Neural Computing and Applications (2021) 33:5313–5326
https://doi.org/10.1007/s00521-020-05322-7

REVIEW ARTICLE

Bidirectional generative transductive zero-shot learning

Xinpeng Li[1] · Dan Zhang[1] · Mao Ye[1] · Xue Li[2] · Qiang Dou[1] · Qiao Lv[1]

Received: 11 March 2020 / Accepted: 2 September 2020 / Published online: 12 September 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract
Most zero-shot learning (ZSL) methods aim to learn a mapping from visual feature space to semantic feature space or from both visual and semantic feature spaces to a common joint space and align them. However, in these methods the visual and semantic information are not utilized sufficiently and the useless information is not excluded. Moreover, there exists a strong bias problem that the instances from unseen classes always tend to be predicted as some seen classes in most ZSL methods. In this paper, combining the advantages of generative adversarial networks (GANs), a method based on bidirectional projections between the visual and semantic feature spaces is proposed. GANs are used to perform bidirectional generations and alignments between the visual and semantic features. In addition, cycle mapping structure ensures that the important information are kept in the alignments. Furthermore, in order to better solve the bias problem, pseudo-labels are generated for unseen instances and the model is adjusted along with them iteratively. We conduct extensive experiments at traditional ZSL and generalized ZSL settings, respectively. Experiment results confirm that our method achieves the state-of-the-art performances on the popular datasets AWA2, aPY and SUN.

Keywords Zero-shot learning · Transductive · Bidirectional generation · CycleGAN

1 Introduction

Image recognition trained on a large number of labeled instances can get good results at present, but it takes a lot of manpower and resources to collect these labeled images. Especially, it requires experts to give identification for fine-grained classification. How to complete image recognition with only a few labeled instances or even some categories without labels has become a very challenging and realistic task.

Zero-shot learning (ZSL) [22, 33, 41] is an effective method to solve the above problem. Zero-shot learning is a special unsupervised domain adaptation method. Its purpose is to learn a model based on a set of labeled source data, and then transfer the learned knowledge to the target domain to identify another set of unlabeled data. In zero-shot learning setting, the data categories in these two domains are assumed completely non-overlapping.

Because the source data during training are labeled, we usually call the classes in source domain as seen classes, and the classes in target domain as unseen classes. Zero-shot learning can be divided into traditional ZSL and generalized zero-shot learning (GZSL), which are called ZSL and GZSL, respectively. The difference is that, in the test, ZSL only classifies instances from target domain without labeled visual samples, while GZSL classifies all instances from both source and target domains. Zero-shot learning can also be divided into two categories as inductive ZSL and transductive ZSL. For inductive ZSL, we can only use the labeled data from source domain for training; while for transductive ZSL, we can use not only the labeled data from source domain but also the unlabeled data from target domain at the time of training. For inductive ZSL, the predictions of instances from target domain depend entirely on the knowledge learned from source domain. But for transductive ZSL, the unlabeled data from target domain can be used to adjust the trained model iteratively.

Since the unseen classes do not appear at all during the training, we need some auxiliary information, that is,

✉ Mao Ye
maoye@uestc.edu.cn

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, People's Republic of China

[2] School of Information Technology and Electronic Engineering, The University of Queensland, Brisbane, QLD 4072, Australia

semantic description. These auxiliary information can be semantic attributes vectors [1, 8], word2vec [29] and human gaze [18], etc. For example, semantic attributes vectors define some common characteristics between the seen and unseen classes. If both the seen and unseen classes are animals, the semantic attributes vectors will be fur, color, and stripes, etc. By semantic attributes vectors, the visual features from the seen or unseen classes can be bridged. Thus, only auxiliary information are needed, which greatly reduces the collection difficulty of labeled data.

In the semantic embedding research direction, some of existing zero-shot methods map the visual features to the semantic space [4, 9, 21, 33]. But in this way, they reduce the expression ability of visual information. Some methods map the semantic features to the visual space [20, 35, 44]. However, the expression ability of semantic attributes vectors is reduced and the noise will be introduced that are not visual descriptions at all [6, 7, 32]. The remaining methods project the visual and semantic features into a common space [5, 26, 45] and align them. However, some simple and rough alignments, such as the shortest Euclidean distance between them, are usually adopted. We call such alignment hard alignment. The visual and semantic feature distributions by such hard alignments are not well aligned at the overall level. Meanwhile, there exist an obvious bias problem when incorrectly bridging visual and semantic information as shown in Fig. 1. When classifying instances from target domain, they are always predicted to be some seen classes in source domain, which is a serious issue that exists in many zero-shot learning methods.

In order to solve the above mentioned problems, we propose a bidirectional mapping method. With the bidirectional projections, we can make full use of the information from two domains without introducing too much noise. Motivated by the idea of cycleGAN [47], a couple of GANs [13] are used to solve the problem of hard alignment. Two generators realize the bidirectional mappings between the visual features and semantic features. At the same time, we remap the information that has been mapped to another domain back to the domain it belongs to, and compare it with the information before the mapping. The error between them is called cycle loss. Cycle loss and classification task loss further guarantee that the important information is kept and the alignment is correct. In order to solve the bias problem better, a transductive method is proposed to use pseudo-labels for model correction iteratively. At the test phase, we do not give a classification result based on the features in only one space. The features in both the visual and semantic feature spaces are combined to give a decision. In summarization, this paper has the following contributions:

1. A transductive method of bidirectional projections is proposed. The method makes the visual features more consistent with the corresponding semantic features and greatly weaken the bias problem. Extensive experimental results show that our model achieves the state-of-the-art performance at both ZSL and GZSL settings.
2. We propose a new zero-shot classifier based on the bidirectional projection method. The classifier combines both visual and semantic features to give the final prediction, which makes full use of visual and semantic information to reduce discriminant bias.

The reminder of the paper is arranged as follows. Section 2 introduces the related works of transductive zero-shot learning and zero-shot learning based on GANs. We detail our BGT model in Sect. 3. The experimental results are shown in Sect. 4 and conclusion is given in Sect. 5.

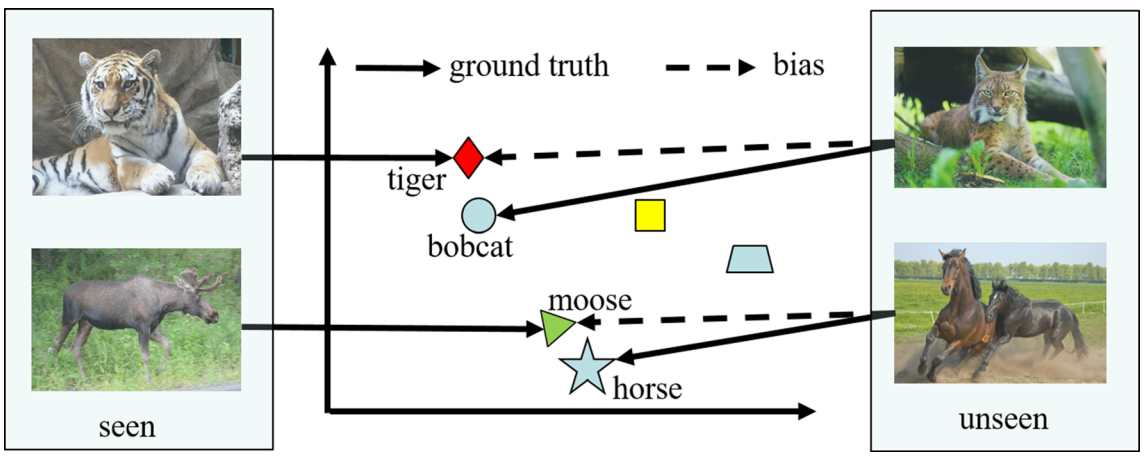


Fig. 1 Visualization of bias problem. The labeled instances are from seen classes at the stage of training, so the unseen instance will often be predicted as a similar seen class. As shown in the figure, the bobcat from unseen classes has a high probability of being predicted as tiger which is from seen classes

2 Related work

Our approach is a transductive method based on GANs, so we will firstly introduces some common transductive methods and then some GAN-based methods. At the same time, the similarities and differences between these methods and ours will be introduced.

2.1 Transductive zero-shot learning

Unlike the standard ZSL, transductive ZSL uses target domain data during training phase to reduce the ubiquitous domain offset problem. It does not violate the "zero-shot" setting because the data from target domain are unlabeled.

The transductive methods use unseen instances in multiple ways. Some methods first train a model with source data, then use the trained model to get pseudo-labels of the instances from target domain. On this basis, they use the obtained pseudo-labels to further adjust the model [3, 15, 34, 46]. Our method follows this research line. It will also use pseudo-labels and unseen instances to further train our model after getting the trained model to make the model more suitable for unseen instances. However, the performance of this approach largely depends on the predictive ability of the trained model on unseen instances. So the unseen instances are also used in the phase of training in our method. We project the visual features of the seen instances to the semantic domain and then project them back to the visual domain. And it is required that the distance between the projected visual feature and the original visual feature should not be too large. In this way, the model also learns how to map the features of the unseen instance without losing information, thus, the model could predict more accurate pseudo-labels on unseen instances. Thereby the overall effect of the model is improved.

Other methods are devoted to making the model more adaptable to the target domain through special training or prediction methods. In [10], classifications of the instances from unseen classes are implemented in two steps. First, canonical correlation analysis (CCA) is used to project visual features and semantic features into a multi-view embedding space, and then unseen unlabeled instances are used to construct a hypergraph to achieve transfer from the seen classes to the unseen classes. Kodirov et al. [20] proposes to use a space shared by the seen and unseen classes to improve the performance of the model on the unseen class. Recently, Verma and Rai [39] proposes to learn the data distribution from the attributes of the seen and unseen classes, and then use unseen instances to adjust the parameters of the distribution.

2.2 Zero-shot learning based on GANs

In recent years, a lot of research related GANs has appeared [14, 27, 28], and GANs has performed well in many scenarios [11, 12, 25, 43]. At the same time, some GAN-based zero-shot learning methods are proposed. In [30, 42, 48], semantic attributes vectors and random noise are used directly to generate unseen instances. But simply using noise and semantic attributes vectors to generate unseen instances has great uncertainty because of GANs property. Tong et al. [38] uses GANs to generate samples with specified semantic attributes vectors to mitigate the bias problem. In order to solve the problem of generative diversity and reliability, LisGAN [24] introduces soul samples and make all generated unseen instances similar to them.

Different from these methods, our method does not use GANs for feature generation, but uses GANs for feature alignment. That is, a bidirectional generation motivated by cycleGAN [47] is used in our method to project visual and semantic information to each other's domain. Then, through adversarial learning, our model makes the projected features and the original features follow a similar distribution and align them.

3 The proposed approach

3.1 Problem definition

Suppose that we have a set of N_s labeled images $D_s = \{(x_i, y_i, z_i)\}_{i=1}^{N_s}$ from C_s seen classes $Y_s = \{1, 2, \ldots, C_s\}$, where $x_i \in X_s \subset \mathbb{R}^{m \times N_s}$ is the visual feature of the ith instance in D_s and m is the dimensionality of visual feature space; $y_i \in Y_s$ is the corresponding label and $z_i \in Z_s \subset \mathbb{R}^{n \times C_s}$ is the corresponding attributes vector where n is its dimension. There is a corresponding relationship between Y_s and Z_s; each column of Z_s represents a semantic attributes vector corresponds to a class in Y_s. We also have a set $D_u = \{(x_j, y_j, z_j)\}_{j=1}^{N_u}$ from C_u unseen classes $Y_u = \{C_s + 1, C_s + 2, \ldots, C_s + C_u\}$, where $x_j \in X_u \subset \mathbb{R}^{m \times N_u}$ is the visual feature of the jth instance in D_u; $y_j \in Y_u$ and $z_j \in Z_u \subset \mathbb{R}^{n \times C_u}$ is the corresponding label and semantic attributes vector. While $y_j \in Y_u$ and $z_j \in Z_u \subset \mathbb{R}^{n \times C_u}$ are unavailable during training. Similarly, there is a corresponding relationship between Y_u and Z_u. The goal of ZSL problem is to learn a function $f : X_u \rightarrow Y_u$. For the GZSL problem, the goal is to learn a function $f : \{X_s, X_u\} \rightarrow Y_s \cup Y_u$. It is worth noting that $Y_s \cap Y_u = \emptyset$. Table 1 shows the main notations used here in after.

Table 1 Notation used in our approach

Notation	Description
N_s	Number of seen instances
N_u	Number of unseen instances
C_s	Number of seen classes
C_u	Number of unseen classes
m	Dimensionality of visual feature space
n	Dimension of semantic attributes vectors
D_s	Sourse dataset
X_s	Seen instance matrix
X_u	Unseen instance matrix
Z_s	Semantic attributes vectors of seen classes
Y_s	Ground truth label set of seen classes
Z_u	Semantic attributes vectors of unseen classes
Y_u	Ground truth label set of unseen classes
λ	Hyper-parameter
r	Self-marking ratio

3.2 Bidirectional generative transductive (BGT) model

3.2.1 Overall idea

The overall framework is shown in Fig. 2. First, the visual features of both the source and target data are extracted by a convolution neural networks as x^s and x^u, respectively. Then semantic attributes vectors are projected into a semantic space by a function Φ approximated by a neural networks. There are two generators G_{va} and G_{av} which map from the visual feature space to the semantic feature space and vice versa, respectively. The fake semantic and visual features a^s_{fake} and x^s_{fake} are generated from the source visual and semantic features x^s and a^s, respectively. Then we judge whether these are fake by the semantic and visual feature domain classifiers D_a and D_v, respectively. By this bidirectional projections, the source visual features are aligned with the semantic features in both the visual and semantic spaces. For the visual feature of the target data, we do similar operations which make the visual features of target data consistent with the semantic features. The implementation process at this stage is summarized in Sect. 3.2.6.

In the test phase, we need to combine the divisions in both the visual and semantic spaces to give final predictions instead of giving judgments only in one space as before. Next we will describe each part of our model in details. In Sect. 3.3 we will show how to make further adjustments to the model using pseudo-attributes vectors of target samples.

3.2.2 Generator loss

Since in the source dataset the visual features have the corresponding semantic features, two generators are designed to realize the bidirectional projections which

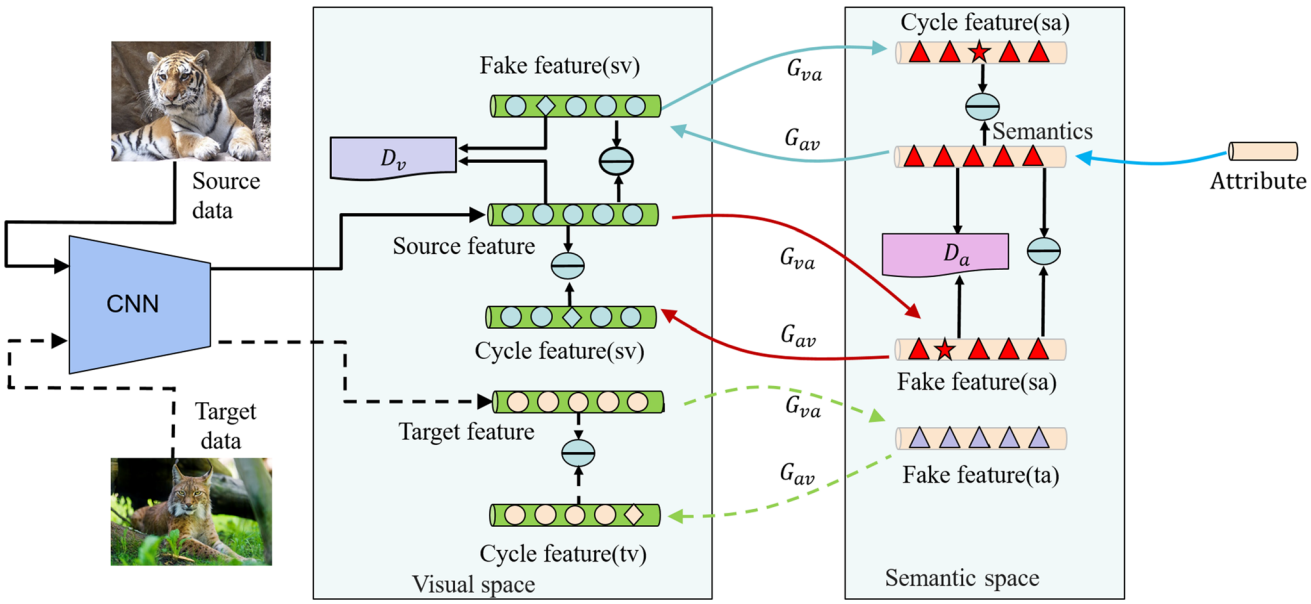


Fig. 2 The overall architecture. The symbol − in the circle is the calculation in Euclidean distance. We train a couple of generative networks to bidirectionally generate visual features from semantic features and projected semantic features from visual features, respectively. In the test phase, by combining the information from the visual and semantic spaces, the category of a target sample is predicted

align visual and semantic features in two spaces. The generator losses are defined as follows:

$$L_{G_{va}} = \frac{1}{N_s}\sum_{i=1}^{N_s} -\log(D_a(G_{va}(x_i^s))), \tag{1}$$

$$L_{G_{av}} = \frac{1}{N_s}\sum_{i=1}^{N_s} -\log(D_v(G_{av}(\Phi(z_i^s)))) \tag{2}$$

where the function Φ is a mapping from the semantic attributes vector z_i^s to the semantic space and x_i^s the visual feature of the ith instance in the source dataset. By defining loss in this way, we can make the generated feature as similar as the original feature in the source domain. In the end, the total loss is

$$L_G = L_{G_{va}} + L_{G_{av}}. \tag{3}$$

3.2.3 Discriminator loss

The discriminators D_a and D_v are used to determine whether the generated feature is real. The losses of discriminators are defined as follows:

$$L_{D_a} = \frac{1}{N_s}\sum_{i=1}^{N_s}(-\log(1 - D_a(G_{va}(x_i^s))) - \log(D_a(\Phi(z_i^s)))), \tag{4}$$

$$L_{D_v} = \frac{1}{N_s}\sum_{i=1}^{N_s}(-\log(1 - D_v(G_{av}(\Phi(z_i^s)))) - \log(D_v(x_i^s))). \tag{5}$$

Through these losses, the discriminator can bee learned to identify that the generated feature is fake, and the original is true. In the end, the total loss is

$$L_D = L_{D_a} + L_{D_v}. \tag{6}$$

3.2.4 Cycle loss

In order to ensure that the generator does not lose important information, cycle loss for the source data is defined as follows:

$$L_{C_a} = \frac{1}{N_s}\sum_{i=1}^{N_s}\left\|\Phi(z_i^s) - G_{va}(G_{av}(\Phi(z_i^s)))\right\|_2, \tag{7}$$

$$L_{C_v} = \frac{1}{N_s}\sum_{i=1}^{N_s}\left\|x_i^s - G_{av}(G_{va}(x_i^s))\right\|_2. \tag{8}$$

At the same time, in order to ensure that these generators have a good performance on the target dataset, we also introduce the cycle loss to the target dataset during the training process, which is defined as follows,

$$L_{C_v^u} = \frac{1}{N_u}\sum_{j=1}^{N_u}\left\|x_j^u - G_{av}(G_{va}(x_j^u))\right\|_2, \tag{9}$$

where x_j^u the visual feature of the jth instance in the target dataset. Finally the final cycle loss is the following:

$$L_C = L_{C_a} + L_{C_v}. \tag{10}$$

We participate in training with L_G and L_C as a whole, thus the combination loss is:

$$L_{GC} = L_G + L_C. \tag{11}$$

3.2.5 Task loss

After ensuring that our generators can do a good job of mapping between the two spaces, we need to further match the visual and semantic features. To achieve accurate classification, we define the task losses in both the visual and semantic spaces, respectively, as follows:

$$L_{T_a} = \frac{1}{N_s}\sum_{i=1}^{N_s}\left\|\Phi(z_i^s) - G_{va}(x_i^s)\right\|_2 + \lambda\|W_{va}\|, \tag{12}$$

$$L_{T_v} = \frac{1}{N_s}\sum_{i=1}^{N_s}\left\|x_i^s - G_{av}(\Phi(z_i^s))\right\|_2 + \lambda\|W_{av}\| \tag{13}$$

where W_{av}, W_{va} are the learning parameters in G_{av} and G_{va}, and λ is a regularization parameter which is a constant. Through the regularization we can effectively reduce the bias problem and improve the effect. In the end, the total loss is defined as follows,

$$L_T = L_{T_a} + L_{T_v}. \tag{14}$$

3.2.6 Training process

The training process is shown in Algorithm 1. The training steps for each epoch are: First we train the two generators and Φ according to the L_T. Then the two generators, discriminator and ϕ are trained according to L_{GC} and L_D. Finally we train the two generators and Φ according to the $L_{C_v^u}$. We repeat the above steps until the model converges.

Algorithm 1 The process of training.

Input:

source dataset: $D_s = \{(x_i, y_i, z_i\}_{i=1}^{N_s}$,
unlabelled instances in target domain: $X_u = \{x_j\}_{j=1}^{N_u}$,
regularization parameter: λ.

Output:

$\{G_{av}^*, G_{va}^*, D_v^*, D_a^*, \Phi^*\}$.

Training:

1: **repeat**
2: updata $\{G_{av}, G_{va}, \Phi\}$ according to Eq. (14) using D_s;
3: updata $\{G_{av}, G_{va}, D_v, D_a, \Phi\}$ according to Eq. (11) using D_s;
4: updata $\{G_{av}, G_{va}, D_v, D_a, \Phi\}$ according to Eq. (6) using D_s;
5: updata $\{G_{av}, G_{va}, \Phi\}$ according to Eq. (9) using X_u;
6: **until** The model convergence.

3.2.7 Classification

Finally, we combine the information in both the visual and semantic spaces to give a prediction, as shown in Fig. 3. For an instance x, its predicted semantic attributes vector at the ZSL setting is

$$\arg\min_{z \in Z_u} \|x - G_{av}(\Phi(z))\|_2 + \|\Phi(z) - G_{va}(x)\|_2. \tag{15}$$

For GZSL, it is

$$\arg\min_{z \in Z_s \cup Z_u} \|x - G_{av}(\Phi(z))\|_2 + \|\Phi(z) - G_{va}(x)\|_2. \tag{16}$$

Since the semantic attributes vector has a clear correspondence with the class, we can get its corresponding label through the semantic attributes vector. So the above formulas can be used for classification. Whether using this classifier is more effective than traditional single-domain classifiers will be discussed further in Sect. 4.4.2.

3.3 Self-labeled strategy

The process of self-labeled strategy is summarized in Algorithm 2. At the first stage, the model is trained using the datasets D_s and X_u. When the model converges, we get

$$\{G_{av}^*, G_{va}^*, D_v^*, D_a^*, \phi^*\} = \arg\min\{L_{GC}, L_D, L_{C_v^u}, L_T\}, \tag{17}$$

where $\{G_{av}^*, G_{va}^*, D_v^*, D_a^*, \phi^*\}$ represents the optimal generators, discriminators, and semantic mapping learned in the training phase. At the second stage, the different strategies for ZSL and GZSL are used. In the following two subsections we will give a detailed introduction.

3.3.1 ZSL

We use $\{G_{av}^*, G_{va}^*, D_v^*, D_a^*, \phi^*\}$ as initial parameters. Then the prediction is performed for the instances in X_u according to Eq. (14), and the predicted semantic attributes vector is referred to as pseudo-semantic attributes vector.

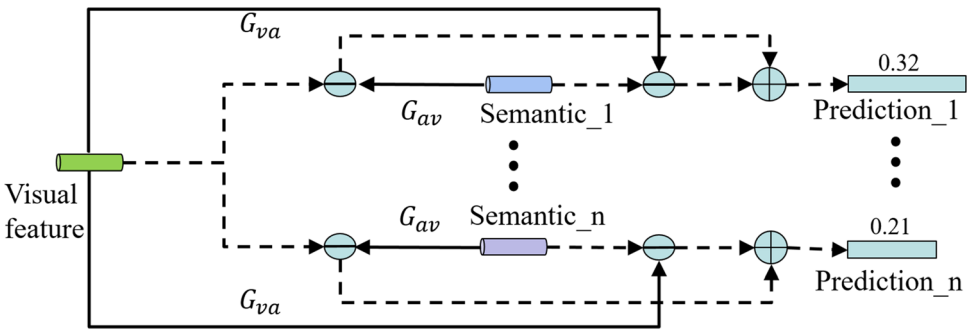


Fig. 3 Visualization of the classifier. Given an instance x, we take a semantic attributes vector z and map z to the semantic space through Φ to get the semantic feature $\Phi(z)$. Then $\phi(z)$ is mapped to the visual domain through the generator G_{av} to get the generated visual feature, then we calculate the distance between it and x. At the same time, we map x to the semantic domain through the generator G_{va} to obtain the generated semantic feature and calculate the distance between it and $\Phi(z)$. Finally, two distances are added to get the score of z for the instance x. We predict the label of instance x as the corresponding class of semantic attributes vector with the lowest score

On this basis we use Eq. (14) to compute the task losses of all instances in target domain, and sort all instances and their pseudo-semantic attributes vectors according to their task losses. After setting a self-marking ratio r, then $r \times N_u$ instances with the smallest task loss are selected and the same number of samples in D_s are replaced. The parameters $\{G^*_{av}, G^*_{va}, D^*_v, D^*_a, \phi^*\}$ are updated according to Eqs. (11) and (14). The above steps are repeated until the training converges.

3.3.2 GZSL

As mentioned earlier in this paper, the instances in the unseen classes are always classified to some seen categories. So we use pseudo-semantic attributes vectors to make adjustment for our model. Since in the adjustment process, the performance of the model will gradually be biased towards the unseen classes. So we further use D_s to update the parameters of $\{G^*_{av}, G^*_{va}, D^*_v, D^*_a, \phi^*\}$ by the losses (11) and (14) after doing the same operations as that in ZSL.

Algorithm 2 The process of self-labeled strategy.

Input:

source dataset: $D_s = \{(x_i, y_i, z_i)\}_{i=1}^{N_s}$,
unlabelled instances: $X_u = \{x_j\}_{j=1}^{N_u}$,
semantic attributes vectors of unseen classes: Z_u,
regularization parameter: λ,
self-marking ratio r_Z for ZSL,
self-marking ratio r_G for GZSL.

Output:

optimal $\{G_{av}, G_{va}, \Phi\}$ for ZSL,
optimal $\{G_{av}, G_{va}, \Phi\}$ for GZSL.

Training for ZSL:

Initialize $\{G_{av}, G_{va}, \Phi\}$ through $\{G^*_{av}, G^*_{va}, \Phi^*\}$;
repeat
  Predict pseudo-attributes vectors according Eq. (15) for all x_j in X_u;
  Calculate the L_t between each instance and their pseudo-attributes vectors according to Eq. (14), and use the resulting L_t as the score for this instance;
  Sort all the scores, select the top $r_Z \times N_u$ instances with lowest scores and their pseudo-attributes vectors to form the set D_p;
  updata $\{G_{av}, G_{va}, \Phi\}$ according to Eq. (14) using D_p;
  updata $\{G_{av}, G_{va}, \Phi\}$ according to Eq. (11) using D_p;
until The model converges again.

Training for GZSL:

Initialize $\{G_{av}, G_{va}, \Phi\}$ through $\{G^*_{av}, G^*_{va}, \Phi^*\}$;
repeat
  Predict pseudo-attributes according Eq. (15) for all x_j in X_u;
  Calculate the L_t between each instance and their pseudo-attributes vectors according to Eq. (14), and use the resulting L_t as the score for this instance;
  Sort all the scores, select the top $r_G \times N_u$ instances with the lowest scores and their pseudo-attributes vectors to form the set D_p;
  updata $\{G_{av}, G_{va}, \Phi\}$ according to Eq. (14) using D_p;
  updata $\{G_{av}, G_{va}, \Phi\}$ according to Eq. (11) using D_p;
  updata $\{G_{av}, G_{va}, \Phi\}$ according to Eq. (14) using D_s;
  updata $\{G_{av}, G_{va}, \Phi\}$ according to Eq. (11) using D_s;
until The model converges again.

4 Experiments

4.1 Datasets and setting

4.1.1 Datasets

AWA2 (Animal With Attribute 2) includes 30,475 instances from 50 classes, 40 of which are used as seen classes and 10 classes as unseen classes, and their semantics are described as 85-dimensional attributes.

aPY (aPascal-aYahoo) includes 15339 instances from 32 classes. We use 20 classes of data as seen classes, and the remaining 12 classes as the unseen classes. Its semantics are described as 64-dimensional attributes.

SUN (SUN Attribute) includes 14,340 instances from 717 categories. Among them, 645 classes are used as the seen classes and 72 classes are used as the unseen classes. Its semantics are described as 102-dimensional attributes.

In this paper, the original image is not used as the training data, but the 2048-dimensional feature extracted by resnet101 [17] pre-trained on ImageNet is used as the visual feature. More details are shown in Table 2. And these datasets have two splits as SS and PS which are same with the previous work [41].

4.1.2 Methods for comparisons

Our method is based on GANs, so we chose some methods that are also based on GANs for comparison. They are: generative adversarial approach for zero-shot learning (GAZSL) [48], Wasserstein GAN with a Classification Loss(f-CLSWGAN) [42], Leveraging invariant side GAN(LisGAN) [24]. At the same time, our method is also a transductive method, so we also selected some transductive methods for comparison. They are: transductive multi-view zero-shot learning (TMV) [10], shared model space (SMS) [16], quasi-fully supervised learning (QFSL) [37]. Some other methods are also selected for comparison which do not have many similarities with ours. Because they have greatly promoted the development of ZSL research and they are often regarded as baselines by other researchers. They are: direct attribute prediction (DAP) [23], deep visual semantic embedding (DEVISE) [9], cross modal transfer (CMT) [36], convex combination of semantic embeddings (CONSE) [31], semantic similarity embedding (SSE) [45], structured joint embedding (SJE) [2], embarrassingly simple approach to zero-shot learning (ESZSL) [33], latent embeddings (LATEM) [40], attribute label embedding (ALE) [2], synthesized classifiers (SYNC) [5], semantic autoencoder (SAE) [21], generative framework for zero-shot learning (GFZSL) [39], deep embedding model (DEM) [44].

These methods use a variety of strategies to accomplish ZSL and GZSL tasks. GAZSL [48] uses Wikipedia to generate features of unseen classes and use these generated features for training. The f-CLSWGAN [42] generates features of unseen classes for training and optimizes the wasserstein distance. LisGAN [24] introduces soul samples to ensure GANs's generation diversity and generation reliability, thereby improves the performance of model. TMV [10] proposes a transductive multiview embedding space to solve the problem of mapping offset and uses the multi-view information of visual features in this space. SMS [16] realizes knowledge transfer by learning model sharing space of multiple models. QFSL [37] uses labeled data to train the relationship between visual information and semantic information, and uses unseen data to reduce bias. DAP [23] learns an attribute probability classifier, and then uses this classifier for classification. DEVISE [9] uses pairwise ranking objective method to make predictions. For the first time, CMT [36] projects the image into semantic space and align it with the class name. CONSE [31] maps the image to semantic space through a convex combination of the label embedding vectors and then aligns them. SSE [45] compares the similarity between visual information and semantic information in visual space and semantic space at the same time. SJE [2] optimizes a structural SVM loss to learn a bilinear compatibility. ESZSL [33] learns a bilinear compatibility and explicitly regularizes the objective frobenius norm using square loss. LATEM [40] extends the SJE [2] to be a piecewise linear mappings. ALE [2] uses a ranking loss to learn a bilinear compatibility function between the visual space and the attributes space. SYNC [5] uses a linear combination of multiple classifiers learned by seen classes to construct an unseen classifier. SAE [21] uses a semantic auto-encoder to reconstruct the image features. GFZSL [39] models each class as a gaussian model, and then learns a regression function to project them into a common space. DEM [44] projects visual information into visual space, and then uses a multi-modality fusion method to combine more semantic information.

4.1.3 Evaluation metrics

We use the similar accuracy evaluation metrics as [41]. For ZSL, the top-1 accuracy average of per-class is computed in the following way,

$$\text{acc}_Y = \frac{1}{|Y|}\sum_{c=1}^{|Y|}\frac{\#\text{correct predictions in } c}{\#\text{samples in } c} \tag{18}$$

where $|Y|$ is the total number of categories at the time of testing. For GZSL, we need to consider the performance on both the seen and unseen classes. The harmonic mean of

Table 2 The details of the datasets we used

Datasets	AWA2	aPY	SUN
#Images	30,475	15,339	14,340
#Seen classes	40	20	645
#Unseen classes	10	12	72
#Attributes	85	64	102

# Indicates the size

accuracies respect to seen and unseen classes is calculated as follows,

$$H_{\text{acc}} = \frac{2 \times \text{acc}_{Y_s} \times \text{acc}_{Y_u}}{\text{acc}_{Y_s} + \text{acc}_{Y_u}} \tag{19}$$

where acc_{Y_s} is the accuracy on the seen data and acc_{Y_u} is the accuracy on the unseen data.

4.2 Implementation

The generators are all composed of three fully connected layers. Each fully connected layer is followed by ReLU layer. All generators from the visual space to the semantic space share weights, and all generators from the semantic space to the visual space share weights. Discriminators also use a three fully connected layers. The first two layers are activated by ReLU function, and the last layer is activated by Sigmod function.

This paper uses the semantic attributes vectors provided by the dataset as auxiliary information. Then, we use the mapping Φ which is composed by a fully connected layer activated by ReLU to map the original attributes vector into the semantic space mentioned before. Then in the training process, training will stop once convergence is achieved, because excessive training will aggravate the bias problem. The regularization coefficient λ in task losses is set to 1e$-$4 when training with AWA or aPY and is set to 0 when training with SUN. The learning rate is set to 1e$-$5, and we use Adam [19] for training.

In the training phase, the generators and discriminators in our GANs and the Φ are trained synchronously. In the phase of training, at each epoch we first use the source dataset for training, and then use the target dataset for training. For the source data for each batch, we first minimize L_{T} and then minimize L_{GC} and then minimize L_{D}. After an epoch, we train the model by minimizing $L_{\text{C}_{\text{v}}^{\text{u}}}$. In the self-labeled phase, we still train the generator and discriminator and Φ at the same time. For ZSL seting, we first replace the corresponding part of D_{s} with the selected pseudo-labeled instances. Then the losses L_{T} and L_{GC} are minimized successively. For GZSL after we do the same operation as ZSL does, and the losses L_{T} and L_{GC} on D_{s} are minimized successively.

4.3 Comparison results

4.3.1 Comparisons at ZSL setting

Table 3 shows that our method has good performance compared with existing methods known to us. When we use SS split for SUN, we achieve similar results to the current best methods. For other situations our method improved by 1.6–14.9% over the best method.

We found that GAN-based ZSL methods such as GFZSL, f-CLSWGAN, LisGAN and our method tend to have better results than traditional embedding methods. This shows that GANs could be a powerful tool for ZSL research. At the same time we can use GANs in many ways in ZSL research. GFZSL, f-CLSWGAN, LisGAN all use GANs to generate visual features, our method uses the characteristics of GANs to achieve flexible alignment between visual and semantic information. These two approaches do not conflict, so combining these two kind of methods may achieve good results.

4.3.2 Comparisons at GZSL setting

Table 4 shows the comparison results. On the dataset AWA and aPY, the H_{acc} of our method is 4.4% and 5.3% higher than the current best method. Our method also achieves good results on the SUN. The main reason for this result is that our model performs well when predicting unseen instances. This shows that generalization ability of our model is better. And we found that many previous methods such as DAP, ESZSL and SAE have a good performance in the traditional ZSL problem but their performance drops sharply in the GZSL problem. Therefore, these models will be greatly limited in practical applications, and our models do not have to worry about this.

We use H_{acc} as the main evaluation index of GZSL, which is a more objective method. It is affected by the prediction accuracy of both the seen and unseen instances. Although some methods such as SAE, GFZSL, DEM, GAZSL guarantee a high prediction accuracy of seen instances, the prediction accuracy of unseen instances is very low. While f-CLSWGAN and LisGAN have achieved high H_{acc}, but in order to obtain higher prediction accuracy on unseen instance, the prediction accuracy on the seen instances is sacrificed. So to sum up, our method can achieve better performance because we consider both two aspects. First of all, we have the structure of cycleGAN to ensure better binding of visual information and semantic information, and introduce the reconstruction process of unseen visual information in the training phase. These

strategies ensure that the knowledge learned from the seen domain could be smoothly transferred to the unseen domain, thereby ensuring that a high prediction accuracy on unseen instances could be obtained. On the other hand, in order not to sacrifice too much prediction accuracy on the seen instances, we adopt a strategy different from the strategy of ZSL setting in the self-labeled stage. While using the unseen instances to adjust the model, the seen instances are also used to adjust the model.

Our model has achieved good performance in the SUN dataset but does not have the same advantages in the AWA and aPY datasets. This is because the number of categories in the SUN dataset is much larger than the other two datasets. So the model has to face more diverse data, and it is more difficult to generate a similar distribution respect to the visual or semantic features.

4.4 Model analysis

4.4.1 Parameter sensitivity

Our model has an important parameter, which is the ratio of the unlabeled instances in the target dataset we used in the self-labeled phase. Figure 4 shows our experimental analysis about this parameter.

At the ZSL setting, the accuracy on AWA is gradually stabilized with the increase of the ratio. For the aPY and SUN datasets, the classification accuracy first increases with the increase of ratio, but after reaching a peak, the classification accuracy decreases with the increase of ratio. This is because the model does not have a particularly good classification ability for the unseen instances in the datasets aPY and SUN. When the ratio is increased to a certain extent, too many false predictions are introduced which may spoil the learned mode. At the GZSL setting, the harmonic mean accuracies have similar phenomenons.

The results show that the optimal ratios are 0.85, 1, 0.8, 0.2, 0.9 and 0.8 for AWA(ss), AWA(PS), aPY(ss), aPY(PS), SUN(SS), SUN(PS) at the ZSL setting, respectively. The optimal ratios are 0.8, 0.8 and 0.9 for AWA, aPY and SUN at the GZSL setting, respectively. However, as shown in Fig. 4, these settings are not absolute, and good results can also be achieved by floating around the optimal ratios.

4.4.2 Significance test

In general, we can not know the generalization accuracy of the model, and we can only approximate the generalization accuracy by the mean value of multiple experimental results. As shown in Tables 3 and 4, the results are the mean values given after many experiments. We assume that there is more than 95% confidence that there is no significant difference between the results given in this paper and the generalization accuracy. In order to verify the hypothesis we put forward, the "Student's t test" is used to verify our hypothesis. Specifically, we conducted 10 repeated experiments and obtained 10 sets of Top-1 accuracy and H_{acc} under ZSL and GZSL for different datasets, respectively. Then we use these data and the results in Tables 3 and 4 to perform a significant test through "Student's t test" with the statistical significance level $\alpha = 0.05$. The results are shown in Table 5.

From Table 5, we can see that the p value of the results on each dataset under ZSL or GZSL is greater than 0.05, which proves that our hypothesis is accurate. That is, the results given in Tables 3 and 4 and the model's generalization accuracy are not significantly different.

Table 3 Top-1 accuracies of different methods on three datasets with two splits

Method	T/I	AWA		aPY		SUN	
		SS	PS	SS	PS	SS	PS
DAP	I	58.7	46.1	35.2	33.8	38.9	39.9
DEVISE	I	68.6	59.7	35.4	39.8	57.5	56.5
CMT	I	66.3	37.9	26.9	28.0	41.9	39.9
CONSE	I	67.9	44.5	25.9	26.9	44.2	38.8
SSE	I	67.5	61.0	31.1	34.0	54.5	51.5
SJE	I	69.5	61.9	32.0	32.9	57.1	53.7
ESZSL	I	75.6	58.6	34.4	38.3	57.3	54.5
LATEM	I	68.7	55.8	34.5	35.2	56.9	55.3
ALE	I	80.3	62.5	30.9	39.7	59.1	58.1
SYNC	I	71.2	46.6	39.7	23.9	59.1	56.3
SAE	I	80.7	54.1	8.3	8.3	42.4	40.3
GFZSL	I	79.3	63.8	51.3	38.4	**62.9**	60.6
DEM	I	–	68.4	–	35.0	–	61.9
GAZSL	I	–	68.2	–	41.4	–	61.3
f-CLSWGAN	I	–	68.2	–	40.5	–	60.8
LisGAN	I	–	70.6	–	43.1	–	61.7
TMV	T	–	–	–	–	61.4	–
SMS	T	–	–	39.0	–	60.5	–
QFSL	T	84.8	79.7	–	–	61.7	58.3
BGT(ours)	T	**95.6**	**82.4**	**57.9**	**49.8**	62.2	**63.5**

The best results are shown in bold. T means the corresponding method is transductive; I means the corresponding method is inductive

4.4.3 Ablation

We adopt a Bidirectional Generative method, and put forward the classifier shown in Sect. 3.2.6 which is called VSC. In order to better verify the effectiveness of our classifier, two kinds of settings are designed to complete the ablation experiment.

Table 4 Comparisons at GZSL setting

Method	Venue	AWA			aPY			SUN		
		ts	tr	H	ts	tr	H	ts	tr	H
DAP	TPAMI, 2013	0	84.7	0	4.8	78.3	9.0	4.2	25.7	7.2
CMT	NIPS, 2013	0.5	90.0	1.0	1.4	85.2	2.8	8.1	21.8	11.8
DEVISE	NIPS, 2013	17.1	74.7	27.8	4.9	76.9	9.2	16.9	27.4	20.9
CONSE	ICLR, 2014	0.5	90.6	1.0	0.0	**91.2**	0.0	6.8	39.9	11.6
SSE	ICCV, 2015	8.1	82.5	14.8	0.2	78.9	0.4	2.1	36.4	4.0
SJE	CVPR, 2015	8.0	73.9	14.4	3.7	55.7	6.9	14.7	30.5	19.8
ESZSL	ICML, 2015	5.9	77.8	11.0	2.4	70.1	4.6	11.0	27.9	15.8
LATEM	CVPR, 2016	11.5	77.3	20.0	0.1	73.0	0.2	14.7	28.8	19.5
ALE	TPAMI, 2016	14.0	81.8	23.9	4.6	73.7	8.7	21.8	33.1	26.3
SYNC	CVPR, 2016	10.0	**90.5**	18.0	7.4	66.3	13.3	7.9	**43.3**	13.4
SAE	CVPR, 2017	1.1	82.2	2.2	0.4	80.9	0.9	8.8	18.0	11.8
GFZSL	ECML, 2017	2.5	80.1	4.8	0.0	83.3	0.0	0.0	39.6	0.0
DEM	CVPR, 2017	30.5	86.4	45.1	11.1	75.1	19.4	20.5	34.3	25.6
GAZSL	CVPR, 2018	19.2	86.5	31.4	14.2	78.6	24.0	21.7	34.5	26.7
f-CLSWGAN	CVPR, 2018	**57.9**	61.4	59.6	32.9	61.7	42.9	42.6	36.6	39.4
LisGAN	CVPR, 2019	52.6	76.3	62.3	34.3	68.2	45.7	**42.9**	37.8	**40.2**
BGT(ours)		56.2	82.2	**66.7**	**39.3**	72.9	**51**	40.2	30.4	34.6

The symbols ts and tr are the top-1 accuracies of the unseen and seen instances, respectively. The symbol H is the harmonic mean respect to the seen and unseen instances. The best results are shown in bold

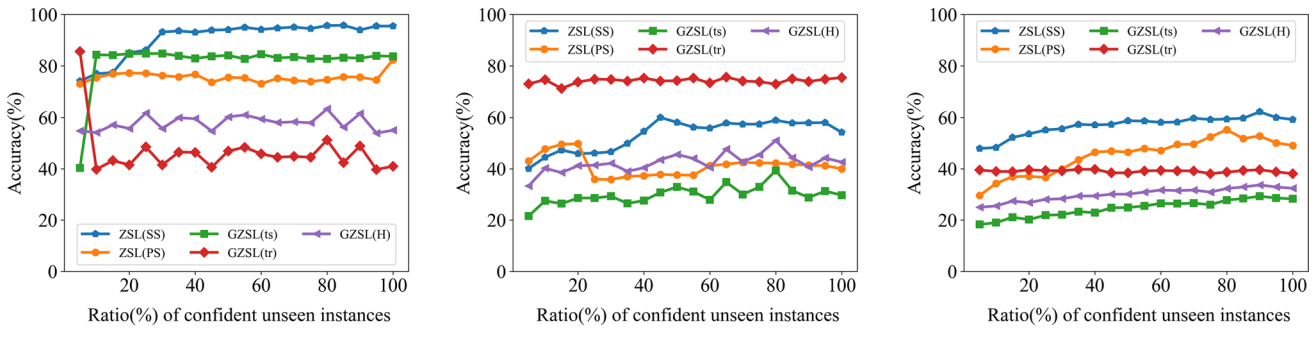
1. The classifier only depends on the distance in the visual domain. As shown in Eq. (20), the classifier in this setting is called VC,

$$\begin{cases} \arg\min_{z\in Z_u} & \|x^t - G_{av}(\Phi(z))\|_2, \quad \text{for} \quad \text{ZSL}, \\ \arg\min_{z\in Z_s\cup Z_u} & \|x^t - G_{av}(\Phi(z))\|_2, \quad \text{for} \quad \text{GZSL}. \end{cases} \tag{20}$$

2. The classifier only depends on the distance in the semantic domain. As shown in Eq. (21), the classifier in this setting is called SC,

$$\begin{cases} \arg\min_{z\in Z_u} & \|\Phi(z) - G_{va}(x^t)\|_2, \quad \text{for} \quad \text{ZSL}, \\ \arg\min_{z\in Z_s\cup Z_u} & \|\Phi(z) - G_{va}(x^t)\|_2, \quad \text{for} \quad \text{GZSL}. \end{cases} \tag{21}$$

The experimental results are shown in Table 6. According to the experimental results, it is found that our classifier is significantly better than those which only use the distance in a single domain as the classification basis, which also proves the opinion mentioned before in



(a) AWA **(b)** aPY **(c)** SUN

Fig. 4 Parameter sensitivity. The horizontal axis represents the proportion of unlabeled instances we used in the self-labeled phase for all unseen instances

Table 5 The result of Student's *t* test

	ZSL						GZSL		
	AWA		aPY		SUN		AWA	aPY	SUN
	SS	PS	SS	PS	SS	PS			
t-value	− 1.44	− 0.35	− 1.31	0.13	− 0.25	− 1.16	0.53	0.06	0.6
p-value	0.18	0.73	0.22	0.89	0.81	0.27	0.61	0.95	0.56

For ZSL we analyze Top-1 accuracies for different datasets, and for GZSL we analyze harmonic mean accuracies

Table 6 Comparison results of different classifiers

classifier	ZSL/GZSL	AWA		aPY		SUN	
		SS	PS	SS	PS	SS	PS
VC	ZSL(top-1)	75.1	69.6	43.7	40.3	46.3	58.4
	GZSL(Hacc)	–	57.3	–	40.3	–	26.6
SC	ZSL(top-1)	78.8	73.6	41.6	35.5	41.4	17.7
	GZSL(Hacc)	–	44.8	–	24.4	–	4.3
VSC	ZSL(top-1)	**95.6**	**82.4**	**59.9**	**49.8**	**62.0**	**63.5**
	GZSL(Hacc)	–	**66.7**	–	**51**	–	**34.6**

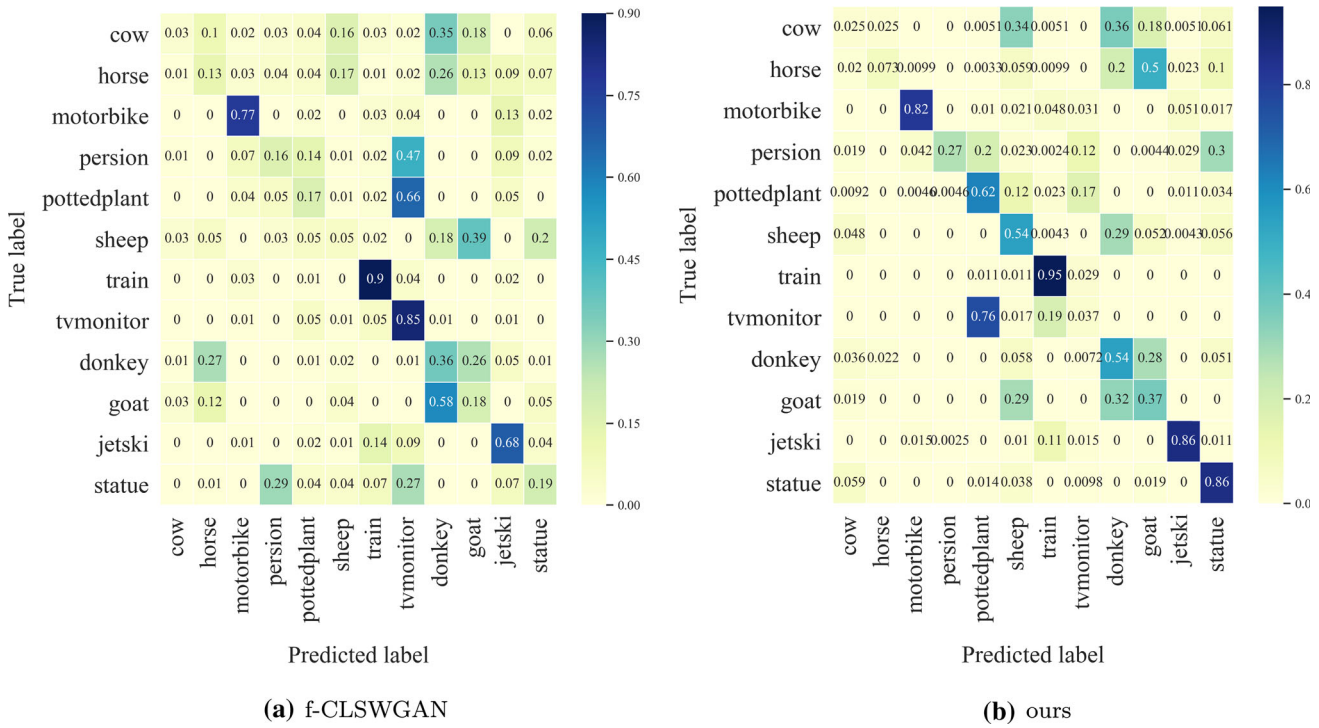
At ZSL setting, the top-1 acc defined by Eq. (18) is shown for different classifiers

At GZSL setting, Hacc defined by Eq. (19) is shown for different classifiers

this paper, i.e., the bidirectional mapping will retain more useful information and better classification results can be obtained by adopting the classifier proposed in this paper.

4.4.4 Class-wise accuracy

To analyze the sensitivity of our model to different categories of images, we analyze classification results of our model for different categories. In order to obtain a more objective evaluation, we choose a same GAN-based model f-CLSWGAN for comparison, which offers confusion matrix in their paper. Figure 5 is the confusion matrix on the aPY dataset. From Fig. 5, we can observe that our method have better performance. Especially for some categories, f-CLSWGAN can not give reasonable judgments,



(a) f-CLSWGAN **(b)** ours

Fig. 5 The confusion matrices on the aPY dataset. The subfigure (**a**) and subfigure (**b**) are the confusion matrices of f-CLSWGAN and our method, respectively

but our method can classify them well. Especially, for potted plant, sheep, statue, the classification accuracies of our method are 45%, 49% and 67% higher than f-CLSWGAN.

There also exists an interesting phenomenon in the experimental results. In terms of ZSL, researchers usually think that misclassification is because two things are visually similar, such as goat and donkey. However, we find that whether using our model or f-CLSWGAN, when classifying tvmonitor and pottedplant, there is always a high probability of misclassifying them as each other. While it is obvious that these two things are not visually similar. The reason for this misclassification is that they usually appear in similar environments, that is, their visual features contain the similar background information. This extra unnecessary background information influences the judgment of our model. So how to eliminate the influence of background information on our model may become one of our future research directions.

5 Conclusions

This paper proposes a zero-shot learning method based on bidirectional projections, which are used to map visual features and semantic features to each other and align their distributions. And it also ensures that no effective information is lost in the mapping process. At the same time, we introduce the cycle loss of unseen unlabeled data in the training process and the predicted pseudo-labels of these samples to correct the model, which greatly alleviates the bias problem in zero-shot learning. Experimental results on three popular datasets show that our method is superior to most of the existing state-of-the-art methods.

Acknowledgements This work was supported in part by the National Key R&D Program of China (2018YFE0203900), National Natural Science Foundation of China (61773093), Important Science and Technology Innovation Projects in Chengdu (2018-YF08-00039-GX) and Research Programs of Sichuan Science and Technology Department (17ZDYF3184).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Akata Z, Perronnin F, Harchaoui Z, Schmid C (2013) Label-embedding for attribute-based classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 819–826
2. Akata Z, Perronnin F, Harchaoui Z, Schmid C (2015) Label-embedding for image classification. IEEE Trans Pattern Anal Mach Intell 38(7):1425–1438
3. Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory. ACM, pp 92–100
4. Bucher M, Herbin S, Jurie F (2016) Improving semantic embedding consistency by metric learning for zero-shot classification. In: European conference on computer vision. Springer, pp 730–746
5. Changpinyo S, Chao WL, Gong B, Sha F (2016) Synthesized classifiers for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5327–5336 (2016)
6. Changpinyo S, Chao WL, Sha F (2017) Predicting visual exemplars of unseen classes for zero-shot learning. In: Proceedings of the IEEE international conference on computer vision, pp 3476–3485
7. Demirel B, Gokberk Cinbis R, Ikizler-Cinbis N (2017) Attributes2classname: a discrimiative model for attribute-based unsupervised zero-shot learning. In: Proceedings of the IEEE international conference on computer vision, pp 1232–1241
8. Farhadi A, Endres I, Hoiem D, Forsyth D (2009) Describing objects by their attributes. In: 2009 IEEE conference on computer vision and pattern recognition, pp 1778–1785
9. Frome A, Corrado GS, Shlens J, Bengio S, Dean J, Ranzato M, Mikolov T (2013) Devise: a deep visual-semantic embedding model. In: Advances in neural information processing systems, pp 2121–2129
10. Fu Y, Hospedales TM, Xiang T, Gong S (2015) Transductive multi-view zero-shot learning. IEEE Trans Pattern Anal Mach Intell 37(11):2332–2345
11. Gan Y, Liu K, Ye M, Zhang Y, Qian Y (2019) Generative adversarial networks with denoising penalty and sample augmentation. In: Neural computing and applications, pp 1–11
12. Gan Y, Liu K, Ye M, Qian Y (2019) Generative adversarial networks with augmentation and penalty. Neurocomputing 360:52–60
13. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680
14. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC (2017) Improved training of Wasserstein gans. In: Advances in neural information processing systems, pp 5767–5777
15. Guo Y, Ding G, Han J, Gao Y (2017) Zero-shot recognition via direct classifier learning with transferred samples and pseudo labels. In: Thirty-First AAAI conference on artificial intelligence (2017)
16. Guo Y, Ding G, Jin X, Wang J (2016) Transductive zero-shot recognition via shared model space learning. In: Thirtieth AAAI conference on artificial intelligence
17. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
18. Karessli N, Akata Z, Schiele B, Bulling A (2017) Gaze embeddings for zero-shot image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4525–4534
19. Kingma DP, Ba J, Adam (2014) A method for stochastic optimization. arXiv preprint arXiv:1412.6980
20. Kodirov E, Xiang T, Fu Z, Gong S (2015) Unsupervised domain adaptation for zero-shot learning, In Proceedings of the IEEE international conference on computer vision, pp: 2452–2460

21. Kodirov E, Xiang T, Gong S (2017) Semantic autoencoder for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3174–3183
22. Lampert CH, Nickisch H, Harmeling S (2009) Learning to detect unseen object classes by between-class attribute transfer. In: 2009 IEEE conference on computer vision and pattern recognition, pp 951–958. IEEE (2009)
23. Lampert CH, Nickisch H, Harmeling S (2013) Attribute-based classification for zero-shot visual object categorization. IEEE Trans Pattern Anal Mach Intell 36(3):453–465
24. Li J, Jing M, Lu K, Ding Z, Zhu L, Huang Z (2019) Leveraging the invariant side of generative zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7402–7411
25. Liu L, Zhang H, Xu X, Zhang Z, Yan S (2019) Collocating clothes with generative adversarial networks cosupervised by categories and attributes: a multidiscriminator framework. IEEE Trans Neural Netw Learn Syst (2019)
26. Lu Y (2015) Unsupervised learning on neural network outputs: with application in zero-shot learning. arXiv preprint arXiv:1506.00990
27. Mao X, Li Q, Xie H, Lau RY, Wang Z, Paul Smolley S (2017) Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2794–2802
28. Mao X, Li Q, Xie, Lau RY, Wang Z Smolley SP (2018) On the effectiveness of least squares generative adversarial networks. IEEE Trans Pattern Anal Mach Intell 41(12):2947–2960
29. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781
30. Mishra A, Krishna Reddy S, Mittal A, Murthy HA (2018) A generative model for zero shot learning using conditional variational autoencoders. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 2188–2196
31. Norouzi M, Mikolov T, Bengio S, Singer Y, Shlens J, Frome A, Corrado GS, Dean J (2013) Zero-shot learning by convex combination of semantic embeddings. arXiv preprint arXiv:1312.5650
32. Qiao R, Liu L, Shen C, Hengel Avd (2017) Visually aligned word embeddings for improving zero-shot learning. arXiv preprint arXiv:1707.05427
33. Romera-Paredes B, Torr P (2015) An embarrassingly simple approach to zero-shot learning. In: International conference on machine learning, pp 2152–2161
34. Saito K, Ushiku Y, Harada T (2017) Asymmetric tri-training for unsupervised domain adaptation. In: Proceedings of the 34th international conference on machine learning, vol 70, pp 2988–2997
35. Shigeto Y, Suzuki I, Hara K, Shimbo M, Matsumoto Y (2015) Ridge regression, hubness, and zero-shot learning. In: Joint European conference on machine learning and knowledge discovery in databases, pp 135–151
36. Socher R, Ganjoo M, Manning CD, Ng A (2013) Zero-shot learning through cross-modal transfer. In: Advances in neural information processing systems, pp 935–943
37. Song J, Shen C, Yang Y, Liu Y, Song M (2018) Transductive unbiased embedding for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1024–1033
38. Tong B, Klinkigt M, Chen J, Cui X, Kong Q, Murakami T, Kobayashi Y (2018) Adversarial zero-shot learning with semantic augmentation. In: Thirty-second AAAI conference on artificial intelligence
39. Verma VK, Rai P (2017) A simple exponential family framework for zero-shot learning. In: Joint European conference on machine learning and knowledge discovery in databases, pp 792–808
40. Xian Y, Akata Z, Sharma G, Nguyen Q, Hein M, Schiele B (2016) Latent embeddings for zero-shot classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 69–77
41. Xian Y, Lampert CH, Schiele B, Akata Z (2018) Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. IEEE Trans Pattern Anal Mach Intell
42. Xian Y, Lorenz T, Schiele B, Akata Z (2018) Feature generating networks for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5542–5551
43. Zhang H, Sun Y, Liu L, Wang X, Li L, Liu W (2018) Clothingout: a category-supervised gan model for clothing segmentation and retrieval. In: Neural computing and applications, pp 1–12
44. Zhang L, Xiang T, Gong S (2017) Learning a deep embedding model for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2021–2030
45. Zhang Z, Saligrama V (2015) Zero-shot learning via semantic similarity embedding. In: Proceedings of the IEEE international conference on computer vision, pp 4166–4174
46. Zhou ZH, Li M (2005) Tri-training: exploiting unlabeled data using three classifiers. IEEE Trans Knowl Data Eng 11:1529–1541
47. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2223–2232
48. Zhu Y, Elhoseiny M, Liu B, Peng X, Elgammal A (2018) A generative adversarial approach for zero-shot learning from noisy texts. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1004–1013

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.