Simple is Better: A Global Semantic Consistency Based End-to-End Framework for Effective Zero-Shot Learning

Fan Wu[1], Shuigeng Zhou[1] *, Kang Wang[1], Yi Xu[1]
Jihong Guan[2], and Jun Huan[3]

[1] Shanghai Key Lab of Intelligent Information, and School of Computer Science, Fudan University, China
{fanwu15,sgzhou,kangwang17,yxu17}@fudan.edu.cn
[2] Department of Computer Science & Technology, Tongji University, China
jhguan@tongji.edu.cn
[3] Big Data Lab, Baidu Reaseach
huanjun@baidu.com

Abstract. In image recognition, there are many cases where training samples cannot cover all target classes. Zero-shot learning (ZSL) addresses such cases by classifying the samples of unseen categories that have no corresponding samples contained in the training set via class semantic information. In this paper, we propose a novel and simple end-to-end framework, called Global Semantic Consistency Network (GSC-Net for short), which makes complete use of the semantic information of both seen and unseen classes to support effective zero-shot learning. We also employ a soft label embedding loss to further exploit the semantic relationships among classes and use a seen-class weight regularization to balance attribute learning. Moreover, to adapt GSC-Net to the setting of Generalized Zero-shot Learning (GZSL), we introduce a parametric novelty detection mechanism. Experiments on all the three widely-used ZSL datasets show that GSC-Net performs better than most existing methods under both ZSL and GZSL settings. Especially, GSC-Net achieves the state of the art performance on two datasets (AWA2 and CUB). We explain the effectiveness of GSC-Net from the perspectives of class attribute learning and visual feature learning, and discover that the validation accuracy of seen classes can serve as an indicator of ZSL performance.

Keywords: Zero-shot learning · Global semantic consistency · Label embedding loss.

1 Introduction

In some real computer vision applications, such as species classification [3], activity recognition and anomaly detection [22], labeled training samples cannot cover all target classes. Zero-shot Learning (ZSL) [1] provides a systematic way

* Correspondence author.

to address this type of problems by utilizing the semantic information of classes. Such class semantic information, including annotated attributes [9], label word vectors [16] *etc.*, can be uniformly encoded in attribute vectors [20, 32]. This process is also referred to as class embedding or (label) semantic embedding.

ZSL uses the samples of the seen classes (those having training samples) for training and tests on the samples of the unseen classes (those having no training samples). The semantic embeddings of both seen and unseen classes are used as the bridge connecting them. The essence of ZSL is to learn the association between the visual features of samples (images) and the class embeddings, which is then transferred to the samples of unseen classes.

In the test stage, ZSL considers only classifying new images of unseen classes. However, in some real-world applications, an image classification system usually needs to recognize new images from both seen and unseen classes of the application domain. This is addressed by the so-called *generalized zero-shot learning* (GZSL). Fig. 1 illustrates both ZSL and GZSL tasks. Most of the existing ZSL

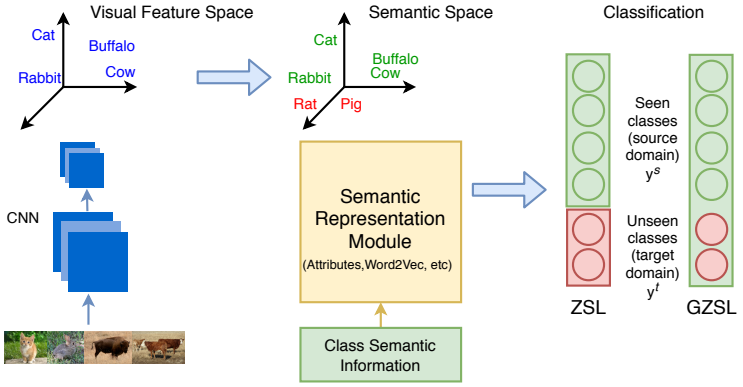


Fig. 1: Illustration of ZSL and GZSL tasks. Available data are labeled images of the seen classes (source domain, $\mathcal{Y}^s$) and semantic information of both seen and unseen classes ($\mathcal{Y}^{s+t}$). In essence, both ZSL and GZSL learn the mapping or compatibility between visual feature space and semantic space, then apply it to unseen classes (target domain, $\mathcal{Y}^t$) . At the test stage, ZSL model is only evaluated on unseen classes ($\mathcal{Y}^t$) whereas GZSL recognizes images from both seen and unseen classes ($\mathcal{Y}^{s+t}$).

methods [29] can be grouped into three types:

The 1st-type of works includes these that learn a compatibility function between the image features and the class embeddings, and treat ZSL classification as a compatibility score ranking problem [2, 10, 26]. However, these methods suffer from the following drawbacks: the attribute annotations are pointwise rather than pairwise, compatibility scores are unbounded, and ranking may fail to learn some semantic structures due to the fixed margin [4].

Methods of the 2nd-type project the visual features and semantic embeddings into a shared space and treat ZSL training as ridge regression. The shared space can be visual space, semantic space or a common space of visual features

and semantic embeddings. The prediction process of these methods is a nearest neighbor search in the shared space, which may cause *hubness problems* [19].

Most of recent works fall into the 3rd-type. They either employ deep neural networks [5, 15, 30], or use generative models [13, 28, 33], to pursue better performance. For example, Morgado *et al.* [17] adopted a semantically consistent regularization of the last fully-connected (FC) layer's weights of the neural network in end-to-end training, based on the attribute matrix of the seen classes. These methods are usually complex and hard to be deployed in general situation, and very time-consuming to be trained.

To overcome the limitations of existing ZSL methods, in this paper we propose a novel and simple end-to-end framework, called *global semantic consistency network* (GSC-Net) to exploit the semantic embeddings of both seen and unseen classes while preserving the global semantic consistency. By treating the global semantic consistency layer as a fully-connected (FC) layer with fixed weights, we can easily employ all kinds of CNN techniques such as the dropout policy, sigmoid activation, and cross entropy loss. The softmax layer and loss layer in GSC-Net are both over all classes of the learning problem domain, which thus makes full use of the semantic information in training.

Furthermore, we employ the label embedding loss to exploit the semantic relationships among classes and propose a seen-class weight regularization to balance the training,which thus guides the net to learn a more comprehensive representation. Moreover, We design a parametric novelty detection mechanism for adapting GSC-Net to the GZSL task. Experimental results over three widely-used datasets show that GSC-Net performs better than most existing methods under both ZSL and GZSL settings. We also explain the effectiveness of GSC-Net from the perspectives of class attribute learning and visual feature learning, and discover that the validation accuracy of seen classes can be an indicator of ZSL performance.

2 Method

2.1 Problem Formulation

Assume there are n_s seen classes (denoted by set $\mathcal{Y}^s$) and n_t unseen classes (denoted by set $\mathcal{Y}^t$) in a problem domain, where seen classes and unseen classes are disjoint, *i.e.*, $\mathcal{Y}^s \cap \mathcal{Y}^t = \varnothing$. So the number of total classes $n_c = n_s + n_t$. In the seen class space $\mathcal{Y}^s$, given a dataset with N_s labeled samples, $\mathcal{D}_s = \{(\mathbf{I}_i,\ y_i), i = 1, \ldots, N_s\}$ where $\mathbf{I}_i$ is the i-th training image, and $y_i \in \mathcal{Y}_s$ is the label of $\mathbf{I}_i$. Given the class attribute matrix $\mathbf{A} = [\mathbf{A}^s, \mathbf{A}^t]$ where $\mathbf{A}^s \in \mathbb{R}^{L \times n_s}$ corresponds to the seen classes, $\mathbf{A}^t \in \mathbb{R}^{L \times n_t}$ corresponds to the unseen classes, L is the attribute dimension.

Now, given a new test image $\mathbf{I}_j$, the goal of ZSL is to predict its label $\hat{y}_j$ just among the unseen classes, *i.e.*, $\hat{y}_j \in \mathcal{Y}^s$, while the goal of GZSL is to predict its label $\hat{y}_j$ among all classes, *i.e.*, $\hat{y}_j \in \mathcal{Y}^{s+t}$ where $\mathcal{Y}^{s+t} = \mathcal{Y}^s \cup \mathcal{Y}^t$.

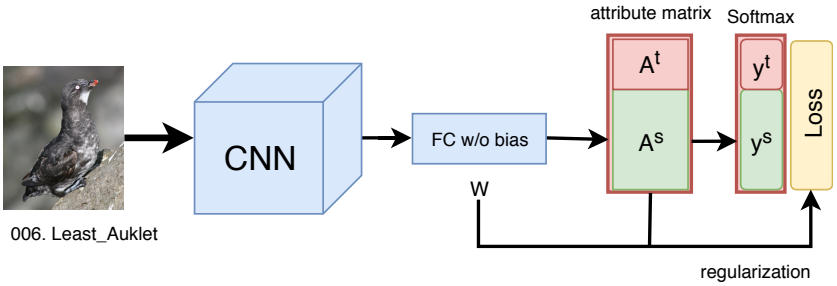


Fig. 2: The GSC-Net architecture. The class attribute matrix $\mathbf{A} = [\mathbf{A}^s, \mathbf{A}^t]$ where $\mathbf{A}^s$ is for the seen/training classes and $\mathbf{A}^t$ is for the unseen/test classes. Though no training images belong to the unseen classes, the Global Semantic Consistency (GSC) Layer, softmax layer and loss layer are designed for all classes $\mathcal{Y}^{s+t}$.

2.2 Global Semantic Consistency Network

Architecture To exploit the semantic attributes of both seen and unseen classes for training, we propose a novel, simple yet effective end-to-end approach, called *Global Semantic Consistency Network* (GSC-Net for short) for the ZSL task, and adapt it to the GZSL task later. Fig. 2 is the architecture of GSC-Net, which consists of four major components as follows:

1. **CNN block**: $x = CNN(\mathbf{I})$. In this paper, we use the pretrained resnet50 [11] as the CNN by default. The pretrained CNN acts as a feature extractor, with the original last fully-connected (FC) layer being dropped. For fast end-to-end training, we freeze this block's parameters in the first 5 epochs.
2. **FC w/o bias**: $x_a = \mathbf{W}x$. This FC layer (its weight matrix is $\mathbf{W}$ and bias is 0) maps the CNN features into a L-dimensional space. Its output can be interpreted as the image embedding in attribute space.
3. **Global Semantic Consistency (GSC) Layer**: $y^{out} = \mathbf{A}x_a$. Here, $\mathbf{A}$ is the class attribute matrix (it can also be label word embeddings). [32] discussed how to fuse multiple semantic vectors together. If the auxiliary information needs a neural encoding layer, then we can include this layer in end-to-end co-training. Since the semantic information is usually about classes and can be fixed for different samples, like the class attribute matrix, we can freeze it in the net, which thus makes it equivalent to a fully connected network with no bias. In this framework, the prediction process can be almost the same in both the training stage and the test stage by just taking the class with the maximum score.
4. **Loss**: First, we normalize the output score vector to [0, 1] with a softmax $\hat{y} = softmax(y^{out})$. Then, we adopt a *global attribute balancing loss* to handle the imbalance problem between the attributes of seen classes and that of unseen classes. This will be detailed in the next section.

Semantic consistency *vs.* global semantic consistency In order to investigate whether GSC can give a better supervision on both seen and unseen classes, we also design a *semantic consistency network* (SC-Net) for comparison. In SC-Net, $\mathbf{A}^s$ and $\mathbf{A}^t$ are respectively used in the training stage and the test stage, which means the semantic manifold formed by seen classes ($\mathbf{A}^s$) is not aware of the unseen class information ($\mathbf{A}^t$).

In GSC-Net, as we use unseen class information ($\mathbf{A}^t$) in the training stage, though unseen class images are not input to the net, we can still use the global softmax training to form a more comprehensive discriminant space. Intuitively, this can improve performance not only on the ZSL task, but also on the GZSL task that recognizes both training and test classes ($\mathcal{Y}^{tr+ts}$) at the same time. Furthermore, the softmax and cross entropy loss are also applied to the $(n_{tr} + n_t)$-dimension output vector $\hat{y}$. Therefore, GSC-Net pays more attention to the attributes mainly owned by unseen classes, which can make the learned features more discriminative among the unseen classes.

2.3 Global Attribute Balancing Loss

Since the class attribute matrix is given as the only term connecting seen and unseen classes, the key of ZSL or GZSL is to make the net learn a suitable embedding x_a on the L-dimension (attribute) space. In GSC-net, there are two major reasons that may leave the embedding x_a extremely imbalanced on different attributes: 1) only the seen classes are supervised positively in GSC-net; 2) there may be domain shift between seen and unseen classes. Taking these into account, we propose a *global attribute balancing loss* (GAB-loss) for GSC-Net as follows:

$$L_{GAB} = \alpha L_{CE} + (1-\alpha) L_{SLE} + \lambda ||\mathbf{W}\mathbf{A}^s||_2^2 + \beta ||\mathbf{W}||_2^2 \tag{1}$$

where the 1st term is the standard one-hot target cross entropy loss L_{CE}, and the 4th term is a simple weight decay on $\mathbf{W}$ for better generalization. Our contributions lie in the 2nd term and the 3rd term. Concretely, the 2nd term is the soft target cross entropy loss L_{SLE}, and the 3rd term is a L_2 regularization to constrain the weights of seen classes, where $\mathbf{A}^s$ is the seen class attribute matrix. It is actually to balance the attributes of seen and unseen classes, so we call it *attribute balancing regularization*, or *AB-regularization* for short. In what follows, we give detailed explanations on these terms.

Cross entropy loss L_{CE} Formally, it is

$$L_{CE} = q(\hat{y}, y^{true}) \tag{2}$$

where $q(\cdot)$ is a typical cross entropy loss function, $\hat{y}$ is the output vector of the net, y^{true} is the one-hot vector of the target label. Here, we do not use weighted approximate ranking loss [1] because the class semantic matrix used in experiments is point-wisely labeled and cross entropy loss performs better in various experiments.

Soft label embedding loss L_{SLE} With the GSC-Net, less seen class images will be misclassified into unseen classes in GZSL, but more unseen class images will be misclassified into seen classes. This is because the training samples all fall into seen classes y^s, making the weights corresponding to y^s larger and larger than those corresponding to y^t during training process.

As the one-hot supervision will cause the net to 'lazily' learn a smaller weight for these attributes on which unseen classes have high scores (in the class attribute matrix), so we add a soft label guide to the original cross entropy loss as in [23]:

$$L_{SLE} = q(\hat{y}, Y^l_{emb}) \tag{3}$$

where l is the true label index and Y^l_{emb} is the l-th row of soft label embedding matrix Y_{emb}. We have to utilize the semantic information again to generate the soft label embedding Y_{emb} for all classes $\mathcal{Y}^{s+t}$. Inspired by label propagation, we use the class attribute matrix $\mathbf{A}$ to build a label graph, and employ the adaptive scale policy [31] to compute the class similarity. The similarity between two classes is

$$S_{ij} = \begin{cases} e^{-\eta \frac{||A_i - A_j||^2}{h(A_i)h(A_j)}}, & A_j \in \mathcal{N}(A_i); \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

$\mathcal{N}(A_i)$ is the neighbor set of A_i, which can be evaluated by setting a distance threshold to reduce the computation cost. We can also directly replace the values of relatively small A_{ij} with 0. The **local scale function** $h(x)$ is defined as

$$h(x) = ||x - x^{(k)}|| \tag{5}$$

where $x^{(k)}$ is the k-th nearest neighbor of point x. In experiments, we find that it is good enough to set k to 1 or 2.

η in Eq. (4) is a hyperparameter to control the centralization degree of S. The larger η is, the farther a node is away from its neighbors, then Y_{emb} will degenerate to the naive one-hot label. Since the local scale function $h(x)$ actually normalizes the numerator term of Eq. (4), it can be easy to set η to get an appropriate similarity.

Normalizing S by row, then we get the normalized class embedding matrix $Y_{emb} \in \mathbb{R}^{n_c \times L}$, each row can be viewed as the soft label.

Attribute balancing regularization We have two L_2 regularizations (the third and the fourth terms) in Eq. (1). The two terms can be derived as follows:

Inspired by [12], we can minimize the reconstruction error and the regression term as follows:

$$\min_{X_i, W} ||W^T X_i - A_i||^2 + \lambda ||W A_i - X_i||^2 + \beta ||W||^2 \tag{6}$$

where X_i is the CNN feature of the i-th sample while A_i is the attribute vector of the sample's corresponding class. Since A_i is fixed, this formula can be rewritten as:

$$\begin{aligned} &\min_{X_i, W} -2(1+\lambda) A_i^T W^T X_i + ||W^T X_i||^2 + ||X_i||^2 \\ &+\lambda ||W A_i||^2 + \beta ||W||^2 \end{aligned} \tag{7}$$

Through simple deduction, the optimization directions of the first two terms in Eq. (7) are consistent with L_{CE}. We can approximately replace the first two terms in Eq. (7) with L_{CE}. Furthermore, the third term $||X_i||^2$ is restricted by batch normalization. Since only seen class samples are put into the training pipeline, the regularization on A_i can be generalized into A^s. Then the target function turns out to be:

$$\min_{X,y^*} L_{CE} + \lambda||WA^s||^2 + \beta||W||^2 \tag{8}$$

which matches the GAB-loss in Eq. (1).

Overall, GAB-loss can be applied to many problems with unbalancing training data. In the GAB-loss of Eq. (1), α is a hyperparameter falling in [0, 1]. A large α will degenerate the loss to a standard cross entropy. We set it around 0.5 if no prior knowledge. If $\alpha = 0$, L_{SLE} dominates GAB-loss. If the FC layers are randomly initialized at the beginning, the projection on each class is almost the same, so L_{SLE} will make the learning process slow at the starting stage. By increasing the value of α, we can make training faster and get higher accuracy for seen classes. Since the training samples all belong to seen classes, L_{GAB} puts more positive supervision to the unseen class attributes.

Relationship to existing deep ZSL models. Many methods [14, 1, 32] map the visual features and the label semantic vectors into a shared space, then do classification by computing the nearest label embedding vector:

$$c = \arg\min_c ||\theta(x) - \mathbf{A}_y^c||^2 \tag{9}$$

where $\mathbf{A}_y^c$ is the embedding vector of the c-th class. This nearest search method can be clearly visualized and easy to interpret. However, the mean square error is less effective than cross entropy loss in end-to-end training. So we actually transform the search into a softmax classification. Since $\phi(\mathbf{I}_j)$ is independent of classification, Eq. (9) can be written as

$$c = \arg\min_c -\theta(x)^T\mathbf{A}_y^c + \frac{1}{2}||\mathbf{A}_y^c||^2. \tag{10}$$

Since $\mathbf{A}_y^c$ is set statistically equal for each class, Eq. (10) can be simplified to

$$c = \arg\max_c \theta(x)^T\mathbf{A}_y^c \tag{11}$$

where $\theta(x)^T\mathbf{A}_y^c$ can be seen as expression score on class c. Eq. (11) is equivalent to the last FC layer with no bias in GSC-Net. This maximization process can be integrated into a softmax layer and trained with cross entropy loss.

2.4 Parametric Novelty Detection for GZSL

Here we adapt our model for the generalized zero-shot learning (GZSL) task by adding a parametric novelty detection (PND) mechanism. In GSC-Net, unseen

Table 1: Details of the ZSL datasets with the proposed splits

Dataset	No. of attributes	No. of seen classes	No. of unseen classes	No. of samples	No.of samples (Train)	No. of samples from unseen classes (Test)	No. of samples from seen classes (Test)
SUN [18]	102	645	72	14340	10320	1440	2580
AWA2 [27]	85	40	10	37322	23527	7913	5882
CUB [25]	312	150	50	11788	7057	2967	1764

class images still have relatively high scores on seen classes, which means in most cases $y^{Seen} > y^{Unseen}$ in the output vector. Therefore, we set a hyperparameter γ to control the novelty detection as in [7]. When

$$\max_i y_i^{Seen} < \gamma \cdot \max_j y_j^{Unseen}, \tag{12}$$

we say an unseen class image is detected, and take the maximum y^{Unseen} term as the predicted class. So the prediction method with controllable novelty detection goes as follows:

$$c = \begin{cases} \text{argmax}_i\, y_i^{Seen}, & \max_i y_i^{Seen} \geq \gamma \cdot (\max_j y_j^{Unseen}); \\ \text{argmax}_j\, y_j^{Unseen}, & \text{otherwise.} \end{cases} \tag{13}$$

In experiments, γ must be larger than 1. The larger the γ value is, the higher the accuracy on unseen classes is. Our PND mechanism can be easily applied to a typical deep ZSL model. When applied to a certain method, we just append '*' to the method's name for notation.

3 Performance Evaluation

3.1 Datasets and Experimental Settings

Datasets: Xian *et al.* [27] gave a comprehensive evaluation on the existing ZSL methods on several widely used datasets, and proposed an adapted dataset Animals with Attributes 2 (**AwA2**) as well as some suggestions on dataset splits for these ZSL datasets. Since our target is to develop a unified end-to-end ZSL framework, we choose 3 datatsets that have open original images and class attribute annotations: **AwA2** [27], CUB-200-2011 (**CUB**) [25] and Scene UNderstanding (**SUN**) [18]. Table 1 shows more details about these datasets.

In order to make our approach more practical and applicable to more scenarios, we utilize only the class attribute annotations rather than individual samples' attributes. It is common in the datasets that the numbers of images in some classes are much larger than that in the other classes. Therefore, we use the average per-class accuracy to present our results.

Settings: The 2-stage methods use the 2048-D Resnet101 [11] features provided by [27] for all the datasets. To show that our framework can get better

results on even smaller CNN base models, we use pretrained Resnet50 [11] as our CNN module, which also outputs 2048-D vectors. In the beginning epochs, since CNN is well pretrained on ImageNet, we can freeze the CNN parameters and train the FC layers only.

Training policy: We use AdaGrad optimizer [8] with a learning rate 10^{-3}. Regularization ratios λ and β are set to 0.1 and 0.005. L_{CE} ratio α is set to 0.5 by default. For our 3 datasets, to avoid tuning parameters according to test results, we set the affinity factor $\eta = 1.4$, and novelty factor $\gamma = 1.4$. Since we use local function, $\eta \in [1.2, 1.8]$ is suitable enough. In real applications, γ can be set to meet different requirements. If the number of training samples per class is large, which means the seen classes overwhelm unseen classes, γ needs to be large. If α is small, the target label will be soft, then small γ is considered. We run experiments on Titan Xp GPUs with early stopping policy.

3.2 Ablation Study

To testify the benefit of each component in GSC-net, we consider 3 comparison cases: SC-Net, GSC-Net without L_{SLE} (setting λ=0.1 and α=1.0) and GSC-Net without attribute balancing regularization (setting λ=0 and α=0.5).

ZSL results The results are presented in Table 2. The upper part shows the 2-stage methods whose results were reported in [27]. ALE [2] is simple but effective on all datasets. These methods all use 2048-D ResNet101 features. The lower part stands for end-to-end approaches. Under the same protocol, we implemented Deep-SCoRe, DEM, and our models SC-Net and GSC-Net on resnet50. The result of S^2GA [30] is directly cited from the original paper where it was evaluated only on CUB.

On the basis of SC-Net, GSC-Net improves performance a lot by making full use of the total class attribute matrix and boosting the feature learning for unseen classes. With L_{SLE}, GSC-Net further lifts the performance. Overall, GSC-Net surpasses the existing methods and achieves the state-of-the-art performance on all the three datasets.

Comparing the end-to-end (E2E) methods and 2-stage (2S) methods, we can easily discover that E2E methods exceed 2S methods significantly on AWA2 and CUB, but hit a draw on SUN. The reasons may be: 1) there are only 16 images per seen class in SUN, which does not contribute much to CNN finetuning. 2) There are 717 classes but only 102 attributes annotated in SUN. Note that the dimension of the class attribute matrix W, *i.e.*, the last FC weights, is 717×102, therefore the feature dimensionality of 102 is not large enough for 717-way classification.

GZSL results In GZSL setting, the search space contains both the seen classes and the unseen classes. We use the same evaluation protocol as in [27]. Let **ts** be GZSL accuracy on unseen classes and **tr** GZSL accuracy on seen classes. **H** is the harmonic mean between **ts** and **tr**. **H** pays attention to the smaller one

Table 2: Average per-class accuracy (top-1 in %) for the ZSL task

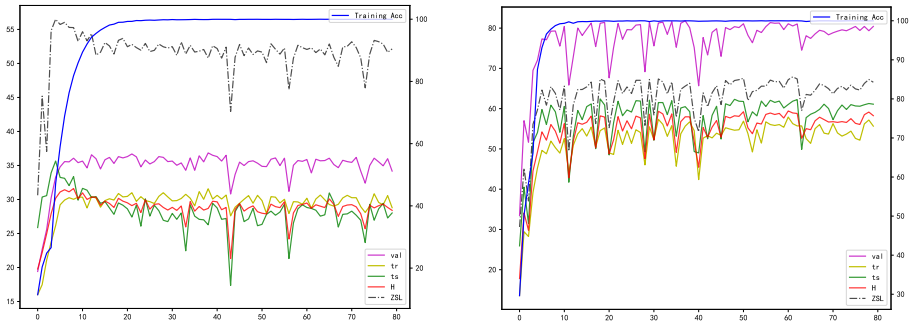
Method	SUN	AWA2	CUB
LATEM [26]	55.3	55.8	49.3
ALE [2]	58.1	62.5	54.9
DEVISE [10]	56.5	59.7	52.0
SJE [3]	53.7	61.9	53.9
ESZSL [21]	54.5	58.6	53.9
SYNC [6]	56.3	46.6	55.6
SAE [12]	40.3	54.1	33.3
Deep-SCoRe [17](Resnet50)	51.7	69.5	61.0
DEM [32](Resnet50)	51.1	68.7	60.1
RELATION NET [24](GoogleNet)	-	-	62.0
S^2GA [30]	-	-	68.9
SC-Net (baseline, Resnet50)	52.7	71.2	61.4
GSC-Net without L_{SLE} (Resnet50)	56.9	73.7	65.1
GSC-Net without *AB-regularization* (Resnet50)	58.1	74.5	68.2
GSC-Net (Resnet50)	**58.3**	**75.4**	**69.2**

Table 3: Results on the GZSL task. '*' refers to employing our novelty detection mechanism.

	SUN			AWA2			CUB		
Method	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**
LATEM [26]	14.7	28.8	19.5	11.5	77.3	20.0	15.2	57.3	24.0
ALE [2]	21.8	33.1	26.3	14.0	81.8	23.9	23.7	62.8	34.4
DEVISE [10]	16.9	27.4	20.9	17.1	74.7	27.8	23.8	53.0	32.8
SJE [3]	14.7	30.5	19.8	8.0	73.9	14.4	23.5	59.2	33.6
ESZSL [21]	11.0	27.9	15.8	5.9	77.8	11.0	12.6	63.8	21.0
SYNC [6]	7.9	**43.3**	13.4	10.0	90.5	18.0	11.5	**70.9**	19.8
SAE [12]	8.8	18.0	11.8	1.1	82.2	2.2	7.8	54.0	13.6
DeepSCoRe* [17]	17.3	30.8	22.2	8.8	91.1	16.0	20.3	65.8	31.0
f-CLSWGAN with softmax [28]	**42.6**	**36.6**	**39.4**	-	-	-	43.7	57.7	49.7
SC-Net* (baseline)	10.3	33.4	15.8	3.8	**93.4**	7.3	15.0	**70.1**	24.7
GSC-Net* without L_{SLE}	35.3	30.1	32.5	27.0	72.9	39.4	51.9	59.7	59.1
GSC-Net* without *AB-regularization*	30.7	**35.3**	32.8	21.3	90.8	34.5	50.4	61.3	55.3
GSC-Net*	**37.5**	31.5	**34.2**	**40.2**	80.5	**53.7**	**53.6**	68.9	**60.3**

between **tr** and **ts**, it is a balanced evaluation for the GZSL task. Table 3 reports the results of GZSL on the three datasets. Some results of existing approaches are obtained from [27]. In the upper part, we can see that most existing ZSL methods perform very poorly on GZSL task in terms of **H** and **ts**. Comparing with these methods, our method can effectively boost the **H** accuracy on all 3 datasets by a large margin.

For the three datasets, GSC-Net improves performance most significantly on CUB, with **H** increasing from 24.7% to 60.3%, mainly due to better attribute



(a) SUN (b) CUB

Fig. 3: GSC-Net ($\alpha = 0.5$) training processes for ZSL task and GZSL task on (a) SUN and (b) CUB respectively. The X-axis is the number of training epochs. The left Y-axis means ZSL (GZSL) accuracy while the right Y-axis is training accuracy. The blue and purple lines indicate training accuracy and validation accuracy on seen classes.

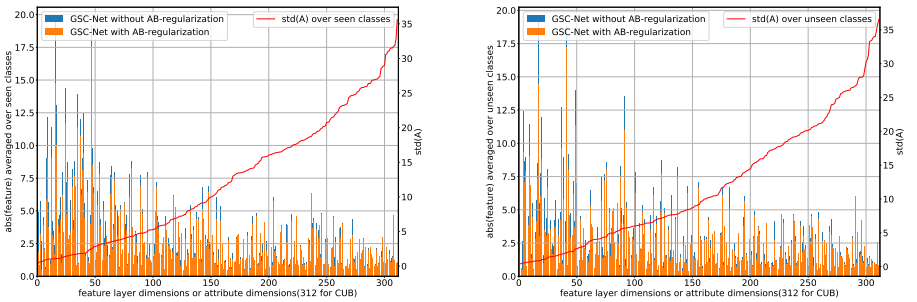
balancing between seen and unseen classes. For SUN, there are too many classes and only 16 images per training seen class, which makes it a challenging problem to get high accuracy on both **ts** and **tr**, since the small number of images per class in SUN cannot support end-to-end finetuning well on this setting. It is worthy to notice that [28] uses pretrained ResNet101 features, so it gets better results on SUN. On the other hand, AWA2 faces an extremely unbalancing situation: the number of images in each seen class is quite large, which may make many test images of unseen classes be classified into seen classes in GZSL. Nevertheless, our method still significantly improves the performance on AWA2, lifting **H** from 7.3% (baseline) to 53.7%.

Fig. 3 shows the training processes of GSC-Net (α=0.5) for ZSL task and GZSL task on SUN and CUB respectively. We can see that **ts** for unseen classes in GZSL is much lower than ZSL accuracy for seen classes, which shows that GZSL is a much harder task than ZSL.

The model reaches a high accuracy in less than 20 epochs and then oscillates irregularly, so we save the earlier models with early stopping policy. Fig. 3 also shows that *ZSL/GZSL accuracy fluctuates with the validation accuracy val* (purple line in Fig. 3) *almost in the same pace*. This obviously reveals that better feature learning gives better ZSL/GZSL prediction. Therefore, we can refer to the *validation accuracy for seen classes* to select the saved models in real scenarios. This can effectively alleviate the situation that the previous deep learning methods of ZSL have to leave some of seen classes as unseen validation set. So this discovery can help exploit the full power of training data.

3.3 Effectiveness of the AB-regularization

Here, we investigate how GSC-Net and AB-regularization work on the CUB dataset. First, we get the feature vectors (x_a layer) for validation images of



(a) *abs*(feature) averaged over seen classes in CUB

(b) *abs*(feature) averaged over unseen classes in CUB

Fig. 4: Attribute analysis on CUB. Attributes (dimensions) are sorted by std(A).

the 200 CUB classes and compute the average features for each class. Thus, we can get a 200*312 matrix $\mathbf{X}_a$ by concatenating these 200 vectors, which can be compared with the class attribute matrix $\mathbf{A}$. Then, we compute the standard deviation of $\mathbf{A}$, i.e. $std(A)$, for both seen and unseen classes. As shown in Fig. 4, the attributes are sorted in ascending order by $std(A)$. From left to right, $std(A)$ goes bigger, which in some extent means that the classes are more distinguishable on those attributes of the *right* part in Fig. 4(a) and (b). From Fig. 4, we can see that GSC-Net with AB-regularization tends to learn balanced features, rather than biased to a small part of attributes when learning on samples from seen classes. The AB-regularization makes the network tend to utilize more attributes, and the features more balanced and effective, with larger values on the *right* part attributes of unseen classes, as shown by the orange histograms in Fig. 4(b). Moreover, for seen and unseen classes, both the attribute feature distributions and $std(\mathbf{A})$ in corresponding positions are nearly similar, which can explain why ZSL works well on CUB.

4 Conclusion

In this work, we try to make full use of the global class semantic information to improve the classification performance of ZSL and GZSL. We first propose a novel end-to-end model with a neural weighted unit to increase the learning ability under a global semantic constraint. We then employ a soft label embedding loss with attribute balancing regularization to further exploit the semantic relationships between classes, which thus enables the neural network to transfer more knowledge to unseen classes without overfitting either the seen classes or their highly related attributes. We show the effectiveness and advantage of the proposed method by extensive experiments for both ZSL and GZSL tasks. We also discover that the validation accuracy on seen classes can be an indicator for ZSL performance, which can be a practical guide for training and early stopping.

Acknowledgement. This work was supported by the Science and Technology on Complex System Control and Intelligent Agent Cooperation Laboratory.

References

1. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for attribute-based classification. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 819–826. IEEE (2013)
2. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for image classification. IEEE transactions on pattern analysis and machine intelligence **38**(7), 1425–1438 (2016)
3. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2927–2936. IEEE (2015)
4. Annadani, Y., Biswas, S.: Preserving semantic relations for zero-shot learning. arXiv preprint arXiv:1803.03049 (2018)
5. Annadani, Y., Biswas, S.: Preserving semantic relations for zero-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
6. Changpinyo, S., Chao, W.L., Gong, B., Sha, F.: Synthesized classifiers for zero-shot learning. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5327–5336 (2016)
7. Chao, W.L., Changpinyo, S., Gong, B., Sha, F.: An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In: European Conference on Computer Vision. pp. 52–68. Springer (2016)
8. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(Jul), 2121–2159 (2011)
9. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1778–1785. IEEE (2009)
10. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al.: Devise: A deep visual-semantic embedding model. In: Advances in neural information processing systems (NIPS). pp. 2121–2129 (2013)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
12. Kodirov, E., Xiang, T., Gong, S.: Semantic autoencoder for zero-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
13. Kumar Verma, V., Arora, G., Mishra, A., Rai, P.: Generalized zero-shot learning via synthesized examples. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
14. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) **36**(3), 453–465 (2014)
15. Li, Y., Zhang, J., Zhang, J., Huang, K.: Discriminative learning of latent features for zero-shot recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)

16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems (NIPS). pp. 3111–3119 (2013)
17. Morgado, P., Vasconcelos, N.: Semantically consistent regularization for zero-shot recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 9, p. 10 (2017)
18. Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2751–2758. IEEE (2012)
19. Radovanović, M., Nanopoulos, A., Ivanović, M.: Hubs in space: Popular nearest neighbors in high-dimensional data. Journal of Machine Learning Research **11**(Sep), 2487–2531 (2010)
20. Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. pp. 49–58 (2016)
21. Romera-Paredes, B., Torr, P.: An embarrassingly simple approach to zero-shot learning. In: International Conference on Machine Learning (ICML). pp. 2152–2161 (2015)
22. Socher, R., Ganjoo, M., Manning, C.D., Ng, A.: Zero-shot learning through cross-modal transfer. In: Advances in neural information processing systems (NIPS). pp. 935–943 (2013)
23. Sun, X., Wei, B., Ren, X., Ma, S.: Label embedding network: Learning label representation for soft training of deep networks. arXiv preprint arXiv:1710.10393 (2017)
24. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. arXiv preprint arXiv:1711.06025 (2017)
25. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds200-2011 dataset. California Institute of Technology (2011)
26. Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B.: Latent embeddings for zero-shot classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 69–77 (2016)
27. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. IEEE transactions on pattern analysis and machine intelligence (2018)
28. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
29. Xian, Y., Schiele, B., Akata, Z.: Zero-shot learning - the good, the bad and the ugly. In: IEEE Computer Vision and Pattern Recognition (CVPR) (2017)
30. Yu, Y., Ji, Z., Fu, Y., Guo, J., Pang, Y., Zhang, Z.: Stacked semantic-guided attention model for fine-grained zero-shot learning. arXiv preprint arXiv:1805.08113 (2018)
31. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in neural information processing systems (NIPS). pp. 1601–1608 (2005)
32. Zhang, L., Xiang, T., Gong, S.: Learning a deep embedding model for zero-shot learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3010–3019 (2016)
33. Zhu, Y., Elhoseiny, M., Liu, B., Peng, X., Elgammal, A.: A generative adversarial approach for zero-shot learning from noisy texts. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)