Generalized Zero-shot Learning using Open Set Recognition

Omkar Gune
guneomkar@ee.iitb.ac.in
Indian Institute of Technology Bombay
Mumbai, India

Amit More
amitmore@ee.iitb.ac.in
Indian Institute of Technology Bombay
Mumbai, India

Biplab Banerjee
getbiplab@gmail.com
Indian Institute of Technology Bombay
Mumbai, India

Subhasis Chaudhuri
sc@ee.iitb.ac.in
Indian Institute of Technology Bombay
Mumbai, India

Abstract

Generalized Zero-shot Learning (GZSL) aims at identifying the test samples which can belong to previously *seen* (training) or *unseen* visual categories by leveraging the side information present in the form of class semantics. In general, GZSL is a difficult problem in comparison to the standard Zero-shot Learning (ZSL) given the model bias towards the seen classes. In this paper, we follow an intuitive approach to solve the GZSL problem by adhering ideas from the Open Set Recognition (OSR) literature. To this end, the proposed model acts as a pre-processing module for the GZSL inference stage which decides whether a given test sample belongs to seen or unseen class (domain). In order to comprehend the same, we generate *pseudo unseen visual* samples from the available seen data and further train a domain classifier for on-the-fly domain label assignment for the test samples. The domain specific inference modules are then applied subsequently for improved classification. We experiment on standard benchmark AWA1, APY, FLO, and CUB datasets which confirm superior performance over the existing state of the art.

1 Introduction

Zero-shot Learning (ZSL) [7, 21] for visual recognition is inspired by the human learning ability where the goal is to identify *unseen* object classes by using previously *seen* class data along with semantic information. In the simplest form, the ZSL model learns an embedding function which regresses the visual features of seen classes to their corresponding semantic features (also known as class prototypes, semantic embeddings or simply prototypes). During testing, the learned embedding function is then applied to visual features of unseen classes to obtain the respective semantic embeddings. These semantic embeddings are subsequently compared with the ground truth unseen class prototypes using established distance measures for label prediction. Generalized Zero-shot Learning (GZSL) [5] extends the ZSL paradigm by allowing the test samples to come from either seen or unseen classes while making prototypes of both the seen and unseen classes available during testing. Since the model is trained on the visual-semantic information specific to the seen classes, it is largely

found to be biased towards seen/training data. Hence, when a test sample comes from one of the unseen classes, it is highly probable that it gets misclassified to one of the seen classes.

Existing methods address this issue by preserving the class neighborhood structure of the seen classes in both the visual and semantic domains [2, 13, 18, 34] which results in the better alignment of the semantic embeddings of unseen visual samples to their respective class prototypes. As it is difficult to anticipate the model behaviour on unseen data by training the model on only seen class data, data augmentation techniques [8, 20, 41] have gained a lot of attention recently. These techniques make use of generative models to simulate visual samples of unseen classes using unseen class prototypes which leads to an alternative supervised formulation to the (G)ZSL problem. However, their superior performance can largely be attributed to the robustness of the additional supervised classifier modules than the actual data generation stage. This is because the generative models cannot capture the distributions of the unseen classes which were absent during training.

As discussed, the performance of GZSL degrades when the seen and unseen class data are jointly considered during inference. However, if the GZSL inference can be split into separate domain-specific inference settings to handle data potentially coming from the two domains, i.e., seen or unseen classes, an overall performance enhancement can be guaranteed. Nonetheless, it is non-trivial to determine the domain labels of the test samples on-the-fly without any prior information regarding the unseen classes. Recently the paradigm of Open Set Recognition (OSR) has received significant attention for highlighting the test samples which come from outside (open set) of the training classes. Hence, the idea of OSR can be adopted for domain classification in GZSL given the availability of *approximate* unseen class information prior to inference. To obtain the same, we propose to learn novel *pseudo-unseen class* (PUC) semantic prototypes from the available seen class prototypes while simultaneously generating visual samples of seen classes and *PUC*s using a generative model. We note that for every seen class there is corresponding PUC and all such PUCs together act as a representative of open set while seen classes correspond to the closed set. Further, we train a domain classifier (DC) to discriminate between the visual samples of seen classes and PUCs. The trained DC is further adopted to classify the test samples into seen and unseen classes during the GZSL inference. Since the PUC visual samples are constrained to lie within close vicinity of the corresponding seen class visual samples, it is expected that the domain classifier will be able to classify the seen from the actual test unseen classes. For further improving the performance of the domain classification, we propose to utilize the difference between top two scores of softmax activation of the learned DC, referred as the *confidence* of the DC, on the test samples and accordingly predict the class label of a test sample using prototypes of seen, unseen or seen *and* unseen classes. We summarize the major contributions as:

- We propose a DC-based pre-processing strategy which can decide on whether a test sample comes from the seen or unseen (domains) classes during GZSL inference.
- In order to train the DC, we propose a novel OSR driven approach to generate PUC visual samples from a set of learnable PUC prototypes which marginally differ from the corresponding seen class prototype. We address a much harder problem of generating many PUC visual samples using only a few seen class semantic prototypes by introducing novel constraints.
- We tackle the possible misclassification by the DC in the hard-label assignment by intelligently analyzing its confidence on the classification scores for the test samples.

- We experiment on standard benchmark datasets for GZSL and show superior performance on AWA1 [22], APY [7], Flower [27], and CUB [38] datasets.

2 Related Work

ZSL and GZSL: ZSL makes use of semantic side information in the form of class attributes, textual description or word embeddings [25]. Existing methods learn a high dimensional regression function by mapping from visual to semantic domain or vice versa [1, 9, 19, 31, 42, 43] while few methods [13, 39] seek to learn a shared latent embedding space to carry out the ZSL inference. Expressing the unseen classes in terms of seen classes has been studied in [4, 28, 37, 43, 44]. Likewise, end-to-end deep models have recently been explored in this regard to account for the data-driven feature learning in ZSL [29, 42]. On the other hand, a detailed study on different experimental settings for GZSL is given in [5, 40]. To tackle the aforesaid model bias towards the seen classes, [8, 20, 41] use data augmentation techniques using generative models where unseen class samples are first generated and subsequently GZSL problem is dealt with under the fully-supervised setting. In contrast, a few non-data augmentation techniques involve calibration of the deep networks based on the confidence of seen classes and uncertainty of target unseen classes [24], preserving the class neighborhood structures both in the visual and semantic domains [2, 13, 18, 34] and use of a meta-learning strategy [36].

Open Set Recognition (OSR): OSR [32] problem has been introduced to overcome the limitations of closed set learning and to cater to realistic scenarios. In this regard, [17, 33] inherit concepts from the Extreme Value Theory to predict the probability of a given sample coming from previously known or unknown class distributions. Deep networks are extended to learn in an open set framework by [3]. Adversarial data augmentation strategy is used in [10] to extend the OpenMax [3] model to identify the samples from the unknown classes. From a different point of view, detection of the out-of-distributions (OOD) examples [15] partially resembles the OSR paradigm but differs in the design of the experimental protocols.

Comparison with existing methods: As already stated, we take inspirations from the problem of OSR in general [3, 10, 15, 23, 26] but we utilize the visual and semantic information together for generating the representatives of open set data unlike visual data based OSR. In addition, i) different from [26], we imperatively address the much harder problem of generating many visual samples of PUCs using the corresponding seen class semantic prototypes (one prototype per seen class), ii) unlike [26], our proposed pre-processing model is end-to-end trainable: the open set data generation and domain classification are jointly trained which, we show experimentally, is superior to using the pre-trained model for open set data generation, and iii) we address the problem of OSR in the GZSL framework where [23] points out that identifying the seen and unseen classes from the same dataset (detection on the same manifold) is a difficult problem to address which is judiciously handled in our case.

Specific to the GZSL problem, [35] uses the hard assignment of a test sample to seen or unseen classes under the notion of *outlier detection*. However, the hard classification may be erroneous for fine-grained seen-unseen classes. We overcome this problem by considering the confidence of DC based on its softmax scores on the test samples. In particular, we make use of temperature scaling followed by a simple softmax score thresholding [16, 23] to demarcate the seen and (pseudo)-unseen class samples effectively. Similarly, we differ from data augmentation based GZSL methods [20, 41] as we carry out inference within the setting of regressor driven visual-semantic mapping in place of the sophisticated classification

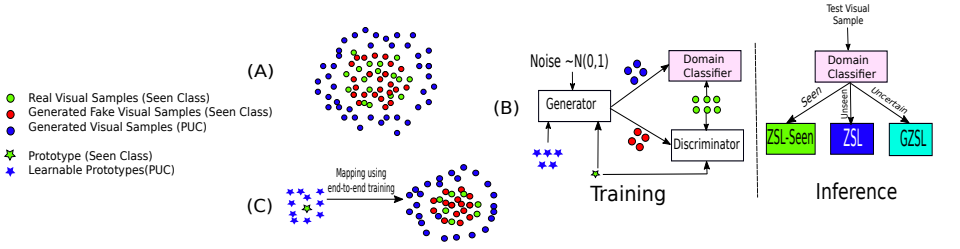

Figure 1: (A) Illustrative figure showing the visual samples of PUC form a boundary around the corresponding seen class. (B) The proposed model is shown here. Training stage uses an adversarial framework between Generator and Discriminator to generate visual samples of PUC using learnable PUC prototypes. The model also use co-operative training between Generator and Domain Classifier to regularise the mapping of PUCs to visual space. The inference stage uses the learned DC as a pre-processing step to label the domain of a test sample. Based on the domain label, as seen, unseen or uncertain, the class label is inferred using ZSL-seen, ZSL or GZSL framework, respectively. (C) Illustrative figure showing the semantic to visual mapping learned by an end-to-end model after training.

mechanism adopted by them solely in the visual domain.

3 The Problem of GZSL

Consider a dataset $\mathcal{D}$ comprising of N_s seen and N_u unseen visual classes such that the total number of classes in $\mathcal{D}$ is $N = N_s + N_u$. Let $\mathcal{D}_s = \{\mathbf{x}_i, y_i\}_{i=1}^{n_s}$ be the training/seen data of n_s samples with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathcal{Y}_s = \{1, 2, \ldots, N_s\}$ representing the visual feature and label corresponding to the i^{th} sample, respectively. During training, semantic class prototypes, $\mathcal{A}_s = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{N_s}\}$, are accessible for each of the seen classes where $\mathbf{a}_i \in \mathbb{R}^k$. Let $\mathcal{D}_s = \mathcal{D}_{tr} \bigcup \mathcal{D}'_{tr}$ such that $\mathcal{D}_{tr}$ and $\mathcal{D}'_{tr}$ denote the training data and the held-out training data sampled from the seen classes, respectively. Let $\mathcal{D}_u = \{\mathbf{x}_j, y_j\}_{j=1}^{n_u}$ be the unseen data with n_u samples such that label $y_j \in \mathcal{Y}_u = \{N_s + 1, N_s + 2, \ldots, N_s + N_u\}$. In ZSL, $\mathcal{Y}_s \cap \mathcal{Y}_u = \phi$. Finally, let $A_u = \{\mathbf{a}_{N_s+1}, \mathbf{a}_{N_s+2}, \ldots, \mathbf{a}_{N_s+N_u}\}$ denote the set of unseen class prototypes which are made available only during testing. Under this setup, the goal of ZSL is to learn embedding functions $f_v(\cdot)$ and $f_s(\cdot)$ using $\mathcal{D}_{tr}$ and $\mathcal{A}_s$ in order to model a measure of compatibility $\mathcal{F}(f_v(\mathbf{x}), f_s(\mathbf{a}))$ which produces high value if $\mathbf{x}$ and $\mathbf{a}$ share the same class label and a low value instead. During inference of ZSL, the label for a test sample $\mathbf{x}_t \in \mathcal{D}_u$ is obtained as

$$\hat{y} = \arg\max_{y \in \mathcal{Y}_u} \mathcal{F}(f_v(\mathbf{x}_t), f_s(\mathbf{a}_y)). \tag{1}$$

GZSL considers that for the test sample $\mathbf{x}_t \in \mathcal{D}'_{tr} \bigcup \mathcal{D}_u$, $y \in \mathcal{Y}_s \bigcup \mathcal{Y}_u$. We also define the supervised learning setup in the framework of GZSL as *ZSL-Seen* where $\mathbf{x}_t \in \mathcal{D}'_{tr}, y \in \mathcal{Y}_s$.

4 GZSL using OSR

We briefly explain the motivation behind the proposed model. Our goal is to carry out the harder GZSL inference in terms of rather easy to handle ZSL and ZSL-Seen problems.

This can be made possible if some prior information regarding the domain (seen or unseen) of a test sample is available. One straight-forward way to obtain this prior information is through deploying a DC capable of predicting the domain of a test sample. To train the DC, one can annotate the visual data coming from seen classes as depicting the closed set while visual data coming from unseen classes as forming the open set within a typical OSR framework. However, in the case of GZSL, the main difficulty in carrying out such training is due to the unavailability of the unseen class data (both visual and semantic) during the training stage. To overcome this difficulty, we generate the visual samples of *PUC*s (using the prototypes of corresponding PUCs) which are considered to be the representatives of an open set. We refer to the PUC corresponding to a given seen class as the one which *moderately resembles the seen class data yet differs semantically*. In this regard, we propose to generate the PUC visual samples in such a way that they lie within a certain territory outside the support of the seen class in the visual feature space. For every seen class there is corresponding PUC and all such PUCs together act as a representative of an open set. We note that PUC visual-semantic data is generated solely using the available visual-semantic seen data. To this end, we use adversarial training strategy to generate the visual samples of both seen classes and PUCs. Visual samples of the seen classes are generated using given seen prototypes (one prototype per seen class) whereas visual samples of PUCs are generated using their *learnable* prototypes (many prototypes per PUC). A single prototype for a seen class is useful in forming the compact cluster in the visual space. However, it may be difficult to generate the visual samples of PUC that lie around the seen class, using only one PUC prototype. Hence, we learn many prototypes for each PUC. Moreover, one might think the necessity of generating visual samples of seen classes as they are already available. We emphasize that as we adversarially generate visual samples for each seen class, it helps in formulating the dense cluster for the corresponding seen class and hence, in turn, assists in defining the discriminative boundary around that class. This further helps in generation of visual samples of corresponding PUC. (Figure 1(A)).

We train the aforementioned DC along with Generative Adversarial Network (GAN) [11] to generate visual samples of seen classes and PUCs in an end-to-end manner. The proposed model is shown in Figure1(B) where during the inference stage, the trained DC is used to infer the domain label of a test sample as seen, unseen or uncertain domain. The uncertain domain is introduced to avoid any miss-classification due to the hard division of domains into seen and unseen. In case of the uncertain domain of a test sample, its class label is obtained using the standard GZSL paradigm. On the contrary, when DC predicts the domain as seen or unseen, the corresponding class label can then be inferred using the ZSL-Seen or ZSL problem, respectively. In the following, we explain the process of generating visual samples corresponding to PUC. The OSR based domain classification is henceforth dealt with.

4.1 Generating PUC Samples

We make use of GAN to generate the samples of PUCs. GANs have been successful in ZSL for generating visual samples from the corresponding attribute vectors by extending an adversarial game between a feature generator (G-Net) and a discriminator (D-Net) networks. In our experiments, we rely on a conditional version of WGAN [12] (CWGAN) which optimizes the following objective,

$$\min_{G}\max_{D}\mathcal{L}_{WGAN} = \mathbb{E}\left[D(\mathbf{x},\mathbf{a})\right] - \mathbb{E}\left[D(G(\mathbf{z},\mathbf{a}),\mathbf{a})\right] - \lambda GPT, \tag{2}$$

where GPT is the gradient penalty term proposed in [12], $\mathbf{a}$ is a semantic prototype of a class to which visual feature $\mathbf{x}$ belongs, λ is a hyper-parameter, $G(\cdot)$ and $D(\cdot)$ represent outputs of G-Net and D-Net, respectively, $\mathbf{z}$ is a Gaussian noise, and $\mathbb{E}[\cdot]$ is an expectation operation.

The model referring to Eq.2 learns to generate the visual samples belonging to the seen classes only. However, we are interested in generating visual samples of PUCs. For achieving the same, we initially learn PUC prototype corresponding to a given seen class prototype. We note from the (G)ZSL literature that the unseen classes can be represented in terms of seen classes since the underlying semantic space is shared between them [37, 43]. We further extend this idea to generate PUC prototypes by exploiting the seen class prototypes and minimizing the following objective function,

$$\min_{\mathbf{w}_{lc}} \mathcal{L}_{PSU} = \max\left[\alpha - \|\mathbf{a}_i - \mathbf{A}_s\mathbf{w}_{lc}\|_2^2, 0\right] + \max\left[\|\mathbf{a}_i - \mathbf{A}_s\mathbf{w}_{lc}\|_2^2 - \beta, 0\right], \tag{3}$$

such that $\mathbf{A}_s = [\mathbf{a}_1\mathbf{a}_2\ldots\mathbf{a}_{Ns}] \in \mathbb{R}^{k\times N_s}$ and $\mathbf{a}_i^* = \mathbf{A}_s\mathbf{w}_{lc}$ is a PUC prototype corresponding to the prototype of seen class i. Here, $\mathbf{w}_{lc} \in \mathbb{R}^{N_s}$ is a *learnable* set of coefficients used to obtain the linear combination of seen prototypes, α and β are hyper-parameters which govern the region where the PUC prototype $\mathbf{a}_i^*$ can lie. We denote the loss incurred in optimizing above objective function by $\mathcal{L}_{PSU}$. The above margin loss ensures that the PUC prototype is close to its corresponding seen class prototype but does not collapse onto it. We generate multiple such prototypes for a PUC corresponding to a seen class. During training, these PUC prototypes are then passed through the G-Net of the CWGAN to generate visual samples of corresponding PUC. Note that, D-Net is trained on real and fake visual samples of seen classes by conditioning on only seen class prototypes. The process of generating PUC prototypes can in principle be made offline. However, we incorporate it into our end-to-end model by following the alternate optimization strategy (details in supplementary material).

4.2 Cooperative Training with DC

The real visual samples of seen classes and generated visual samples of corresponding PUCs are subsequently used to train the DC which essentially learns to separate the seen classes from PUCs. We also train the G-Net, here, to cooperate and generate PUC samples to help the DC. The loss incurred during this cooperative training regularizes the mapping from semantic to visual space in such a way that PUC visual samples lie outside the boundary of corresponding seen class. This training strategy is effective than simply training G-Net (trained on seen class visual-semantic data) to generate the visual samples of PUCs in adversarial setting with D-Net. In general, it is possible that for some of the prototypes of PUC the corresponding visual samples may fall inside the class boundary. This can be expected because by design PUC prototypes are close to corresponding seen class prototype. However, when G-Net is trained along with DC, such a situation will be penalized hence guiding the generator to generate the PUC visual samples that lie outside the corresponding seen class boundary (Figure1(C)). We validate this fact experimentally where we get better performance by incorporating the cooperative training for G-Net than simply using adversarial. Specifically, we train the $N_s + 1$-class multi-class DC where N_s seen classes form the closed set and an extra class is used to represent an open set of all the PUCs. Ideally, one can use a binary classifier to separate seen and (pseudo)unseen classes. But we experimentally found that multi-class classifier performs better. We use $\mathcal{L}_{DOM}$ to denote the standard multi-class cross-entropy loss incurred during the training of the DC module having parameters W_{DOM}.

Overall Objective: The overall objective function to be optimized in an end-to-end setup is

$$\min_{G}\max_{D}\mathcal{L}_{WGAN}+\min_{\mathbf{w}_{lc}}\mathcal{L}_{PSU}+\min_{W_{DOM}}\mathcal{L}_{DOM}. \tag{4}$$

4.3 Inference in GZSL

Once trained, we use the DC as a pre-processing step during GZSL inference. We first pass a test sample to the DC and observe the scores of N_s+1 classes. These scores act as a prior belief used to decide the domain label for the test sample. If the score of one of the first N_s nodes is highest, then the test sample belongs to one of the seen classes, and its label is inferred using the ZSL-Seen settings. If the score of the open set class $(N_s+1)^{th}$ is the maximum, then it is declared that the test sample belongs to one of the unseen classes and its label is inferred using the prototypes of only unseen classes. Note that this is the classical setup which may propagate error if DC fails to classify the seen/unseen samples precisely.

To overcome this difficulty regarding the hard division of GZSL problem into ZSL-Seen and ZSL problems, we make use of relative softmax score of DC along with its predicted label. We experimentally observe that top two scores at the output of softmax classifier provide important information about the confidence of DC in deciding the domain label. Let $\mathcal{S}_{top2}$ denote the difference between highest and second highest softmax score of DC for a given test sample. We refer to $\mathcal{S}_{top2}$ as the *confidence* of the DC. Let S_D and U_D denote that DC has labeled a test sample as belonging to a seen and unseen domain, respectively using the above mentioned classical setup. We use the following decision rule to infer the class label $\hat{y}$ of a test sample, using ZSL-Seen, ZSL or GZSL framework,

$$\hat{y}\in\begin{cases}\mathcal{Y}_s & \text{if } \mathcal{S}_{top2}>\gamma \text{ and } \mathbb{1}\{S_D\}\\ \mathcal{Y}_u & \text{if } \mathcal{S}_{top2}>\gamma \text{ and } \mathbb{1}\{U_D\}\\ \mathcal{Y}_s\bigcup\mathcal{Y}_u & \text{if } \mathcal{S}_{top2}\leq\gamma,\end{cases} \tag{5}$$

where γ is the threshold which is fixed using validation. In case of the first condition in Eq.5, although the DC can infer the class label directly, we stick to the ZSL-Seen framework to make the inference using 1-Nearest Neighbour (1-NN) criterion which is consistent with the existing (G)ZSL literature.

5 Experiments

Datasets and Features: We use the standard benchmark datasets AWA1[22], CUB [38], FLO [27], and APY[7] to test the efficacy of our model. Animals With Attributes (AWA1) is a coarse-grained dataset containing 40 seen and 10 unseen classes of animals along with 85-dim attribute vector for each class and total 30,475 images. Attribute Pascal and Yahoo (APY) dataset contains 12,051 images from 32 classes with 20 seen and 12 unseen with 64-dim attributes per class. The fine-grained Caltech University Birds (CUB) dataset has 200 classes divided into 150 seen and 50 unseen categories, a total of 11,788 images and 312-dim attribute vector per class. The other fine-grained Flowers (FLO) dataset consists of a total of 8189 images of 102 different flower classes with 82 seen and 20 unseen classes. We use 2048-dim 101-ResNet [14] features as visual embedding while the experimental and evaluation protocols of [40] is thoroughly considered. We use only attributes as semantic features for AWA1, CUB, and APY. *We note that we do not use any sentence description for*

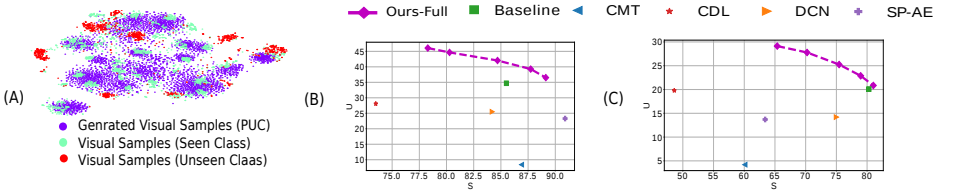

Figure 2: (A) AWA1: t-SNE plot of real visual samples of seen classes (green), unseen classes (red), and generated visual samples of PUCs (violate). One may now observe how PUCs surround seen classes separating them from unseen classes. (B), (C) AWA1, APY: The effect of threshold γ on seen (S) and unseen (U) accuracy.

CUB [38] but use attributes for FLO given by [20] using [30].

Model Architecture: We use the CWGAN [12] for adversarially generating visual samples of seen classes and PUCs. Our G-Net and D-Net model architecture is similar to [41]. For DC we use fully connected neural network based softmax classifier which takes d-dimension input visual features (of seen classes and PUCs) and gives $N_s + 1$ dimension output softmax score. Due to space constraint, we give the details of model architecture in the supplementary material. We use trained DC along with the GZSL model in [13] to evaluate the effect of domain classification. We use GZSL model in [13] as the *Baseline* model as it is a shallow, simple, easy to train and delivers a good performance. We use *Ours-PRT* to denote the model which uses pre-trained G-Net (pretrained on only seen data with adversarial setup but without cooperative training) and decision rule in Eq.5. We use *Ours-Base* to denote the model trained using both adversarial and cooperative training (section 4.2) which infers the labels of test samples by hard division of GZSL into ZSL-Seen and ZSL using the DC. We use *Ours-Full* to denote the model trained similarly but infers the class labels of the test samples using Eq.5.

Fixing α and β: Parameters α and β in Eq.3 define the region where PUC prototypes should lie which in turn play an important role in generating visual samples of PUC. A smaller value of α would make PUC prototypes semantically very similar to the corresponding seen prototypes ($\alpha = 0$ would collapse PUC prototypes onto the corresponding seen class prototype). Further, it is important that β is not too large as it may result in D-Net to win over G-Net. We fix the value of β to the half of the minimum distance between any two seen prototypes and $\alpha = 0.5\beta$. Figure 2 (A) clearly shows that visual samples of PUCs form the boundary around the visual samples of corresponding seen classes thus helping the DC to discriminate between seen and unseen class samples during the GZSL inference stage.

Training and Stopping Criterion: Our proposed DC is trained in an end-to-end manner along with CWGAN using the standard stochastic gradient descent. We alternate between adversarial and cooperative training iterations. We use $\mathcal{D}_{tr}$ and generated visual samples of PUCs to train the DC. We make use of Wasserstein distance as the stopping criterion for training as validation data for PUCs is not available. We provide more details about the training regime in the supplementary material.

Results: We compare our results with the many state-of-the-art methods. We note that although the data augmentation techniques in [20, 41] use generative models, they also resort to the training of additional classifiers and supervised classification in the visual domain. Hence, their methods are not directly comparable with ours and other existing methods which

Method	AWA1			FLO			APY			CUB		
	S	U	H	S	U	H	S	U	H	S	U	H
ALE [39]	76.1	16.8	27.5	61.6	13.3	21.9	73.7	4.6	8.7	62.8	23.7	34.4
DZSL [42]	84.7	32.8	47.3	-	-	-	75.1	11.1	19.4	57.9	19.6	29.2
PSR [2]	-	-	-	-	-	-	51.4	13.5	21.4	54.3	24.6	33.9
SP-AE [8]	90.9	23.3	37.1	-	-	-	63.4	13.7	22.6	70.6	34.7	46.6
RN [36]	-	-	-	-	-	-	-	-	-	61.1	38.1	47.0
CDL [18]	73.5	28.1	40.6	-	-	-	48.6	19.8	28.1	55.2	23.5	32.9
⋆ FGN [41]	57.2	47.6	52.0	60.3	54.3	57.1	-	-	-	59.3	40.2	†47.9
• FGN [41]	61.4	57.9	59.6	73.8	59.0	65.6	*49.3	*29.6	*37.0	57.7	43.7	†49.7
• SEGZSL [20]	67.8	56.3	61.5	-	-	-	-	-	-	53.3	41.5	46.7
CMT [35]	86.9	8.4	15.3	-	-	-	60.1	4.2	8.7	74.2	10.9	19.0
DCN [24]	84.2	25.5	39.1	-	-	-	75.0	14.2	23.9	60.7	28.4	38.7
Baseline [13]	85.5	34.7	49.4	*91.2	*30.3	*45.5	*80.3	*20.1	*32.2	60.7	30.2	40.3
Ours-PRT	73.7	43.3	54.6	89.2	31.7	46.7	51.3	27.5	35.8	47.2	39.7	43.1
Ours-Base	74.9	46.6	57.5	77.7	43.8	56.0	60.9	29.9	40.1	34.7	44.9	39.1
Ours-Full	78.3	46.1	58.0	79.9	43.8	56.6	65.4	29.1	40.3	45.4	43.5	44.4

Table 1: Per class average accuracy comparison of GZSL on different datasets. Accuracy on Seen (S) and Unseen (U) classes, H: Harmonic mean. * indicates our implementation where original results are not available. † uses sentence descriptions along with attributes. • use additional supervised classifiers, hence comparing with them may not be fair. ⋆ is the best model of [41] in the 1-NN setting. Numbers in Red: the best performance. Numbers in blue: Second best performance.

use 1-NN during the inference in (G)ZSL literature. As can be observed from Table 1, we improve by 8.6 (AWA1), 11.1(FLO), 8.1 (APY), and 4.1 (CUB) over the baseline GZSL model in [13] which shows that pre-processing using DC is very effective. We note that our method helps in reducing the bias towards the seen classes as can be seen by improved performance on unseen test samples. We also note that rectification of possible miss-classification by DC results in improved performance of *Ours-Full* model over *Ours-Base* model. Lastly, we greatly improve by a huge absolute margin of 18.9 (AWA1), 16.4(APY), and 5.7(CUB) over recent model [24]. Except for CUB, in the 1-NN setting of testing, we outperform both data-augmentation techniques as well as non-data-augmentation techniques in case of AWA1 and APY while delivering the comparable results on FLO. It can be observed that *Ours-Full* outperforms *Ours-PRT* by 3.4 (AWA1), 9.9 (FLO), 4.5 (APY), and 1.3 (CUB). This shows that cooperative training using DC is effective in learning better semantic to visual mapping.

Ablation Study: We also experiment to analyze the effect of threshold γ on the accuracy of seen (S) and unseen (U) classes on two datasets as shown in Figure2 (B)(C). The figure plots the accuracy for S and U for different values of γ for proposed method along with some of the state-of-the art methods. For better performance, both S and U should be high, *i.e.* a point must lie in the top right corner of the figure. We observe that the proposed method consistently gives better performance for different values of γ in comparison to the other methods. One can further use this analysis to tune the parameter γ as per the requirements.

6 Conclusions

In this paper, we have proposed a pre-processing module in the form of DC which can be used in any standard GZSL framework during the inference step. The goal of the DC is to

separate the seen and unseen classes (domains) which is formulated as an OSR problem. The DC is trained on the visual samples of seen classes and generated visual samples of PUCs. The visual samples of PUCs are generated using adversarial training. We experimentally showed that the DC output which acts as a prior belief about the domain (seen or unseen) of a test sample does help in boosting the overall performance. To avoid performance degradation due to the possible miss-classification by DC, we use the DC confidence along with its decision. We validate our proposed framework and achieve impressive performance on the benchmark GZSL datasets. As a future direction, we are interested in imposing constraints on PUCs in visual space rather than in the semantic space.

Acknowledgement

We acknowledge the support provided for this work by Bharti Centre for Communication in Indian Institute of Technology (IIT) Bombay, India.

References

[1] Zeynep Akata, Mateusz Malinowski, Mario Fritz, and Bernt Schiele. Multi-cue zero-shot learning with strong supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 59–68, 2016.

[2] Yashas Annadani and Soma Biswas. Preserving semantic relations for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7603–7612, 2018.

[3] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016.

[4] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5327–5336, 2016.

[5] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *European Conference on Computer Vision*, pages 52–68. Springer, 2016.

[6] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1043–1052, 2018.

[7] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009.

[8] 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 (ECCV)*, pages 21–37, 2018.

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

[10] ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. In *BMVC*, 2017.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf.

[13] Omkar Gune, Biplab Banerjee, and Subhasis Chaudhuri. Structure aligning discriminative latent embedding for zero-shot learning. In *BMVC*, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[17] Lalit P Jain, Walter J Scheirer, and Terrance E Boult. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision*, pages 393–409. Springer, 2014.

[18] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Learning class prototypes via structure alignment for zero-shot recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 118–134, 2018.

[19] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. *arXiv preprint arXiv:1704.08345*, 2017.

[20] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4281–4289, 2018.

[21] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 951–958. IEEE, 2009.

[22] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.

[23] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *European Conference on Computer VisionICLR*, 2017.

[24] Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Generalized zero-shot learning with deep calibration network. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2005–2015. Curran Associates, Inc., 2018.

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[26] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[27] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[28] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

[29] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016.

[30] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016.

[31] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161, 2015.

[32] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772, 2013.

[33] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11): 2317–2324, 2014.

[34] Yi-Ren Yeh Shih-Yen Tao, Yao-Hung Hubert Tsai and Yu-Chiang Frank Wang. Semantics-preserving locality embedding for zero-shot learning. In *BMVC*, 2017.

[35] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.

[36] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[37] Donghui Wang, Yanan Li, Yuetan Lin, and Yueting Zhuang. Relational knowledge transfer for zero-shot learning. In *AAAI*, volume 2, page 7, 2016.

[38] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

[39] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 69–77, 2016.

[40] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *arXiv preprint arXiv:1707.00600*, 2017.

[41] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551, 2018.

[42] Li Zhang, Tao Xiang, Shaogang Gong, et al. Learning a deep embedding model for zero-shot learning. 2017.

[43] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4166–4174, 2015.

[44] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via joint latent similarity embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6034–6042, 2016.