Zero Shot Domain Generalization

Udit Maniyar[1]
es16btech11024@iith.ac.in

Joseph K J[1]
cs17m18p100001@iith.ac.in

Aniket Anand Deshmukh[2]
Aniket.Deshmukh@microsoft.com

Urun Dogan[2]
urundogan@gmail.com

Vineeth Balasubramanian[1]
vineethnb@iith.ac.in

[1] Indian Institute of Technology, Hyderabad, India

[2] Microsoft, Sunnyvale, CA, USA

Abstract

Standard supervised learning setting assumes that training data and test data come from the same distribution (domain). Domain generalization (DG) methods try to learn a model that when trained on data from multiple domains, would generalize to a new unseen domain. We extend DG to an even more challenging setting, where the label space of the unseen domain could also change. We introduce this problem as *Zero-Shot Domain Generalization* (to the best of our knowledge, the first such effort), where the model generalizes across new domains and also across new classes in those domains. We propose a simple strategy which effectively exploits semantic information of classes, to adapt existing DG methods to meet the demands of Zero-Shot Domain Generalization. We evaluate the proposed methods on CIFAR-10 [17], CIFAR-100 [17], F-MNIST [31] and PACS [19] datasets, establishing a strong baseline to foster interest in this new research direction.

1 Introduction

Generalization is a key desideratum for machine learning models to scale to the dynamic nature of the real world. The standard supervised learning framework assumes that train and test data are from the same distribution (domain). Domain generalization techniques [5, 14, 19, 20, 22] demand to train a model in such a way that it can generalize to a novel domain at inference, by gracefully handling domain shift. However, current domain generalization methods assume the same classes to be present in all domains (including unseen test domains), which is a restriction on the application of such methods. Our work attempts to relax this assumption, and allow novel test domains to have new classes that were not present in any training domain. We introduce this harder problem as *Zero-Shot Domain Generalization*, and to the best of our knowledge, is the first such effort (Fig 1 illustrates the setting). We note that the standard zero-shot learning problem [3, 4, 12, 29, 30] provides a model to generalize to unseen classes, but assumes that datapoints come from a single known domain.

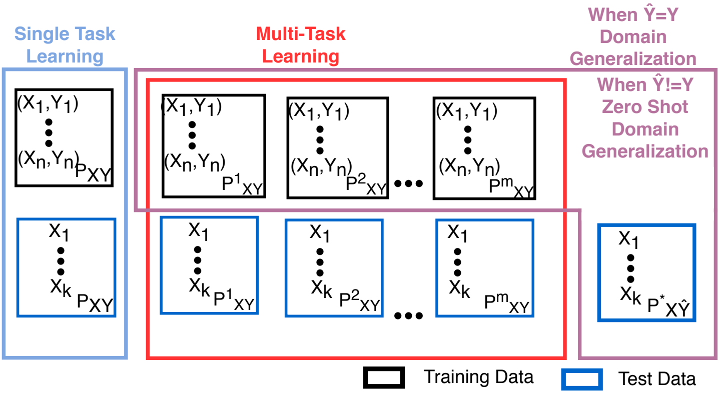

Figure 1: In a single-task learning (light blue), both training data-points and test data-points comes from the same distribution P_{XY}. In multi-task learning (red), data from all distributions are available for training ($P^1_{XY} \cdots P^m_{XY}$). In domain generalization (pink), the model is evaluated on data from unseen distribution P^*_{XY}, while in Zero-Shot DG, the model needs to be performant on any unseen $P^*_{X\hat{Y}}$, where $\hat{Y} \neq Y$.

We hypothesize that learning a domain-invariant feature representation, with explicit class information, would help address Zero-Shot Domain Generalization. Encoding class information in the feature space ensures smooth transition of information from the classes that are seen during training on multiple source domains, to the unseen set of classes in the new domain. To this end, we adapt state-of-the-art domain generalization methods - in particular, Feature Critic Network (FC) [22] and Multi-Task Auto Encoder (MTAE) [14] - to ensure that their intermediate feature representations are semantically consistent. Our experimental evaluation on domain generalization variants of CIFAR-10 [17], CIFAR-100 [17], F-MNIST [31] and PACS [19] provide promise, and establish baselines for this new research problem.

2 Related Work

Domain Generalization: Multi-Domain Learning, Domain Generalization, Domain Adaptation are closely related topics. All of these deal with scenarios in which a model is trained on a source distribution and is used in the context of a different (but related) target distribution. In Multi-Domain Learning the target domain and train domains are the same. In Domain Generalization the target domain is different from that of train domains. In Domain Adaptation also the target domain is different from the train domain but we have access to unlabeled data from the target domains during training.

In domain generalization (DG), we are given training data from different domains and the objective is to generalize to a novel domain. Blanchard *et al.* 2011 [5, 6] proposed a kernel mean embedding based algorithm to get similarity between different domains and transfer the learning. The same kernel mean embedding idea was used to extend this DG framework to a multiclass setting in [8]. Ghifary *et al.* 2015 [14] proposed an autoencoder based method called Multi-Task Auto Encoder (MTAE) to learn domain-invariant features. In MTAE, encoder for all domains is same but decoder is different which forces encoder to learn a common feature space for all domains. Ghifary *et al.* 2017 [15] proposes scatter component analysis (SCA) to solve domain generalization problem. SCA finds representation that trades between maximizing the separability of classes and data, and minimizing the mismatch between domains through scatter. Motiian *et al.* 2017 [26] proposed classification and contrastive semantic alignment (CCSA) to learn a domain-invariant embedding by minimizing the sum of classification loss, confusion alignment loss, and semantic alignment loss. Confusion alignment loss is a distance between distribution and semantic alignment loss makes sure that samples from different domains and with different labels are mapped as far apart as possible in the embedding space. Li *et al*. 2017 [19] proposed a method that takes advantage of the robustness of deep learning models to domain shift, and developed a low-rank parameterized CNN model for end-to-end DG learning. Li *et al*. 2017 [20] proposed a meta-learning approach to solve for DG by simulating train-test domain shifts during training by synthesizing virtual test domains. Li *et al*. 2019 [22] extended this approach to produce a more general feature extractor that can be used with any classifier, by simultaneously learning an auxiliary loss function that trains the feature extractor for improved domain invariance. None of these efforts attempted Zero-Shot DG, setting our work apart.

Zero-Shot Learning (ZSL): On the other hand, Zero-Shot learning is a supervised learning method, in which the classes of images seen during training is disjoint from the classes present during testing. Successful zero-shot models should maintain very high accuracy when presented test examples from those unseen classes. In generalized zero-shot setting, the model should maintain good performance on all the classes which includes both the base classes and new zero-shot classes. The main idea in most zero-shot algorithms is to learn a semantically consistent correlation between seen and unseen classes, using semantic information between classes. Semantic information can be either manually specified as attributes [10, 11, 16] or imparted from word embeddings such as Word2Vec [24] or GloVe vectors[28]. Since attribute based methods use binary mapping they result in loss of information and hence have been shown to be suboptimal[3]. Semantic embedding based algorithms first learn an embedding function which maps visual space to semantic space, to aid in this task. Socher *et al*. 2013 [29] minimizes Euclidean distance loss with word vectors in semantic space as objective, and learns a two-layer neural network for classification. Attribute Label Embedding (ALE) [3], Deep Visual Semantic Embedding (DEVISE) [12] and Structured Joint Embedding (SJE) [1] learn a bilinear compatibility function to map image to semantic space. To learn the compatibility function, different objectives functions have been explored: DEVISE[12] uses pairwise ranking objective, [4] takes a dot product between the embedded visual feature and semantic vectors considering three training losses, including a binary cross entropy loss, hinge loss and Euclidean distance loss. Yongqin *et al*. 2017 [30] presents a survey of zero-shot learning methods.

Even though both DG and ZSL have been explored by the community in isolation, to the best of our knowledge, no existing work in literature has attempted to solve the Zero-Shot DG problem. This setting can have practical use in applications such as robotics, medical image analysis and general scene understanding. We identify this problem in our work, formalize the same and provide a methodology that can serve as a baseline.

3 Zero-Shot Domain Generalization

3.1 Problem Definition

We define a domain as the joint distribution of feature and label space. Let the training data for the i^{th} domain be: $(X_{ij}, Y_{ij}) \sim P^i_{XY}$ and $P^i_{XY} \sim \mu$. In a DG setting, the test data comes from the same distribution μ: $(X^T_j, Y^T_j) \sim P^T$ and $P^T \sim \mu$, while in zero-shot domain generalization, it comes from a different distribution ν: $(X^T_j, Y^T_j) \sim P^T$ and $P^T \sim \nu$, where ν

contains additional unseen labels compared to μ (note that feature space and common label space follow the same data generation structure as μ but it differs only for and when unseen classes are seen).

We assume all (X,Y) pairs are drawn i.i.d. from their respective distributions. In particular, let Y^{tr} and Y^{ts} represent the set of classes in training and test data respectively, such that $Y^{tr} \cap Y^{ts} = \emptyset$. The training data for the i^{th} domain is given by $D_i = \{X_{ij}, Y_{ij}\}_{1 \leq j \leq n_i}$ and $Y_{ij} \in Y^{tr}$, and the test data set be $D^T = \{X_j^T, Y_j^T\}_{1 \leq j \leq n_T}$ and $Y_j^T \in Y^{ts}$ where n_i is the number of images in the i^{th} domain. The main objective of ZSDG problem is to train a model on all training domains $D = \{D_1, D_2, \dots D_N\}$ to perform well on D^T.

3.2 Proposed Approach

We propose a generic approach to extend the state-of-the-art DG techniques to solve zero-shot domain generalization. The key insight is to bind the intermediate domain-invariant feature representations to a semantic space that is shared across the seen classes of the old domains and the unseen classes of the new domain. Such use of semantic space has been successfully used in recent ZSL methodologies [12, 29, 30].

Existing DG methods [14, 22] can be considered as a composition of a feature extractor function f_θ and classifier function g_ϕ: $(g \circ f)(I)$, where I is the input data-point. In this paper, we only consider DG methods which generate domain invariant features and train a common classifier on those features. We restrict domain invariant features to semantic space, forcing the model to be semantically consistent along with domain invariance. Our proposed method can be extended to any DG method which learn domain invariant features but does not apply to methods which learn different features for different domains. While training the proposed semantically constrained DG approach, the features generated by f_θ are projected to a semantic space of labels. Images of similar classes are grouped together, and dissimilar classes spaced apart by a semantic alignment loss given in Equation 1. By using the semantic embedding and restricting lower-level features of the model to a semantic space, a shared invariant representation is learned where semantic alignment is accounted for. We make an inherent assumption that classes which appear visually similar should also be semantically similar (as in other ZSL methods). Thus using semantic space helps us in the visual classification task. We use word embeddings of classes - in particular, simple GloVe embeddings [28] trained on Common Crawl corpus - as the semantic space in this work. One could use more complex embedding functions to study this even further.

During inference, we depart from traditional DG methods which train a separate classifier (neural network or SVM) on the domain-invariant features. We instead use nearest neighbour (NN) search in the semantic space. Our method is lightweight and can scale to large numbers of unseen classes using existing efficient NN search methods. We now describe how we infuse semantic information into three DG methods, to solve zero-shot DG.

Our approach focuses on DG methods which learn domain-invariant features and cannot be any combination of DG + ZSL methods. For example existing SOTA ZSL methods, based on generative models, cannot be directly extended this way. The proposed method is novel in terms of how one does training + inference in ZSDG, diff from existing DG methods.

3.2.1 Semantic AGG (S-AGG)

Aggregation (AGG) is a simple baseline, which has strong DG performance [19, 22]. Here, we group data from all domains and train the network on this multi-domain dataset. In line with our generic definition for DG functions, the model is split into two parts: feature extractor f_θ and classifier g_ϕ. f_θ contains a series of convolutional layers followed by fully connected layers which map the image from a higher dimension $\mathbb{R}^x$ to class-vector dimension $\mathbb{R}^h$. g_ϕ is also a classifier (neural network) which maps from $\mathbb{R}^h$ to number of classes. We now define our semantic loss in this case as follows:

$$\mathcal{L}^{Semantic}(\theta) = ||(f_\theta(X_{ij}) - w[Y_{ij}])||^2 \tag{1}$$

where $w[Y_{ij}]$ denotes the word embedding of the label of image X_{ij}. We modify the aggregation based method to include a semantic loss in a low dimension feature space $\mathbb{R}^h$. Semantic AGG (S-AGG) is hence trained to minimize the following loss function:

$$\min_{\theta,\phi} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \mathcal{L}^{CE}(g_\phi(f_\theta(X_{ij})), Y_{ij}) + \lambda \cdot \mathcal{L}^{Semantic}(\theta) + \eta \mathcal{R}(\theta, \phi) \tag{2}$$

where $\mathcal{L}^{CE}$ is cross-entropy loss, λ is weighing factor and $\mathcal{R}$ is a regularizer.

3.2.2 Semantic MTAE (S-MTAE)

Multi-task Auto Encoder (MTAE) [14] learns domain-invariant features using an autoencoding framework. In MTAE, the encoder acts as a feature extractor. A single encoder (h_Θ) is maintained across domains, which projects each image to a latent representation. Domain-specific decoders ($g_{\Phi_1}, g_{\Phi_2}, \dots g_{\Phi_N}$) take these representations and regenerate the image back to each domain. This implicitly forces the encoder to learn an unbiased, domain-agnostic projection function. The final classification function is learned using these domain-invariant features with an SVM or a simple MLP. We adapt MTAE for Zero-Shot DG by restricting the latent representation to the semantic embedding of the class labels.

MTAE ensures self-domain reconstruction and inter-domain reconstruction. For every pair of domains $(D_i, D_j)_{1 \le i,j \le N}$, images of domain D_j are generated from images of domain $D_i : (g_{\Phi_j} \circ h_\Theta)(I)$ where I is an image of class k in domain D_j that is regenerated (decoded) from images of class k in domain D_i. While doing the above reconstruction the objective is the features captured by the feature extractor are domain invariant and can be used further for classification. In standard MTAE we only have a reconstruction loss term. Our semantic loss restricts the feature space of MTAE to the semantic embedding space of classes, along with standard reconstruction loss. By doing so, these features also act as a criteria for classification i.e we do not need additional neural network or SVM for classification. The loss function for S-MTAE while training domain i is hence:

$$\mathcal{L}(\Theta, \Phi_1, \dots, \Phi_N, i) = \sum_{j=1}^{N} \mathcal{L}^{MSE}(g_{\Phi_j}(h_\Theta(D_i)), D_j) + \lambda \cdot \mathcal{L}^{Semantic}(h_\Theta(D_i), w[Y_{D_i}]) \tag{3}$$

where $w[Y_{D_i}]$ is the word embedding of classes in Y_{D_i}. We minimize the following objective to train across N domains:

$$\min_{\Theta,\Phi_1,\dots,\Phi_N} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\Theta, \Phi_1, \dots, \Phi_N, i) + \eta \mathcal{R}(\Theta, \Phi_1, \dots, \Phi_N) \tag{4}$$

3.2.3 Semantic FC (S-FC)

Feature Critic Networks (FC) [22] provides a meta-learning approach to DG. It learns a domain-invariant feature extractor by meta-learning an auxiliary loss that ‘criticizes’ the effectiveness of features generated by the feature extractor, when dealing with an unseen domain. The network is trained by simulating training-to-testing domain shift by splitting the source domains into virtual training and testing meta-domains, following standard meta-learning practice. The goal is to train a model which can perform well both across new domains and also across zero-shot classes. We achieve both these objectives in a novel way i.e model learns to generalize horizontally(across domains) by simulating domain shift and generalize vertically(across zero-shot classes) by exploiting the semantic information of classes.

Continuing with our generic architecture of domains generalization methods, FC can also be viewed as a composition of a feature extractor f_θ and classifier g_ϕ. While training, we split D into $D^{tr}_{meta}, D^{ts}_{meta}$ such that $D^{tr}_{meta} \cap D^{ts}_{meta} = \emptyset$. We train the model on D^{tr}_{meta} but expect it to perform well on D^{ts}_{meta}. The model is trained using the following loss function:

$$\min_{\theta,\phi} \sum_{D_i \in D^{tr}_{meta}} \sum_{X,Y \in D_i} \mathcal{L}^{CE}(g_\phi(h_\theta(X)),Y) + \lambda \cdot \mathcal{L}^{Semantic}(h_\theta(X),w[Y]) + \mathcal{L}^{Aux} \tag{5}$$

$w[Y]$ refers to word vectors of classes of Y, $\mathcal{L}^{CE}$ is cross-entropy loss, and $\mathcal{L}^{Semantic}$ is the semantic loss defined in Eqn 1. $\mathcal{L}^{Aux}$ is the meta-loss that encourages feature extractor to generate domain-agnostic features. We refer readers to [22] for specifics of the meta-training strategy.

4 Experiments and Results

4.1 Datasets

We evaluate the proposed methods on four different datasets: CIFAR-10 [17], Fashion-MNIST [31], CIFAR-100 [17] and PACS [19]. PACS is the only ready made DG dataset and since PACS contains only 4 domains even its role is limited. We are thus forced to generate new datasets based on rotations. Similar to [9, 14, 21] six different domains are obtained with 0° to 75° rotation and a step of 15° for each non DG dataset (CIFAR-10, Fashion-MNIST, CIFAR-100). For each non DG dataset $\mathcal{D}$, we have $(\mathcal{D}_0, \mathcal{D}_{15}, \mathcal{D}_{30}, \mathcal{D}_{45}, \mathcal{D}_{60}, \mathcal{D}_{75})$ where $\mathcal{D}_x$ = images rotated by x degrees. Images are zero-padded as required after rotation.

PACS[19] and Rotated-MNIST[18] are often used in earlier DG work [9, 14, 20, 21, 22]. However, we restrain from using Rotated-MNIST because of the lack of connection between the visual space and semantic space of numbers here.

We perform leave-one-out experiments with domains for all datasets. We measure standard DG performance as well as Zero-Shot DG performance on the left-out domain in each experiment. Our experiments are carried out on multiple settings as described below (each setting denotes the zero-shot classes used in the experiment on each dataset):
(i) *CIFAR-10:* Setting 1: (cat, truck); Setting 2: (cat, dog); Setting 3: (deer, ship); Setting 4: (car, deer); Setting 5: (airplane, car); Same zero-shot classes were used in [29].
(ii) *Fashion-MNIST:* Setting 1: (t-shirt, sandal); Setting 2: (sandal, shirt); Setting 3: (t-shirt, boot); Setting 4: (sandal, Boot).
(iii) *CIFAR-100:* Setting 1: (whale, fish, rose, can, orange, lamp, couch, beetle, tiger, skyscraper, mountain, kangaroo, fox, snail, man, snake, squirrel, pine-tree, motorcycle, streetcar); Setting 2: (seal, shark, poppy, bottle, apple, keyboard, table, caterpillar, lion, bridge, forest, camel, raccoon, crab, girl, dinosaur, rabbit, maple, bicycle, tractor). These classes were randomly selected. Only two settings are considered due to the complexity of the dataset.

(iv)*PACS:* Setting 1: (horse, house); Setting 2: (dog,house); Setting 3: (giraffe,person); Setting 4: (elephant,house).

In each setting, independently, ZSDG performance is measured on data of unseen classes from target domain, and DG performance is measured on data of seen classes from target domain. We considered diff settings to study robustness.

4.2 Implementation Details

The partition size of D^{tr} and D^{ts} in Semantic FC is 3:2; i.e three domains are chosen as meta-train and two as meta-test. We compare the performance across 6 different methods: AGG, S-AGG, FC, S-FC, MTAE, S-MTAE. Here AGG, FC and MTAE, we mean the vanilla method without semantic loss and S-{AGG, MTAE, FC} denote their semantic counterpart. The ZSDG accuracies are computed using distance in the semantic space to the class vectors. The vanilla method helps us in understanding the usefulness of adding the semantic loss. The reported results are averaged across 5 seeds. All the codes, videos and other resources will be available at https://github.com/aniketde/ZeroShotDG.

We meticulously designed fair experiments for this new setting. As defined in Sec 4.1, we remove some classes (chosen randomly) from domains used for training, from known DG datasets. Trained ZSDG models are evaluated on these unseen classes in unseen domains. Vanilla DG methods form the baseline (which is fair, considering lack of any other methods at this time). The network, dataset, training strategies in vanilla methods and semantic counterparts are all kept consistent for fair comparison.

Choice of Word Vectors: Word2Vec[25] uses a two-layer neural net that processes text by vectorizing words. GloVe[28] is an unsupervised learning algorithm which uses word-word co-occurrence statistics from a corpus to obtain word vectors. We use the pre-extracted Word2Vec and GloVe vectors from wikipedia provided by [2]. After comparing with different semantic vectors we found GloVe embeddings [28] to be much more meaningful and hence help in getting good results.

4.3 Results

Tables 2, 3, 4 and 5 present Domain Generalization (DG) and Zero-Shot Domain Generalization (ZSDG) results on F-MNIST [31], CIFAR-10 [17], CIFAR-100 [17] and PACS [19] datasets respectively. For brevity, we average the accuracy across the domains and present the average accuracy in these tables. The complete results with standard deviations across our multiple runs are in the Supplementary material. We see in the tables that the proposed semantically consistent adaptations of DG methods perform better both on DG and ZSDG. On rotations of F-MNIST, S-MTAE performs better than all other methods with the exception of Setting 4. On rotations of CIFAR-10, S-MTAE and S-FC both perform the best. On rotations of CIFAR-100, S-FC performs better when compared to other methods. From the results, one can hypothesize that for simpler datasets, basic DG methods such as MTAE are sufficient and yield good performance, but when the complexity in the dataset increases (as in CIFAR-100), more complex methods such as FC are required for better performance. For purposes of better understanding, we also present in Table 1 the ZSL performance on the considered settings on the CIFAR-10 dataset using the method proposed by Socher *et al.* [29]. We note that the comparatively lower numbers in Table 3 is because we only use 4000

TARGET	Setting 1		Setting 2		Setting 3		Setting 4	
	DG	ZSDG	DG	ZSDG	DG	ZSDG	DG	ZSDG
AGG	67.16	61.51	70.24	51.59	67.21	57.13	60.87	53.36
S-AGG	69.11	57.673	73.19	56.87	68.08	41.88	62.37	52.5
MTAE	18.12	73.105	18.41	70.79	17.50	79.44	17.62	**64.26**
S-MTAE	**72.97**	**92.45**	**77.29**	**89.54**	**72.17**	**89.00**	**65.56**	52.71
FC	66.17	56.61	69.03	52.33	66.14	53.18	59.18	51.41
S-FC	66.53	49.03	69.93	54.24	66.33	61.04	58.83	53.94

Table 2: Domain Generalization (DG) and Zero-Shot Domain Generalization (ZSDG) performance of different domains on Fashion-MNIST dataset.

TARGET	Setting 1		Setting 2		Setting 3		Setting 4		Setting 5	
	DG	ZSDG	DG	ZSDG	DG	ZSDG	DG	ZSDG	DG	ZSDG
AGG	51.58	48.94	50.85	49.87	48.79	42.63	49.42	52.40	47.93	51.21
S-AGG	48.55	79.77	48.68	53.58	46.04	**81.75**	46.26	82.59	44.94	65.86
FC	**52.18**	54.82	51.87	50.19	**49.56**	51.69	**49.95**	45.00	**48.37**	52.40
S-FC	51.35	**81.1**	50.98	55.3	48.53	77.37	49.29	81.59	47.79	71.15
MTAE	12.14	54.55	11.74	51.19	12.67	56.55	12.41	52.91	12.66	54.62
S-MTAE	51.92	80.12	**51.94**	**55.35**	49.13	79.94	49.42	**83.2**	47.95	**71.63**

Table 3: Domain Generalization (DG) and Zero-Shot Domain Generalization (ZSDG) performance of different domains on CIFAR-10 dataset.

images from the different domains of CIFAR-10, which is lower than the original dataset. We performed a Wilcoxon signed rank sum test and found that our results of semantic variants are statistically significant at a p-value of 0.04.

PACS is widely used in the Domain Generalization papers. But PACS contains only four domains and seven classes, nevertheless we have performed some experiments on PACS and the results are in table 5. On PACS S-AGG has the best ZSDG accuracy whereas FC has the best DG accuracy. MTAE & S-MTAE did not perform as expected and hence we have not reported these results. We hypothesize that since PACS contains too few classes, current methods are not suitable for task at hand.

Target:	Setting 1	Setting 2	Setting 3	Setting 4	Setting 5
zero-shot	93.17	53.55	86.06	88.31	67.54

Table 1: Zero-Shot classification accuracies on CIFAR-10 using our implementation of Socher *et al.* [29].

5 Ablation Studies

5.1 Changing Weighting Co-efficients

A very direct ablation study is changing the weighing factor of the semantic loss and observing the performances. Consider the below general loss function, where λ is the weighing factor.

$$\min_{\theta} \mathcal{L}^{model}(\theta) + \lambda \cdot \mathcal{L}^{Semantic}(\theta) \tag{6}$$

Different values for λ have been considered and accuracies are averaged over five runs and results are in the Table 6. We infer that as the weight of the semantic loss increases

	Setting 1		Setting 2	
	DG	ZSDG	DG	ZSDG
AGG	80.31	5.87	80.47	6.08
S-AGG	74.98	19.99	75.5	20.11
FC	**83.62**	5.5	**83.62**	5.52
S-FC	83.47	**20.17**	**83.62**	**20.7**
MTAE	1.45	5.00	1.29	5.44
S-MTAE	82.03	19.26	82.16	19.24

Table 4: Domain Generalization (DG) and Zero-Shot Domain Generalization (ZSDG) performance of different domains on CIFAR-100 dataset.

	Setting 1		Setting 2		Setting 3		Setting 4	
	DG	ZSDG	DG	ZSDG	DG	ZSDG	DG	ZSDG
AGG	**74.33**	48.03	77.39	50.9	75.48	55.47	71.64	44.06
S-AGG	69.96	**80.13**	72.125	**77.94**	71.24	50.21	68.2	**66.84**
FC	72.93	45.53	**77.76**	46.39	**75.68**	57.37	**71.72**	43.59
S-FC	66.11	76.22	67.55	74.97	67.6	**58.45**	64.03	63.08

Table 5: Domain Generalization (DG) and Zero-Shot Domain Generalization (ZSDG) performance on PACS dataset.

	λ	Setting 1		Setting 2		Setting 3		Setting 4		Setting 5	
		DG	ZSDG	DG	ZSDG	DG	ZSDG	DG	ZSDG	DG	ZSDG
S-AGG	0.1	57.42	**83.44**	56.87	56.18	53.91	80.34	54.39	86.46	52.75	73.61
	0.5	57.84	82.39	57.41	56.15	54.27	80.20	54.88	86.62	52.53	73.83
	1	51.92	82.42	52.40	53.67	49.24	**84.20**	49.47	84.75	47.78	67.67
	5	59.04	81.69	58.62	56.30	55.66	80.25	56.46	87.48	53.89	75.34
	10	**60.09**	82.45	**59.68**	**56.36**	**56.48**	80.26	**57.54**	**87.62**	**54.59**	**75.36**
S-FC	0.1	53.79	82.39	53.51	54.85	51.21	77.5	50.79	82.53	49.64	70.19
	0.5	54.25	**83.69**	53.56	55.29	51.30	79.84	51.18	83.93	49.72	72.42
	1	54.27	82.92	54.27	55.63	51.80	**81.57**	**51.59**	83.86	50.08	71.83
	5	54.92	82.4	54.83	56.24	52.18	80.94	**51.59**	84.00	49.99	72.58
	10	**55.09**	82.99	**55.79**	**56.63**	**52.69**	80.47	51.49	**85.59**	**50.11**	**74.51**
S-MTAE	0.1	56.68	83.14	55.76	56.44	52.17	81.74	52.99	86.83	49.83	75.49
	0.5	54.24	79.52	54.36	56.83	52.16	81.41	52.08	84.52	48.03	71.84
	1	57.22	82.03	56.36	56.34	52.70	81.26	53.17	84.85	50.77	71.98
	5	**58.98**	82.39	58.38	56.58	**55.82**	**83.68**	**55.59**	**86.86**	53.27	**73.60**
	10	58.09	**82.74**	**58.55**	**56.90**	55.50	83.65	56.30	87.1	**53.78**	73.38

Table 6: CIFAR-10 Dataset; left-out-domain = D_3; Domain Generalization and Zero-Shot Domain Generalization (ZSDG) performances when the weighing factor of the semantic loss is changed.

the performance of the model increases. In case of S-MTAE since it is a simpler method the performance goes down when λ is very high and optimal performance is observed when λ is 5. In case of S-FC and S-AGG the performance improves when the weighing factor increases from 5 to 10 also.

5.2 t-SNE Visualization of Semantic Space

We visualise the semantic space that is learned by the domain generalization methods which has been adapted to solve Zero-Shot Domain Generalization, using our proposed methodology. We use t-SNE [23] to project the latent space of the ZSDG methods to two dimension. Figure 2 shows the semantic space visualization of models trained on Fashion-MNIST, CIFAR10, CIFAR100 and PACS datasets. Both zero-shot and other classes are plotted in these figures. For fair analysis of different methods, we choose S-MTAE and S-FC for experiments with F-MNIST and PACS dataset, while semantic space of S-AGG is visualised of CIFAR10

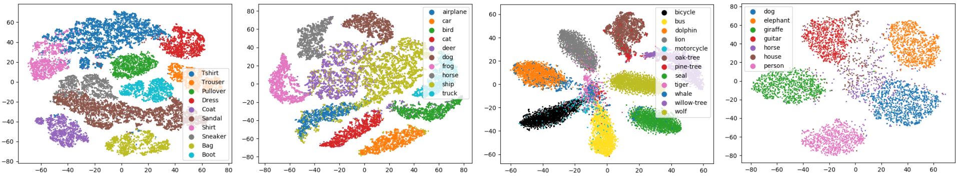

Figure 2: t-SNE plots of the latent space of ZSDG methods on Fashion-MNIST, CIFAR10, CIFAR100 and PACS datasets respectively (from left to right).

and CIFAR100 datasets.

The domains and the zero-shot classes that are selected are as follows: Plot 1: F-MNIST trained using S-MTAE; target domain: D_3; zero-shot classes: t-shirt, sandal. Plot 2: CIFAR10 trained using S-AGG; target domain: D_2; zero-shot classes: deer, ship. Plot 3: CIFAR100 trained using S-AGG; target domain: D_4; zero-shot classes: Setting 1. Though CIFAR100 contains 100 classes, we randomly sample three zero-shot classes and nine seen classes for the t-SNE plot, for easy visualization. Plot 4: PACS trained using S-FC; target domain: cartoon; zero-shot classes: horse, house.

From Figure 2, we observe that images from the zero shot classes are clustered around semantically meaningful classes. In the first plot, the zero-shot class *t-shirt* is closer to shirt, dress, while the other zero-shot class *sandal* is close to sneaker, boot. In the second plot, the zero-shot classes *ship* and *deer* is closer to (truck, airplane) and (dog, horse, frog) respectively. Similar observation is found for the other two datasets too. These results allow us to conclude that the semantic loss enforces alignment in the latent space. This enables graceful transition of DG methodologies to solve ZSDG task.

6 Conclusion

We introduce a novel Zero-Shot Domain Generalization (ZSDG) problem, where a model is expected to generalize to new classes in an unseen domain. This realistic problem setting is harder than Domain Generalization [8, 14, 22, 26], Domain Adaptation [7, 13, 27] and Zero-Shot Learning [4, 12, 29] as ZSDG models are expected to generalize over novel classes in novel domains, without access to corresponding training data-points. We find that learning semantically consistent domain-invariant features helps address this challenging problem. We adapt current state-of-the-art DG methods [14, 22] to this setting to reveal the efficacy of our proposed approach, as well as provide a baseline for further efforts on this problem.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions. This work was partly supported by DST, Govt of India, through the IMPRINT program.

References

[1] Zeynep Akata, Honglak Lee, and Bernt Schiele. Zero-shot learning with structured embeddings. *CoRR*, abs/1409.8403, 2014. URL http://arxiv.org/abs/1409.8403.

[2] Zeynep Akata, Honglak Lee, and Bernt Schiele. Zero-shot learning with structured embeddings. *CoRR*, abs/1409.8403, 2014. URL http://arxiv.org/abs/1409.8403.

[3] Zeynep Akata, Florent Perronnin, Zaïd Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *CoRR*, abs/1503.08677, 2015. URL http://arxiv.org/abs/1503.08677.

[4] Jimmy Ba, Kevin Swersky, Sanja Fidler, and Ruslan Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. *CoRR*, abs/1506.00511, 2015. URL http://arxiv.org/abs/1506.00511.

[5] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *NeurIPS*, pages 2178–2186. 2011.

[6] Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *arXiv preprint arXiv:1711.07910*, 2017.

[7] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[8] Aniket Anand Deshmukh, Srinagesh Sharma, James W Cutler, and Clayton Scott. Multiclass domain generalization. In *NIPS workshop on Limited Labelled Data*, 2017.

[9] Aniket Anand Deshmukh, Yunwen Lei, Srinagesh Sharma, Urun Dogan, James W Cutler, and Clayton Scott. A generalization error bound for multi-class domain generalization. *arXiv preprint arXiv:1905.10392*, 2019.

[10] Kun Duan, Devi Parikh, David Crandall, and Kristen Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, 2012.

[11] Vittorio Ferrari and Andrew Zisserman. Learning visual attributes. In *NeurIPS*, pages 433–440. 2008.

[12] 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 *NeurIPS*, pages 2121–2129. 2013.

[13] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[14] M. Ghifary, W. Bastiaan Kleijn, M. Zhang, and D. Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *ICCV*, pages 2551–2559, 2015.

[15] Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7): 1414–1430, 2016.

[16] Pichai Kankuekul, Aram Kawewong, Sirinart Tangruamsub, and Osamu Hasegawa. Online incremental attribute-based zero-shot learning.

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[18] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

[19] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, pages 5542–5550, 2017.

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

[21] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, pages 624–639, 2018.

[22] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In *ICML*, pages 3915–3924, 2019.

[23] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[25] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41, 1995.

[26] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, pages 5715–5725, 2017.

[27] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010.

[28] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

[29] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *NeurIPS*, pages 935–943, 2013.

[30] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *CoRR*, abs/1707.00600, 2017. URL http://arxiv.org/abs/1707.00600.

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.