Self-supervised Domain-aware Generative Network for Generalized Zero-shot Learning

Jiamin Wu[1], Tianzhu Zhang[1], Zhengjun Zha[1], Jiebo Luo[2], Yongdong Zhang[1], Feng Wu[1]
[1] University of Science and Technology of China
[2] University of Rochester
jiaminwu@mail.ustc.edu.cn, {tzzhang, zhazj, zhyd73, fengwu}@ustc.edu.cn
jluo@cs.rochester.edu

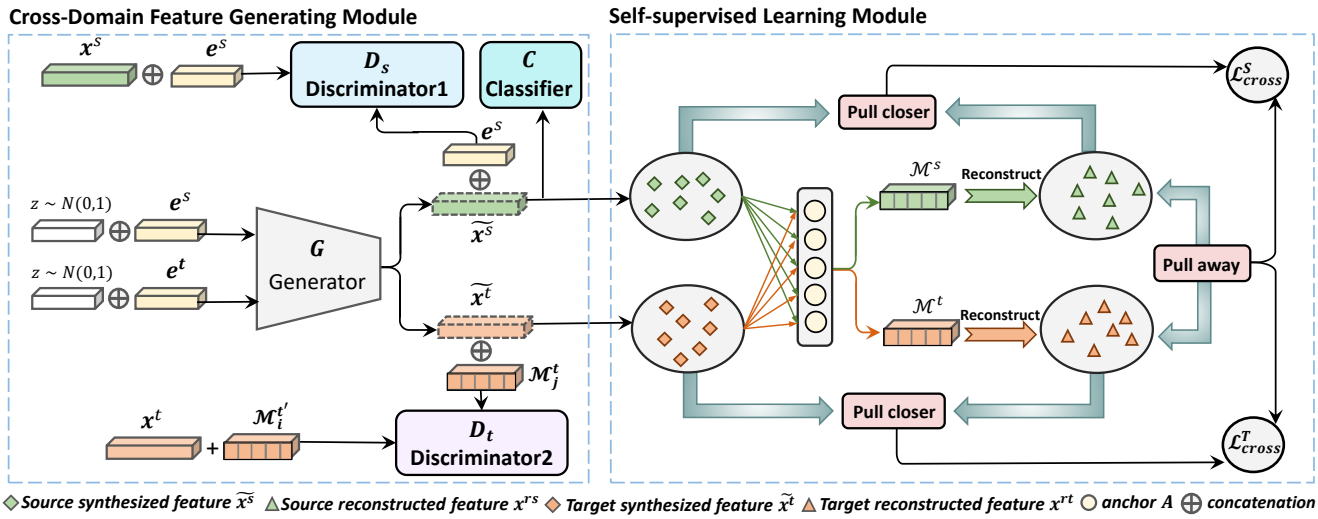


Figure 1. The architecture of our SDGN, which is composed of two modules. (1) The cross-domain feature generating module is trained to synthesize features $\tilde{x}^s$ and $\tilde{x}^t$ for seen and unseen classes. It consists of one generator G, and two discriminators D_s and D_t for source and target domains respectively. The multi-labels are combined with target features to form pairs. They act as the input for the additional target domain discriminator D^t to help preserve domain-label consistency. (2) The self-supervised learning module takes $\tilde{x}^t$ and $\tilde{x}^s$ as an input, and compares them with anchors A to derive the soft multi-labels $\mathcal{M}^s$ and $\mathcal{M}^t$ for feature reconstructions. $\tilde{x}$ and its reconstructed feature x^r in the same domain will be pulled closer while features from different domains will be pulled away. Constrained by $\mathcal{L}^{\mathcal{T}}_{cross}$, $\mathcal{L}^{\mathcal{S}}_{cross}$ and D^t, $\tilde{x}$ can evolve towards the highly-discriminative domain-aware representation.

In the supplementary material, we first introduce the details of datasets, our model architectures and parameters on the five datasets, respectively. Then, we show more t-SNE results on the generated features. Finally, we include our source code for testing and give a brief introduction.

1. Model Architectures and Parameters

The overall architecture of the proposed model is shown in Figure 1. To demonstrate the effectiveness of the proposed model, we conduct experiments on five standard datasets including: CUB [4], AwA1 [1], AwA2 [5], SUN [3] and FLO [2]. Here, we present the exact model architectures used for each experiment along with hyperparameters used to reproduce results.

1.1. More Details on Datasets

We give more detailed introduction of datasets used in the generalized zero-shot learning. CUB is a fine-grained dataset consisting of 200 species of birds, among which 150 classes are regarded as seen categories, and the other 50 classes are

Table 1. The details of benchmark datasets used in GZSL.

Dataset	CUB	AwA1	AwA2	SUN	FLO
Granularity	fine	coarse	coarse	fine	fine
Size	11K	30K	37K	14K	8K
$\mathcal{Y}$	200	50	50	717	102
$\mathcal{Y}_{\mathcal{S}}$	150	40	40	645	82
$\mathcal{Y}_{\mathcal{U}}$	50	10	10	72	20
Att	312	85	85	102	1024

regarded as unseen. It is a medium scale dataset with total 11K images. AwA1 and AwA2 are coarse-grained datasets consisting of 50 kinds of animals, with 40 classes as seen categories and 10 classes as unseen categories. The size of AwA2 is larger than AwA1, i.e., 37K vs 30K. SUN is a fine-grained and medium-scale dataset with respect to both number of images and number of classes, i.e., SUN contains 14K images coming from 717 types of scenes annotated with 102 attributes. We use 645 classes to compose source domain, and 72 classes to compose target domain. FLO is a fine-grained dataset with 8K images from 102 types of flowers, among which 82 classes used as seen categories and 20 classes are used as unseen categories. More details about datasets are shown in Table 1.

1.2. On the CUB Dataset

Hyperparameters:

- Batch size is 64.
- Base learning rate is 0.0001.
- The loss coefficient λ_C of pre-trained classifier is 0.01.
- The loss coefficient λ_t of target domain classifier is 1.
- The loss coefficient λ_a of self-supervised learning module is 0.01.
- Parameters are initialized from zero centered normal distribution with a standard deviation 0.02.
- The synthesized feature number is 300.
- The margin μ in the triplet mining loss is 0.3.
- The number of epoch is 300.

Generator G:

- It has 2 fully connected layers. The first fully connected layer is followed by a LeakyReLU layer with a negative slope set as 0.2. the second fully connected layer is followed by a ReLU layer.
- The input is the concatenation of a Gaussian noise vector and a class embedding vector.
- The dimension of Gaussian noise is set as the same as the dimension of the class embedding.
- We use the ADAM optimizer with $\beta = 0.5$.

Discriminator D_s:

- It has 2 fully connected layers. The first fully connected layer is followed by a LeakyReLU layer with a negative slope set as 0.2.
- The input is the concatenation of a source image feature and its class embedding.
- We use the ADAM optimizer with $\beta = 0.5$.

Discriminator D_t:

- It has 2 fully connected layers. The first fully connected layer is followed by a LeakyReLU layer with a negative slope set as 0.2.
- The input is the concatenation of a target image feature and its multi-label.
- We use the ADAM optimizer with $\beta = 0.5$.

Classifier C:

- It consists of one linear softmax layer.
- We use the ADAM optimizer with $\beta = 0.5$.

1.3. On the AwA Dataset

The parameters of AwA1 and AwA2 are identical.
Hyperparameters:

- Batch size is 64.
- Base learning rate is 0.00001.
- The loss coefficient λ_C of pre-trained classifier is 0.01.
- The loss coefficient λ_t of target domain classifier is 1.
- The loss coefficient λ_a of self-supervised learning module is 0.01.
- Parameters are initialized from zero centered normal distribution with a standard deviation 0.02.
- The synthesized feature number is 1800.
- The margin μ in the triplet mining loss is 0.3.
- The number of epoch is 300.

The architectures of G, D_s, D_t, C are the same as those on the CUB dataset.

1.4. On the SUN Dataset

Hyperparameters:

- Batch size is 64.
- Base learning rate is 0.0002.
- The loss coefficient λ_C of pre-trained classifier is 0.01.
- The loss coefficient λ_t of target domain classifier is 1.
- The loss coefficient λ_a of self-supervised learning module is 0.01.
- Parameters are initialized from zero centered normal distribution with a standard deviation 0.02.
- The synthesized feature number is 400.
- The margin μ in the triplet mining loss is 0.3.
- The number of epoch is 300.

The architectures of G, D_s, D_t, C are the same as those on the CUB dataset.

1.5. On the FLO Dataset

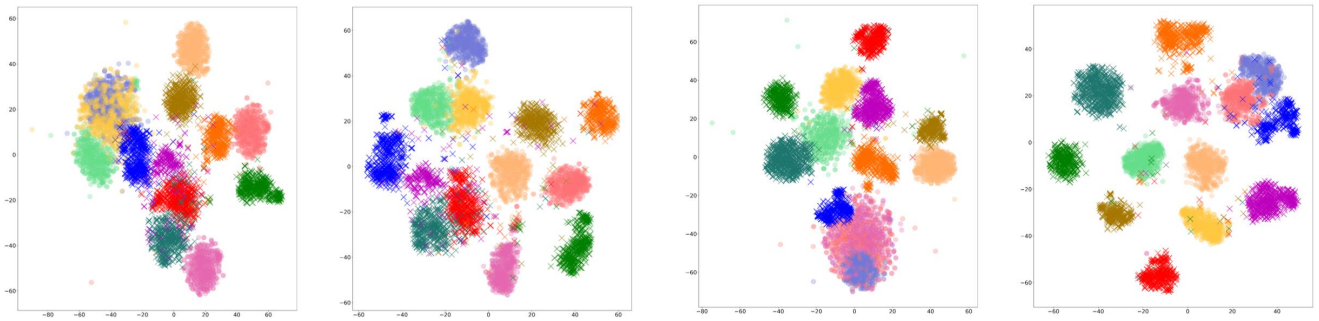
Hyperparameters:

- Batch size is 64.
- Base learning rate is 0.000001.
- The loss coefficient λ_C of pre-trained classifier is 0.01.
- The loss coefficient λ_t of target domain classifier is 1.
- The loss coefficient λ_a of self-supervised learning module is 0.01.
- Parameters are initialized from zero centered normal distribution with a standard deviation 0.02.
- The synthesized feature number is 1200.
- The margin μ in the triplet mining loss is 0.3.
- The number of epoch is 300.

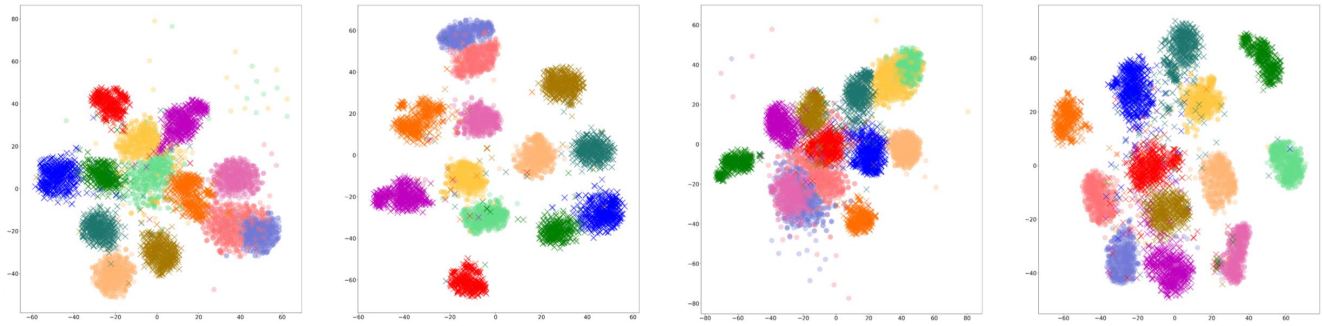
The architectures of G, D_s, D_t, C are the same as those on the CUB dataset.

2. More Visualization Results

Here, we give more t-SNE visualization results on generated features of baseline and the proposed SDGN. We show 10 groups of comparison results. In each group, the left side is the baseline fake features, while the right side is the SDGN fake features. We randomly choose different source classes and target classes to take a insight into the data distribution of generated features.



(a) (b)



(c) (d)

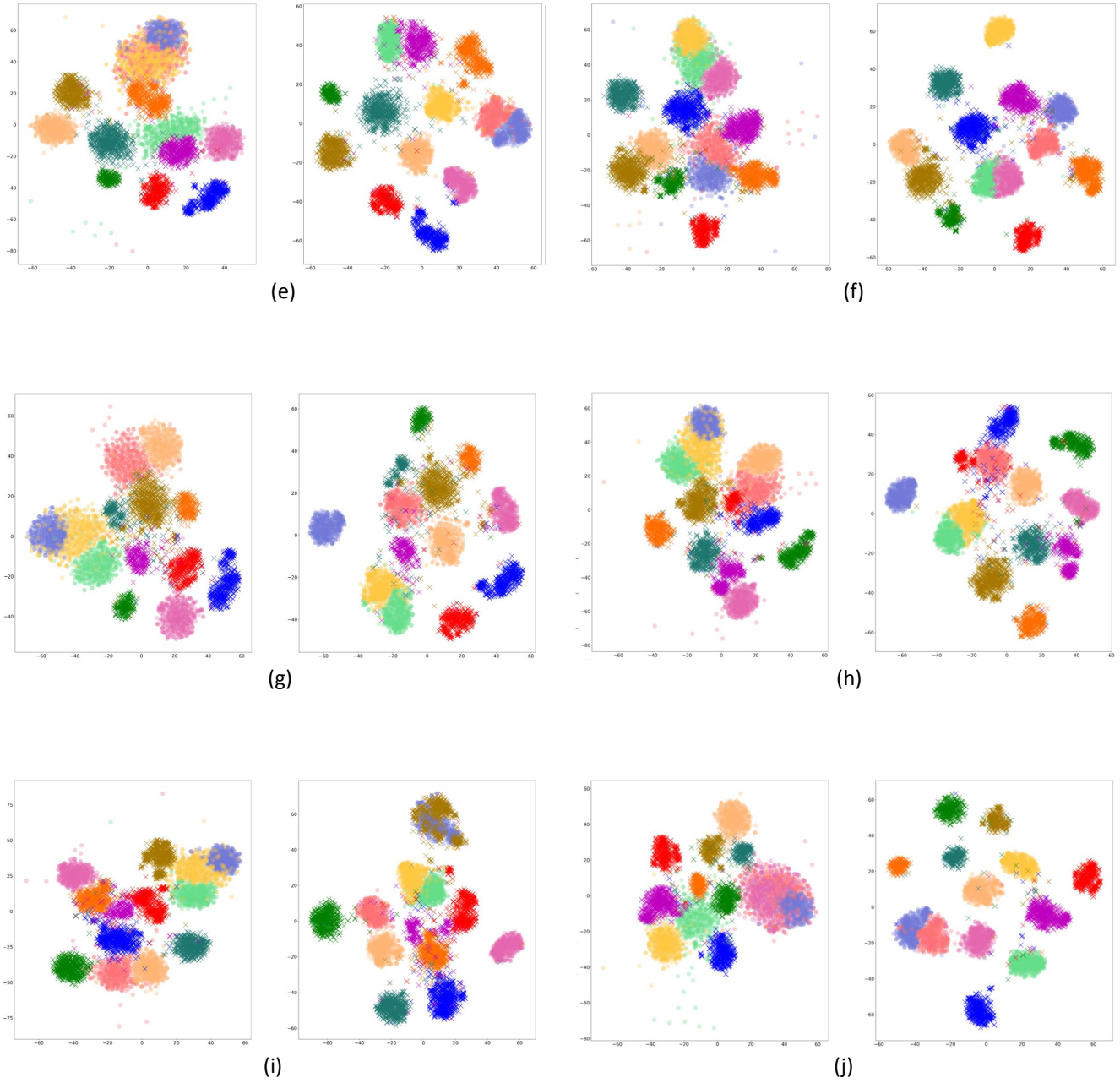


Figure 2. The t-SNE visualization results of generated features of the baseline method (left side) and our proposed SDGN (right side). $\times$ denotes the features of source classes, while $\bullet$ denotes the features of target classes.

As shown in Figure 2, across 10 groups of comparison results we can see consistent characteristics of feature distribution of source and target classes. The synthesized features of the baseline method are crowding together, while in our SDGN model, the target features are evidently pushed away from source features, and features have more segregated clusters for both source and target domains.

3. Source Code

Our source code for testing is shared in folder "Code-paper5641", which includes five files, i.e., 'demo.py', 'classifier.py', 'model.py', 'util.py' and 'ReadMe.md'.

1. 'demo.py' contains the test code for evaluating the model performance on GZSL and ZSL.

2. ‘classifier.py’ contains the code for constructing the final softmax classifier by use of fake features synthesized by a trained generator. It also provides the specific procedures to calculate the seen accuracy, unseen accuracy and harmonic mean.

3. ‘model.py’ contains the various kinds of model components for the generator and discriminators.

4. ‘util.py’ provides the code for loading data for various datasets.

5. ‘ReadMe.md’ introduces the implementation process in details.

For more details, please refer to the ‘ReadMe.md’.

References

[1] 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, 2013.

[2] 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.

[3] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.

[4] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[5] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *CVPR*, 2017.