COCOA: Context-Conditional Adaptation for Recognizing Unseen Classes in Unseen Domains

Puneet Mangla* [1] Shivam Chandhok* [2] Vineeth N Balasubramanian[1] Fahad Shahbaz Khan[2,3]
[1]Indian Institute of Technology, Hyderabad [2]Mohamed bin Zayed University of AI, UAE
[3]Linköping University, Sweden

Abstract

Recent progress towards designing models that can generalize to unseen domains (i.e domain generalization) or unseen classes (i.e zero-shot learning) has embarked interest towards building models that can tackle both domain-shift and semantic shift simultaneously (i.e zero-shot domain generalization). For models to generalize to unseen classes in unseen domains, it is crucial to learn feature representation that preserves class-level (domain-invariant) as well as domain-specific information. Motivated from the success of generative zero-shot approaches, we propose a feature generative framework integrated with a COntext COnditional Adaptive (COCOA) Batch-Normalization layer to seamlessly integrate class-level semantic and domain-specific information. The generated visual features better capture the underlying data distribution enabling us to generalize to unseen classes and domains at test-time. We thoroughly evaluate our approach on established large-scale benchmarks – DomainNet, DomainNet-LS (Limited Sources) – as well as a new CUB-Corruptions benchmark, and demonstrate promising performance over baselines and state-of-the-art methods. We show detailed ablations and analysis to verify that our proposed approach indeed allows us to generate better quality visual features relevant for zero-shot domain generalization.

1. Introduction

The dependence of deep learning models on large amounts of data and supervision creates a bottleneck and hinders their utilization in practical scenarios. This is a major problem in computer vision, where it is difficult to acquire large amounts of labeled data due to practical feasibility constraints or high annotation costs. There is thus a need to equip deep learning models with the ability to generalize to unseen tasks or classes at test-time using data from other related tasks or classes (where data is abundant)

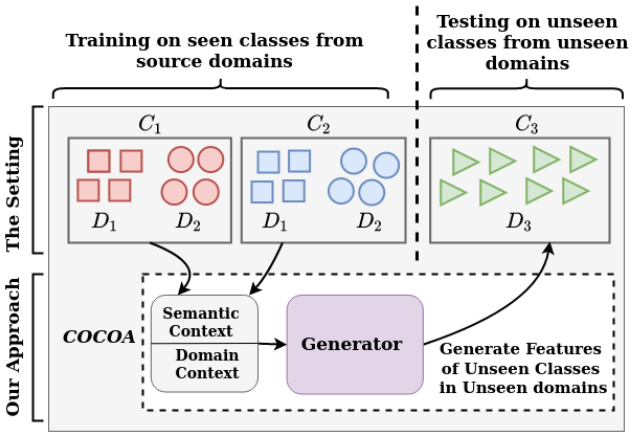

Figure 1. Overview of our approach. Shapes represent different domains and colors represent different classes. *(Top)* We show the zero-shot domain generalization problem setting in which we train on seen classes from source domains and model has to recognize unseen classes in unseen domains at test-time. *(Bottom)* Encoding semantic and domain context using COntext COnditional Adaptive (COCOA) Batch-Normalization layer helps to instill semantic and domain information into generated features. By jointly fusing both semantic and domain context, our method achieve best generalization to unseen classes in unseen domains at test-time.

[21, 32, 54, 52, 33, 34, 39, 11, 38]. Two main scenarios where this problem is most prevalent are: (1) when extensive labeled data is present in some related domains but not in the target domain of interest (i.e domain shift); or (2) there is data available for few classes but not all classes of interest (i.e semantic shift).

Domain shift challenges occur when the collection of data in the target domain or of all possible backgrounds/views of the depiction of objects is infeasible or tedious [25, 43, 15]. Semantic shift challenges occur when there are rare objects or long-tailed distributions which frequently occur in real-world scenarios [35]. There has been great interest and corresponding efforts in recent years towards tackling *domain shift* and *semantic shift* separately. However, applications in the real world do not guarantee that only one of them will occur – there is a need to make systems robust to the domain and semantic shifts simultaneously. The problem of tackling domain shift and semantic

*Equal Contribution

shift, as considered herein, together can be grouped as the Zero-Shot Domain Generalization (which we call ZSLDG) problem setting [27, 28]. In ZSLDG, the model has access to a set of seen classes from multiple source domains and has to generalize to unseen classes in unseen target domains at test time. Note that this problem of recognizing unseen classes in unseen domains (ZSLDG) is much more challenging than tackling zero-shot learning (ZSL) or domain generalization (DG) separately [27] and growingly relevant as deep learning models get reused across tasks.

Feature generation methods [52, 36, 42, 30] have recently shown significant promise in traditional zero-shot or few-shot learning settings and are among the state-of-the-art today for such problems. However, these approaches assume that the source domains (during training) and the target domain (during testing) are the same and thus aim to generate features that only address semantic shift. They rely on the assumption that the visual-semantic relationship learned during training will generalize to unseen classes at test-time. However, there is no guarantee that this holds for images in domains unseen during training [27]. Thus, when used for addressing ZSLDG, the aforementioned methods lead to suboptimal performance. To tackle domain shift at test-time, it is important to generate feature representations that encode both domain-invariant and domain-specific information at a class level [10, 44]. Thus we conjecture that generating features that are consistent with only class-level semantics is not sufficient for addressing ZSLDG. However, encoding domain-specific information along with class information in generated features is not straightforward [27].

Drawing inspiration from these observations, we propose a unified generative framework that uses COntext COnditional Adaptive (COCOA) Batch-Normalization to seamlessly integrate semantic and domain information to generate visual features of unseen classes in unseen domains. Depending on the setting, "*context*" can qualify as domain (for Domain Generalization), semantics (for Zero-Shot Learning), or both (in case of Zero-Shot Domain Generalization). Since this work primarily deals with the ZSLDG setting, 'context' in this work refers to both domain and semantic information (unless specified explicitly). Figure 1 provides an overall view of our framework and setting. Our key contributions are as follows:

- We devise a new feature generative framework integrated with COntext COnditional Adaptive (COCOA) batch normalization to encode both semantic and domain information for recognizing unseen classes in unseen domains.
- In addition, we use different regularization techniques to regulate the presence of semantic and domain information in generated features in order to achieve better generalization at test-time.
- We perform comprehensive experiments on standard benchmarks: DomainNet and DomainNet-LS (Limited Sources). Further, we introduce another fine-grained ZSLDG benchmark, CUB-Corruptions, that incorporates domain-shifts encountered in practical real-world scenarios. We demonstrate that our proposed methodology provides state-of-the-art performance for the ZSLDG setting on both established and proposed benchmarks.
- We perform extensive analysis to characterize the efficacy of COntext COnditional Adaptation (COCOA) in encoding both semantic and domain-specific information to generate features that capture the given context better.

To the best of our knowledge, this is the first generative approach to address the ZSLDG problem setting.

2. Related Work

Our work is situated at the intersection of ZSL and DG, and we hence review each of these kinds of methods first, before discussing the very few ZSLDG methods (two to be precise) that have been developed so far.

Domain Generalization: Deep learning models suffer from dataset bias which results in poor generalization when training and testing data come from different domains [47, 17]. Domain generalization methods aim to alleviate dataset bias and enable models to generalize to unseen domains at inference. Most such approaches can be broadly categorized into two groups: (1) Methods that attempt to develop a domain-agnostic feature representation that is robust to real-world image variations [32, 54, 21, 15, 25]; and (2) Methods that aim to utilize domain-dependent information for generalization. The former group includes traditional approaches to enforce generalization to domains [32, 54, 21], learning invariance via autoencoders [15] or using adversarial learning strategies to alleviate discrepancies between sources [25, 26]. Other approaches tackle domain shift by introducing specific training policies to simulate scenarios faced during test time [22] [23, 5] or using self-supervision [6] based techniques which enable the model to focus on the content of the image more than the domain. The common idea behind these approaches is that they tend to discard domain-specific information for generalization.

In contrast to this, the latter group comprises methods that aim to develop a domain-dependent representation [43, 44]. These approaches aim to collect independent domain statistics for each domain and interpolating them to infer the statistics for the domain of test samples at test time [43]. The main idea is to leverage the domain-specific information by mapping different domains to different points instead of leveraging the domain agnostic information.

Zero-Shot Learning Most methods that address the zero-shot learning problem aim to leverage the shared semantic space between the seen and unseen classes. The semantic representations usually comprise manually annotated at-

tributes, sentence embeddings, or word-vectors which essentially capture class-level characteristics associated with the object class. Traditional methods [7] strive to channel knowledge through the semantic attribute layer to model the joint and facilitate knowledge transfer from semantic to image space. Other approaches aim to learn a projection function from visual to semantic space [40, 4, 13], semantic to visual space [49, 20, 56, 46] or a common intermediate embedding space [48, 57], that can generalize to unseen classes at test-time.

Furthermore, recent efforts use generative models to addressing generalization to unseen classes by synthesizing visual features for unseen classes conditioned on their semantic representations. These methods aim to construct a generative model that can capture the modes of the data distribution well and facilitate knowledge transfer between visual and semantic spaces to ensure better generalization to unseen classes. To this end, recent approaches aim to generate discriminative visual features [52, 36, 42, 30, 12, 18, 19], combine strengths of different generative models like VAE and GAN [53, 33], the model joint likelihood of visual and semantic features in a generative framework [8, 36] or use other likelihood-based generative models such as normalizing flows [45] to generate visual features.

Zero-Shot Domain Generalization The problem of generalization to unseen classes in unseen domains i.e ZSLDG has been recently introduced [28, 27] and has attracted attention due to its usefulness in practical real-world scenarios encountered in computer vision. [28] showed results for previously proposed DG methods on the new ZSLDG setting. However, different domains were restricted here to different rotations of the same objects, which is limiting in practice. A more recent approach, CuMix [27], proposed a methodology that mixes up multiple source domains and categories available during training to simulate semantic and domain shift at test time. This work also established a benchmark dataset, DomainNet, for this setting with an evaluation protocol, which we follow in this work for a fair comparison. The DomainNet dataset is also a significant improvement over previous DG datasets, in terms of complexity and variety in domains.

3. COCOA: Proposed Methodology

We begin with a discussion of important considerations when designing an approach for ZSLDG.

Domain-shift: As discussed earlier, most methods that address only DG aim to map training data onto a domain-invariant manifold to tackle generalization to new domains [25, 55, 32, 54]. However, in practice, it is difficult to eliminate domain-specific cues to achieve a perfect domain-invariant representation. In practice, it is observed that feature representations that encode both domain-invariant and domain-specific information achieve improved recognition performance and generalization to new domains [10, 44].

Semantic-shift: Previous works that aim to learn an embedding function [13, 41, 2, 50, 48, 3, 9] to enable generalization to unseen classes typically suffer from bias towards training data (seen classes), thereby resulting in poor generalization to unseen classes. Generative methods alleviate this issue by synthesizing unseen class features [52, 53, 36, 42, 30, 12, 18, 19], thus reducing ZSL to a standard supervised learning setup. In practice, it is observed that generative approaches outperform other methods (such as embedding-based methods) enabling better generalization to unseen classes at test time.

Drawing motivation from these observations, we propose a generative learning-based model to address the problem of ZSLDG. Since both class-level domain-invariant features (common across domains), as well as unique individual domain-specific characteristics, are important for generalization to unseen classes in unseen domains [10, 44], we aim to learn a generative model which can encode both class-level semantic and domain-specific information in the generated features.

3.1. Problem Setting and Overall Framework

Our goal is to train a classifier $\mathcal{C}$ which can tackle domain-shift as well as semantic shift simultaneously and recognize unseen classes in unseen domains at test-time. Let $S^{Tr} = \{(\mathbf{x}, y, \mathbf{a}^s_y, d) | \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}^s, \mathbf{a}^s_y \in \mathcal{A}, d \in \mathcal{D}^s\}$ denote the training set, where $\mathbf{x}$ is a seen class image in the visual space ($\mathcal{X}$) with corresponding label y from a set of seen class labels $\mathcal{Y}^s$. $\mathbf{a}^s_y$ denotes the class-specific semantic representation for seen classes. We assume access to domain labels d for a set of source domains $\mathcal{D}^s$ with cardinality K. The test-set is denoted by $S^{Ts} = \{(\mathbf{x}, y, \mathbf{a}^u_y, d) | \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}^u, \mathbf{a}^u_y \in \mathcal{A}, d \in \mathcal{D}^u\}$ where $\mathcal{Y}^u$ is the set of labels for unseen classes and $\mathcal{D}^u$ represents the set of unseen target domains. Note that standard zero-shot setting (tackles only semantic-shift) complies with the condition $\mathcal{Y}^s \cap \mathcal{Y}^u \equiv \emptyset$ and $\mathcal{D}^s \equiv \mathcal{D}^u$. The standard DG setting (tackles only domain-shift), on the other hand, works under the condition $\mathcal{Y}^s \equiv \mathcal{Y}^u$ and $\mathcal{D}^s \cap \mathcal{D}^u \equiv \emptyset$. In this work, our goal is to address the challenging ZSLDG setting where $\mathcal{Y}^s \cap \mathcal{Y}^u \equiv \emptyset$ and $\mathcal{D}^s \cap \mathcal{D}^u \equiv \emptyset$. We aim to learn a mapping from $\mathcal{X} \rightarrow \mathcal{Y}^u$ such that $\mathbf{x} \in \mathcal{X}$ is sampled from a domain $d \in \mathcal{D}^u$ and has class label $y \in \mathcal{Y}^u$, given that, the model is trained using only images from seen classes $\mathcal{Y}^s$ in seen domains $\mathcal{D}^s$.

Our overall framework to address ZSLDG is summarized in Figure 2. We employ a three-stage pipeline, where in the first stage, we train a visual encoder to extract visual features $\mathbf{f}$ that encode discriminative information (described in Sec 3.2). This information consists of both class-level semantic information (domain-invariant) and domain-specific characteristics that help improve generalization [10]. Next,

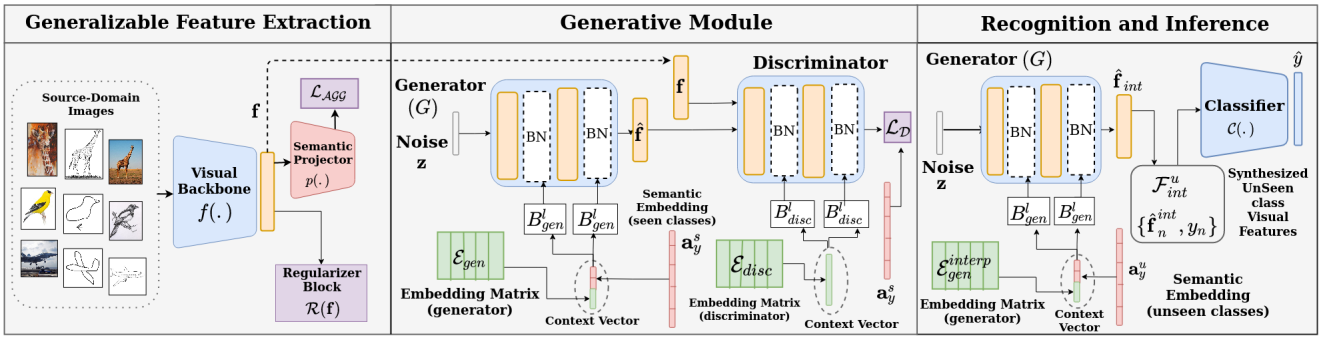

Figure 2. Overall architecture of our approach. The proposed pipeline consists of three stages: (1) Generalizable Feature Extraction; (2) Generative Module; and (3) Recognition and Inference. The first stage trains a visual backbone $f(.)$ to extract visual feature $\mathbf{f}$ that encodes discriminative class-level (domain-invariant) and domain-specific information. The second stage learns a generative model G which uses COntext COnditioned Adaptive (COCOA) batch-normalization layers to fuse and integrate semantic and domain-specific information into the generated features $\hat{\mathbf{f}}$. Lastly, the third stage generates visual features for unseen classes across domains and trains a softmax classifier on generated features

we train a generative model to generate features $\mathbf{f}$ from noise with the help of COntext COnditional Adaptation (COCOA) (described in Sec 3.3). This enables our model to capture both class-level and domain-specific information in generated features $\hat{\mathbf{f}}$ and generate features relevant for the task. In the third stage, we use our trained generative model to generate features for unseen classes across domains and train a classifier that can handle domain shift as well as semantic shift simultaneously at test-time (described in Sec 3.4).

3.2. Generalizable Feature Extraction

Here, we describe the first stage in our proposed pipeline. Since the semantic representations of both seen and unseen class labels lie in the same space, capturing class-level semantic (domain-invariant) information in visual features becomes important for generalization to unseen classes. In addition to domain-invariant attributes, domain-specific information is also important for generalization to new domains [10]. To this end, we extract visual features $\mathbf{f}$ that encode discriminative class-level cues by training a visual encoder $f(.)$, which is then used to train a semantic projector $p(.)$. Both these modules are trained with images from all source domains available. The visual encoder and the semantic projector are trained to minimize the following loss:

$$\mathcal{L}_{AGG} = \mathbb{E}_{(\mathbf{x},y)\sim S^{Tr}}\left[\mathcal{L}_{CE}(\mathbf{a}^T p(f(\mathbf{x})), y)\right] \quad (1)$$

where $\mathcal{L}_{CE}$ is standard cross-entropy loss and $\mathbf{a} = [\mathbf{a}^s_1, ..., \mathbf{a}^s_{|\mathcal{Y}^s|}]$. To further improve the visual features and enhance their generalizability, we also propose to use a regularization block resulting in an added regularizer loss term $\mathcal{R}(\mathbf{f})$. The regularizer block primarily uses rotation prediction (viz. providing a rotated image as input, and predicting rotation angle), although we attempt other strategies such as CuMix [27] or domain classification in our experiments in Sec 4. The overall loss function to train the visual encoder hence becomes:

$$\mathcal{L}_R = \mathcal{L}_{AGG} + \mathcal{R}(\mathbf{f}) \quad (2)$$

where $\mathcal{R}(\mathbf{f})$ denotes the regularization loss term.

3.3. Generative Module

We now describe the next step in our pipeline i.e learning the generative network. To this end, we learn a generative model which comprises of a generator $G : \mathcal{Z} \times \mathcal{A} \times \mathcal{D} \rightarrow \mathcal{F}$ and a projection discriminator [31] (which uses a projection layer to incorporate conditional information into the discriminator) $D : \mathcal{F} \times \mathcal{A} \times \mathcal{D} \rightarrow \mathbb{R}$ as shown in Figure 2. We represent this discriminator as $D = D_l \circ D_f$ ($\circ$ denotes composition) where D_f is discriminator's last feature layer and D_l is the final linear layer. Both the generator and the projection discriminator are conditioned through a Context Conditional Adaptive (COCOA) Batch-Normalization module, which provides class-specific and domain-specific modulation at various stages during the generation process. Modulating the feature information by such semantic and domain-specific embeddings helps to integrate respective information into the features, thereby enabling better generalization. We use the available (seen) semantic attributes $\mathbf{a}^s_y$ that capture class-level characteristics, for encoding semantic information. However, such a representation that encodes domain information is not present for individual source domains. We hence define learnable (randomly initialized and optimized during training) domain embedding matrices $\mathcal{E}_{gen} = \{\mathbf{e}^{gen}_1, \mathbf{e}^{gen}_2, \mathbf{e}^{gen}_3 ..., .\mathbf{e}^{gen}_K\}$ and $\mathcal{E}_{disc} = \{\mathbf{e}^{disc}_1, \mathbf{e}^{disc}_2, \mathbf{e}^{disc}_3 ....\mathbf{e}^{disc}_K\}$ for the generator and the discriminator respectively.

The generator G takes in noise $\mathbf{z} \in \mathcal{Z}$ and a context vector $\mathbf{c}$, and outputs visual features $\hat{\mathbf{f}} \in \mathcal{F}$. The context vector $\mathbf{c}$, which is the concatenation of class-level semantic attribute $\mathbf{a}^s_y$ and domain-specific embedding $\mathbf{e}^{gen}_i$, is provided as input to a BatchNorm estimator network $B^l_{gen} : \mathcal{A} \times \mathcal{D} \rightarrow \mathcal{R}^{2\times h}$ (h is the dimension of layer l activation vectors) which outputs batchnorm paramaters, γ^l_{gen}

and β^l_{gen} for the l-th layer. Similarly, the discriminator's feature extractor $D_f(.)$ has a separate BatchNorm estimator network $B^l_{disc} : \mathcal{D} \rightarrow \mathcal{R}^{2\times h}$ to enable domain-specific context modulation of its batchnorm parameters, $\gamma^l_{disc}, \beta^l_{disc}$ as shown in Figure 2.

Formally, let $\mathbf{f}^l_{gen}$ and $\mathbf{f}^l_{disc}$ denote feature activations belonging to domain d and semantic attribute $\mathbf{a}^s_y$, at l-th layer of generator G and discriminator D respectively. We modulate $\mathbf{f}^l_{gen}$ and $\mathbf{f}^l_{disc}$ individually using Context Conditional Adaptive Batch-Normalization as follows:

$$\begin{aligned}
&\gamma^l_{gen}, \beta^l_{gen} \leftarrow B^l_{gen}(\mathbf{c}) \text{ where } \mathbf{c} = [\mathbf{a}^s_y, \mathbf{e}^{gen}_d] \\
&\mathbf{f}^{l+1}_{gen} \leftarrow \gamma^l_{gen} \cdot \frac{\mathbf{f}^l_{gen} - \mu^l_{gen}}{\sqrt{(\sigma^l_{gen})^2 + \epsilon}} + \beta^l_{gen} \\
&\gamma^l_{disc}, \beta^l_{disc} \leftarrow B^l_{disc}(\mathbf{e}^{disc}_d) \\
&\mathbf{f}^{l+1}_{disc} \leftarrow \gamma^l_{disc} \cdot \frac{\mathbf{f}^l_{disc} - \mu^l_{disc}}{\sqrt{(\sigma^l_{disc})^2 + \epsilon}} + \beta^l_{disc}
\end{aligned} \tag{3}$$

Here, $(\mu^l_{gen}, (\sigma^l_{gen})^2)$ and $(\mu^l_{disc}, (\sigma^l_{disc})^2)$ are the mean and variance of activations of the mini-batch (also used to update running statistics) containing $\mathbf{f}^l_{gen}$ and $\mathbf{f}^l_{disc}$ respectively. $\mathbf{c}$ denotes the context vector composed of semantic and domain embeddings and $[\cdot, \cdot]$ denotes the concatenation.

Finally, the generator and discriminator are trained to optimize the adversarial loss given by:

$$\begin{aligned}
\mathcal{L}_D = &\mathbb{E}_{(\mathbf{x}, \mathbf{a}^s_y, d)\sim S^{Tr}} \left[\max(0, 1 - D(f(\mathbf{x}), \mathbf{a}^s_y, \mathbf{e}^{disc}_d))\right] \\
&+ \mathbb{E}_{y\sim\mathcal{Y}^s, d\sim\mathcal{D}^s, \mathbf{z}\sim\mathcal{Z}} \left[\max(0, 1 + D(\hat{\mathbf{f}}, \mathbf{a}^s_y, \mathbf{e}^{disc}_d))\right]
\end{aligned} \tag{4}$$

$$\begin{aligned}
\mathcal{L}_G = &\mathbb{E}_{y\sim\mathcal{Y}^s, d\sim\mathcal{D}^s, \mathbf{z}\sim\mathcal{Z}}[-D(\hat{\mathbf{f}}, \mathbf{a}^s_y, \mathbf{e}^{disc}_d) \\
&+ \lambda_G \cdot \mathcal{L}_{CE}(\mathbf{a}^T p(\hat{\mathbf{f}}), y)]
\end{aligned} \tag{5}$$

where $D(\mathbf{f}, \mathbf{a}, \mathbf{e}) = \mathbf{a}^T D_f(\mathbf{f}, \mathbf{e}) + D_l(D_f(\mathbf{f}, \mathbf{e}))$ is the projection term , $\hat{\mathbf{f}} = G(\mathbf{z}, \mathbf{c})$ denotes generated feature. The second term, $\mathcal{L}_{CE}(\mathbf{a}^T p(\hat{\mathbf{f}}), y)$ in $\mathcal{L}_G$ ensures that the generated features have discriminative properties (λ_G is a hyperparameter). $p(.)$ is the semantic projector that was trained in the previous stage.

3.4. Recognition and Inference

In this section, we describe the training procedure of the classifier $\mathcal{C}$ and the inference mechanism for our proposed model. Given a generative network G trained on the training set S^{Tr} as described in Sec 3.3, we freeze the parameters of the generator and aim to synthesize visual features for unseen classes in different domains. To this end, we concatenate the domain embeddings $\mathbf{e}^{gen}_i, (i \in \mathcal{D}^s)$ of the source domains from matrix $\mathcal{E}_{gen}$ (learned end-to-end with the generative model) and the semantic attributes/representations of unseen classes $\mathbf{a}^u_y$ to get the context vector $\mathbf{c}$, which in turn is input to the trained batchnorm predictor network B^l_{gen}. The output batchnorm parameters $(\gamma^l_{gen}, \beta^l_{gen})$ are used in the batchnorm layers of the pre-trained generator to generate features $\hat{\mathbf{f}}$. This enables us to generate features that are consistent with unseen class semantics and also encode domain information from individual source domains in the training set. After obtaining batch-norm parameters, the new set of unseen class features are generated as follows:

$$\begin{aligned}
&\hat{\mathbf{f}}_n = G(\mathbf{z}_n, \mathbf{c}) \\
&\text{where } \mathbf{c} = [\mathbf{a}^u_{y_n}, \mathbf{e}^{gen}_n], \mathbf{z}_n \sim \mathcal{Z}, y_n \sim \mathcal{Y}^u, \mathbf{e}^{gen}_n \sim \mathcal{E}_{gen}
\end{aligned} \tag{6}$$

To improve generalization to new domains at test-time and make the classifier domain-agnostic, we synthesize embeddings of newer domains by interpolating the learned embeddings of the source domains via a mix-up operation.

$$\mathcal{E}^{gen}_{interp} = \lambda \cdot \mathbf{e}^{gen}_i + (1-\lambda) \cdot \mathbf{e}^{gen}_j \text{ where } i, j \sim \mathcal{D}^s, \lambda \sim \mathcal{U}[0, 1] \tag{7}$$

where $\mathbf{e}^{gen}_i$ and $\mathbf{e}^{gen}_j$ refer to the domain embeddings of i-th and j-th source domains. The unseen class features generated using interpolated domain embeddings $\mathcal{F}^u_{int} = \{(\hat{\mathbf{f}}^n_{int}, y_n)\}^N_{n=1}$ are generated as follows:

$$\begin{aligned}
&\hat{\mathbf{f}}^n_{int} = G(\mathbf{z}_n, \mathbf{c}_{interp.}) \text{ where } \mathbf{c}_{interp.} = [\mathbf{a}^u_{y_n}, \mathbf{e}^{gen}_{interp.}], \\
&\mathbf{z}_n \sim \mathcal{Z}, y_n \sim \mathcal{Y}^u, \mathbf{e}^{gen}_{interp.} \sim \mathcal{E}^{gen}_{interp}
\end{aligned} \tag{8}$$

Next, we train a MLP softmax classifier, $\mathcal{C}(.)$ on the generated multi-domain unseen class features dataset $\mathcal{F}^u_{int}$ by minimizing:

$$\mathcal{L}_{CLS} = \mathbb{E}_{(\hat{\mathbf{f}}_{int}, y)\sim\mathcal{F}^u_{int}}[\mathcal{L}_{CE}(\mathcal{C}(\hat{\mathbf{f}}_{int}), y)] \tag{9}$$

Classifying image at test time. At test-time, given a test image $\mathbf{x}_{test}$, we first pass it through the visual encoder $f(.)$ to get the discriminative feature representation $\mathbf{f}_{test} = f(\mathbf{x})$. Next we pass this feature representation to the classifier to get the final prediction.

$$\hat{y} = \arg\max_{y\in\mathcal{Y}^u} \mathcal{C}(\mathbf{f}_{test})[y] \tag{10}$$

4. Experiments and Results

In this section, we describe the experimental setup and validate the performance of our proposed methodology with previously proposed baselines and models addressing the problem of ZSLDG.

Datasets: We evaluate our method with all baselines on the DomainNet and DomainNet-LS benchmark datasets, described briefly below. In the Appendix, we provide description and first results on a newer dataset - CUB-Corruptions

DomainNet: Recently, [27] proposed DomainNet dataset as a benchmark for the ZSLDG problem setting. DomainNet is a diverse large-scale coarse-grained dataset and currently the established benchmark for the ZSLDG problem setting [27]. Originally proposed for domain adaptation

[37], DomainNet is a dataset with 0.6 million images belonging to 345 different categories divided into 6 different domains i.e *painting*, *real*, *clipart*, *infograph*, *quickdraw* and *sketch*. The *infograph* and *quickdraw* domains are considered harder to generalize to, since they involve the most evident domain-shift at test time. We follow the seen-unseen class training-testing splits and protocol as established in [27] where we train on 5 domains and test on the left-out domain. We consider each domain as target domain individually through a separate experiment and report average unseen classification accuracy over all such experiments. Also, following [27], we use *word2vec* representations [29] as the semantic representations for class labels.

DomainNet-LS: This benchmark, also established in [27], refers to the setting where source domains are limited to only *real* and *painting* domains. Compared to DomainNet, this is a more challenging setting since the model may overfit to the fewer source domains [25], thus making it difficult to learn generalizable domain-invariant features.

Baselines. For a comprehensive evaluation, we compare our approach with the following set of baselines which include simpler baselines derived from standard ZSL/DG literature (following recent work in ZSLDG [27]) as well as the recent ZSLDG methods.

- Simpler baselines include ZSL methods like *SPNet* [51], *DeViSE* [13], *ALE* [1] and their coupling with well-known DG methods (DANN [14], EpiFCR [24]) (as in [27]).
- State-of-the-art ZSLDG method, *CuMix* [27] as well as its variants: (1) *Mixup-img-only* where mixup is done only at image level without curriculum; (2) *Mixup-two-level* where mixup is applied at both feature and image level without curriculum; (3) *CuMix-reverse* where the curriculum strategy is applied but the order of semantic and domain mixing is switched.
- Feature generation baselines which include f-clsWGAN [52] (standard ZSL only approach) and its combination with visual backbone trained using AGG (Equation 1) objective (called *AGG + f-clsWGAN*), as well as CuMix [27] methodology (*CuMix + f-clsWGAN*) and self-supervised rotation [16] feature extraction method (*ROT + f-clsWGAN*).

Note that our proposed approach can be seamlessly integrated into other generative ZSL frameworks, which we leave for future work.

Implementation Details Due to space constraints, we describe the implementation details and hyperparameter settings for our method in the Appendix.

Results on DomainNet Benchmark. Table 1 shows the performance comparison of our method with the aforementioned baselines. We observe that a simple classification-based feature extraction (using only $\mathcal{L}_{\mathcal{AGG}}$) without any regularization when coupled with our generative module i.e *COCOA*$_{AGG}$ is able to outperform the current state-of-the-art, CuMix [27]. In addition, when we use rotation prediction as a a regularizing auxiliary task [16] (referred as *COCOA*$_{ROT}$), we observe that it achieves the best performance on all domains individually as well as on average (with significant improvement especially on hard domains like *quickdraw* where there is large domain shift encountered at test-time), thus outperforming [27] by a margin of about 2% average across domains (which corresponds to a $\sim$9% relative increase). We believe this is because the auxiliary task enables better representation learning. More analysis with other regularizers is shown later in the paper.

Utilizing generative ZSL baselines as it is provides sub-optimal generalization performance for the harder ZSLDG problem setting. Also, we observe that combining such generative ZSL frameworks (which generate features with only class level semantics) with different visual backbones including CuMix results in inferior average performance when compared with our approach.

Results on DomainNet-LS Benchmark. Table 2 shows the individual domain and average performance of our method and baselines on this limited-source setting. We observe that even in this challenging setting, our method outperform [27] by a margin of $\sim 0.5\%$ average across all domains (corresponds to $\sim 3\%$ relative improvement).

Evaluation on Corrupted-CUB benchmark, Standard DG and ZSL settings: Owing to space constraints, we report the results of our method on Corrupted-CUB (a newly introduced dataset for ZSLDG), standard ZSL and DG benchmarks in the Appendix.

Component-wise Ablation Study. Table 3 shows the component-wise performance for our method. We use the DomainNet dataset [27] and rotation-based self-supervision regularization for this study.

- $S1$ corresponds to the performance of the feature extractor $f(.)$ trained using using rotation prediction as a regularizer (without generative stage 2).
- $S2$ corresponds to the performance achieved by learning a generator G without using domain embeddings as an input. Specifically, this implies that the BatchNorm estimator networks B^l_{gen} is given only the class-level semantic attribute as input i.e context vector $\mathbf{c} = \mathbf{a}^s_y$.
- $S3$ denotes the use of $S2$, with domain embeddings as an additional input in the context vector provided to B^l_{gen} i.e $\mathbf{c} = [\mathbf{a}^s_y, \mathbf{e}^{gen}_d]$. Note that both $S2$ and $S3$ follow the inference mechanism described in Sec 3.4 using domain embeddings $\mathcal{E}_{gen}$, without the use of interpolated domain embeddings $\mathcal{E}^{gen}_{interp}$ (Eqn 6)
- Lastly, $S4$ denotes our complete model, where we train the final classifier also on features generated by using interpolated domain embeddings $\mathcal{E}^{gen}_{interp}$ ((Eqn 8).

Method		Target Domain					
DG	ZSL	*clipart*	*infograph*	*painting*	*quickdraw*	*sketch*	Avg.
-	DEVISE [13]	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 [51]	26.0	16.9	23.8	8.2	21.8	19.4
DANN [14]	DEVISE [13]	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 [51]	25.9	15.8	24.1	8.4	21.3	19.1
EpiFCR [24]	DEVISE [13]	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 [51]	26.4	16.7	24.6	9.2	23.2	20.0
Mixup-img-only		25.2	16.3	24.4	8.7	21.7	19.2
Mixup-two-level		26.6	17	25.3	8.8	21.9	19.9
CuMix[27]		27.6	17.8	25.5	9.9	22.6	20.7
f-clsWGAN [52]		20.0	13.3	20.5	6.6	14.9	15.1
AGG + f-clsWGAN		27.4	17.0	25.9	11.0	23.8	21.0
CuMix + f-clsWGAN		27.3	17.9	26.5	11.2	24.8	21.5
ROT + f-clsWGAN		27.5	17.4	26.4	11.4	24.6	21.4
COCOA$_{AGG}$		27.6	17.1	25.7	11.8	23.7	21.2
COCOA$_{ROT}$		**28.9**	**18.2**	**27.1**	**13.1**	**25.7**	**22.6**

Table 1. Performance comparison with established baselines and state-of-art methods for ZSLDG problem setting scenario on benchmark DomainNet dataset. We report performance on individual domains as well as average performance. For fair comparison, all reported results follow the backbones, protocol and splits as established in [27]. Best results are highlighted in bold and second best results are underlined

Method	*Clipart*	*Infograph*	*Quickdraw*	*Sketch*	Avg
SPNet	21.5	14.1	4.8	17.3	14.4
Epi-FCR + SPNet	22.5	14.9	5.6	18.7	15.4
MixUp img only	21.2	14.0	4.8	17.3	14.3
MixUp two-level	22.7	16.5	4.9	19.1	15.8
CuMix reverse	22.9	15.8	4.8	18.2	15.4
CuMix	23.7	**17.1**	5.5	19.7	16.5
COCOA$_{AGG}$	23.8	15.6	6.7	20.1	16.55
COCOA$_{ROT}$	**23.82**	15.9	**7.3**	**20.75**	**16.94**

Table 2. ZSLDG performance comparison with state-of-art methods on DomainNet-LS setting. For fair comparison, all reported results follow the protocol and splits as established in [27]. Best results are highlighted in bold and second best results are underlined.

From Table 3, we infer that learning a generative network and training final classifier $C(.)$ on synthesized unseen class features (i.e. $S2$) brings significant improvement in the average performance when compared with standalone feature extractor-based pipeline $S1$. Also, we observe that modulating the intermediate features in the generative network with batchnorm parameters conditioned on both domain and semantic embeddings (i.e $S3$) improves performance on all domains individually as well as enhances average performance when compared with only semantic embeddings-based context in $S2$. This corroborates our hypothesis that conditioning on both domain and semantic embeddings enables the classifier to discriminate between the distribution of features $\mathbf{f}$ better by encoding both domain-specific and class-level information in generated features $\hat{\mathbf{f}}$ (we discuss this further via feature visualization as well). Furthermore, interpolating source domain embeddings to get new domain representations and training the final classifier using features generated by these interpolated domain embeddings

Variant	*Clipart*	*Infograph*	*Painting*	*Quickdraw*	*Sketch*	Avg
$S1$	27.5	17.8	25.4	9.5 7	22.5	20.54
$S2$	27.6 7	17.36	27.08	11.57	24.97	21.716
$S3$	28.5	17.6	26.8	12.7	25.48	22.2
$S4$	28.9	18.2	27.1	13.1	25.7	22.6

Table 3. Ablation study for different components of our framework on DomainNet dataset

Fusion	**Input**	**BN**	*Clipart*	*Infograph*	*Painting*	*Quickdraw*	*Sketch*	Avg
F1	$[\mathbf{z}, \mathbf{a}_y, \mathbf{e}^{gen}]$	-	27.75	14.77	23.93	10.79	25.31	20.51
F2	$\mathbf{z} + [\mathbf{a}_y, \mathbf{e}^{gen}]$	-	28.23	14.99	24.34	10.0	23.47	20.2
F3	$\mathbf{z} + \mathbf{a}_y$	-	24.87	16.09	23.52	11.85	22.89	19.84
F4	$[\mathbf{z}, \mathbf{a}_y]$	$\mathbf{e}^{gen}$	26.56	16.9	24.89	13.0	24.9	21.25
F5	$\mathbf{z} + \mathbf{a}_y$	$\mathbf{e}^{gen}$	27.74	16.19	26.38	11.04	24.13	21.1
F6	$\mathbf{z}$	$[\mathbf{a}_y, \mathbf{e}^{gen}]$	28.9	18.2	27.1	13.1	25.7	22.6

Table 4. Performance comparison to analyse different potential ways to fuse and encode class-level semantic and domain-specific information in generated features $\hat{\mathbf{f}}$ while training the generative network G. Symbols $\mathbf{z}, \mathbf{a}_y, \mathbf{e}^{gen}$ represents *noise*, *attribute* and *domain embedding* respectively.

$S4$ further improves performance by alleviating the bias towards source domains and enhancing the generalization capabilities of the model.

Fusion Techniques. In this subsection, we validate our choice of using batchnorm-based fusion of domain-specific and class-level semantic information. Specifically, we compare the model performance for different choices of potential ways that could be used to encode this information in the generated features. We show this analysis in Table 4. The input layer column corresponds to the vector input to the generative model and the BN layer corresponds to the embeddings that are input to the BatchNorm estimator network B^l_{gen}. Furthermore, [.] denotes the concatenation of vectors and $'+'$ denotes the addition operation between vectors.

We observe that simply concatenating the domain and semantic embeddings together with noise and providing it as input directly to the generative model (instead of conditional batchnorm) i.e $F1$ leads to lower performance than the proposed COCOA approach $F6$. Also, concatenating only semantic embeddings with noise in the input layer and having batchnorm parameters conditioned solely on domain embeddings i.e $F4$ also exhibits lower average performance when compared with $F6$. We observe that using $'+'$ operator deteriorates performance when compared with the concatenation-based counterpart for all cases. Overall, we achieve the best performance with the proposed method.

Visualization of Generated Features. We now present the visualization of features generated by our trained generative model. We sample semantic attributes $\mathbf{a}_y^u$ for randomly selected unseen classes and use domain embeddings (learned end-to-end) of the five source domains (*Real*, *Infograph*, *Quickdraw*, *Clipart*, *Sketch*) used during training. We individually visualize the features generated of each unseen class using only semantic embeddings/context i.e $\mathbf{c} = \mathbf{a}_y^u$ (Fig 3, Row 1) and concatenation of both semantic and domain embeddings/context i.e $\mathbf{c} = [\mathbf{a}_y^u, \mathbf{e}_d^{gen}]$ (Fig 3, Row 2) when estimating the batchnorm parameters of the generator. We notice that conditioning the batchnorm parameters only on semantic context (Fig 3, Row 1) collapses the generated features of different domains to the same point in the cluster and suffers from *mode dropping* as observed in [30]. This occurs since the model is unable to explain the variance in features due to domain-specific information. On the other hand, when both semantic and domain context are used (Fig 3, Row 2), the model better captures the modes of the original data distribution. It can be seen that the model can better retain domain-specific variance (associated with the five source domains) within a specific class cluster when both semantic and domain embeddings are both used. More such results are shown in the Appendix.

Analysis of Regularizers. As in Sec 3.2, our model is trained with a regularizer block using an additional loss term $\mathcal{R}(\mathbf{f})$, which helps our model to generalize well, as well as regulate the presence of semantic and domain information in visual features. We now analyze the choice of rotation prediction as a regularizer, and study other regularizers for their relevance to this setting, in particular: (1) CuMix [27] that provides feature regularization and has been studied for the ZSLDG setting; we refer to this variant as *COCOA*$_{CuMix}$; and (2) A domain classification regularization to understand whether regulating the amount of domain-specific information also brings any improvement. We refer to this variant as *COCOA*$_{DOM(\alpha)}$ and the objective is given by:

$$\mathcal{L}_{DOM(\alpha)} = \mathcal{L}_{AGG} + \alpha \cdot \mathbb{E}_{(\mathbf{x},y,d)\sim S^{Tr}}[\mathcal{L}_{CE}(g(f(\mathbf{x})), d)] \tag{11}$$

where $g(.)$ is a domain classifier (implemented as single linear layer). By varying the hyperparameter α, we can regulate the amount of domain-information in features f. From Table 5, we notice that when CuMix-based regularization is used, features generated using i.e *COCOA*$_{CuMix}$ leads to improvement in the average performance relative to *COCOA*$_{AGG}$. We also observe that adding domain classification as an auxiliary prediction task does not bring enhancement in the performance (as α is varied from 0 to 0.1). We hypothesize that while domain information is relevant for better generalization, more such information can cause features to lose their class-discriminative capability. Further analyzing better regularizers to learn better base features is a potential direction for future work.

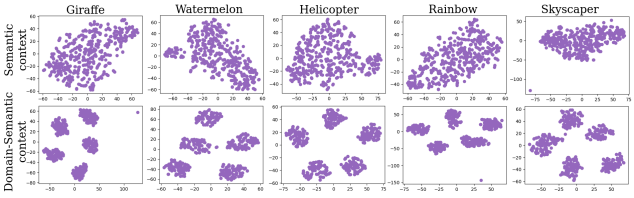

Figure 3. Individual t-SNE visualization of synthesized image features by COCOA for randomly selected unseen classes (*Giraffe*, *Watermelon*, *Helicopter*, *Rainbow*, *Skyscaper*) using only semantic context (Row 1) and both domain-semantic context (Row 2) *(Best viewed in color, zoomed in).*

Variant		*Clipart*	*Infograph*	*Painting*	*Quickdraw*	*Sketch*	Avg
COCOA$_{DOM(\alpha)}$	$\alpha = 0$	27.6	17.1	25.7	11.8	23.7	21.2
	$\alpha = 0.01$	28.14	16.1	27.19	10.55	23.8	21.1
	$\alpha = 0.05$	26.26	16.7	26.88	11.2	23.6	20.9
	$\alpha = 0.1$	26.48	15.5	26.9	11.21	23.7	20.7
COCOA$_{CuMix}$		27.7	17.5	**27.8**	12.6	25.6	22.2
COCOA$_{ROT}$		**28.9**	**18.2**	27.1	**13.1**	**25.7**	**22.6**

Table 5. Analysis on the choice of Regularizer $R(\mathbf{f})$. Note that COCOA$_{DOM(\alpha)}$ with $\alpha = 0$ is same as COCOA$_{AGG}$

5. Conclusion

In this work, we propose a unified generative framework for the ZSLDG problem setting that uses an elegant approach to encode class-level (domain-invariant) and domain-specific information. Our approach uses context conditional batch-normalization to integrate class-level semantic and domain-specific information into generated visual features, thereby enabling better generalization. We conduct extensive experiments on benchmark ZSLDG datasets and demonstrate the effectiveness of the proposed method. Furthermore, we show extensive analysis to validate our choice of using conditional batch-normalization to fuse semantic and domain-dependent characteristics. Our future work will include the development of better methods to effectively fuse and regulate the presence of semantic and context information to further improve generalization performance for unseen classes in unseen domains.

Acknowledgement: This work has been partly supported by the funding received from DST through the IMPRINT program (IMP/2019/000250)

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2013.

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label embedding for image classification. *TPAMI*, 2016.

[3] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. *CVPR*, 2015.

[4] Yashas Annadani and Soma Biswas. Preserving semantic relations for zero-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7603–7612, 2018.

[5] Y. Balaji, S. Sankaranarayanan, and R. Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, 2018.

[6] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, B. Caputo, and T. Tommasi. Domain generalization by solving jigsaw puzzles. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2224–2233, 2019.

[7] Lampert CH, Nickisch H, and Harmeling S. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[8] Shivam Chandhok and V. Balasubramanian. Two-level adversarial visual-semantic coupling for generalized zero-shot learning. *WACV*, 2021.

[9] S. Changpinyo, W.-L. Chao, B.; Gong, and F. Sha. Synthesized classifiers for zero-shot learning. *CVPR*, 2016.

[10] Prithvijit Chattopadhyay, Y. Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. volume abs/2008.12839, 2020.

[11] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. Object counting and instance segmentation with image-level supervision. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12389–12397, 2019.

[12] Rafael Felix, Vijay BG Kumar, Ian Reid, and Gustavo Carneiro. Multi-modal cycle-consistent generalized zero-shot learning. *In Proceedings of the European Conference on Computer Vision*, 2018.

[13] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

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

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ArXiv*, abs/1803.07728, 2018.

[17] Boqing Gong, K. Grauman, and Fei Sha. Reshaping visual datasets for domain adaptation. In *NIPS*, 2013.

[18] He Huang, Changhu Wang, Philip S Yu, and Chang-Dong Wang. Generative dual adversarial network for generalized zero-shot learning. *CVPR*, 2019.

[19] Rohit Keshari, Richa Singh, and Mayank Vatsa. Generalized zero-shot learning via over-complete distribution. *CVPR*, 2020.

[20] Elyor Kodirov, Tao Xiang, Zhenyong Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2452–2460, 2015.

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

[22] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018.

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

[24] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1446–1455, 2019.

[25] Haoliang Li, Sinno Jialin Pan, S. Wang, and A. Kot. Domain generalization with adversarial feature learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018.

[26] Y. Li, X. Tian, Mingming Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, 2018.

[27] Massimiliano Mancini, Zeynep Akata, E. Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *ECCV*, 2020.

[28] Udit Maniyar, K. J. Joseph, A. Deshmukh, Ü. Dogan, and V. Balasubramanian. Zero-shot domain generalization. *ArXiv*, abs/2008.07443, 2020.

[29] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[30] Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A Murthy. A generative model for zero shot learning using conditional variational autoencoders. *CVPRW*, 2018.

[31] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018.

[32] Krikamol Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. *ArXiv*, abs/1301.2115, 2013.

[33] Sanath Narayan, A. Gupta, F. Khan, Cees G. M. Snoek, and L. Shao. Latent embedding feedback and discriminative features for zero-shot classification. *ArXiv*, abs/2003.07833, 2020.

[34] Sanath Narayan, Akshita Gupta, Salman Khan, Fahad Shahbaz Khan, Ling Shao, and Mubarak Shah. Discriminative region-based multi-label zero-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8731–8740, October 2021.

[35] Jian Ni, Shanghang Zhang, and Haiyong Xie. Dual adversarial semantics-consistent network for generalized zero-shot learning. *NeurIPS*, 2019.

[36] Jian Ni, Shanghang Zhang, and Haiyong Xie. Dual adversarial semantics-consistent network for generalized zero-shot learning. *NeurIPS,*, 2019.

[37] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[38] Jathushan Rajasegaran, Munawar Hayat, Salman Hameed Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for continual learning. In *NeurIPS*, 2019.

[39] Jathushan Rajasegaran, Salman Hameed Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. itaml: An incremental task-agnostic meta-learning approach. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13585–13594, 2020.

[40] Scott E. Reed, Zeynep Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 49–58, 2016.

[41] B. Romera-Paredes and P. H. Torr. An embarrassingly simple approach to zero-shot learning. *ICML*, 2015.

[42] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. *CVPR*, 2019.

[43] Mattia Segu, A. Tonioni, and Federico Tombari. Batch normalization embeddings for deep domain generalization. *ArXiv*, abs/2011.12672, 2020.

[44] Seonguk Seo, Yumin Suh, D. Kim, Jongwoo Han, and B. Han. Learning to optimize domain specific normalization for domain generalization. 2020.

[45] Yuming Shen, J. Qin, and L. Huang. Invertible zero-shot recognition flows. In *ECCV*, 2020.

[46] Yutaro Shigeto, Ikumi Suzuki, K. Hara, M. Shimbo, and Y. Matsumoto. Ridge regression, hubness, and zero-shot learning. In *ECML/PKDD*, 2015.

[47] A. Torralba and Alexei A. Efros. Unbiased look at dataset bias. *CVPR 2011*, pages 1521–1528, 2011.

[48] Y.-H. H. Tsai, L.-K. Huang, and R. Salakhutdinov. Learning robust visual-semantic embeddings. *ICCV*, 2017.

[49] Ziyu Wan, Dongdong Chen, Y. Li, Xingguang Yan, Junge Zhang, Y. Yu, and Jing Liao. Transductive zero-shot learning with visual structure constraint. In *NeurIPS*, 2019.

[50] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. *CVPR*, 2016.

[51] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8256–8265, 2019.

[52] Yongqin Xian, Tobias Lorenz, Bernt Schiele, , and Zeynep Akata. Feature generating networks for zero-shot learning. *CVPR*, 2018.

[53] Y. Xian, S. Sharma, B. Schiele, and Z. Akata. A feature generating framework for any-shot learning. *CVPR*, 2019.

[54] Zheng Xu, W. Li, Li Niu, and Dong Xu. Exploiting low-rank structure from latent domains for domain generalization. In *ECCV*, 2014.

[55] P. Yang and Wei Gao. Multi-view discriminant transfer learning. In *IJCAI*, 2013.

[56] L. Zhang, Tao Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3010–3019, 2017.

[57] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via joint latent similarity embedding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6034–6042, 2016.