Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 7480712, 13 pages
https://doi.org/10.1155/2021/7480712

Research Article

Attention-Based Graph Convolutional Network for Zero-Shot Learning with Pre-Training

Xuefei Wu,[1] Mingjiang Liu,[1] Bo Xin ,[1] Zhangqing Zhu ,[1] and Gang Wang[2]

[1]*Department of Control and Systems Engineering, School of Management and Engineering, Nanjing University, Nanjing 210093, China*
[2]*Nanjing Research Institute for Agricultural Mechanization, Ministry of Agriculture and Rural Area, Nanjing 210014, China*

Correspondence should be addressed to Bo Xin; xinbo@nju.edu.cn and Zhangqing Zhu; zzqing@nju.edu.cn

Received 1 July 2021; Accepted 10 November 2021; Published 7 December 2021

Academic Editor: Francesco Lolli

Zero-shot learning (ZSL) is a powerful and promising learning paradigm for classifying instances that have not been seen in training. Although graph convolutional networks (GCNs) have recently shown great potential for the ZSL tasks, these models cannot adjust the constant connection weights between the nodes in knowledge graph and the neighbor nodes contribute equally to classify the central node. In this study, we apply an attention mechanism to adjust the connection weights adaptively to learn more important information for classifying unseen target nodes. First, we propose an attention graph convolutional network for zero-shot learning (AGCNZ) by integrating the attention mechanism and GCN directly. Then, in order to prevent the dilution of knowledge from distant nodes, we apply the dense graph propagation (DGP) model for the ZSL tasks and propose an attention dense graph propagation model for zero-shot learning (ADGPZ). Finally, we propose a modified loss function with a relaxation factor to further improve the performance of the learned classifier. Experimental results under different pre-training settings verified the effectiveness of the proposed attention-based models for ZSL.

1. Introduction

Image classification can be viewed as the task to correctly classify the given image into its class. There are many supervised models that have achieved significant success in image classification, such as K-nearest neighbors (KNN) [1] and support vector machines (SVM) [2]. Especially in recent years, deep learning techniques have made great progress in image classification. However, most existing recognition models require a large amount of training samples and can only classify instances belonging to the classes covered by the training data. There are about 30,000 classes that humans can recognize [3], where the workload is quite huge to label all classes and the classes may be growing over time. In contrast, humans are very good at recognizing the unseen classes via reasoning. For example, if we have seen cats and spotted dogs, we will look for an animal called a leopard, which is a cat with spots. Hence, it is important for the agents to acquire the ability of recognizing the unseen classes and zero-shot learning (ZSL) is proposed accordingly.

Zero-shot learning [4] is an inevitable trend of target classification, whose general idea is to transfer the knowledge contained in the training instances to the task of testing instance classification. As no labeled instances belonging to the unseen classes are available, some auxiliary information is necessary to be involved. The auxiliary information involved by the existing ZSL methods is usually some semantic information [5]. Semantic attributes and semantic word vector are two typical semantic information, while we have to learn the mapping from semantic space to visual space when using the two semantic information, which will make it difficult for the model to learn semantic vector representation from structured information.

As a non-Euclidean space data structure, knowledge graph cannot be processed well by the traditional convolutional neural network (CNN). In order to solve this problem, the graph convolutional network (GCN) [6] was proposed with local graph operators. In a GCN, the influence of neighbor nodes on the central node is the same, and the GCN was affected by Laplacian oversmoothing, which makes the GCN a shallow network. In order to solve the problem that the central node can accept the distant node, Michael Kampffmeyer proposed the DGP model [7]. However, there is no good explanation for the contribution of neighbor nodes to the central node. Hence, we apply the attention mechanism to the GCN for enhancing the interpretability of the model and the model can well evaluate the contribution of different neighbor nodes to the central node.

Zero-shot learning aims at recognizing unseen classes by training. Therefore, the classes of testing dataset cannot be included in the training dataset. In recent studies, many models have adopted a pretrained model [8], and we consider whether the pretrained model affects the model. It is clear that when the model is being trained, more samples will help the model test to get better results. In zero-shot learning, we only consider the relationship between the training set and testing set, but do not consider the influence of pre-training. In many models, there are small-scale datasets, such as Animals with Attributes 2 (AWA2) [9] used for the zero-shot learning task, and the model will use the pretrained model of the ImageNet dataset. However, the classes of the ImageNet training set are often more than that of the training classes of AWA2 and other datasets. When we only know a small-scale dataset for zero-shot learning, the task should only be carried out in the training classes of the small-scale dataset. Therefore, we divide the zero-shot learning into three settings, that is, small-scale setting, classifier setting, and large-scale setting, according to the pre-training methods, and integrate the results of the three settings to make the evaluation of the task model more accurate for more practical tasks.

In this article, we proposed the attention-based graph convolutional network for zero-shot learning with pre-training to improve the performance of the task for unseen classes and improve the generalization ability of the model. For the unseen classes, we use the relationship of the classes to establish a connection between the seen classes and the unseen classes. We use knowledge graph as a prior knowledge of agents, which allows the agents to learn to reason. Then, we use the GCN to process the knowledge graphs and train the classifier for the unseen classes. The main contributions of this article are threefold:

We integrate the attention mechanism and graph convolutional network for zero-shot learning. Specifically, we propose two attention-based models, AGCNZ and ADGPZ, to learn adaptive connection weights of the nodes to achieve more accurate predictions.

We present a modified loss function with a relaxation factor, which has a positive effect on the performance.

We have a complete discussion of the setting of ZSL and propose three settings to certify the effect of pre-training for zero-shot learning. Extensive experiments show that the proposed attention-based models can effectively improve the performance of zero-shot learning.

The rest of the article is organized as follows. Section 2 introduces the related work of ZSL. In Section 3, the proposed approach is presented with the overall framework followed by specific algorithms. In Section 4, the three pre-training settings are introduced and the experimental results demonstrate the success of the proposed algorithms. Conclusions are given in Section 5.

2. Preliminaries

Zero-shot learning (ZSL) was first proposed in 2009 [10, 11] and has become one of the important fields of machine learning for that ZSL can identify specific unseen classes and meets the future demand for target recognition. In ZSL, seen classes and unseen classes are connected in a high-dimensional semantic representation space, which includes the attribute space, word vector space, and text description space. The attribute space is firstly introduced in ZSL, where the essential idea is to train a classifier with each attribute of the input, use the trained classifier to predict attributes, and pay more attention to the correlation between learning attributes during the training stage. For example, DAP [12] first estimates the posterior value of each attribute in the image and predicts the class label by learning the probabilistic attribute classifier. Later, for the limitations of the DAP model, Akata et al. [13] introduced a function to measure the compatibility between the image and the label embedding, whose parameters are learned from a set of training samples to ensure that the correct classes rank higher than the incorrect classes in a given image. Li et al. [5] also pay attention to attribute ZSL, and an end-to-end network that automatically discovers discriminative regions by a zoom network and learns the discriminative semantics of user-defined and latent attributes in augmented space is represented.

As for the word vector space, Socher et al. [14] can recognize objects in an image using an unsupervised large text corpus without training data. Frome et al. [8] presented a new deep visual semantic embedding model that uses labeled image data and semantic information extracted from unlabeled text to identify visual objects. Inheriting the DeViSE method, Norouzi et al. [15] proposed a simple method to construct an image embedding system from the existing n-way image classifier as a result of a semantic word embedding model containing n class tags. In the text description [16–18], text description is used to classify unseen classes, and Kodirov et al. [19] proposed to solve the drift of the zero-shot field by using a learning semantic autoencoder (SAE). Wang et al. [20] introduced the GCN in their research, using structured information and complex relationships to generate classifiers for unseen classes. Knowledge graph is a semantic network that represents the relationship between entities, and each class is represented as an entity on the knowledge graph, for example, as shown in Figure 1. In the zero-shot learning semantic representation space, attribute descriptions require attribute annotations and text descriptions require sentence descriptions, and a large number of manual annotations are required. Therefore,

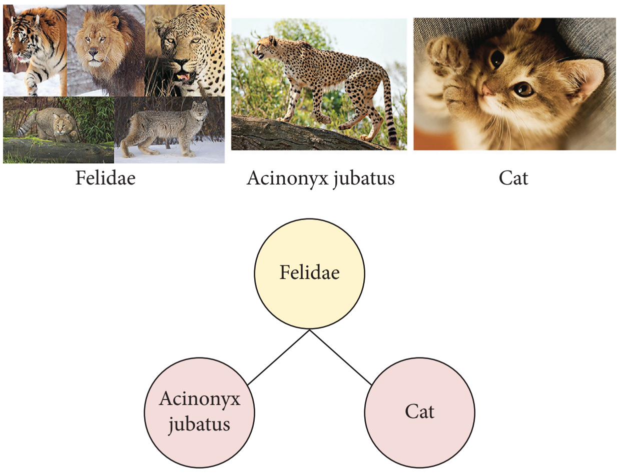


Figure 1: A knowledge graph can be established to represent the relationships between different species.

the cost is relatively high and the advantages shown by the word vector space are considerably attractive.

The graph convolution network (GCN) has become a hot spot of research in recent years. In the GCN, the number of neighbors around the central node is different in non-Euclidean data. Hence, many scholars have begun to study how to deal with graph data structures. A GCN is a kind of the network structure models that can process graph structure data, and the most important part is its convolutional kernel. Like a CNN, the GCN also aims to be able to define convolutions on graphs. Therefore, the essence of the graph convolution is to find a learning convolution kernel suitable for graphs. Bruna et al. [21] first proposed spectral convolutional neural networks. Spectral domain graph convolutional networks implement convolution operations on topological graphs through the theory of graphs, but the method has disadvantages such as computational complexity and nonlocal connection. In addition, Defferrard et al. [22] proposed to fit the convolution kernel using Chebyshev polynomials to reduce computational complexity. Based on the previous works, Kipf and Welling [6] proposed a simple and effective layered propagation method via first-order approximation, which became the pioneering work of the graph convolutional network (GCN). Because of the advantage of the GCN to process the graph data, GCN is gradually applied to a wide range of research fields [23–25] and there are also some studies on graphs, such as Deepwalk [26] and Node2vec [27].

3. Problem Statement

In this section, a schematic framework of the proposed approach is shown in Figure 2 ied loss funct with specific methods of introducing the attention mechanism to different GCN models for the zero-shot learning. In addition, a modifion is also proposed between the predicted classifier and the ground-truth classifier. Then, the algorithms are presented in detail as shown in Algorithms 1 and 2.

3.1. Attention-Based Graph Convolution Network for Zero-Shot Learning.

Here given a graph G, each node on the G represents a category. The adjacency matrix is expressed as $A \in \mathbb{R}^{N\times N}$, which is used to characterize the relation between categories. The propagation formula between GCN layers is defined as

$$H^{(m+1)} = \sigma\left(\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}H^{(m)}W^{(m)}\right), \tag{1}$$

where I is the identify matrix, $\widetilde{A} = A + I$, degree matrix is expressed as $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$, $\sigma(\cdot)$ is the nonlinear activation function, and $W \in \mathbb{R}^{D\times F}$ is the weight matrix with F-feature map in the output layer. $H^{(m)} \in \mathbb{R}^{N\times D}$ is the matrix of activations in m^{th} layer, where N is the number of nodes and D is the feature dimension [6].

In the above formula, each vertex not only has its own neighbor, but also has a self-connection. Laplace smooths the new feature of the vertex, that is, the weighted average of the vertex itself and its neighbors. Because the vertices of the same cluster tend to be more tightly connected, this makes the classification task easier. In GCNs, although using a convolution is already very effective, two-layer GCNs are much better than one-layer GCNs. Because smoothing on the first level of activation makes vertex characteristics in the same category more similar and classification tasks easier. However, as the number of GCN layers increases, the performance will decrease. The reason is that additional Laplace smoothing will be performed as the number of layers increases. Consequently, we can generally use a 2-layer network in this article.

3.1.1. Attention Mechanism.

As an important concept in neural networks, the attention mechanism was first used in machine translation [28]. There are many applications in various fields, such as computer vision [29–31] and natural language processing [28, 32, 33]. Attention mechanism, whether in computer vision or natural language processing, can be classified as giving more attention to the target areas that need to be focused on. In this article, when solving ZSL tasks with knowledge graphs, we represent each category as

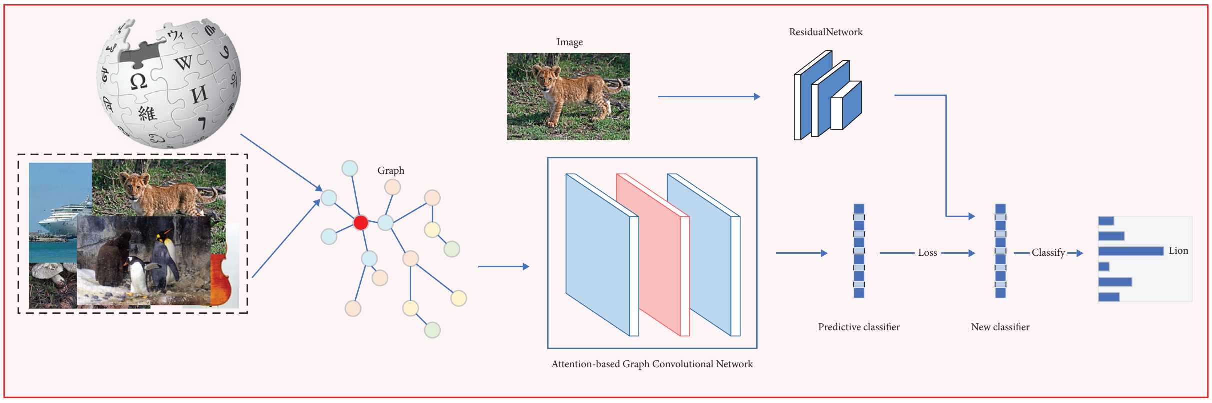


Figure 2: Structure of attention-based graph convolutional network for zero-shot learning. Each node in the knowledge graph is represented by the word embedding vector of each category. The word embedding vector is output by the GloVe model trained in Wikipedia.

each node on the knowledge graph and then use GCN to process the knowledge graph. Therefore, it is very crucial whether the result of GCN processing knowledge graphs can fully express the real situation of each neighbor node's influence on the central node. Therefore, we use cosine distance to calculate the attention of the node [34] and to capture the degree of association between node j and node i, as shown in Figure 3, and then use the improved GCN to process the knowledge graph for ZSL.

3.1.2. Loss Function for Predicted Classifier. A node represents a class in the knowledge graph, and then, we use a word embedding vector for each node. The word embedding vectors of all nodes in the knowledge graph are the input to the graph convolutional network. There are N nodes, M-dimensional vectors, input $X \in \mathbb{R}^{N\times M}$, y is the ground-truth for seen classes, and the loss function [7] can be represented as

$$L = \frac{1}{2N}\sum_{i=1}^{N}\left(GCN_i(X) - y_i\right)^2. \tag{2}$$

The optimized loss function is

$$L = \frac{1}{2N}\sum_{i=1}^{N}\left(GCN_i(X) - y_i + \delta\right)^2, \tag{3}$$

where $GCN_i(X)$ represents the output of the graph convolutional network model and δ is the parameter to adjust the error between the ground-truth and the predicted classifier. We hope to calculate the error of the difference δ at least, where the optimized loss function utilizes a relaxation factor to enhance the generalization ability of the model. We use the ground-truth to train the predicted classifier that can classify unseen classes, add a relaxation factor to enhance the generalization ability of the model, and do not have to be exactly the same as the ground-truth.

3.1.3. Pre-Training Zero-Shot Learning Setting. We propose three pre-training settings for zero-shot learning to better evaluate the model. The architecture of the proposed three pre-training settings is given in Figure 4. We use the ResNet50 [35] model, which has been pretrained on the large-scale dataset. Based on this, for the classifier parameters of the pretrained model, large-scale setting continues to use the classifier parameters of the large-scale dataset, and classifier setting is that we use the classifier parameters of trained by the training set of the small-scale dataset used to test. Small-scale setting is that the training set of the small-scale dataset is trained with the ResNet50 model to get the pretrained model.

3.2. Attention Graph Convolutional Network for Zero-Shot Learning (AGCNZ). In zero-shot learning tasks, we consider the relationship between the training set (seen classes) D_{tr} and the testing dataset (unseen classes) D_{te} in dataset D and $D_{te} \cap D_{tr} = \varnothing$. The ground-truth is trained on the training set to get the classifier parameters. The knowledge graph is established by using the classes of ImageNet and AWA2, which reflects the relationship between each class.

In the GCN, we introduce the attention mechanism and use cosine distance to calculate the similarity between nodes. The propagation formula [34] of the first layer is given as follows:

$$H^{(1)} = \sigma\left(XW^{(0)}\right), \tag{4}$$

where $H^{(0)} = X$. The introduced parameter $\theta^{(l)} \in \mathbb{R}$ in the layer is guided by the attention mechanism, and the rule [34] of AGCNZ propagation for the attention layer is

$$H^{(l+1)} = Att^{(l)}H^{(l)}, \tag{5}$$

where $Att^{(l)}$ is the propagation matrix and l denotes the layer index. The output row vector [34] of node i is recorded as

$$Att_i^{(l)} = soft\max\left(\left[\theta^{(l)}\cos\left(H_i^{(l)}, H_j^{(l)}\right)\right]_{j\in E(i)\cup\{i\}}\right), \tag{6}$$

where $E(i)$ is the neighborhood of node i. In order to ensure that the sum of each row of the propagation matrix is 1, the softmax function is used so that the influence of nodes adjacent to the central node is 1. In summary, the attention [34] between node i and node j is

```
Input: Adjacency matrix A, Number of nodes N, Input node features $X$, Pretrained ResNet50 model classifier parameters $P_p$
Output: Classifier parameter $P_{ag}$, Predicted categories of Unseen classes $Y_{tep}$.
Initializes: the graph convolutional network parameters.
while not converged do
    Update by equation (4);
    for Attention-layer do
        Update by equation (7);
        Update by equation (5);
    end for
    Loss = LossFunction ($P_{ag}$, $P_p$), LossFunction update by equation (2) or (3);
    Loss.backward;
end while
return $P_{ag}$
$Y_{tep}$ is obtained by using $P_{ag}$ as classifier parameter of classification $X_{te}$.
```

Algorithm 1: AGCNZ algorithm.

```
Input: Graph $G$, Number of nodes $N$, Input node characteristics $X$, Pretrained ResNet50 model classifier parameters $P_p$
Output: Classifier parameter $P_{agd}$, Predicted categories of Unseen classes $Y_{tep}$.
Initializes: the graph convolutional network parameters.
Change the Graph $G$ to a dense Graph $G_D$, get the adjacency matrix A.
while not converged do
    Update by equation (4);
    for Attention-layer do
        Update by equation (10);
        Update by equation (9);
    end for
    Loss = LossFunction ($P_{agd}$, $P_p$), Loss Function update by equation (2) or (3);
    Loss.backward;
end while
return $P_{agd}$
$Y_{tep}$ is obtained by using $P_{agd}$ as classifier parameter of classification $X_{te}$.
```

Algorithm 2: ADGPZ algorithm.

$$Att_{ij}^{(l)} = \left(\frac{1}{C}\right) e^{\theta^{(l)} \cos\left(H_i^{(l)}, H_j^{(l)}\right)}, \tag{7}$$

where $C = \sum_{j \in E(i) \cup \{i\}} e^{\theta^{(l)} \cos(H_i^{(l)}, H_j^{(l)})}$. It calculates the similarity between node i and node j, and pays more attention to nodes with more similar central nodes. The AGCNZ algorithm is shown in Algorithm 1. Meanwhile, the architecture of the attention is shown in Figure 5.

3.3. Attention Dense Graph Propagation for Zero-Shot Learning (ADGPZ).

GCN is limited to shallow layer; that is, in the experiment, only two-layer GCN is the best, so the central node cannot receive the information from the remote node. To solve this problem, Kampffmeyer et al. [7] proposed a dense graph propagation (DGP) model to solve this problem and achieved good performances. However, we hope that we can better balance the weight relationship between different neighbors. Because not all edges represent the same degree of association, it is desired to focus on those nodes that are more related to the center node. At this time, the attention mechanism tends to choose those neighbor nodes with the same class as the central node, giving stronger association strength.

Instead of directly processing the knowledge graph with GCN, the DGP model transforms the knowledge graph into a graph in which ancestors and descendants are directly connected with the central node, and then, the dense graph is processed by the GCN. For a given graph, the DGP layer to layer propagation mode [7] is

$$H = \sigma\left(D_a^{-1} A_a \sigma\left(D_d^{-1} A_{d} X W_{d}\right) W_a\right), \tag{8}$$

where $A_a \in \mathbb{R}^{N \times N}$ and $A_{d} \in \mathbb{R}^{N \times N}$ are used to denote the adjacency matrix directly connected to ancestors and descendants, respectively. The ADGPZ propagation rule of the attention layer is

$$H^{(l_{\text{den}}+1)} = Att_a\left(Att_d H^{(l_{\text{den}})}\right), \tag{9}$$

where l_{den} represents the layer index; $Att_{d(ij)}$ represents the attention of descendants:

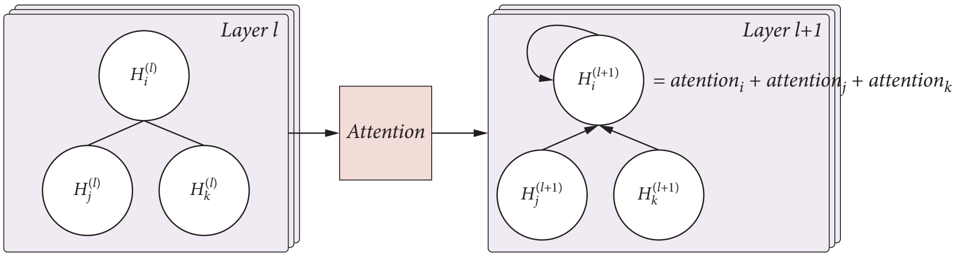


Figure 3: Attention mechanism is used to make the central node obtain different contribution degrees from neighbor nodes.

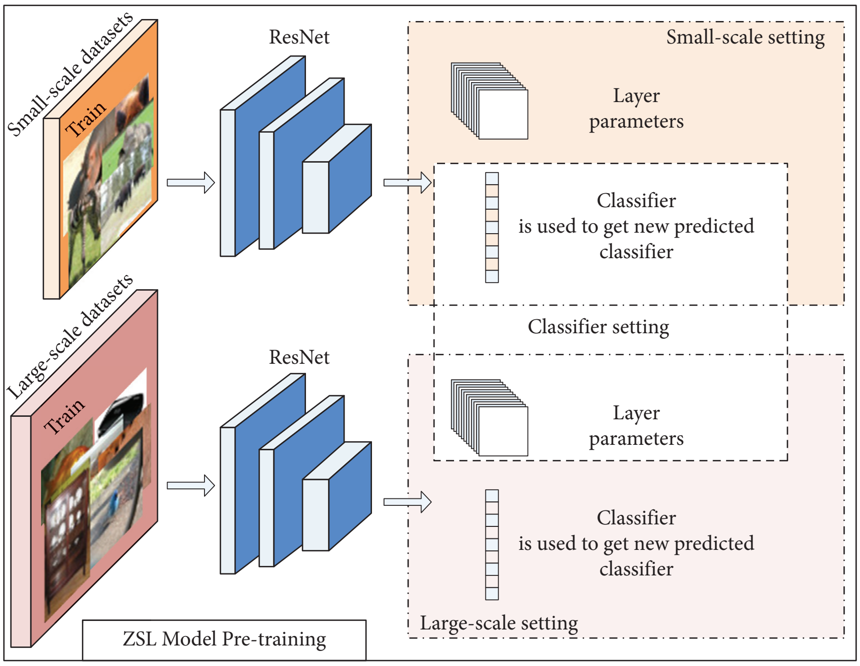


Figure 4: The architecture of the proposed three pre-Training settings for zero-shot learning.

$$Att_{d(ij)} = \left(\frac{1}{C}\right) e^{\theta_d \cos\left(H_{d(i)}, H_{d(j)}\right)}, \qquad (10)$$

where $C = \sum_{j \in E(i) \cup \{i\}} e^{\theta_d \cos(H_{d(i)}, H_{d(j)})}$. In the same way, we can get $Att_{a(ij)} = (1/C) e^{\theta_a \cos(H_{a(i)}, H_{a(j)})}$ and $C = \sum_{j \in E(i) \cup \{i\}} e^{\theta_a \cos(H_{a(i)}, H_{a(j)})}$.

The introduction of attention into the model provides some explanation information. At the same time, the acquired propagation matrix $Att_{d(ij)}$ can also reflect the attention of center node i to neighbor node j in the process of feature aggregation, which represents the influence of node j on node i in the classification process. The ADGPZ algorithm is shown in Algorithm 2, and the architecture of the attention is shown in Figure 6.

4. Experiment

4.1. Datasets. We carried out several groups of experiments on both of large-scale and small-scale datasets. ImageNet dataset [36] contains 140 million images, which are divided into more than 20 000 classes (synsets), including 1000 training sets. We used 2-hops for the test, with 1549 classes. Animals with Attributes 2 (AWA2) [9] contains 50 kinds of animal species, of which 40 species are training sets and 10 species are test sets. The training set contains 29 409 pictures, and the test set contains 7913 pictures. Attribute Pascal and Yahoo (aPY) [9] are 32 classes, 20 classes from Pascal are used as training, and 12 classes are from Yahoo as test. Experimental settings are guaranteed that the ImageNet dataset training set does not contain unseen classes of the testing set and that the classes of the dataset are in the knowledge graph. Three classes from ImageNet and AWA2 are added as a supplement for the unavailable data in the split aPY testing set.

4.2. Experimental Settings. The knowledge graph of the relationship between classes is established by using the ImageNet dataset and AWA2 dataset class names and WordNet. The GloVe [37] text model trained with the Wikipedia dataset represents that each class represents word embedding vectors. In the experiment, we only use a half of the graph without attributes and reconcile the words by

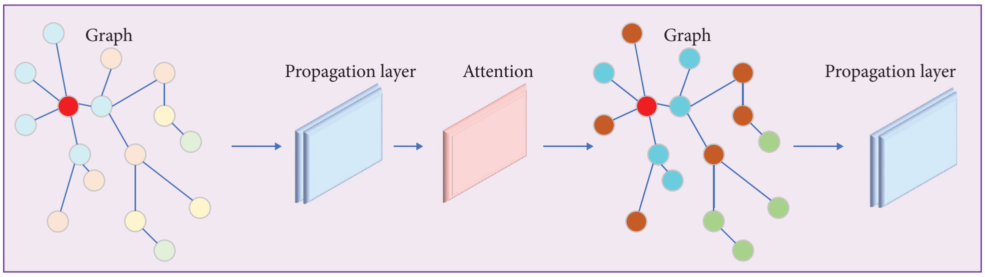


FIGURE 5: Part attention-based graph convolutional network of AGCNZ. The knowledge graph composed of word embedding vectors of each category is used as input through the propagation layer and then output to get the predicted classifier after passing through the attention layer.

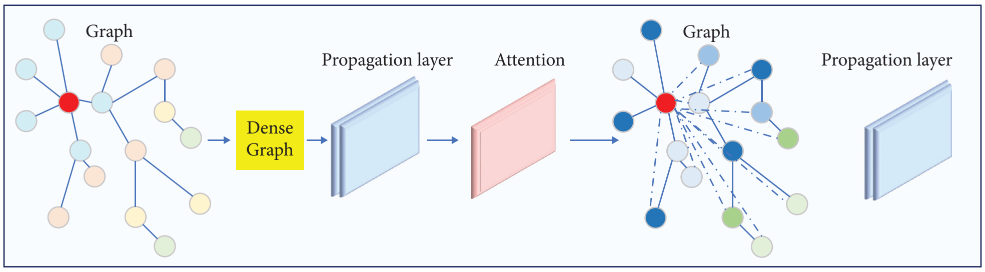


FIGURE 6: Part attention-based graph convolutional network of ADGPZ. Before AGCNZ processes the knowledge graph, ADGPZ has to densify the knowledge graph, that is, connect the ancestors and descendants of each central node of the knowledge graph directly with the central node and then use them as input.

WordNet. Our models are trained and tested in PyTorch using an Adam optimizer [38], the learning rate of 1×10^{-3} and the weight decay of 5×10^{-4}. The nonlinear activation function uses the ReLU function with dropout set to 0.5. We use a two-layer model in the GCN model. In the ADGPZ model and DGP model, we only consider the different effects within 5-hop neighbors on the central node. To better compare and discuss the effect of attention mechanism and pre-training for ZSL in the experiment, the fine-tuning method in this article [7] is not used in the full-text experiment.

4.3. Results on Small-Scale Datasets. The results of all the comparisons are shown in Table 1 and show that our models outperform the baseline and other methods, where the annotation ⋆ means from [9] and † means from [7]. OL and ML stands for original loss and modified loss. These methods use pretrained models that have been trained on the ImageNet datasets. It can be seen from the table that the classification effect is significantly improved in ImageNet setting with the attention mechanism. Our model AGDPZ$_l$ outperforms the best model DGP by 4.8% on the AWA2 dataset, and AGCNZ$_l$ shows better performance on the aPY dataset.

To demonstrate the effectiveness of our methods, we compare the results in different settings. In Tables 2 and 3, all is small-scale setting and classifier is classifier setting on the small-scale dataset. We compared the accuracy of the four methods in the classification of unseen classes under the small-scale setting and classifier setting. No matter AGCNZ$_l$ or AGDPZ$_l$, the classification accuracy of 50.7%, 37.0%, 55.6%, and 39.6% under the two settings is better than that of baseline (43.9% and 36.5%) on the AWA2 dataset. Similar performances can be found for the aPY testing set. The classification accuracy of AGCNZ$_l$ and AGDPZ$_l$(66.8%, 50.8%, 65.6% and 48.3%) is better than that of the baseline method. Among them, the best model is 6.8% better than baseline method.

4.3.1. Effect of Pre-Training on the Model. We further design comparative experiments to demonstrate the effect of pre-training for ZSL. Compared with Tables 1–3, we can find that the accuracy of the ADGPZ$_l$ model can go up to 82.1% on the AWA2 dataset and around 91% on the aPY dataset. Among the three settings, the classification accuracy of the large-scale setting is the best one. The results show that the effect of classifier parameters trained with small-scale datasets is not as good as that of pre-training with the large-scale datasets. The model parameters pretrained with ImageNet training set are actually equivalent to training with 1000 classes. Although the 1000 classes do not contain the same class in the test set, it is clear that the effect on the classification of unseen classes is affected. To more intuitively compare the influence of pre-training on the model, we show it in Figure 7. It is clear that under the three pre-training settings, no matter on the AWA2 dataset or aPY dataset, the classification accuracy of classifier setting is higher than that of small-scale setting, and the classification accuracy of large-scale setting is higher than that of classifier setting. In contrast, different pre-training settings will produce different results for ZSL, which further indicates that pre-training has an impact on the model. In future, it is

Table 1: Top-1 accuracy of different models on the AWA2 and aPY datasets in large-scale setting using the ImageNet dataset.

	Method	AWA2 (%)	aPY (%)
OL	Con SE†	44.5	26.4
	DeViSE †	59.7	37.0
	SSE†	61.0	35.0
	SE-GZSL*	69.2	—
	GCNZ*	70.7	—
	DGP*	77.3	91. 2$_{\text{(ours)}}$
	AGCNZ	79.0	91.2
	ADGPZ	**80.3**	90.4
ML	DGP$_l$	81.7	91.5
	AGCNZ$_l$	78.3	**91.4**
	ADGPZ$_l$	**82.1**	90.6

Table 2: Top-1 accuracy of different models on the AWA2 test set.

Method	Accuracy (%)			
	Original loss		Modified loss	
	Classifier	All	Classifier	All
GCN	38.7	34.8	40.2	35.7
DGP	43.9	36.5	52.2	38.0
AGCNZ	49.7	36.0	50.7	37.0
ADGPZ	50.2	38.4	**55.6**	**39.6**

Table 3: Top-1 accuracy of different models on the aPY test set. To observe the effect of pre-training classes for ZSL, the ImageNet samples of aPY 20 training classes were used as the training set.

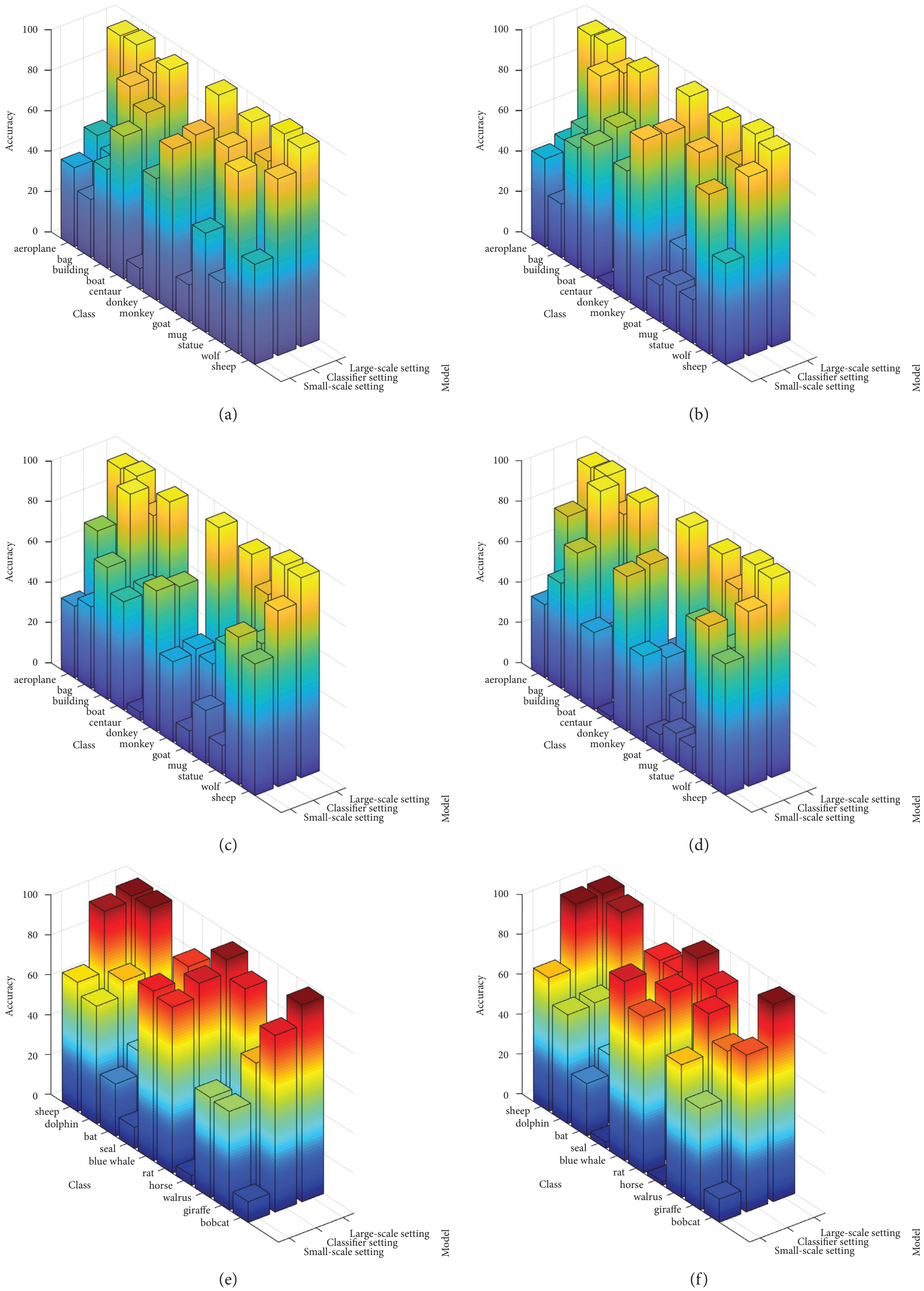
Method	Accuracy (%)			
	Original loss		Modified loss	
	Classifier	All	Classifier	All
GCN	48.0	43.6	64.8	45.4
DGP	60.0	43.5	64.2	47.5
AGCNZ	66.1	49.9	**66.8**	**50.8**
ADGPZ	62.3	45.9	65.6	48.3

necessary to consider different pre-training settings in the evaluation of model competence.

4.3.2. Effect of Modified Loss Function on the Model. In Table 1, when the DGP model used the modified loss function, its classification accuracy is improved by nearly 4%. In all the tables, it is clear that the model with the optimized loss function is better than the original loss function. Without the modified loss function, the accuracy of ADGPZ classification was improved by 3% over the baseline method. The attention mechanism introduced in the baseline method is significantly better than the baseline method as exhibited in Tables 2 and 3. It is proved that when calculating the errors of the predicted classifier and the ground-truth, the modified loss function by adding a parameter to adjust the errors between them to obtain the classifier of the unseen classes can better improve the performance of classification of the unseen classes. The accuracy is also shown in Figure 8. It is clear that on the two datasets, the classification accuracy of the modified loss function of each method is higher than that of the original loss function. The experimental results show that the relaxation factor by introducing the loss function can make the model better classify in ZSL.

4.4. Discussion on Large-Scale Datasets. We further test the proposed models on large-scale datasets, and the experimental results of AGCNZ$_l$ and ADGPZ$_l$ were not as good as those of GCN and DGP, and the experimental results of ADGPZ$_l$ were the worst, shown in Table 4. The reason is that in a large-scale datasets, the number of classes of the training set is more than that of the small-scale dataset. For AGDPZ, its model is the most complex, and the number of added parameters will be more than that of the small-scale datasets; thus, the model overfits. In Table 4, using the model of the modified loss function improves the accuracy of unseen classes, where ‡, $\ast$, and ≀ indicate the results from [20, 39, 40].

4.5. Further Analysis. We further analyzed which parameters were more sensitive to changes using the modified loss function model. We implemented experiments with the learning rate and the weight decay, where the implementation details are kept consistent except for the more important parameters. The experimental results show that the model is more sensitive to changes in the learning rate. Meanwhile, the ADGPZ model is more sensitive on parameter variation, which is due to the more



(a)

(b)

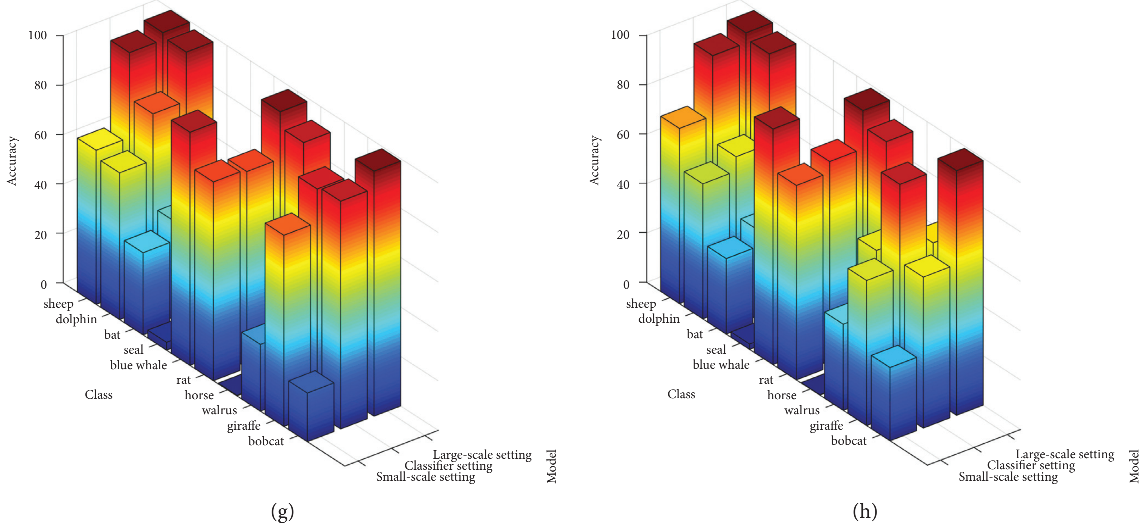
(c)

(d)

(e)

(f)

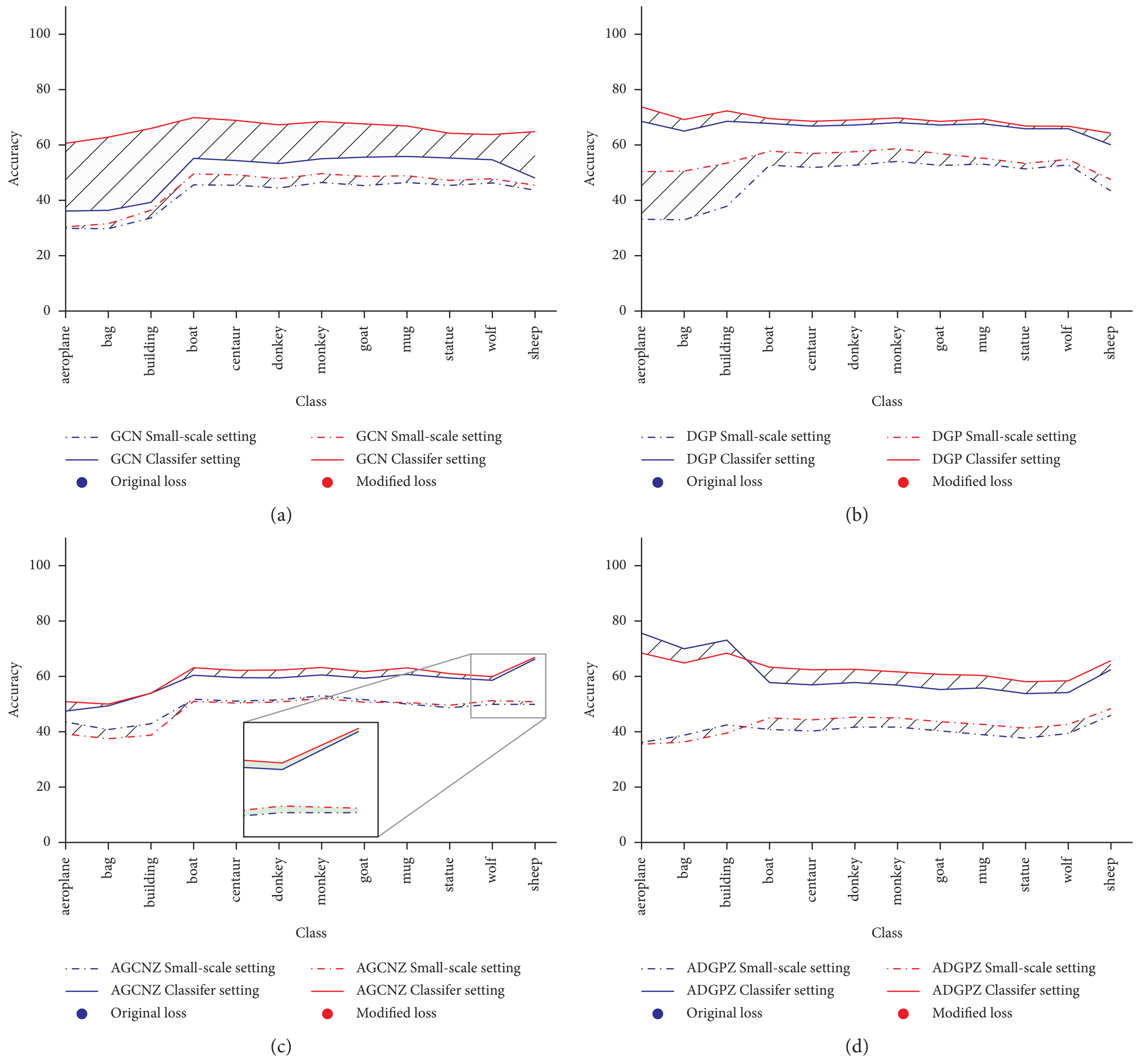
Figure 7: Continued.



(g)

(h)

FIGURE 7: Comparison of classification accuracy of the small-scale setting, classifier setting, and large-scale setting. Among them, (a)–(d) are the results on the aPY dataset, and (e)–(h) are the results on the AWA2 dataset.



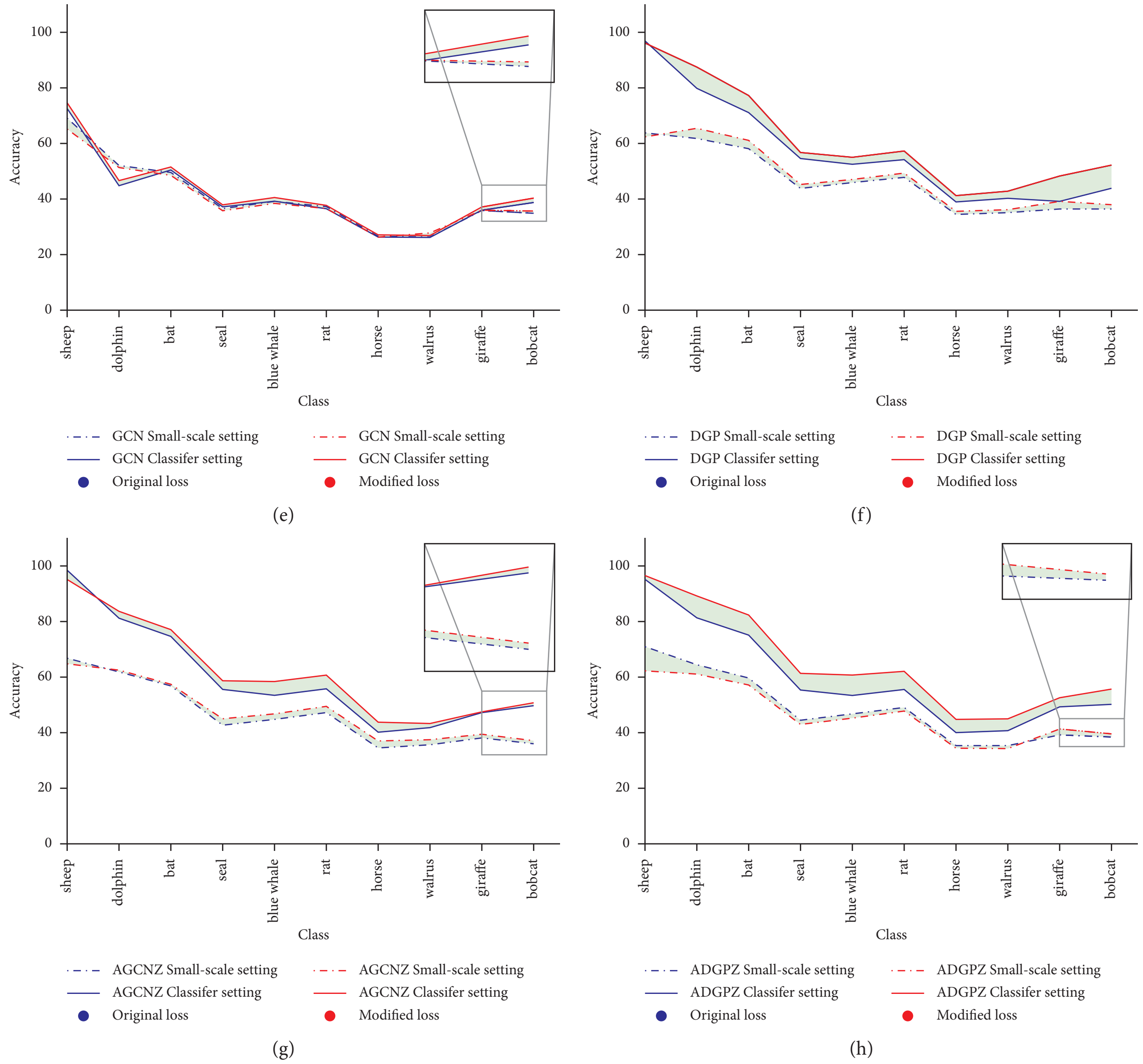
(a)

(b)

(c)

(d)

FIGURE 8: Continued.



(e)

(f)

(g)

(h)

Figure 8: Comparison of classification accuracy of two loss functions. Among them, (a)–8(d) are the results on the aPY dataset, and (e)–(h) are the results on the AWA2 dataset.

Table 4: Top-k accuracy of different models on the ImageNet dataset.

	Method	Accuracy (%)				
		1	2	5	10	20
OL	Con SE‡	8.3	12.9	21.8	30.9	41.7
	SYNC‡	10.5	17.7	28.6	40.1	52.0
	EXEM*	12.5	19.5	32.3	43.7	55.2
	GCNZ$^{\wr}$	19.8	33.3	53.2	65.4	74.6
	GCN$_{\text{(ours)}}$	24.5	37.8	57.1	69.7	79.5
	DGP$_{\text{(ours)}}$	24.6	37.4	56.7	69.5	79.1
	AGCNZ	22.2	34.8	53.6	66.5	77.5
	ADGPZ	20.0	31.6	49.4	63.1	74.3
ML	**GCN**	**24.8**	**38.4**	**57.5**	**69.7**	79.4
	DGP	**24.7**	**38.1**	**57.1**	**69.6**	**79.3**
	AGCNZ	**22.3**	**34.9**	**53.6**	**66.7**	**77.8**
	ADGPZ	**20.2**	**31.7**	**49.3**	**63.2**	**74.8**

complex propagation of the ADGPZ attention layer than AGCNZ.

In the experiments, we found that the effect of unseen classes classifier obtained by using the large-scale dataset pretrained parameters is much better than that obtained by using the small-scale dataset training parameters. Hence, we hold the opinion that using a large number of training samples for the pre-training is more likely to improve the classification of unseen classes.

In real life, we have different application scenarios for large-scale and small-scale datasets. For small-scale datasets, it is enough to identify and classify a specific domain; while for large-scale datasets, it can be applied to a wide range of scenarios. In the experiments, we found that ResNet50 model pretrained with the large-scale dataset has 30% higher classification accuracy for unseen classes than the model trained with the small-scale training set. It is clear that the more classes the agent has seen in the training, the better it can recognize for the unseen classes. More categories stored for the training of the agents may help identify unseen classes for later ZSL tasks in an incremental learning paradigm.

5. Conclusion

In this article, we combine the attention mechanism with GCN, propose two models of AGCNZ and ADGPZ with a modified loss function, and propose three pre-training settings for the zero-shot learning. The experimental results demonstrate the success of the attention mechanism and the proposed models with the modified loss function in three pre-training settings, which is proved to be an influencing factor for evaluating the model in ZSL. Extended experiments also provide more characteristics of the proposed approach with detailed discussion. The emergence of the ZSL task avoids the cost of labeling and training when new categories are added and enables the model to have reasoning ability to recognize unknown categories, which promotes the development of the image recognition. Our future work will consider more ways to improve the loss function, not just by introducing a relaxation factor. We will also focus on more applications of the attention-based GCN aiming at specific fields and algorithm improvement with online adaptation.

Data Availability

The underlying data supporting the results of this study can be found at https://github.com/xf-wu/ZSL.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was partially supported by the National Key Research and Development Program of China (No. 2018AAA0101100) and the National Natural Science Foundation of China (No. 62073160).

References

[1] T. M. Cover, “Rates of convergence for nearest neighbor procedures,” in *Proceedings of the Hawaii International Conference on System Sciences*, Kauai, HI, USA, 1968.

[2] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[3] Z. Ji, H.-R. Wang, Y.-L. Yu, and Y.-W. Pang, “Summary of zero sample image classification: ten years of progress,” *SCIENTIA SINICA Informationis*, vol. 49, no. 10, pp. 1299–1320, 2019.

[4] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, “Learning to compare: relation network for few-shot learning,” in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1199–1208, Salt Lake City, UT, USA, June 2018.

[5] Y. Li, J. Zhang, J. Zhang, and K. Huang, “Discriminative learning of latent features for zero-shot recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7463–7471, Salt Lake City, UT, USA, June 2018.

[6] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France, April 2017.

[7] M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, and E. P. Xing, “Rethinking knowledge graph propagation for zero-shot learning,” in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11487–11496, Long Beach, CA, USA, June 2019.

[8] A. Frome, G. S. Corrado, J. Shlens et al., “Devise: a deep visual-semantic embedding model,” in *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pp. 2121–2129, Lake Tahoe, NV, USA, December 2013.

[9] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, “Zero-shot learning-A comprehensive evaluation of the good, the bad and the ugly,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2251–2265, 2019.

[10] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, “Zero-shot learning with semantic output codes,” in *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pp. 1410–1418, Vancouver, Canada, December 2009.

[11] C. H. Lampert, H. Nickisch, and S. Harmeling, “Learning to detect unseen object classes by between-class attribute transfer,” in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 951–958, Miami, FL, USA, June 2009.

[12] C. H. Lampert, H. Nickisch, and S. Harmeling, “Attribute-based classification for zero-shot visual object categorization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, 2014.

[13] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, “Label-embedding for attribute-based classification,” in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 819–826, Portland, OR, USA, June 2013.

[14] R. Socher, M. Ganjoo, H. Sridhar, O. Bastani, C. D. Manning, and A. Y. Ng, “Zero-shot learning through cross-modal transfer,” in *Proceedings of the International Conference on*

Learning Representations (ICLR), Scottsdale, AZ, USA, May 2013.

[15] M. Norouzi, T. Mikolov, S. Bengio et al., “Zero-shot learning by convex combination of semantic embeddings,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, Canada, April 2014.

[16] L. J. Ba, K. Swersky, S. Fidler, and R. Salakhutdinov, “Predicting deep zero-shot convolutional neural networks using textual descriptions,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4247–4255, Santiago, Chile, December 2015.

[17] M. Elhoseiny, B. Saleh, and A. M. Elgammal, “Write a classifier: zero-shot learning using purely textual descriptions,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2584–2591, Sydney, Australia, December 2013.

[18] S. E. Reed, Z. Akata, H. Lee, and B. Schiele, “Learning deep representations of fine-grained visual descriptions,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 49–58, 2016.

[19] E. Kodirov, T. Xiang, and S. Gong, “Semantic autoencoder for zero-shot learning,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4447–4456, Honolulu, HI, USA, July 2017.

[20] X. Wang, Y. Ye, and A. Gupta, “Zero-shot recognition via semantic embeddings and knowledge graphs,” in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6857–6866, Salt Lake City, UT, USA, June 2018.

[21] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, Canada, April 2014.

[22] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pp. 3837–3845, Barcelona, Spain, December 2016.

[23] F. Monti, O. Shchur, A. Bojchevski, O. Litany, S. Günnemann, and M. M. Bronstein, “Dual-primal graph convolutional networks,” *CoRR*, vol. abs/1806, Article ID 00770, 2018.

[24] H. Gao, Z. Wang, and S. Ji, “Large-scale learnable graph convolutional networks,” in *Proceedings of the 24 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1416–1424, London, UK, August 2018.

[25] Z. Huang, X. Li, Y. Ye, and M. K. Ng, “MR-GCN: multi-relational graph convolutional networks based on generalized tensor product,” in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1258–1264, New York, NY, USA, July 2020.

[26] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: online learning of social representations,” in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, New York, NY, USA, August 2014.

[27] A. Grover and J. Leskovec, “node2vec: scalable feature learning for networks,” in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, San Francisco, CA, USA, August 2016.

[28] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[29] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, “Recurrent models of visual attention,” in *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pp. 2204–2212, Vancouver, Canada, 2014.

[30] F. Wang, M. Jiang, C. Qian et al., “Residual attention network for image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6450–6458, Honolulu, HI, USA, November 2017.

[31] B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan, “Diversified visual attention networks for fine-grained object classification,” *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1245–1256, 2017.

[32] K. Cho, B. van Merrienboer, Ç. Gülçehre et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014.

[33] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pp. 3104–3112, Montreal, Canada, December 2014.

[34] K. K. Thekumparampil, C. Wang, S. Oh, and L. Li, “Attention-based graph neural network for semi-supervised learning,” *CoRR*, vol. abs/1803, Article ID 03735, 2018.

[35] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[36] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, “Imagenet: a large-scale hierarchical image database,” in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, Miami, FL, USA, June 2009.

[37] J. Pennington, R. Socher, and C. D. Manning, “Glove: global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014.

[38] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, May 2015.

[39] S. Changpinyo, W. Chao, B. Gong, and F. Sha, “Synthesized classifiers for zero-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5327–5336, Las Vegas, NV, USA, June 2016.

[40] S. Changpinyo, W. Chao, and F. Sha, “Predicting visual exemplars of unseen classes for zero-shot learning,” in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3496–3505, Venice, Italy, October 2017.