

SEIC: Semantic Embedding with Intermediate Classes for Zero-Shot Domain Generalization

Biswajit Mondal^{[0000–0002–3444–6321]*} and Soma Biswas^[0000–0002–9068–7023]

Indian Institute of Science, Bangalore, India
mondalb682@gmail.com, somabiswas@iisc.ac.in

Abstract. In this work, we address the Zero-Shot Domain Generalization (ZSDG) task, where the goal is to learn a model from multiple source domains, such that it can generalize well to both unseen classes and unseen domains during testing. Since it combines the tasks of Domain Generalization (DG) and Zero-Shot Learning (ZSL), here we explore whether advances in these fields also translate to improved performance for the ZSDG task. Specifically, we build upon a state-of-the-art approach for domain generalization and appropriately modify it such that it can generalize to unseen classes during the testing stage. Towards this goal, we propose to make the feature embedding space semantically meaningful, by not only making an image feature close to its semantic attributes, but also taking into account its similarity with the other neighbouring classes. In addition, in order to reserve space for the unseen classes in the embedding space, we propose to introduce pseudo intermediate classes in between the semantically similar classes during training. This reduces confusion of the similar classes and thus increases the discriminability of the embedding space. Extensive experiments on two large-scale benchmark datasets, namely DomainNet and DomainNet-LS and comparisons with the state-of-the-art approaches show that the proposed framework outperforms all the other techniques on both the datasets.

1 Introduction

The recent advancement in deep neural networks has achieved enormous success in numerous areas of computer vision, such as classification [49, 13], segmentation [40], retrieval [50, 38, 34], playing Atari games with reinforcement learning [32], etc. In standard supervised training, we assume that the training and the test data belong to the same distribution, and the test data contains only the classes that were seen during training. Such models can fail when they encounter images from classes and domains unseen during the training process, as often encountered in real scenarios. Since it is impractical to collect examples from all possible classes and domains during training, it is important that the learnt models generalize well to these challenging scenarios. This has led to a significant amount of research focused in areas like domain generalization (DG) and zero-shot learning (ZSL). The DG task [25, 27, 48, 39] aims to classify samples from

* Corresponding author.

unseen target domains after learning from multiple source domains which have the same classes as the target data. On the other hand, the ZSL task [33, 23] aims to classify samples belonging to classes unseen during training, but from the same domain as the training data. It is only recently, that researchers have started addressing the more realistic and challenging zero-shot domain generalization (ZSDG) task [28, 8, 29], where during testing, the samples can not only belong to unseen classes, but also unseen domains.

ZSDG being a combination of DG and ZSL tasks, an advancement in any of these fields should translate to an advancement in the ZSDG problem. But recent research [8] indicates that naively combining DG and ZSL approaches does not help to improve the performance on the ZSDG task. In this work, we explore whether a state-of-the-art DG approach can be appropriately modified so that it also achieves state-of-the-art performance for the ZSDG task. Specifically, we build upon MixStyle [51], which computes the convex combination of instance-level feature statistics of different samples to generate diverse domains/styles for training, while keeping the semantic information intact for the DG task.

In this work, we propose simple, yet effective modifications which can generalize the MixStyle [51] framework for classifying unseen classes (from unseen domains) during testing. Towards this goal, we want to make the feature embedding space semantically meaningful, so that unseen class images/features can be matched with their semantic attributes, as well as discriminative, so that the classification performance in this space is satisfactory. To account for both these objectives, we propose two modifications to the original DG approach, namely (1) We introduce intermediate (pseudo) classes between semantically similar classes in the embedding space, to reserve space for the unseen classes during testing; (2) Each image feature is encouraged to be not only close to its true attribute vector, but also at semantically meaningful distances from the attributes of its neighbouring classes. The combined framework is termed as **S**emantic **E**mboding with **I**ntermediate **C**lasses (SEIC). To summarize, our contributions are as follows:

1. We propose a simple, yet effective framework termed SEIC, to address the problem of zero-shot domain generalization.
2. We propose to make the feature embedding space semantically meaningful and discriminative by accounting for the neighbouring class information as well as by introducing intermediate pseudo classes.
3. We show that a state-of-the-art DG method can be appropriately modified to get state-of-the-art result for the related ZSDG problem.
4. Extensive experiments and comparisons on the challenging DomainNet and DomainNet-LS datasets [35] justify the effectiveness of the SEIC framework.

2 Related Work

Here, we briefly describe the related literature on domain generalization (DG), zero-shot learning (ZSL) and finally zero-shot domain generalization (ZSDG).

Domain Generalization: First proposed in [5], domain generalization is a

problem gaining rapid attention in the vision community. A broad category of approaches can be summarized by domain-invariant representation learning, i.e., learning representations that eliminate domain-specific variations within the dataset. This approach was first examined in the context of domain adaptation in [4], which was used to construct a domain-adversarial neural network in [11]. Several algorithms have been proposed for domain generalization via adversarial learning [25, 27, 48, 39]. MixStyle [51] is motivated by the observation that visual domain is closely related to image style (or domain). [20] augments the feature-space by identifying the dominant modes of change in the source domain. [14, 47] transform images into frequency space to perform domain generalization. Single-source DG methods, tackle a more challenging scenario, where only a single source domain is available during training [14, 37, 39, 42]. Some works also address the DG problem during the testing phase [16].

Zero Shot Learning: ZSL [33, 23] aims to transfer the model trained on the seen classes to the unseen ones, usually using a semantic space between seen classes and unseen classes. Early works in ZSL focused on the conventional ZSL [1, 2, 6, 22, 7], where the test data only belongs to the unseen classes, and the predicted class is calculated based on the feature similarity with the attributes of the unseen test classes in the embedding space. In generalized ZSL (GZSL), both seen and unseen classes can be present during testing, making it a more challenging task. The works in [9, 3, 46, 15] addresses the overfitting problem that arises due to training on only the seen classes [44]. Many works [19, 31, 45] employ generative methods for converting the problem into a supervised learning problem using Generative Adversarial Networks (GANs) [12] and Variational Autoencoder (VAEs) [21] to synthesize images of the unseen classes. [18] uses adaptive metric learning to check the compatibility of a sample with the class semantics.

Zero Shot Domain Generalization: In general, the ZSL and DG tasks have been considered separately. But recently, ZSDG is being researched actively because of its more realistic and practical applications. Cumix [28] aims to simulate the test-time domain and semantic shift using images from unseen domains and categories by mixing up the images available in source domains and categories during the training phase. It also uses a curriculum-based mixing policy to generate increasingly complex training samples. SLE-Net [8] uses visual and semantic encoders to learn domain-agnostic structured latent embeddings by projecting images from different domains and their class-specific semantic representations to a common latent space. SnMpNet [34] addresses the problem of image retrieval, where the test data can belong to classes or domains which are unseen during training. Our work is similar in spirit to [29], which effectively exploits semantic information of the classes to adapt the existing DG methods to tackle the ZSDG task. Zero-shot domain adaptation is another research area similar to ZSDG, which aims to transfer the knowledge from a single source domain to a target domain. [26] projects the samples of source and target domains to a common space and then learns unseen class prototypes of the target domain. [17] learns class-agnostic domain feature representations and prevents negative

transfer effects using adversarial learning. [41] introduces a new scenario where labelled samples are available for a subset of target domain classes and proposes a method to transform samples from source domain to target domain without loss of class information.

3 Problem Definition

Zero-shot domain generalization (ZSDG) aims to classify unseen classes in unseen domains. Let \mathcal{X} denote the image space, \mathcal{Y} the set of possible classes and \mathcal{D} the set of possible domains. The classes are divided into two sets, one is used for training or the *seen classes* ($\mathcal{Y}^s \in \mathcal{Y}$) and the other for testing or the *unseen classes* ($\mathcal{Y}^u \in \mathcal{Y}$). Similarly, we have *seen domains* ($\mathcal{D}^s \in \mathcal{D}$) and *unseen domains* ($\mathcal{D}^u \in \mathcal{D}$). For training, we are given the set, $\mathcal{M} = \{(\mathbf{x}, y, \mathbf{a}_y, d) | \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}^s, \mathbf{a}_y \in \mathcal{E}, d \in \mathcal{D}^s\}$, where \mathbf{x} is an image belonging to a seen class and a seen domain, and has a class label y belonging to the seen class set \mathcal{Y}^s . \mathbf{a}_y is the semantic embedding in \mathcal{E} for class y in \mathcal{Y}^s , where \mathcal{E} is the embedding space. d is \mathbf{x} 's domain label from the seen domain set.

During testing, the goal is to classify the test data $\mathcal{N} = \{\mathbf{x}\}$, which belong to an unseen class, i.e. $y \in \mathcal{Y}^u$ and also an unseen domain, i.e. $d \in \mathcal{D}^u$. In standard ZSL, training is done on the set of seen classes and testing on the set of unseen classes which are mutually disjoint, but the domains remain the same, i.e., $\mathcal{Y}^s \cap \mathcal{Y}^u = \phi$ and $\mathcal{D}^s \equiv \mathcal{D}^u$. In DG, training is done on images belonging to a set of domains that is disjoint to the set of domains used for testing, but the set of classes is shared, i.e., $\mathcal{D}^s \cap \mathcal{D}^u = \phi$ and $\mathcal{Y}^s \equiv \mathcal{Y}^u$. Each domain can have different distributions, i.e., $p_{\mathcal{X}}(\mathbf{x}|d_i) \neq p_{\mathcal{X}}(\mathbf{x}|d_j), \forall i \neq j$. Here, we address the more challenging ZSDG problem where testing is done on domains and classes unseen during training, i.e., $\mathcal{D}^s \cap \mathcal{D}^u = \phi$ and $\mathcal{Y}^s \cap \mathcal{Y}^u = \phi$.

4 Proposed Method

Now, we describe the proposed framework, termed SEIC for the ZSDG task. First, we describe the recent state-of-the-art DG technique MixStyle [51] that we use as the backbone for SEIC framework, followed by the proposed modifications.

4.1 Handling Unseen Domains Using Domain Generalization

Here, we briefly describe the MixStyle [51] approach, where given training data from multiple source domains, the goal is to learn a model which can generalize well to unseen target domains. MixStyle regularizes CNN training by perturbing the style information of the samples from the source domains. It mixes the feature statistics of two instances with a random convex weight to simulate new styles. The framework broadly consists of a feature extractor \mathcal{F}^{DG} and a classifier g^{DG} to get the output, $y^{DG} = g^{DG} \circ \mathcal{F}^{DG}$. The mixing is done using the statistics of features from the output of different CNN layers in the feature extractor.

Let \mathbf{f}_i and \mathbf{f}_j be the feature maps corresponding to samples \mathbf{x}_i and \mathbf{x}_j after a particular CNN layer. It computes the mixed style feature statistics for \mathbf{f}_i using \mathbf{f}_j as follows:

$$\mu_{ms}(\mathbf{f}_i; \mathbf{f}_j) = \lambda\mu(\mathbf{f}_i) + (1 - \lambda)\mu(\mathbf{f}_j) \quad (1)$$

$$\sigma_{ms}(\mathbf{f}_i; \mathbf{f}_j) = \lambda\sigma(\mathbf{f}_i) + (1 - \lambda)\sigma(\mathbf{f}_j) \quad (2)$$

where $\lambda \sim \beta(\alpha, \alpha)$ and $\alpha \in (0, \infty)$ is a hyper parameter. $\sigma(\cdot)$ and $\mu(\cdot)$ are standard deviation and mean, respectively, computed along the height and width of each channel. Finally, the style-normalized features $f_{ms}(\mathbf{f}_i; \mathbf{f}_j)$ are computed by using the mixed feature statistics as follows:

$$f_{ms}(\mathbf{f}_i; \mathbf{f}_j) = \sigma_{ms}(\mathbf{f}_i; \mathbf{f}_j) * \mathbf{f}'_i + \mu_{ms}(\mathbf{f}_i; \mathbf{f}_j) \quad (3)$$

$$\text{where, } \mathbf{f}'_i = \frac{\mathbf{f}_i - \mu(\mathbf{f}_i)}{\sigma(\mathbf{f}_i)} \quad (4)$$

The mixing of the statistics does not alter the class information (i.e. class is same as that of \mathbf{x}_i) even if the two features being mixed belong to different classes. This module can be easily plugged in after different layers of the CNN to get more diversity in the source domains and achieve better generalizability for unseen domains.

4.2 Handling Unseen Classes Using the Proposed SEIC Framework

We will now describe the proposed modifications, such that the above model also performs well for unseen classes during testing. Specifically, we make the following three modifications: (i) First, to establish the connection between the seen and unseen classes, we replace the classifier weights using the semantic vectors, which are automatically obtained using the class names. (ii) To reserve space for the unseen classes which will be encountered during testing, we introduce intermediate pseudo-classes in the training process; (iii) We utilize the neighbourhood class information to make the feature embedding space semantically meaningful. These modifications (details below) enable the proposed framework (SEIC) to handle unseen classes as well during the testing stage.

(i) *Utilizing class attributes to link the seen and unseen classes:*

In ZSDG task, since unseen classes can be encountered during testing, it is important to link the seen and unseen classes. Specifically, the goal is to learn the relation between the feature embeddings and the class semantics, such that the class label of the test data can be predicted by comparing it with the semantic embeddings of all the classes. The semantic embeddings can be obtained using unsupervised Natural Language Processing algorithms like Word2Vec [30], GloVe vectors [36], etc. As discussed earlier, in MixStyle, the model architecture has a feature extractor \mathcal{F}^{DG} followed by a classifier g^{DG} . For handling unseen classes, we replace the weights in the classification layer by the semantic vectors of each class, i.e., we want the predicted semantic embeddings extracted from the model

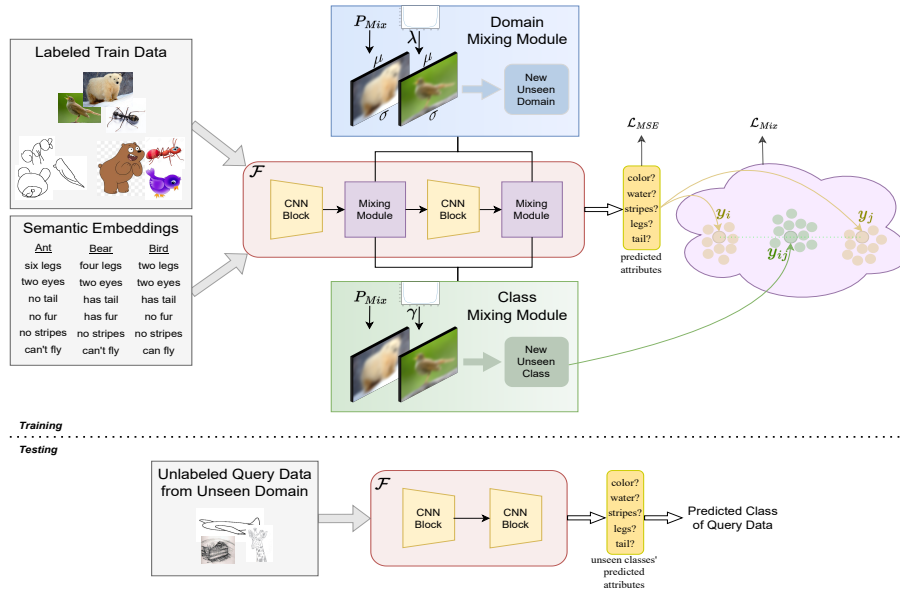


Fig. 1: Depiction of the proposed method. For training, samples from seen domains and seen classes are fed to the feature extractor, which consists of CNN blocks, each followed by a mixing module. The mixing module consists of the domain and class mixing modules to simulate unseen domains and classes. To learn distinctive features, especially between similar classes, the mixing module is given the information of pair-wise mixing probabilities of the classes, P_{Mix} and intermediate pseudo classes are inserted at the output layer. During testing, the model has no mixing module and pseudo-class nodes, and predicts the samples from unseen classes and unseen domains.

to be similar to the semantic embeddings, $g : \mathcal{Y} \rightarrow \mathcal{E}$. The modified feature extractor and embedding function are denoted as $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{E}$ and $g : \mathcal{Y} \rightarrow \mathcal{E}$ respectively. In this work, g is the Word2Vec embedding of the class name. For an image \mathbf{x} , the model predicts the class as follows:

$$y^* = \underset{y}{\operatorname{argmax}} g(y)^T \mathcal{F}(\mathbf{x}). \quad (5)$$

With this modification, the model is capable of recognizing unseen classes. But currently, since the model is trained using only the seen classes, the feature embedding space may not be discriminative enough to distinguish between the seen and unseen classes, which we address using the intermediate pseudo-classes.

(ii) Introducing Intermediate Pseudo Classes: Now, we discuss how we address the challenge of correctly classifying unseen classes during testing. The current model will try to embed the features of the unseen classes to its class attributes, but since these classes were not used for model training, their em-

beddings are usually confused, specifically with those of the semantically similar seen classes. For example, a new *bee* class feature may be easily confused with features of other insects, but will probably not be confused with features from animals like cat or vehicles like buses, etc. To improve class discrimination and thus reduce this confusion, we propose to introduce additional pseudo classes in between the existing training classes, with more emphasis on semantically similar classes. These intermediate classes act as proxies for the unseen classes, that might be encountered during testing. We propose to generate data for the intermediate classes using the available training data as explained below.

Given two classes $y_i, y_j \in \mathcal{Y}^s$ in the seen class set with their respective semantic embeddings as $\mathbf{a}_{y_i}, \mathbf{a}_{y_j} \in \mathcal{E}$, we form an intermediate pseudo class y_{ij} (or equivalently y_{ji}) and assign it a semantic embedding equal to the average of the semantic embeddings of y_i and y_j , i.e.,

$$\mathbf{a}_{y_{ij}} = \frac{\mathbf{a}_{y_i} + \mathbf{a}_{y_j}}{2} \quad (6)$$

To generate the training data for these intermediate pseudo-classes, we propose to mix pairs of samples belonging to classes y_i and y_j in the feature space after different CNN layers. Given the feature maps \mathbf{f}_i and \mathbf{f}_j of two training instances after some CNN layer, we obtain the intermediate class feature f_{ic} as:

$$f_{ic}(\mathbf{f}_i; \mathbf{f}_j) = \gamma \mathbf{f}_i + (1 - \gamma) \mathbf{f}_j \quad (7)$$

where $\gamma \in \mathbb{R}$ is sampled from a uniform distribution, i.e., $\gamma \sim \mathbf{U}(t_1, t_2)$. Since the pseudo classes are generated using the average of two semantic embeddings, as shown in eq. (6), we choose t_1 and t_2 such that the feature of one class does not overshadow the feature of the other class.

We have discussed handling of unseen domains by mixing the statistics and handling the unseen classes by mixing the features. Now, we combine them to get our final mixing module:

$$f_{Mix}(\mathbf{f}_i; \mathbf{f}_j) = f_{ic}(\mathbf{f}'_i; \mathbf{f}'_j) * \sigma_{ms}(\mathbf{f}_i; \mathbf{f}_j) + \mu_{ms}(\mathbf{f}_i; \mathbf{f}_j) \quad (8)$$

where \mathbf{f}'_i and \mathbf{f}'_j are defined as per eq. (4). Since we also want to retain the original samples with their class and domain information, the mixing is done only if a generated random number (r) is less than a pre-set threshold (τ), otherwise the original features with their class information are used for training. This random number is generated independently for each mixing module and for every batch in every epoch. Depending on whether mixing has happened or not, we have two separate losses. The final loss for the image \mathbf{x}_i belonging to class y_i is given as:

$$\mathcal{L}_{Mix}(\mathbf{x}_i; \mathbf{x}_j) = \begin{cases} \mathcal{L}_{CE}(g(y)^T \mathcal{F}(\mathbf{x}_i; \mathbf{x}_j), y_{ij}), & r < \tau \\ \mathcal{L}_{CE}(g(y)^T \mathcal{F}(\mathbf{x}_i), y_i), & \text{otherwise} \end{cases} \quad (9)$$

where \mathcal{L}_{CE} is the cross-entropy loss, $g(y)$ is the set of semantic embeddings corresponding to each class node, \mathcal{F} is the feature extractor consisting of the CNN

layers and the mixing modules inserted between the layers, y_{ij} is the pseudo class in-between the classes corresponding to \mathbf{x}_i and \mathbf{x}_j . When the random number r (generated uniformly in the range 0 to 1) is greater than or equal to the threshold τ , then the mixing module acts as an identity function.

As discussed earlier, to learn distinctive features for closely related classes, we focus more on learning the pseudo classes that are between two semantically close classes. First, we calculate the Euclidean distance between the semantic embeddings of every pair of classes to find the similarity among them. A class is mixed with another class following a probability distribution based on the semantic similarity of the classes. For e.g., if we are given only three classes y_i , y_j and y_k , with their pair-wise distances from \mathbf{a}_{y_i} as, $\text{dist}(\mathbf{a}_{y_i}, \mathbf{a}_{y_j}) = d_{ij}$ and $\text{dist}(\mathbf{a}_{y_i}, \mathbf{a}_{y_k}) = d_{ik}$. Then, the probability of mixing a sample of class y_i with a sample of class y_j will be:

$$P_{Mix}(y_i, y_j; y_k) = \frac{\exp(-d_{ij})}{\exp(-d_{ij}) + \exp(-d_{ik})} \quad (10)$$

The probability of mixing a sample of class y_i with a sample of class y_k can be calculated in a similar manner. Another reason why we mix semantically close classes is that it has more potential to generate meaningful novel classes. For example, mixing two types of insects may produce another insect, but mixing an insect with a dog might not produce anything realistic.

(iii) **Incorporating Information from Neighbouring Classes:** With the above modifications, the model is now capable of handling both unseen classes and unseen domains during testing. But, the success of the unseen class predictions depend upon how semantically meaningful the feature embeddings are. Here, we propose to guide the feature embeddings not only using its correct ground truth attribute (using the classification loss), but also using the information of its semantically similar neighbouring classes. For e.g., the feature embedding of an insect class *wasp* can be guided by its class attribute, and also by its relative distances from the other insect classes. This is specially important during the feature computation of the unseen classes, where in absence of its ground truth attributes, the embedding has to be solely guided by the seen class attributes.

The standard classification loss encourages the model to predict a score of 1 for the ground truth class and 0 for all other classes. In contrast, we propose to calculate the loss not just with respect to the ground truth attribute, but also with respect to the other classes, appropriately weighted by their similarity with the ground truth class. Given an image $\mathbf{x}_i \in \mathcal{X}$, belonging to class $y_i \in \mathcal{Y}^s$ with semantic vector $\mathbf{a}_{y_i} \in \mathcal{E}$, we propose to use an additional loss term as follows:

$$\mathcal{L}_{MSE}(\mathbf{x}_i) = \sum_{y \in \mathcal{Y}^s} \exp\left(-\frac{\|\mathbf{a}_{y_i} - \mathbf{a}_y\|^2}{\max_{z \in \mathcal{Y}^s} \|\mathbf{a}_{y_i} - \mathbf{a}_z\|^2}\right) (\|\mathcal{F}(\mathbf{x}_i) - \mathbf{a}_y\|^2 - \|\mathbf{a}_{y_i} - \mathbf{a}_y\|^2)^2 \quad (11)$$

where \mathbf{a}_y is the semantic vector of an arbitrary class y . $\|\mathcal{F}(\mathbf{x}_i) - \mathbf{a}_y\|^2$ is the distance between the predicted embedding of the sample x_i and the ground

truth semantic vector of class y . Similarly, $\|\mathbf{a}_{y_i} - \mathbf{a}_y\|^2$ is the distance between the ground truth semantic vector of x_i and the ground truth semantic vector of class y . The term inside the second parentheses encourages the embedding of the image feature and its ground truth attribute vector with respect to the attributes of the neighbouring classes to be similar. The term inside the first parentheses is an exponential weighting factor so that this strict relative positioning is mainly applied for the semantically similar classes.

Note that when the mixing module is not activated, we directly follow eq. (11). For the case when mixing is done, we calculate the \mathcal{L}_{MSE} loss by replacing a_{y_i} with the average of the attributes of the mixed classes, i.e., $a_{y_{ij}}$ as given in eq. (6). We combine the two losses to get the final loss as:

$$\mathcal{L} = \mathcal{L}_{Mix} + \eta \mathcal{L}_{MSE} \quad (12)$$

where η is a hyper-parameter to balance the relative effects of the two losses. Similar idea has been explored in SnMpNet [34] for the retrieval task.

5 Experimental Evaluation

Here, we describe in detail the datasets used, implementation details, results and further analysis of the proposed approach.

Datasets Used: For evaluation of our method, we use two large-scale benchmark datasets, namely DomainNet and DomainNet-LS, as used in the recent works in ZSDG [28, 8]. **DomainNet** [35] consists of 345 classes and 6 domains, namely *clipart*, *infograph*, *painting*, *quickdraw*, *real* and *sketch*, spread across approximately 0.6 million images. We follow the same experimental protocol defined in the literature. Out of 345 classes, we use 300 for training as the seen classes and the remaining 45 for testing as unseen classes. Out of 6 domains, we use 5 domains at a time for training as seen domains and the remaining domain is used for testing as unseen domain. We hold-out each domain (except *real*) one-by-one and repeat the training process using the 300 classes in the remaining 5 domains. The testing is not done on the *real* domain, since the backbone is pre-trained on the ImageNet dataset, and thus the *real* domain can not be considered as an unseen domain. For the **DomainNet-LS** benchmark, only *real* and *painting* domains are used for training and the rest are used for testing, the splitting of the classes remains same. Clearly, it is a more challenging setting, since the domain invariant features have to be learnt only using two domains.

Implementation details: For fair comparison with the state-of-the-art approaches, we use the ResNet-50 backbone, which has four CNN blocks. We have the mixing modules only after the 1st three blocks, as done in [51]. To learn the features in the semantic space, we use the 300-dimension semantic vectors from the Word2Vec [30] representation. Following [51], we use $\alpha = 0.1$ as the input parameter of the Beta distribution. Inspired from [52], we sample γ from a distribution uniform in $t_1 = 0.4$ to $t_2 = 0.6$, i.e. $\gamma \sim U(0.4, 0.6)$. Based on the analysis shown in Fig. 3(b), we set $\eta = 1$ to give equal weightage to both the

Table 1: Leave-one-domain-out ZSDG results on DomainNet using average per-class accuracy metric.

Method		Target Domain					Average
DG	ZSL	Clipart	Infograph	Painting	Quickdraw	Sketch	
-	DEWISE [10]	20.1	11.7	17.6	6.1	16.7	14.4
	ALE [1]	22.7	12.7	20.2	6.8	18.5	16.2
	SPNet [43]	26.0	16.9	23.8	8.2	21.8	19.4
DANN	DEWISE [10]	20.5	10.4	16.4	7.1	15.1	13.9
	ALE [1]	21.2	12.5	19.7	7.4	17.9	15.7
	SPNet [43]	25.9	15.8	24.1	8.4	21.3	19.1
Epi-FCR	DEWISE [10]	21.6	13.9	19.3	7.3	17.2	15.9
	ALE [1]	23.2	14.1	21.4	7.8	20.9	17.5
	SPNet [43]	26.4	16.7	24.6	9.2	23.2	20.0
CuMix (img only) [28]		25.2	16.3	24.4	8.7	21.7	19.2
CuMix (two-level) [28]		26.6	17.0	25.3	8.8	21.9	19.9
CuMix [28]		27.6 \pm 0.5	17.8 \pm 0.2	25.5 \pm 0.4	9.9 \pm 0.3	22.6 \pm 0.3	20.7 \pm 0.3
SLE-Net [8]		27.8 \pm 0.3	18.4\pm0.4	26.6 \pm 0.3	11.5 \pm 0.2	25.2 \pm 0.3	21.9 \pm 0.3
Proposed SEIC		29.9\pm0.2	17.4 \pm 0.1	26.7\pm0.4	12.0\pm0.4	27.3\pm0.3	22.7\pm0.3

Table 2: Leave-one-domain-out ZSDG results on DomainNet using standard accuracy metric.

Method	Clipart	Infograph	Painting	Quickdraw	Sketch	Average
CuMix [28]	27.8	16.3	27.6	9.7	25.9	21.5
SLE-Net [8]	29.1	17.6	28.8	11.5	26.3	22.7
Proposed SEIC	32.7	18.3	27.5	11.9	30.4	24.2

losses. For training, we use the Adam optimizer with a learning rate of 10^{-5} and a batch size of 80. We find that setting the probability threshold, with which the mixing module is activated, equal to 0.2, i.e., $\tau = 0.2$ gives the best results, as shown in Fig. 3(a).

5.1 Results on DomainNet and DomainNet-LS Datasets

Here, we perform extensive experiments to evaluate the effectiveness of the proposed SEIC framework for the ZSDG task.

Results on DomainNet dataset: For DomainNet dataset, we compare our results using two metrics: average per-class accuracy and standard accuracy. We follow the same experimental protocol as the previous works in the literature, namely Cumix [28] and SLE-Net [8]. First, we report the results on standalone ZSL methods such as DEWISE [10], ALE [1] and SPNet [43] and combination of the ZSL methods with DG methods, like DANN [11] and Epi-FCR [24]. Along with SLE-Net [8], we report the results of CuMix and its variants: CuMix (img

Table 3: Leave-one-domain-out ZSDG results on DomainNet-LS using average per-class accuracy metric.

Method	Clipart	Infograph	Quickdraw	Sketch	Average
SPNet [43]	21.5	14.1	4.8	17.3	14.4
Epi-FCR+SPNet [43]	22.5	14.9	5.6	18.7	15.4
CuMix (img only) [28]	21.2	14.0	4.8	17.3	14.3
CuMix (two-level) [28]	22.7	16.5	4.9	19.1	15.8
CuMix (reverse) [28]	22.9	15.8	4.8	18.2	15.4
CuMix [28]	23.7	17.1	5.5	19.7	16.5
SLE-Net [8]	24.0	16.0	7.2	20.5	16.9
Proposed SEIC	25.9	16.0	8.5	22.9	18.3

only) where MixUp is applied only at the image level and CuMix (two-level) where MixUp is applied at both image and feature level, as given in [28].

In Table 1, we report the average per-class accuracy for the five test domains using various methods. The results of all the previous approaches have been directly taken from [8]. On using only the standalone ZSL methods DEVISE [10], ALE [1] and SPNet [43], we get 14.4%, 16.2% and 19.4% accuracy, respectively. On integrating the DANN [11] framework with the above ZSL methods, there is a drop in accuracy. Instead of DANN [11], if we combine Epi-FCR [24] with the ZSL methods, the average accuracies improve to 15.9%, 17.5% and 20.0%, respectively. The proposed SEIC framework outperforms the state-of-the-art SLE-Net [8] on four out of the five domains with an average accuracy of 22.7%, which is better than [8] by 0.8%. In Table 2, we show the standard accuracies of the proposed approach for the DomainNet dataset and compare it with the previous two ZSDG methods. SLE-Net [8] obtains average accuracy of 22.7%. Here also, our method outperforms the other methods with an average accuracy of 24.2%, which is an increase of 1.5% over SLE-Net [8].

Results on DomainNet-LS dataset: In Table 3, we show the results on the DomainNet-LS dataset, where we train the model only on 2 domains: *real* and *painting*. An average accuracy of 14.4% is attained by SPNet [8]. On combining it with Epi-FCR [24], the accuracy improves by 1.0%. CuMix [28] achieves an average accuracy of 16.5% beating its other variants like CuMix (reverse) [28]. Our method achieves an average accuracy of 18.3% which is better than SLE-Net [8] by 1.4%.

5.2 Additional Analysis

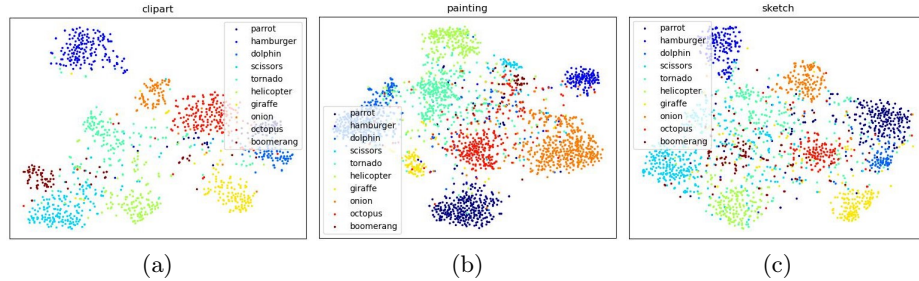
Ablation Study: In Table 4, each of the components is deactivated one-by-one while keeping the others activated. The baseline here is the original backbone with the fixed semantic embeddings as the classifiers. We do this analysis on the DomainNet-LS dataset by taking the following five cases:

- (a) Here, domain mixing, which generates mixed style features is deactivated,

Table 4: Analysis of the contribution of each component in the proposed method using DomainNet-LS dataset.

	f_{ms}	f_{ic}	P_{Mix}	\mathcal{L}_{MSE}	Clipart	Infograph	Quickdraw	Sketch	Average
Case (a):	✗	✓	✓	✓	24.9	15.0	6.5	21.9	17.1
Case (b):	✓	✗	✓	✓	25.8	14.2	6.4	21.7	17.0
Case (c):	✓	✓	✗	✓	24.2	15.4	5.4	20.2	16.3
Case (d):	✓	✓	✓	✗	25.2	15.9	8.0	21.2	17.6
Case (e):	✓	✓	✓	✓	25.9	16.0	8.5	22.9	18.3

Fig. 2: t-SNE plots of the semantic space of test domains: (a) Clipart, (b) Painting and (c) Sketch in the DomainNet dataset for 10 unseen classes.



rest of the components are active. The mixing of two samples is done only on the feature level, the statistics of the features are not altered. Here, the model's ability to generalize to unseen domains would be hampered.

(b) Here, the CE loss corresponding to the intermediate class features f_{ic} is absent, thereby making the model less effective at recognizing unseen test classes.

(c) Our proposed method uses the knowledge of semantically similar classes for generating the pseudo intermediate classes for increasing the class discriminability. Here, we turn off this component making the model inefficient at distinguishing between similar classes. Here, for creating the intermediate classes, two randomly picked samples are used instead of semantically similar classes.

(d) Here, the information provided by the neighbouring classes i.e. \mathcal{L}_{MSE} defined in eq. (11) is not used.

(e) This is the proposed SEIC framework which uses all the components. We observe that all the proposed modules help towards improving the performance of the SEIC framework for the ZSDG task.

Visualization of the Semantic Space: Here, we visualize the feature embedding space which is learnt using the proposed SEIC framework. Fig. 2 shows the t-SNE plots of the feature embeddings for 10 randomly chosen unseen test




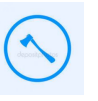

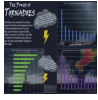







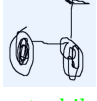






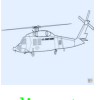

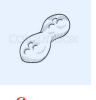

Infograph	marker	peas	skyscraper	axe	grapes	tornado
						
	marker	peas	skyscraper	marker	suitcase	ladder
Painting	axe	beard	skateboard	hurricane	peas	cloud
						
	axe	beard	skateboard	dolphin	asparagus	moon
Quickdraw	megaphone	motorbike	windmill	bread	rollerskates	sweater
						
	megaphone	motorbike	windmill	onion	sweater	windmill
Sketch	boomerang	hamburger	helicopter	parrot	peanut	scissors
						
	boomerang	hamburger	helicopter	dolphin	finger	airplane

Table 5: Model predictions for some test images in 5 domains of the DomainNet dataset. The ground truth is given at the top of each image. The correct (green) and incorrect (red) predictions are shown at the bottom of each image.

classes for the DomainNet dataset. We observe that the unseen test classes form reasonably nice clusters in the embedding space, even though the model has not been trained using data from these classes or domains. Also, the clusters are semantically meaningful, for example, in Fig. 2(c), we observe that semantically similar classes (living creatures) like *parrot*, *dolphin* and *octopus* are closer to each other compared to other different classes like *scissors* and *boomerang*.

Visual Examples of Correct and Incorrect Predictions: Table 5 shows few examples which are correctly and wrongly classified by the proposed SEIC framework. These images are from two domains of the DomainNet dataset. We observe that many of the wrong predictions are quite intuitive and may be wrongly classified even by humans. For example, the last image of painting, i.e. *cloud* is wrongly predicted as *moon*, the last image of quickdraw, i.e. *sweater* is wrongly predicted as *windmill*, etc.

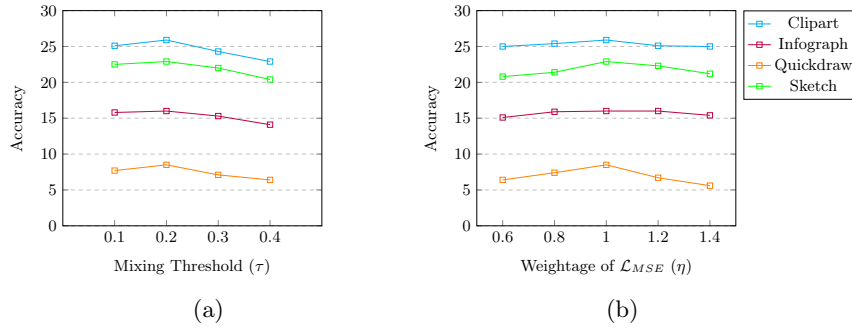


Fig. 3: Effect of variations in hyperparameters.

Effect of Varying Different Hyperparameters: First, we show the variation in performance for different values of τ , which controls how frequently a mixing module is activated. We analyze the performance for $\tau = \{0.1, 0.2, 0.3, 0.4\}$, on DomainNet-LS. In Fig. 3(a), we observe that the best result for each domain is obtained at $\tau = 0.2$. But the degradation of performance for different values of τ is very gradual, indicating that the model performance is quite stable with respect to this hyperparameter. In Fig. 3(b), we show the accuracy variation for different values of η , which is a hyperparameter weighting the importance of the MSE loss term in eq. (12). We analyze the results for $\eta = \{0.6, 0.8, 1.0, 1.2, 1.4\}$. Clearly, the best result is achieved when the value of η is set equal to 1. The trend is consistent across each of the four domains. Therefore, both the loss terms are given equal weightage in the final loss equation in our experiments.

6 Conclusion

In this work, we propose a novel framework termed SEIC, to address the ZSDG task. Specifically, we extend a state-of-the-art DG method capable of generalizing across unseen domains into a ZSDG framework which can handle unknown test classes as well. Generalization across unseen domains is achieved by generating intermediate domains by mixing the feature statistics of the different training samples. Similarly, generalization across unseen classes is handled by generating pseudo classes between similar seen classes using mixed features of the training samples. In addition, we also utilize the information of the neighbourhood classes to learn the semantically meaningful feature embeddings. Extensive experiments on two large-scale benchmark datasets and comparison with the state-of-the-art show the effectiveness of the proposed SEIC framework.

Acknowledgements: This work is partly supported through a research grant from SERB, Department of Science and Technology, Govt. of India.

References

1. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for attribute-based classification. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 819–826 (2013). <https://doi.org/10.1109/CVPR.2013.111>
2. Akata, Z., Reed, S.E., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2927–2936 (2015). <https://doi.org/10.1109/CVPR.2015.7298911>
3. Annadani, Y., Biswas, S.: Preserving semantic relations for zero-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7603–7612 (2018)
4. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.: A theory of learning from different domains. *Machine Learning* **79**, 151–175 (2010)
5. Blanchard, G., Lee, G., Scott, C.: Generalizing from several related classification tasks to a new unlabeled sample. In: *Advances in Neural Information Processing Systems* (2011)
6. Bucher, M., Herbin, S., Jurie, F.: Improving semantic embedding consistency by metric learning for zero-shot classification. In: *European Conference on Computer Vision (ECCV)*. pp. 730–746 (2016). https://doi.org/10.1007/978-3-319-46454-1_44
7. Cacheux, Y.L., Borgne, H.L., Crucianu, M.: Modeling inter and intra-class relations in the triplet loss for zero-shot learning. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 10332–10341 (2019). <https://doi.org/10.1109/ICCV.2019.01043>
8. Chandhok, S., Narayan, S., Cholakkal, H., Anwer, R.M., Balasubramanian, V.N., Khan, F.S., Shao, L.: Structured latent embeddings for recognizing unseen classes in unseen domains. In: *British Machine Vision Conference (BMVC)* (2021)
9. Chao, W., Changpinyo, S., Gong, B., Sha, F.: An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In: *European Conference on Computer Vision (ECCV)*. pp. 52–68 (2016)
10. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: Devise: A deep visual-semantic embedding model. In: *Advances in Neural Information Processing Systems*. pp. 2121–2129 (2013)
11. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.S.: Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **17**, 59:1–59:35 (2016)
12. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems*. pp. 2672–2680 (2014)
13. Huang, C., Li, Y., Loy, C.C., Tang, X.: Learning deep representation for imbalanced classification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016)
14. Huang, J., Guan, D., Xiao, A., Lu, S.: FSDR: frequency space domain randomization for domain generalization. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 6891–6902 (2021)
15. Huynh, D., Elhamifar, E.: Fine-grained generalized zero-shot learning via dense attribute-based attention. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4482–4492 (2020). <https://doi.org/10.1109/CVPR42600.2020.00454>

16. Iwasawa, Y., Matsuo, Y.: Test-time classifier adjustment module for model-agnostic domain generalization. In: *Advances in Neural Information Processing Systems (NeurIPS)*. pp. 2427–2440 (2021)
17. Jhoo, W.Y., Heo, J.: Collaborative learning with disentangled features for zero-shot domain adaptation. In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. pp. 8876–8885. IEEE (2021). <https://doi.org/10.1109/ICCV48922.2021.00877>
18. Jiang, H., Wang, R., Shan, S., Chen, X.: Adaptive metric learning for zero-shot recognition. *IEEE Signal Processing Letters* **26**(9), 1270–1274 (2019). <https://doi.org/10.1109/LSP.2019.2917148>
19. Jurie, F., Bucher, M., Herbin, S.: Generating visual representations for zero-shot classification. In: *2017 IEEE International Conference on Computer Vision Workshops (ICCV)*. pp. 2666–2673 (2017). <https://doi.org/10.1109/ICCVW.2017.308>
20. Khan, M.H., talha Zaidi, S.M., Khan, S., Khan, F.S.: Mode-guided feature augmentation for domain generalization. In: *British Machine Vision Conference (BMVC)*. p. 176 (2021)
21. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: *International Conference on Learning Representations (ICLR)* (2014)
22. Kodirov, E., Xiang, T., Gong, S.: Semantic autoencoder for zero-shot learning. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4447–4456 (2017). <https://doi.org/10.1109/CVPR.2017.473>
23. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 951–958 (2009). <https://doi.org/10.1109/CVPR.2009.5206594>
24. Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y., Hospedales, T.M.: Episodic training for domain generalization. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 1446–1455 (2019). <https://doi.org/10.1109/ICCV.2019.00153>
25. Li, H., Pan, S.J., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5400–5409 (2018). <https://doi.org/10.1109/CVPR.2018.00566>
26. Li, X., Fang, M., Chen, B.: Generalized zero-shot domain adaptation with target unseen class prototype learning. *Neural Computing and Applications* **34** (06 2022). <https://doi.org/10.1007/s00521-022-07413-z>
27. Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., Tao, D.: Deep domain generalization via conditional invariant adversarial networks. In: *Proceedings of the European Conference on Computer Vision (ECCV)* (September 2018)
28. Mancini, M., Akata, Z., Ricci, E., Caputo, B.: Towards recognizing unseen categories in unseen domains. In: *Proceedings of the European Conference on Computer Vision (ECCV)* (August 2020)
29. Maniyar, U., Joseph, K.J., Deshmukh, A.A., Dogan, U., Balasubramanian, V.N.: Zero-shot domain generalization. In: *British Machine Vision Conference (BMVC)* (2020)
30. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: *International Conference on Learning Representations (ICLR)* (2013)
31. Mishra, A., Reddy, M.S.K., Mittal, A., Murthy, H.A.: A generative model for zero shot learning using conditional variational autoencoders. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2188–2196 (2018). <https://doi.org/10.1109/CVPRW.2018.00294>

32. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.A.: Playing atari with deep reinforcement learning. ArXiv preprint arXiv:1312.5602. (2013)
33. Palatucci, M., Pomerleau, D., Hinton, G.E., Mitchell, T.M.: Zero-shot learning with semantic output codes. In: Advances in Neural Information Processing Systems. pp. 1410–1418 (2009)
34. Paul, S., Dutta, T., Biswas, S.: Universal cross-domain retrieval: Generalizing across classes and domains. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12056–12064 (October 2021)
35. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1406–1415 (2019)
36. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014). <https://doi.org/10.3115/v1/d14-1162>
37. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 23–30 (2017). <https://doi.org/10.1109/IROS.2017.8202133>
38. Vo, N., Jiang, L., Sun, C., Murphy, K., Li, L.J., Fei-Fei, L., Hays, J.: Composing text and image for image retrieval - an empirical odyssey. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
39. Volpi, R., Namkoong, H., Sener, O., Duchi, J.C., Murino, V., Savarese, S.: Generalizing to unseen domains via adversarial data augmentation. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 5339–5349 (2018)
40. Wang, K., Liew, J.H., Zou, Y., Zhou, D., Feng, J.: Panet: Few-shot image semantic segmentation with prototype alignment. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
41. Wang, Q., Breckon, T.P.: Generalized zero-shot domain adaptation via coupled conditional variational autoencoders. CoRR **abs/2008.01214** (2020)
42. Wang, Z., Luo, Y., Qiu, R., Huang, Z., Baktashmotlagh, M.: Learning to diversify for single domain generalization. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 814–823 (2021). <https://doi.org/10.1109/ICCV48922.2021.00087>
43. Xian, Y., Choudhury, S., He, Y., Schiele, B., Akata, Z.: Semantic projection network for zero- and few-label semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8256–8265 (2019). <https://doi.org/10.1109/CVPR.2019.00845>
44. Xian, Y., Schiele, B., Akata, Z.: Zero-shot learning - the good, the bad and the ugly. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3077–3086. IEEE Computer Society (2017). <https://doi.org/10.1109/CVPR.2017.328>
45. Xian, Y., Sharma, S., Schiele, B., Akata, Z.: F-VAEGAN-D2: A feature generating framework for any-shot learning. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10275–10284 (2019). <https://doi.org/10.1109/CVPR.2019.01052>
46. Xie, G., Liu, L., Jin, X., Zhu, F., Zhang, Z., Qin, J., Yao, Y., Shao, L.: Attentive region embedding network for zero-shot learning. In: IEEE Conference

- on Computer Vision and Pattern Recognition (CVPR). pp. 9384–9393 (2019). <https://doi.org/10.1109/CVPR.2019.00961>
47. Xu, Q., Zhang, R., Zhang, Y., Wang, Y., Tian, Q.: A fourier-based framework for domain generalization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14383–14392 (2021)
 48. Yang, F., Cheng, Y., Shiau, Z., Wang, Y.F.: Adversarial teacher-student representation learning for domain generalization. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 19448–19460 (2021)
 49. Zhang, H., Liu, S., Zhang, C., Ren, W., Wang, R., Cao, X.: Sketchnet: Sketch classification with web images. In: CVPR (2016). <https://doi.org/10.1109/CVPR.2016.125>
 50. Zhang, J., Shen, F., Liu, L., Zhu, F., Yu, M., Shao, L., Shen, H.T., Van Gool, L.: Generative domain-migration hashing for sketch-to-image retrieval. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
 51. Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Domain generalization with mixstyle. In: ICLR (2021)
 52. Zhu, F., Cheng, Z., Zhang, X., Liu, C.: Class-incremental learning via dual augmentation. In: Advances in Neural Information Processing Systems. pp. 14306–14318 (2021)