Generalized Zero-shot Learning using Open Set Recognition

Supplementary Material

BMVC 2019 Paper Id:35

1 Introduction

In this supplementary material, we provide the details about model architecture, training regime, co-operative training using DC and evolution of DC during the training.

We use the same notations in this supplementary material as the ones used in the main paper.

2 Model Architecture

We give the details about our model in this section.
We use similar architecture as used in [1] except for the classifier which is trained on the generated features. The motivation for the classifier trained on the generated features is not completely clear as the D-net (critic in [1]) has the same goal. Hence, we do not use the classifier.

We use multilayer perceptron (MLP) based neural networks for G-Net, D-Net, and the DC. Our G-Net and D-net are conditional WGANs and conditioned on the class semantic prototypes. G-net consists of total three layers input, hidden and an output layer with following dimensions and activation functions

- Input layer: $2k$ where k is the attribute dimension specific to the dataset
- Hidden layer: 4096, Leaky Relu
- Output layer: d (visual feature dimension)

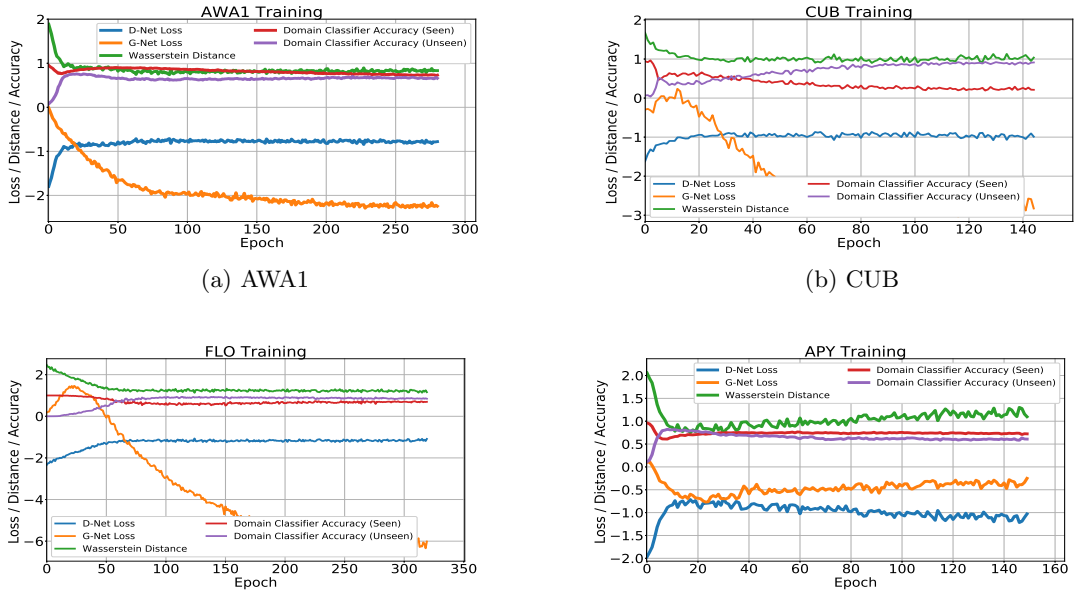
D-net consists of total three layers input, hidden and output layer with following dimensions and activation functions

- Input layer: $d + k$ where k is the attribute dimension specific to the dataset and d is the visual feature dimension
- Hidden layer: 4096, Leaky Relu
- Output layer: 1

The DC has a total of four layers one input, two hidden and one output layer with dimension and activation function as follows.

- Input layer: d where d is the visual feature dimension
- Hidden layer 1: 1000, Relu
- Hidden layer 2: 500, Relu
- Output layer: $N_s + 1$ where N_s is the number of sen classes of the dataset, softmax

We use normalised attributes for all the datasets.



(a) AWA1 (b) CUB

(c) FLO (d) APY

Fig. 1. The training process for different datasets. It can be observed that DC performance on seen and unseen test data saturates when the Wasserstein distance gets saturated. We use Wasserstein distance as the stopping criterion for the training. Note that, DC is not trained on the test seen and unseen data. Its performance is shown only for the visualization purpose. (Best viewed in colors.)

3 Training Regime and Stopping Criterion

Our model is trained in an end-to-end manner in the sense that we train conditional WGAN and the proposed DC in an alternating fashion. One iteration of the training includes the following steps.

- *Training the D-Net:* D-Net is trained on the real and fake visual samples of seen classes both conditioned on their corresponding semantic seen class prototypes.
- *Training the G-Net:* G-Net is trained to generate real-looking visual samples of seen classes conditioned on their corresponding semantic prototypes.
- *Generating pseudo-unseen prototypes:* PUC prototypes are generated by optimizing over the loss function $\mathcal{L}_{PSU}$ given by Eq.(3) in the main paper.
- *Training the DC:* Using the PUC prototypes learned in the previous step, corresponding PUC visual samples are generated using G-Net. These generated PUC visual samples and real visual samples of seen classes are fed to the DC for the training. The loss incurred during the training of the DC is also used to train the G-Net.

We note that unlike the OOD problem where validation data for OOD samples is available during the model training, GZSL setting does not allow to use unseen class data during training. This poses a difficulty in the model training and early stopping. However, we experimentally found that the Wasserstein distance

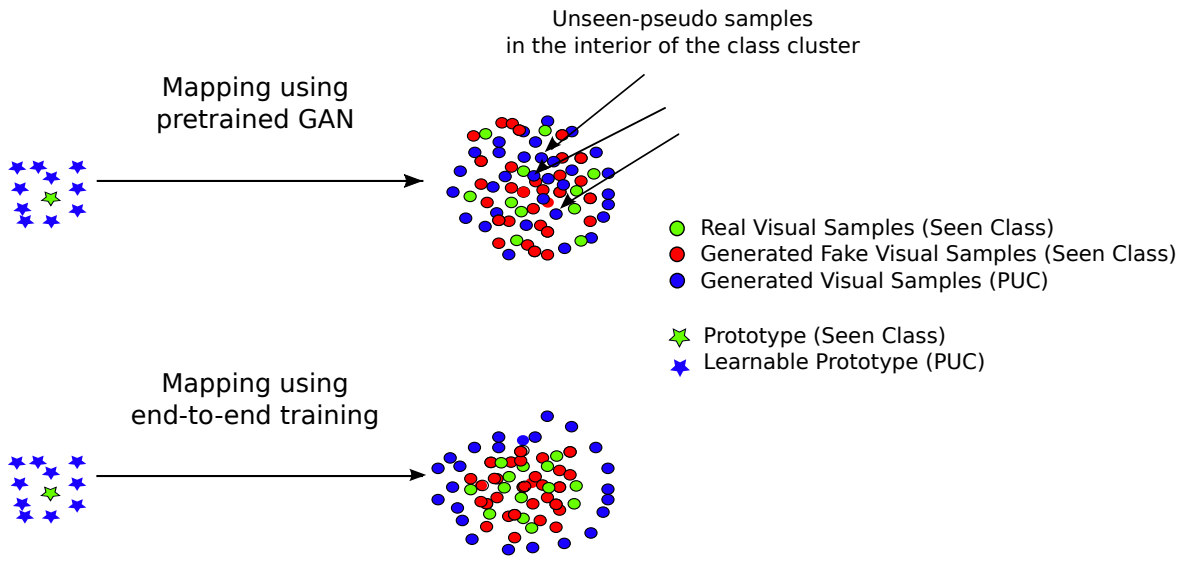


Fig. 2. Illustrative figure showing the effect of co-operative training using DC. With pretrained WGAN it may be possible that for some PUC prototypes corresponding visual samples reside in the interior of the cluster formed by seen class visual samples. Training WGAN in an end-to-end manner allows the DC to help G-Net in learning a better mapping from semantic to visual space. (Best viewed in colors.)

turns out to be the good stopping criterion for our end-to-end model training. This is shown in Figure 1 for different datasets. It can be observed that when Wasserstein distance gets saturated, the DC performance on the seen and unseen class test data also saturates. **Note that during the training of the DC the seen and unseen class test data is not used. The plots in Figure 1 for DC seen and unseen accuracy are only for visualization purpose. These are obtained by saving the DC model from the start (epoch 0) till the Wasserstein distance saturates and evaluating the DC over these epochs on the test seen and unseen class data.**

4 Effect of Co-operative Training using DC

One may ask the question of training the conditional WGAN along with the DC, as it is possible to pretrain the conditional WGAN. We experimented with both the settings, i.e., using pretrained WGAN and end-to-end training. The end-to-end training is effective as can be observed from Table 1 in the main paper. For clarity of explanation, results of Baseline [2], Ours-PRT, Ours-Base, and Ours-Full are re-scripted below. It can be observed that end-to-end training (Ours-Full) outperforms the pretrained scenario (Ours-PRT) by 3.4 (AWA1), 9.9 (FLO), 4.5 (APY), and 1.3 (CUB). We hypothesize that this improvement is due to the DC helping the G-Net in learning a robust mapping from semantic to visual space. If pre-trained WGAN is used (trained on the seen class data), it may be possible that for some unseen-pseudo class prototypes the corresponding visual samples are in the interior of the cluster formed by visual samples of the seen class. *This is possible because by design PUC prototypes are close to the corresponding seen class prototypes and*

Method	AWA1			FLO			APY			CUB		
	S	U	H	S	U	H	S	U	H	S	U	H
Baseline [2]	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	51.32	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. Performance comparison of GZSL on different datasets. S: Accuracy on seen classes, U: Accuracy on unseen classes, H: Harmonic mean. * indicates our implementation where original results are not available. Ours-Full (cooperative/end-to-end training) outperforms the Ours-PRT (CWGAN pretrained on only seen data.)

hence their corresponding visual samples may overlap. However, such a situation is less likely to be possible in case of end-to-end training where the DC guides the G-Net to generate PUC visual samples that form the boundary around corresponding the seen class. Any PUC visual sample falling in the interior would subsequently be penalized by the DC. The resultant mapping learned using this co-operative training can be illustrated as shown in Figure 2.

5 The Capacity of the Baseline Model

It is worth reiterating from the experiments that prior knowledge about the domain (seen or unseen) of a test sample helps in boosting the GZSL performance. To this end, we comment on how much maximum improvement one can get using the given GZSL model. If the DC is assumed to be ideal/perfect with 100% accuracy on test seen and unseen data, then the maximum performance from the GZSL model can be obtained. To know the capacity (the maximum performance of the GZSL model), we assume that the domain labels for test seen and unseen data are given by the *oracle*. Based on this, a test sample is labeled using either ZSL-Seen or ZSL setting. Using this, we evaluate the GZSL model and term the performance of the model as the *Model Capacity*. We compare the model capacity with our results. This is shown in Figure 3. It can be observed that there is still a considerable scope of improving the DC and the simple baseline model in [2] can give excellent performance.

6 The Effect of γ on the Seen and Unseen accuracy

For better harmonic mean H, it is necessary that GZSL model performs equally well on the test seen and unseen data. The inherent model bias often makes the accuracy on the test seen data (S) to be better as compared to the accuracy on the test unseen data (U). One way to evaluate the overall performance is to compare seen and unseen accuracy for different values of model parameters and then choose that value of model parameter which suits the user requirement. For this, we plot the seen-unseen accuracy by varying the γ as shown in Figure 4. One would like to have the point in this plot to lie on the top right corner. This depicts that corresponding model performs equally well on the test seen and unseen. From Figure 4 we observe that, for CUB our model performs equally well on test seen and unseen data data, however there is scope for overall improvement in the model performance. On the other hand, in case of FLO we perform better as compared to other existing methods. This is because for different values of γ our model lies at the top and right of the existing models in the plot.

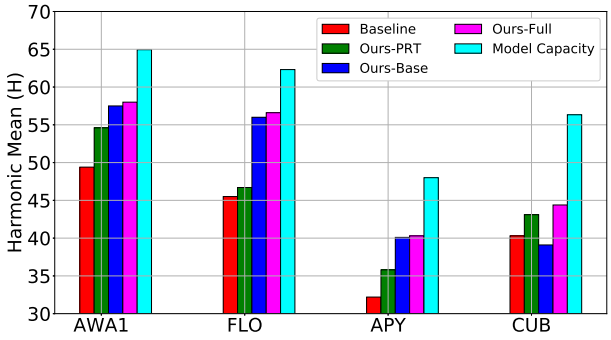


Fig. 3. Model Capacity Vs. performance of our models. It can be observed that there is still a large scope for the improvement in the DC. (Best viewed in colors.)

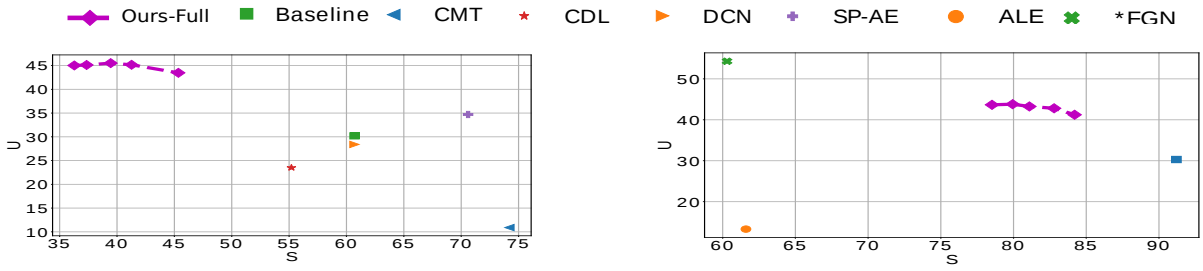
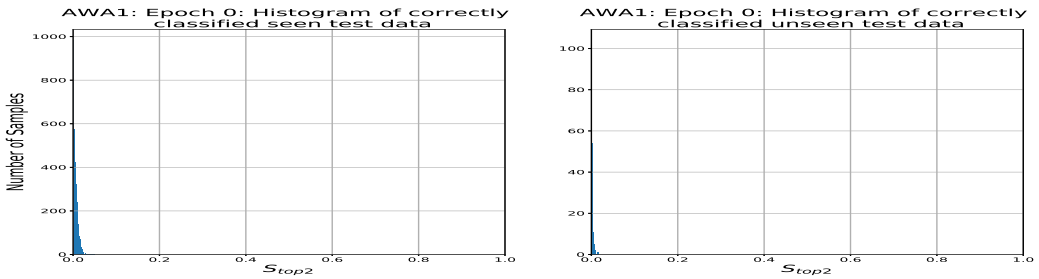


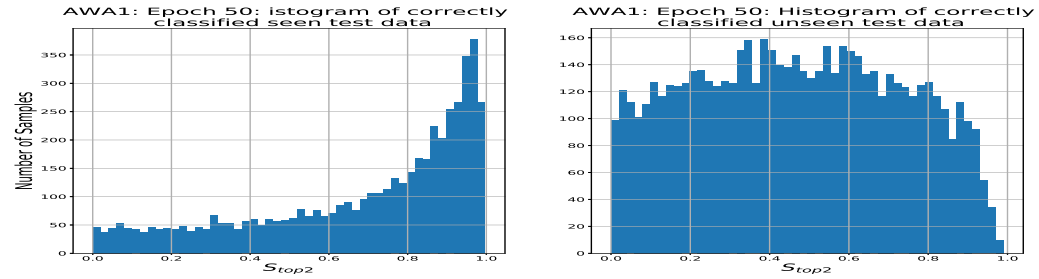
Fig. 4. The effect of γ on seen (S) and unseen (U) accuracy. The proposed model is compared with the different existing models. (Best viewed in colors.)

7 Evolution of the DC

In this section, we show how the *confidence* of the DC evolve over the training. To recapitulate, we note that the confidence of the DC is the difference of the top two scores of softmax activation of the multi-class classifier. We find the histogram of the number of seen and unseen class test samples that are correctly classified by the DC. Bins of the histogram correspond to the DC confidence, i.e., S_{top2}. During the initial phase, the DC is not well trained and hence the number of seen and unseen class test samples that are correctly classified is small. In such a case, the histogram is skewed towards left as shown in Figure 5(a). As the training progresses the DC learns to separate the seen classes and PUC and hence seen and unseen classes in the GZSL inference stage. This can be observed from the histogram in Figure 5 (b) where there is a large number of seen and unseen class test samples which are correctly classified with the higher confidence thus making histograms to be skewed towards the right. Figure 6, 7, and 8 shows the training progress for APY, CUB, and FLO datasets, respectively.



(a) Epoch 0

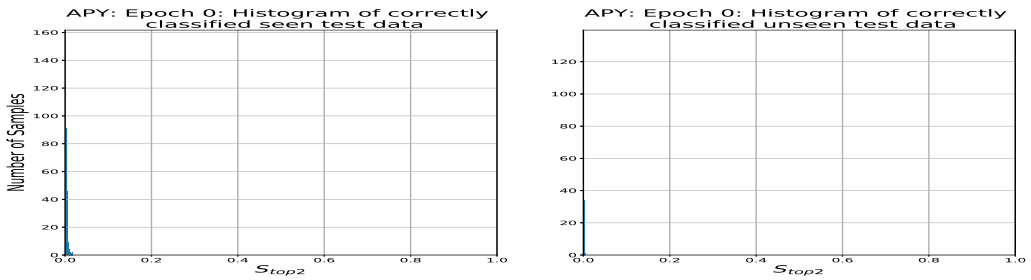


(b) Epoch 50

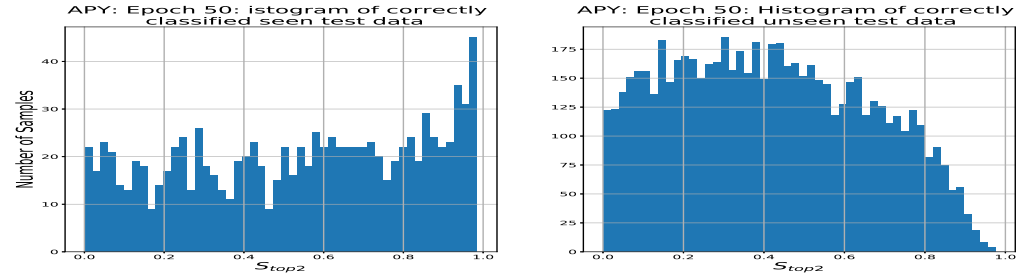
Fig. 5. AWA1 dataset. The histograms of a number of test seen and unseen class samples that are correctly classified. The bins of the histograms correspond to the DC confidence i.e S_{top2} score. As training progresses from epoch 0 to epoch 50, a large number of seen and unseen class test samples are correctly classified with high confidence.

References

1. Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5542–5551.
2. O. Gune, B. Banerjee, and S. Chaudhuri, "Structure aligning discriminative latent embedding for zero-shot learning," in *BMVC*, 2018.

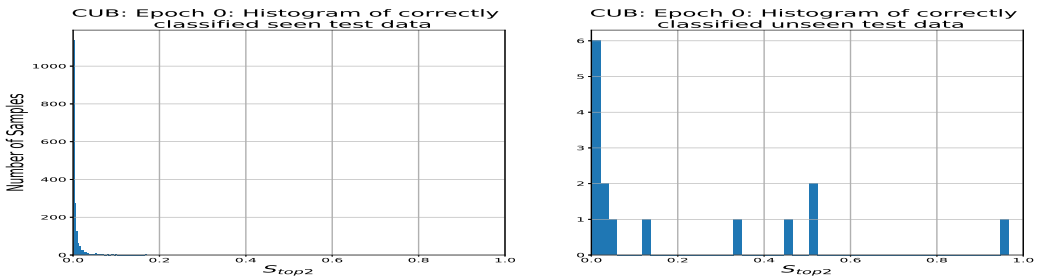


(a) Epoch 0

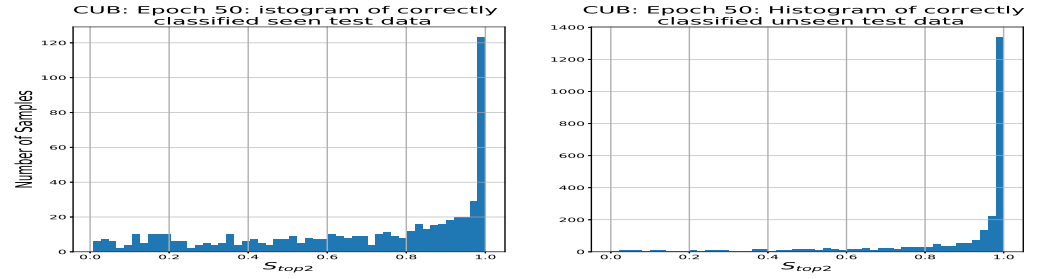


(b) Epoch 50

Fig. 6. APY dataset. The histograms of a number of test seen and unseen class samples that are correctly classified. The bins of the histograms correspond to the DC confidence i.e S_{top2} score. As training progresses from epoch 0 to epoch 50, a large number of seen and unseen class test samples are correctly classified with high confidence.

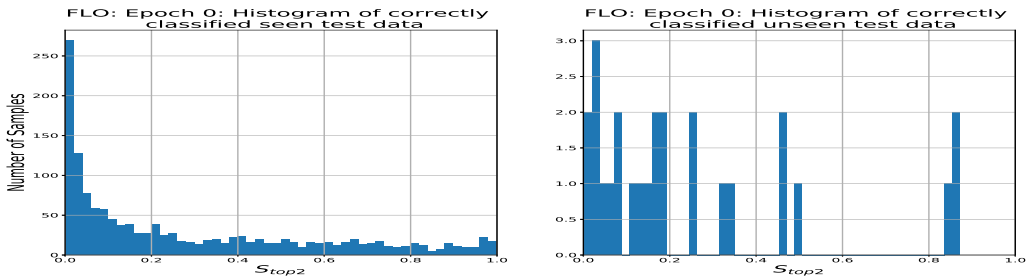


(a) Epoch 0

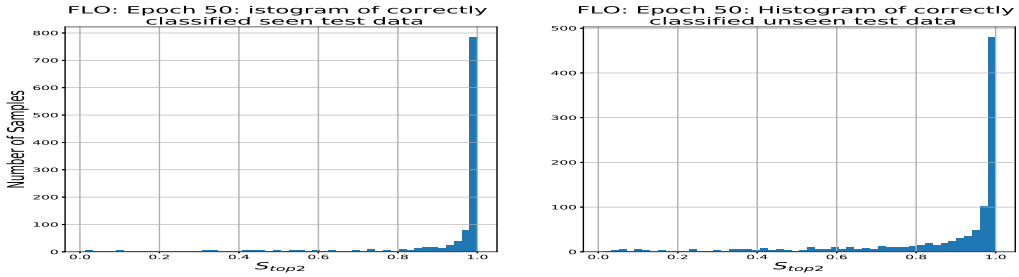


(b) Epoch 50

Fig. 7. CUB dataset. The histograms of a number of test seen and unseen class samples that are correctly classified. The bins of the histograms correspond to the DC confidence i.e S_{top2} score. As training progresses from epoch 0 to epoch 50, a large number of seen and unseen class test samples are correctly classified with high confidence.



(a) Epoch 0



(b) Epoch 50

Fig. 8. FLO dataset. The histograms of a number of test seen and unseen class samples that are correctly classified. The bins of the histograms correspond to the DC confidence i.e S_{top2} score. As training progresses from epoch 0 to epoch 50, a large number of seen and unseen class test samples are correctly classified with high confidence.