domain membership of each sample in an unsupervised fashion to decouple the multi-modal data distributions. ProtoGR and ProtoCCL are built on the basis of prototype representations to model the both intra- and inter-domain semantic relations among different categories, thereby encoding the semantic structure into the learned feature spaces.

3.1. Task Formulation

In vanilla domain generalization problem, we are given M source domains $\mathcal{D}_s = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_M\}$ that are sampled from different probability distributions on the joint space $\mathcal{X} \times \mathcal{Y}$. It requires the presence of domain labels, thus the m -th source domain can be represented as $\mathcal{D}_m = \{(x_i^m, y_i^m, d_i^m)\}_{i=1}^{N_m}$, where N_m denotes the number of samples in this domain. In our work, we focus on the compound domain generalization problem which assumes that the domain labels of source data are not available, *i.e.*, the source data has mixed domains $\mathcal{D}_s = \{(x_i, y_i)\}_{i=1}^N$, where N denotes the number of all source samples. Both vanilla and compound domain generalization assume that different source domains share the identical label space, and we consider that there are K classes in all.

3.2. Latent Domain Discovery

A strong baseline [29] for compound (or vanilla) DG is to train a prediction model by simply combining all source samples. However, this approach inevitably deteriorates the performance on the domain of interest [53] as the intrinsic inter-domain relationships are not considered. This phenomenon indicates that the underlying multi-modal information is crucial for the success of DG methods. On the other hand, discovering the latent domains in the context of DG remains an open question since the typical representation learning method are naturally devised to divide the input data into different semantic categories, and most existing DG approaches require the domain information to implement their frameworks. In addition, using embedded representations along with standard clustering approaches, such as k -means, may not lead to satisfactory estimation results with respect to the domain membership of each sample. Remedying this issue, we propose to uncover the multi-modal distribution patterns with a simple and effective Style-induced Domain-specific Normalization (SDNorm) module, which can be easily plugged into modern deep neural networks and DG methods.

Suppose that we are given feature maps of an image extracted from a CNN layer $\mathbf{f} \in \mathbb{R}^{C \times H \times W}$, where C , H and W represent the channel, height, and width of the feature maps. Technically, we resort to the channel-wise mean and standard deviation μ , σ of \mathbf{f} to obtain its style representation. The definition is formulated as,

$$\mu(\mathbf{f}) = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W f_{h,w} \quad (1)$$

$$\sigma(\mathbf{f}) = \sqrt{\frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W (f_{h,w} - \mu(\mathbf{f}))^2 + \epsilon} \quad (2)$$

Then, we concatenate the computed $\mu(\mathbf{f})$ and $\sigma(\mathbf{f})$ in each channel to represent the style representation, $\text{sty}(\mathbf{f}) \in \mathbb{R}^{2C}$. The style representation $\text{sty}(\mathbf{f})$ is further fed into a domain label predictor F_d ¹ to output a set of domain assignment probabilities. Formally, we optimize this domain prediction module by minimizing the entropy of its output,

$$\mathcal{L}_d = -\frac{1}{N} \sum_{i=1}^N F_d(\text{sty}(\mathbf{f})_i) \log F_d(\text{sty}(\mathbf{f})_i) \quad (3)$$

We further incorporate the computed domain assignment probabilities into Batch Normalization (BN) [24] layer to re-normalize the multi-modal distributions. BN is devised to standardize the features of each layer to zero-mean and univariance. Suppose that we are given an input feature map \mathbf{z} in a certain channel to a BN layer, then BN layer transforms the feature by incorporating the domain information

¹ F_d was pretrained on a portion of samples with pseudo domain labels.

in the following manner,

$$\hat{\mathbf{z}}_m = \frac{\mathbf{z} - \mu_m}{\sqrt{\sigma_m^2 + \epsilon}} \quad (4)$$

where $\mu_m = \sum_{i=1}^{|B|} p_{i,m} \mathbf{z}^i$ and $\sigma_m^2 = \sum_{i=1}^{|B|} p_{i,m} (\mathbf{z}^i - \mu_m)^2$. $p_{i,m}$ represents the predicted probability from F_d that an input image x_i belongs to the m -th domain. Accordingly, the proposed SDNorm can be formulated as a BN-like module,

$$\text{SDNorm}_m(\mathbf{z}; \lambda_m, \beta_m) = \lambda_m \cdot \hat{\mathbf{z}}_m + \beta_m \quad (5)$$

where λ_m and β_m are learnable parameters for domain m . We perform SDNorm for different source domains by using different bias/variance terms by following the prior practices [3, 4, 36, 39, 49, 62] in domain adaptation. Note that we also normalize each channel independently in each mini-batch, but we omit the subscript for the sake of simplicity.

We highlight that the latent domain discovery is an unsupervised process, thus the false domain assignment will inevitably arise during training. Different from representative latent domain discovery methods [17, 21, 40], which assign a deterministic domain label (*i.e.* hard domain label) to each source sample, our approach computes a set of probabilities (*i.e.* soft domain label) for each source sample. In this way, the negative influence of false domain assignment will be mitigated during the feature re-normalization process.

3.3. Prototypical Relation Modeling

Mainstream DG approaches are devoted to enforce semantic consistency in the shared embedding space via alignment-based strategies, such as moment matching [42, 52] and adversarial learning [35, 40]. However, the semantic relations among different categories, which are crucial for depicting the semantic structures, are less investigated. As a consequence, only seeking one-vs-one category alignment cannot ensure learning generalizable representations for the unseen target domain. On the other hand, *prototype* has demonstrated strong efficacy in few-shot learning [51], domain adaptation [5, 6, 64, 67], and unsupervised learning [32] by encoding semantic structure under the circumstances of insufficient or noisy annotated data. In DG, we encounter with a new challenge that how to generalize a learned semantic structure to the unseen domain in an unsupervised fashion. To solve the above issues, we propose to model the *prototypical relations* among different semantic categories via two parallel and complementary modules called Prototypical Graph Reasoning (ProtoGR) and Prototypical Category-aware Contrastive Learning (ProtoCCL). ProtoGR seeks to help each category prototype attend and reason over its neighborhoods' prototypical features instead of using pairwise alignment, thereby capturing the topological information of semantic structure. ProtoCCL compensates for the limited sample size in the prototypical fea-

ture space by contrastively learning the domain relationships and preserving the category discriminability.

First of all, we compute the global prototype c_m^k for each category in different discovered latent domains. The global prototype is a mean vector of the embedded samples in each category via a mapping function f ,

$$c_m^k = \frac{1}{|\mathcal{D}_m^k|} \sum_{(x_i^{\mathcal{D}_m}, y_i^{\mathcal{D}_m}) \in \mathcal{D}_m^k} f(x_i^{\mathcal{D}_m}), \quad (6)$$

where \mathcal{D}_m^k represents the set of samples labeled with class k in the m -th source domain. In order to harmonize the training process, we follow the prior practice [7,64] to update the global class prototype by using a mini-batch sampled from all source domains through an exponential moving average strategy,

$$c_{m(I)}^k \leftarrow \rho c_{m(I-1)}^k + (1 - \rho) \hat{c}_{m(I)}^k \quad (7)$$

where ρ denotes the exponential decay rate and is set to 0.7 in all experiments, $\hat{c}_{m(I)}^k$ is the local class prototype, and I represents the iteration times.

3.3.1 Prototypical Graph Reasoning (ProtoGR)

To implement ProtoGR, we utilize a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with nodes $v_i \in \mathcal{V}$ and edges $(v_i, v_j) \in \mathcal{E}$. In this graph, \mathcal{V} corresponds to $M \times K$ global class prototypes, and the adjacency matrix is defined as $\mathbf{A} \in \mathbb{R}^{MK \times MK}$ which corresponds to each edge (v_i, v_j) with its element A_{ij} . The node features are denoted by $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_{MK}\} \in \mathbb{R}^{MK \times d}$, where each element corresponds to a node in \mathcal{V} and d is the dimension of the prototype feature. Specifically, we define the correlations between nodes,

$$A_{ij} = \mathbb{1}\left(\frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \cdot \|\mathbf{x}_j\|_2} > \delta\right) \cdot \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \cdot \|\mathbf{x}_j\|_2} \quad (8)$$

where \mathbf{x}_i and \mathbf{x}_j denote the i -th and j -th prototype features, and δ is the threshold parameter that control the sparsity of \mathcal{G} . We fix δ to 0.5 in all experiments. $\mathbb{1}(\cdot)$ represents the indicator function.

To model the long-range dependencies and enhance the expressive power of graph nodes, we introduce graph attention mechanism [56] to our model,

$$\mathbf{x}_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(l)} \cdot \mathbf{W} \mathbf{x}_j^{(l)}\right), \quad (9)$$

where $\mathbf{x}_j^{(l)}$ is the hidden feature for node j in the l -th layer, \mathbf{W} is a weight matrix, and σ is a nonlinear function such as ReLU. Given two nodes i and j in \mathcal{G} , their edge weights are formulated as,

$$\alpha_{ij}^{(l)} = \frac{A_{ij} \exp(\text{LReLU}(\mathbf{a}_{(l)}^T [\mathbf{W} \mathbf{x}_i^{(l)} \parallel \mathbf{W} \mathbf{x}_j^{(l)}]))}{\sum_{k \in \mathcal{N}_i} A_{ik} \exp(\text{LReLU}(\mathbf{a}_{(l)}^T [\mathbf{W} \mathbf{x}_i^{(l)} \parallel \mathbf{W} \mathbf{x}_k^{(l)}]))} \quad (10)$$

where \mathcal{N}_i represents the neighborhood of node i in the graph, $\mathbf{a}_{(l)}$ is a learnable weight vector, and \parallel denotes the concatenation operation. In practice, we stack two graph attention layers to form our ProtoGR. Moreover, we combine the hidden features with the input features in the last layer to enhance the discriminability,

$$\mathbf{X}^{(L)} = \mathbf{X}^{(L)} + \mathbf{X}^{(0)} \quad (11)$$

To further strengthen the discriminability of class prototypes, we perform node classification based on \mathcal{G} . To be specific, the last layer of ProtoGR predicts the class label, which can be formulated as,

$$\hat{y} = \text{softmax}(\text{FC}(\text{ProtoGR}(x, \mathcal{G}))), \quad (12)$$

where x is the node features, \hat{y} denotes the predicted label, and FC denotes a fully-connected layer. Finally, the node classification loss of ProtoGR module is defined as $\mathcal{L}_{\text{ProtoGR}}$.

3.3.2 Prototypical Category-aware Contrastive Learning (ProtoCCL)

In addition to the topological relations among prototypes learned by ProtoGR, we further encode the semantic structures into the embedding space by introducing a contrastive learning objective for the task of category discrimination.

Contrastive learning based methods [18, 19, 22, 32, 61] have demonstrated compelling results in (self-supervised) representation learning. Among them, InfoNCE [45] is a representative contrastive loss function,

$$\mathcal{L}_I^{\text{InfoNCE}} = -\log \frac{\exp(\mathbf{v} \cdot \mathbf{v}^+ / \tau)}{\exp(\mathbf{v} \cdot \mathbf{v}^+ / \tau) + \sum_{\mathbf{v}^- \in \mathcal{N}_I} \exp(\mathbf{v} \cdot \mathbf{v}^- / \tau)} \quad (13)$$

where \mathbf{v}^+ is positive embeddings for I , \mathcal{N}_I contains embeddings of negatives, and τ is a temperature hyper-parameter.

In ProtoCCL, we utilize the contrastive loss in a fully supervised way considering the presence of class labels. For a prototype \mathbf{c} (query) with its category label k , the positive samples \mathbf{c}^+ (positive keys) are other prototypes from class k , while the negatives samples \mathbf{c}^- (negative keys) are prototypes from other classes. Formally, ProtoCCL for a prototype from category k can be formulated as follows,

$$\mathcal{L}_k^{\text{ProtoCCL}} = \frac{1}{|\mathcal{C}_k|} \sum_{\mathbf{c}^+ \in \mathcal{C}_k} -\log \frac{\exp(\mathbf{c} \cdot \mathbf{c}^+ / \tau)}{\exp(\mathbf{c} \cdot \mathbf{c}^+ / \tau) + \sum_{\mathbf{c}^- \in \mathcal{N}_k} \exp(\mathbf{c} \cdot \mathbf{c}^- / \tau)} \quad (14)$$

where \mathcal{C}_k and \mathcal{N}_k denote prototype feature collections of the positive and negative samples, for prototype \mathbf{c} .

The proposed ProtoCCL shares the principle of contrastive loss [18] to output small value when prototype \mathbf{c} is similar to its positive key and dissimilar to all other negative keys. It is noteworthy that the constructed memory bank only stores the prototypes instead of all samples, thus significantly reducing the computational costs.

3.4. Training Losses

The overall training process can be divided into two stages. In the first stage, we obtain the pseudo domain labels of all source samples by optimizing Eq. (3) and cross-entropy loss \mathcal{L}_{cls} . Formally, \mathcal{L}_{cls} is defined as follows,

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^K \mathbb{1}[y_i = j] \log(F_c \circ G(x_i)) \quad (15)$$

where G denotes the feature extractor and F_c represents the class label predictor. In the second stage, by using the domain labels predicted in the first stage, we have the following loss for our COMEN model,

$$\mathcal{L}_{\text{COMEN}} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{ProtoPR}} + \gamma \mathcal{L}_{\text{ProtoCCL}} \quad (16)$$

where λ and γ is the trade-off parameters that balance different training losses.

4. Experiments

4.1. Datasets and Evaluation Protocol

We evaluate out-of-domain accuracy of COMEN on four DG benchmarks: **PACS** [29], **Digits-DG** [72], **VLCS** [11], and **Office-Home** [57]. (1) PACS, which includes 9,991 images of seven categories, is the most commonly used DG benchmark due to its large distributional shift across four domains (*Photo*, *Art Painting*, *Cartoon*, and *Sketch*). (2) Digits-DG, which is a handwritten digit recognition benchmark, is composed of four domains (*MNIST* [28], *MNIST-M* [14], *SVHN* [44], and *SYN* [14]) where the domain shifts stem from the variations of background, style, and color. (3) VLCS collects images from *PASCAL VOC 2007* [10], *LabelMe* [47], *Caltech* [12], and *Sun* [63] and contains five categories in all. (4) Office-Home, which is an object recognition dataset in office and home environments, consists of 15,500 images of 65 categories from four domains (*Artistic*, *Clipart*, *Product*, and *Real World*).

To facilitate a fair comparison, we follow the leave-one-domain-out evaluation protocol as in [2, 29, 33], *i.e.*, one domain is chosen as the held out domain and the remaining domains are seen as source domains. For PACS, Digits-DG, and Office-Home, we follow the train and val splits established in [29, 72, 73]. For VLCS, we follow the previous works [2, 40] to split 30% of the source samples as validation datasets. In the testing phase, we use all target samples to compute the classification accuracy of the model that exhibits the best performance in the validation dataset.

4.2. Implementation Details

For PACS and Office-Home, we use ResNet-18 [20] pre-trained on the ImageNet as the backbone architecture. For Digits-DG, we follow [72] to construct the backbone with

Table 1. Domain Generalization results on PACS benchmark.

Method	Art	Cartoon	Photo	Sketch	Avg
<i>w/ Domain Supervision</i>					
CCSA [41]	80.5	76.9	93.6	66.8	79.4
MMD-AAE [34]	75.2	72.7	96.0	64.2	77.0
CrossGrad [50]	79.8	76.8	96.0	70.2	80.7
Metareg [1]	<u>83.7</u>	77.2	95.5	70.3	81.7
MASF [9]	80.3	77.2	95.0	71.7	81.1
Epi-FCR [31]	82.1	77.0	93.9	73.0	81.5
L2A-OT [72]	83.3	78.2	96.2	73.6	82.8
SagNet [43]	83.6	77.7	95.5	76.3	83.3
SelfReg [26]	82.3	78.4	96.2	<u>77.5</u>	83.6
MixStyle [73]	84.1	<u>78.8</u>	<u>96.1</u>	75.9	<u>83.7</u>
<i>w/o Domain Supervision</i>					
DeepAll [31]	77.6	73.9	94.4	70.3	79.1
JiGen [2]	79.4	75.3	96.0	71.6	80.5
MMLD [40]	81.3	77.2	<u>96.1</u>	72.3	81.7
RSC [23]	79.9	76.9	94.6	77.1	81.1
EISNet [60]	81.9	76.4	96.0	74.3	82.2
SFA-A [33]	81.2	77.8	93.9	73.7	81.7
SagNet [43]	83.6	77.7	95.5	76.3	83.3
COMEN (Ours)	82.6	81.0	94.6	84.5	85.7

four 64-kernel 3×3 conv layers and a softmax layer. For VLCS, we use AlexNet [27] pre-trained on the ImageNet as the backbone architecture. We train the networks using SGD with momentum of 0.9 and weight decay of $5e-4$ for 50 epochs. For PACS, VLCS, and Office-Home, the initial learning rate is set to 0.001, which is decayed by 0.1 at 20 epochs. For Digits-DG, we train the network from scratch and the initial learning rate is set to 0.05, which is decayed by 0.1 every 20 epochs. The batch size is set to 128 in all experiments. For Eq. (14), we set $\tau = 0.5$. For Eq. (16), we set $\lambda = 0.1$ and $\gamma = 0.1$. We evaluate the performance in all classes and report the average accuracy over 3 runs with different random seeds. All experiments are implemented on the PyTorch framework using a 1080Ti GPU.

4.3. Comparisons with State-of-the-Arts

Comparison Methods. We extensively compare the proposed COMEN against state-of-the-art DG methods, which can be classified into two groups: **w/ Domain Supervision** and **w/o Domain Supervision**. *w/ Domain Supervision* indicates that these methods relies on the domain information to implement their algorithms. Note that *w/o Domain Supervision* does not imply that the included methods also explicitly discover and model the latent domains. By contrast, they usually treat the compound source domains as a whole. For all the compared approaches, we summarize the domain generalization results reported in their original papers. **DeepAll** represents the standard supervised learning by directly combining all source domains.

Table 2. Domain Generalization results on Digits-DG benchmark.

Method	MNIST	MNIST-M	SVHN	SYN	Avg
<i>w/ Domain Supervision</i>					
CCSA [41]	95.2	58.2	65.5	79.1	74.5
MMD-AAE [34]	96.5	58.4	65.0	78.4	74.6
CrossGrad [50]	96.7	61.1	65.3	80.2	75.8
DDAIG [71]	96.6	64.1	68.6	81.0	77.6
L2A-OT [72]	96.7	63.9	68.6	83.2	78.1
MixStyle [73]	96.5	63.5	64.7	81.2	76.5
<i>w/o Domain Supervision</i>					
DeepAll [72]	95.8	58.8	61.7	78.6	73.7
JiGen [2]	96.5	61.4	63.7	74.0	73.9
SFA-A [33]	96.5	66.5	70.3	85.0	79.6
COMEN (Ours)	97.1	67.6	75.1	91.3	82.3

Table 3. Domain Generalization results on VLCS benchmark.

Method	V	L	C	S	Avg
<i>w/ Domain Supervision</i>					
D-MTAE [16]	63.9	60.1	89.1	61.3	68.6
CCSA [41]	67.1	62.1	92.3	59.1	70.2
DBA-DG [29]	70.0	63.5	93.6	61.3	72.1
CIDDG [35]	64.4	63.1	88.8	62.1	69.6
MMD-AAE [34]	67.7	62.6	94.4	64.4	72.3
MLDG [30]	67.7	61.3	94.4	65.9	72.3
Epi-FCR [31]	67.1	64.3	94.1	65.9	72.9
MASF [9]	69.1	64.9	94.8	67.6	74.1
<i>w/o Domain Supervision</i>					
DeepAll [33]	71.9	59.2	96.9	62.6	72.7
JiGen [2]	70.6	60.9	96.9	64.3	73.2
MMLD [40]	72.0	58.8	96.7	68.1	73.9
SFA-A [33]	70.4	62.0	97.2	66.2	74.0
COMEN (Ours)	72.8	62.6	97.0	67.6	75.0

Table 4. Domain Generalization results on Office-Home.

Method	Art	Clipart	Product	Real	Avg
<i>w/ Domain Supervision</i>					
CCSA [41]	59.9	49.9	74.1	75.7	64.9
MMD-AAE [34]	56.5	47.3	72.1	74.8	62.7
CrossGrad [50]	58.4	49.4	73.9	75.8	64.4
DDAIG [71]	59.2	52.3	74.6	76.0	65.5
DOSN [49]	59.4	45.7	71.8	74.7	62.9
L2A-OT [72]	60.6	50.1	74.8	77.0	65.6
MixStyle [73]	58.7	53.4	74.2	75.9	65.5
<i>w/o Domain Supervision</i>					
DeepAll [72]	58.9	49.4	74.3	76.2	64.7
JiGen [2]	53.0	47.5	71.5	72.8	61.2
RSC [23]	58.4	47.9	71.6	74.5	63.1
SagNet [43]	60.2	45.4	70.4	73.4	62.4
COMEN (Ours)	57.6	55.8	75.5	76.9	66.5

PACS and Digits-DG. The domain generalization results on PACS and Digits-DG benchmarks are reported in Table 1 and Table 2, we can observe that the proposed COMEN outperforms all the baseline methods by a

large margin and improves over state-of-the-art results by +2.0% (PACS) and 2.7% (Digits-DG) on average, clearly demonstrating that COMEN is capable of extracting generalizable and transferable representations for effective out-of-distribution generalization. In particular, COMEN largely improves the classification accuracy on the hard generalization tasks, such as *Sketch* (70.3% \rightarrow 84.5%), *MNIST-M* (58.8% \rightarrow 67.6%), and *SVHN* (61.7% \rightarrow 75.1%), where the source domains are significantly different from the unseen target domain. This encouraging results further reveal the significance of discovering and modeling latent domains.

VLCS and Office-Home. Table 3 and Table 4 display the results on VLCS and Office-Home. Our method consistently and substantially outperforms all the compared methods on most tasks, which indicates that COMEN is versatile and scalable across different domain generalization scenarios. However, we notice that the improvements on VLCS are smaller than the results on the other datasets. The rationale is that the domain shifts of the VLCS dataset stems from the type of camera, and thus the visual appearances are more analogous to each other. In addition, all methods including ours achieve less improvements on Office-Home dataset compared to the vanilla DeepAll baseline. The DeepAll model even outperforms the state-of-the-art *w/o Domain Supervision* methods, such as RSC and SagNet. By comparison, COMEN significantly exceeds the performance of these methods on this scenario.

These experimental results also reveal two important observations. (1) Mixstyle [73] and SFA-A [33], which introduce explicit feature augmentation strategies, is clearly better than the previous works that focus on feature alignment, such as MASF [9] and MMLD [40]. (2) COMEN substantially outperforms the state-of-the-art methods (Mixstyle and SFA-A), which demonstrates the superiority of COMEN by estimating the domain assignment of each source sample and encoding semantic structure into latent space in virtue of prototypical representations.

4.4. Further Empirical Analysis

Ablation Study. To study the relative contribution of each component in the proposed COMEN, we perform an extensive ablation analysis on four DG benchmarks. The results are reported in Table 5, where the first row corresponds to the DeepAll baseline. Specifically, we evaluate all possible combinations of the three proposed components: DSNorm, ProtoGR, and ProtoCCL. We can observe that all the components are designed reasonably and when any one of these components is removed, the performance drops accordingly. Noting that removing the DSNorm module leads to significant performance degeneration, which verifies the importance of discovering the latent domains in compound DG. Combining both ProtoGR and ProtoCCL, which model the prototypical relations in the feature space, shows clear

Table 5. Ablation of COMEN on four DG benchmarks (%).

DSNorm	ProtoGR	ProtoCCL	PACS	Digits-DG	VLCS	Office-Home	Average
-	-	-	79.1	73.7	72.7	64.7	72.6
✓	-	-	81.3	75.4	73.4	65.0	73.8
-	✓	-	82.0	75.6	74.1	65.4	74.3
-	-	✓	81.3	76.1	73.5	65.1	74.0
-	✓	✓	82.6	77.8	74.4	65.6	75.1
✓	✓	-	83.5	79.0	74.3	65.8	75.7
✓	-	✓	84.0	79.3	74.2	66.3	76.0
✓	✓	✓	85.7	82.3	75.0	66.5	77.4

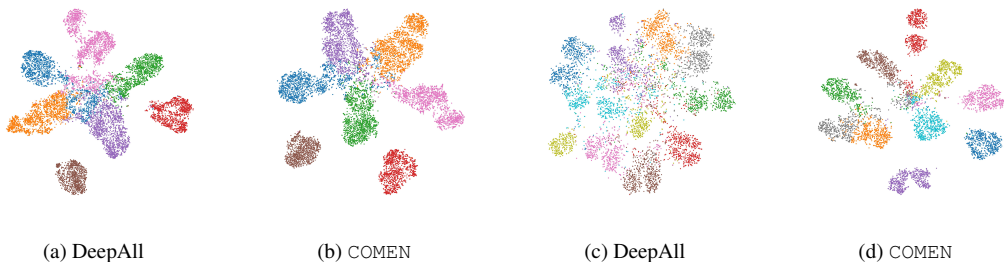


Figure 3. The t-SNE visualization of deep network activations extracted by DeepAll and COMEN on the PACS and Digits-DG datasets.

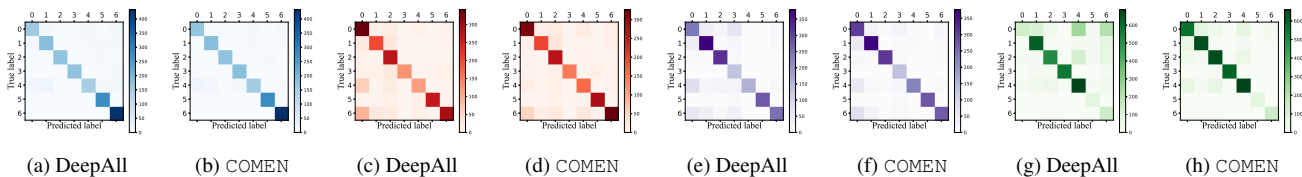


Figure 4. The confusion matrices of DeepAll and COMEN on PACS dataset. From left to right: Photo, Art, Cartoon, and Sketch.

improvement over DeepAll and DSNorm only. For methods without DSNorm, we randomly and uniformly divide the compound source domains into multiple discrete domains so as to conduct the ProtoGR or/and ProtoCCL modules. In this case, we can observe that these two modules are robust to the noisy domain assignment results.

Feature Visualization. We utilize t-SNE [55] for the visualizations of the deep network activations learned by DeepAll and COMEN on the PACS and Digits-DG datasets. The results are shown in Figure 3, where different colors stand for different classes. The features learned by DeepAll on the mixture of source domains cannot be reasonably separated, and the decision boundaries of some classes are ambiguous in the feature space. By contrast, COMEN learns discriminative representations for both source and unseen target domains by simultaneously enlarging inter-class distance and suppressing the intra-class dispersion.

Confusion matrices. Figure 4 respectively plots confusion matrices of DeepAll and COMEN regarding the PACS dataset. We can see that the false prediction results are largely reduced by COMEN especially on the hard generalization task (Cartoon and Sketch), revealing the key impor-

tance of discovering and modeling latent domains.

5. Conclusion

We study a challenging yet practical domain generalization problem, namely compound domain generalization. To solve this problem, we propose a novel COMEN framework to automatically discover and model the distinct underlying domains with three newly proposed modules. In particular, DSNorm re-normalizes the multi-modal distributions to obtain domain information. ProtoGR and ProtoCCL model the prototypical relations by simultaneously encoding semantic structure of feature space and learning discriminative domain-invariant representations. Experiments reveal that COMEN significantly outperforms state-of-the-art DG methods on four benchmark datasets.

COMEN is tailored for the classification task, and cannot be simply extended to other vision tasks, such as object detection and semantic segmentation.

Acknowledgement. This work was partially supported by National Key Research and Development Program of China (No.2020YFC2003902) and Hong Kong Research Grants Council through Research Impact Fund (Grant R-5001-18).

References

- [1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, pages 998–1008, 2018.
- [2] Fabio M Carlucci, Antonio D’Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, pages 2229–2238, 2019.
- [3] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and S Rota Bulò. Autodial: Automatic domain alignment layers. In *ICCV*, 2017.
- [4] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 2019.
- [5] Chaoqi Chen, Jiongcheng Li, Zebiao Zheng, Yue Huang, Xinghao Ding, and Yizhou Yu. Dual bipartite graph learning: A general approach for domain adaptive object detection. In *ICCV*, pages 2703–2712, 2021.
- [6] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *CVPR*, pages 627–636, 2019.
- [7] Chaoqi Chen, Zebiao Zheng, Yue Huang, Xinghao Ding, and Yizhou Yu. I3net: Implicit instance-invariant network for adapting one-stage object detectors. In *CVPR*, pages 12576–12585, 2021.
- [8] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. Self-supervised learning of adversarial examples: Towards good generalizations for deepfake detections. In *CVPR*, 2022.
- [9] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, pages 6447–6458, 2019.
- [10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.
- [11] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *ICCV*, pages 1657–1664, 2013.
- [12] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR-W*, pages 178–178, 2004.
- [13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017.
- [14] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.
- [15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016.
- [16] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *ICCV*, pages 2551–2559, 2015.
- [17] Boqing Gong, Kristen Grauman, and Fei Sha. Reshaping visual datasets for domain adaptation. In *NeurIPS*, pages 1286–1294, 2013.
- [18] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pages 1735–1742, 2006.
- [19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [21] Judy Hoffman, Brian Kulis, Trevor Darrell, and Kate Saenko. Discovering latent domains for multisource domain adaptation. In *ECCV*, pages 702–715, 2012.
- [22] Hanzhe Hu, Jinshi Cui, and Liwei Wang. Region-aware contrastive learning for semantic segmentation. In *ICCV*, pages 16291–16301, 2021.
- [23] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, pages 124–140, 2020.
- [24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.
- [25] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, pages 158–171, 2012.
- [26] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *ICCV*, pages 9619–9628, 2021.
- [27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012.
- [28] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [29] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, pages 5542–5550, 2017.
- [30] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018.
- [31] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *ICCV*, pages 1446–1455, 2019.
- [32] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021.
- [33] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *ICCV*, pages 8886–8895, 2021.

- [34] Shuang Li, Shiji Song, Gao Huang, Zhengming Ding, and Cheng Wu. Domain invariant and class discriminative feature learning for visual domain adaptation. *IEEE Transactions on Image Processing*, 27(9):4260–4273, 2018.
- [35] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, pages 624–639, 2018.
- [36] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. In *ICLR*, 2017.
- [37] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.
- [38] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *ECCV*, pages 466–483, 2020.
- [39] Massimiliano Mancini, Lorenzo Porzi, Samuel Rota Buló, Barbara Caputo, and Elisa Ricci. Boosting domain adaptation by discovering latent domains. In *CVPR*, pages 3771–3780, 2018.
- [40] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *AAAI*, 2020.
- [41] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, pages 5715–5725, 2017.
- [42] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, pages 10–18, 2013.
- [43] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *CVPR*, pages 8690–8699, 2021.
- [44] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bisacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
- [45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [46] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *ICML*, pages 7728–7738, 2020.
- [47] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.
- [48] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.
- [49] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In *ECCV*, pages 68–83, 2020.
- [50] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *ICLR*, 2018.
- [51] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.
- [52] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, pages 443–450, 2016.
- [53] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528, 2011.
- [54] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.
- [55] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008.
- [56] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.
- [57] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017.
- [58] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*, pages 5334–5344, 2018.
- [59] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. *arXiv preprint arXiv:2103.03097*, 2021.
- [60] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. In *ECCV*, pages 159–176, 2020.
- [61] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *ICCV*, pages 7303–7313, 2021.
- [62] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Transferable normalization: Towards improving transferability of deep neural networks. In *NeurIPS*, pages 1951–1961, 2019.
- [63] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492, 2010.
- [64] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *ICML*, pages 5419–5428, 2018.
- [65] Caiming Xiong, Scott McCloskey, Shao-Hang Hsieh, and Jason Corso. Latent domains modeling for visual domain adaptation. In *AAAI*, 2014.
- [66] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *ICLR*, 2021.
- [67] Yabin Zhang, Bin Deng, Kui Jia, and Lei Zhang. Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation. In *ECCV*, pages 781–797. Springer, 2020.

- [68] Yabin Zhang, Bin Deng, Hui Tang, Lei Zhang, and Kui Jia. Unsupervised multi-class domain adaptation: Theory, algorithms, and practice. *IEEE TPAMI*, 2020.
- [69] Yabin Zhang, Minghan Li, Ruihuang Li, Kui Jia, and Lei Zhang. Exact feature distribution matching for arbitrary style transfer and domain generalization. In *CVPR*, 2022.
- [70] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, pages 5031–5040, 2019.
- [71] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, pages 13025–13032, 2020.
- [72] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, pages 561–578, 2020.
- [73] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021.