

# Deep Representation of Hierarchical Semantic Attributes for Zero-shot Learning

Zhaocheng Zhang

School of Information

Renmin University of China, Beijing, China

zhaocheng.zhang@ruc.edu.cn

Gang Yang\*

Key Lab of Data Engineering and Knowledge Engineering

Renmin University of China, Beijing, China

yanggang@ruc.edu.cn

**Abstract**—On account of a large scale of dataset need to be annotated to fit for specific tasks, Zero-Shot Learning(ZSL) has invoked so much attention and got significant progress in recent research due to the prevalence of deep neural networks. At present, ZSL is mainly solved through the utilization of auxiliary information, such as semantic attributes and text descriptions. And then, we can employ the mapping method to bridge the gap between visual and semantic space. However, due to the lack of effective use of auxiliary information, this problem has not been solved well. Inspired by previous work, we consider that visual space can be used as the embedding space to get a stronger ability to express the precise characteristics of semantic information. Meanwhile, we take into account that there are some noise attributes in the annotated information of public datasets that need to be processed. Based on these considerations, we propose an end-to-end method with convolutional architecture, instead of conventionally linear projection, to provide a deep representation for semantic information to solve ZSL. Semantic features would express more detailed and precise information after being feed into our method. Besides, we use word embedding to generate some superclasses for original classes and propose a new loss function for these superclasses to assist in training. Experiments show that our method can get decent improvements for ZSL and Generalized Zero-Shot Learning(GZSL) on several public datasets.

**Index Terms**—zero-shot learning, clustering, superclass loss

## I. INTRODUCTION

We human beings have the ability to recognize an animal with some descriptions even though we have never seen it before. This process sounds not difficult for people but not easy for computers. According to the traditional machine learning process, the model can recognize such pictures in the testing stage only after learning this type of picture in the training stage. However, we do not always have the cost to label each type of picture, and there will always be new categories of objects we have never seen before. For solving this situation, [1] proposed Zero-Shot Learning (ZSL). According to this learning pattern, we only need to know some descriptions of a class, and then we can classify it without learning any specific picture.

ZSL consists of two parts, training stage and testing stage, also known as inference. Moreover, the data would be divided into two disjoint groups, seen class and unseen class. In

This work was supported by the Beijing Natural Science Foundation (No. 4192029), and the National Natural Science Foundation of China (61773385, 61672523). Corresponding author: Gang Yang. Email: yanggang@ruc.edu.cn

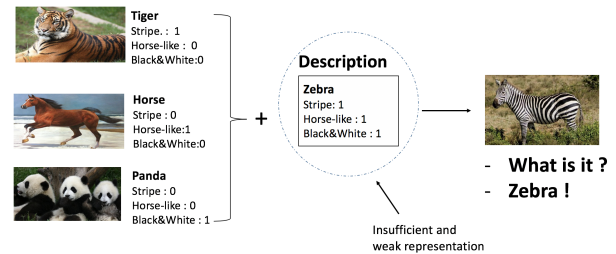


Fig. 1. Usual method in zero-shot learning. Attribute lacking of enough ability to express becomes the main defect in solving this problem.

the training stage, only the images of seen class and their corresponding label are available. Images of unseen class are limited only in the testing stage, where we use a model trained only on seen class to classify unseen class. The most critical part is to use auxiliary information to bridge the gap between seen and unseen class. These types of information are called side-information in this domain, such as semantic attribute and text descriptions. Because whether it is a seen class or an unseen class, we can use an attributes vector to describe it. They share a list of attributes, except that the values within it are different. For example, if the vector of attributes orderly contains stripes, horse-like and black&white, the attributes feature of tiger should be (1,0,0), the horse is (0,1,0), and the panda is (0,0,1).

From this problem was proposed on, two main ways of solving ZSL arose, probability inference and feature projection. [1] proposed to train a separate classifier for each attribute, and use these classifiers to obtain an attributes vector for each test image as its feature in the testing stage, and then compare the similarity with the vector of unseen classes. This type of method uses cross-class attribute and maximum likelihood estimation to figure out the probabilities of unseen classes, which is the basis of probabilistic inference methods. However, it relies too much on the accuracy of attribute prediction, and there is still a large gap between semantic feature and visual feature of instance, which limits the development and application of probabilistic reasoning methods.

With the rapid development of deep neural networks, the use of which to map features has gradually become the mainstream in recent research, and each brought decent improvements

compared to previous work. Regarding the prevailing pattern where the model learns a projection between semantic space and visual space, two major problems exist and hurt the performance of the model. Domain shift [2] and hubness problem [3], [4] are the two crucial problems and need to be solved.

One of the critical points for solving ZSL is the choice of embedding space. Existing works often focus on mapping the visual feature of target images into semantic space then searching for the nearest semantic representation of the unseen class, where the nearest class can be seen as the predicted label. Nevertheless, this type of method still suffers severe hubness problems, which means there are always some class features that will become the hubness of most classes after mapping, which may hurt the performance of ZSL. Unlike conventional mapping of image features to semantic space in SOC [5], EsZSL [6] and [7], DEM [8] proposed to consider the semantic space as the embedding space, which can alleviate the impact of the hubness problem on ZSL to a certain extent. Based on the consideration that the visual space with a much higher dimension has a stronger ability to express the distinguishable feature, we follow this setting in our paper.

A conventional process of addressing the ZSL problem is presented in Figure 1. We need to train our model with images of seen class and only attributes of unseen class. So the scarcity of annotated data is very prominent in our training process. Traditional methods usually use a linear mapping to process features like ALE [9], DeVISE [10], SJE [11] and DEM [8]. However, this linear mapping method cannot essentially optimize the original attributes. [12] argued that in the original labeled data, there are some attributes, such as “inactive”, “smelly” and “solitary”, which may affect the performance on ZSL, because these attributes cannot be clearly expressed on the image features, but are a kind of noise. The method of linear mapping cannot handle these noises well. So we propose to apply a convolutional architecture before the conventionally linear mapping. Our convolutional architecture can filter those weakly correlated attributes with a relu activation function. Meanwhile, those attributes with a strong correlation stand out to make the correct class be isolated from other classes. The hubness problem would be alleviated well in this way.

Besides, we also propose to generate some superclasses for original class with the embedding of class names to assist in training our model. In many classic ZSL works [10], [13], [14], word vectors are used to encode class names, but most of them use it as a feature of the class to map. To our knowledge, we are the first to use word vectors to generate superclasses and assist in training. We think that this processing of class names in a hierarchical manner can make the mapping results more accurate, and the differences between superclasses would be more significant.

Experiments show that our method could achieve superior performance over the state-of-the-art (SOTA) approaches. The main contributions of this paper are summarized as follows:

- We propose a more complicated and practical method to extend the information content of semantic features. This

method makes the semantic feature between classes more distinguishable and representative.

- We consider the centroids of multiple homogeneous classes as the superclasses to get the deep representations of semantic attributes hierarchically and introduce a new loss function into the training phase.

The rest of this paper is organized as follows: we will introduce some related work in the ZSL domain in Section II. Then we formulate the definition of this problem and go into the details of our proposed method in Section III. Details and results of our experiment presented in Section IV. Conclusions and some future work in Section V.

## II. RELATED WORK

### A. ZSL

Zero-shot learning was first put forward by [1], and a classic method DAP was proposed. This method trains a classifier for every attribute once a time, then a test image is inferred by searching the class, which attains the most similar attribute set. This method uses knowledge shared between the seen and unseen classes as a bridge to transfer information and lays the foundation for subsequent research. Moreover, a method [15] like DAP first gets each attribute classifier during the training stage, but estimates the class posteriors through a random forest, which can mitigate those unreliable attributes. This type of probabilistic inference method generally employs the attributes between seen and unseen classes to estimate the maximum likelihood of the probabilities of unseen classes.

So far, with the significant progress of deep neural networks, recent advances in ZSL mainly focus on learning a projection method between visual space and semantic space. SOC [5] maps the image features into the semantic space and then compares and finds the nearest, i.e., the most similar, class embedding vector. DeVISE [10] also learns a linear mapping function between image and semantic attribute, but a designed ranking loss function is added. EsZSL [6] simply uses the square loss to learn the bilinear compatibility and explicitly regularizes the objective function. [16] transfers visual features into the semantic attribute space and then learns a metric to improve the consistency of the semantic embedding. Nevertheless, these methods only connect visual space with semantic space through simple linear mapping and do not deal with attributes effectively. Recently, SAE [17] proposes to enforce the image feature projected to the semantic space to be reconstructed, then gets a semantic autoencoder to regularize its original model. This design allows the mapping of visual features to be trained to be more discriminative during the reconstruction process. While most of the zero-shot learning methods learn the cross-modal mapping between the image and class embedding space with designed losses, there are also some generative models [18]–[21] that represent each class as a probability distribution using GAN [22] or VAE [23] training. These generative models aim at generating pseudo samples for unseen classes from seen classes, making ZSL a supervised task. GFZSL [18] models each class-conditional

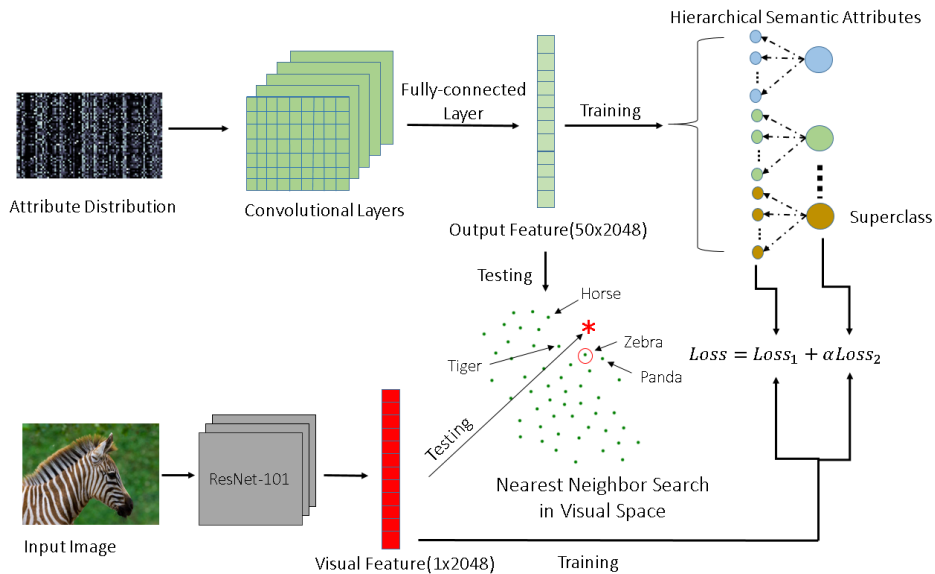


Fig. 2. A whole process of our method. We should transfer the semantic attribute into visual space as same as the dimension of visual feature and then search for the nearest class with the Nearest Neighbor algorithm, considering it as the predicted label for the input image. In the training stage, we obtain some superclasses to calculate another loss between its corresponding feature and the training image feature. Superclass forms the deep representation of hierarchical semantic attributes.

distribution as a Gaussian and learns a regression function that maps a class embedding into the latent space. GLaP [19] assumes that each class’s conditional distribution follows a Gaussian distribution and then generates virtual instances of unseen classes from the learned distribution. [20] acts in a similar way supposing class and visual embeddings of categories are both represented by Gaussian distributions to learn a multimodal projection.

Besides, both *conventional* setting and *generalized* setting are considered as practical in recent work [24], [25]. Conventional setting restricts the search space during testing only consisting of unseen classes, which seems not practical. Therefore generalized setting was proposed in which seen and unseen class are both included during testing. This setting may hurt the performance of the previous ZSL method but often has more practical implications.

### B. Generalized ZSL

Zero-shot learning has been considered as not realistic for being a restrictive set up as it comes with an assumption of the image used during the inference stage can only come from unseen classes. Therefore, generalized zero-shot learning setting [26] was proposed to generalize the zero-shot learning task to the case where both seen and unseen classes are available at test time. Lots of methods on the Generalized ZSL(GZSL) setting have been proposed and get some great results. [27] argues that although the existing method has reached beyond the human performance of the ImageNet classification challenge, we still could not find a generalized method to work at the challenge of object detection. It is a task that not only needs to be able to detect the position and label of the object but also needs to be able to reject unknown

categories. So [28] proposes to use this idea to train an off-line classifier first to classify the input image and then determine whether to use ZSL or GZSL according to the classification result.

## III. PROPOSED METHOD

### A. Problem Formulation

In this paper we formulate the target problem as follows: the seen data  $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$  that consists of  $N^s$  images is used as training set, where  $x_i^s$  is the  $i$ -th image and  $y_i^s \in \mathcal{Y}^s$  is its corresponding label. Similarly the unseen dataset  $\mathcal{U} = \{(x_i^u, y_i^u)\}_{i=1}^{N^u}$  is used as the testing set.

The seen and unseen classes are disjoint, *i.e.*,  $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$ ,  $\mathcal{Y}^s \cup \mathcal{Y}^u = \mathcal{Y}$ . For each  $y \in \mathcal{Y}$ , there is a corresponding attribute vector  $\mathbf{a}_y \in \mathbb{R}^k$ . In the ZSL setting, the target class is restricted only in unseen class where  $\mathcal{Y} = \mathcal{Y}^u$ . While search space is expanded, for GZSL, to which both seen and unseen classes are included where  $\mathcal{Y} = \mathcal{Y}^s \cup \mathcal{Y}^u$ .

### B. Architecture

Compared with previously simply linear projection, we propose to add a newly-designed deep convolutional architecture to map the semantic feature into visual feature space:

$$\mathbf{v}(x) = \mathcal{F}(\mathcal{T}(\mathcal{C}(x))) \quad (1)$$

where  $\mathcal{C}, \mathcal{T}, \mathcal{F}$  are respectively two convolutional layers and one fully connected layer to transform semantic attributes to the same dimensions as visual features. The  $x$  denotes a vector of continuous attribute distribution. Detailed process is presented in Figure 2.

For a given input  $x$ , that is the attribute distribution of one class, the two convolutional layers can heighten the

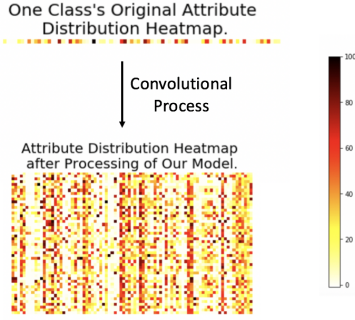


Fig. 3. Heatmap of the intermediate result for one class's attribute distribution. The changes in attribute distribution can be observed. The vertical axis on the right represents the value of the attribute. The darker color equals a higher value of an attribute to this class. The lighter color equals a lower value of an attribute to this class.

dimension of the original attribute to a higher dimension, where these two convolutional approaches can attain more abundant information, and some of the noise attributes would be removed due to the processing of the convolution kernel. In the last stage, there is a fully-connected layer to transfer the flattened vector into the visual space. We can get the target loss by calculating the mean square error between the transferred vector and the training image feature extracted by ResNet-101 [29]. Finally we get an output feature  $v(x)$  in 2048 dimension. We also use a hierarchical process to generalize these features further, which will be explained in Sec III-C later.

We argue that after the processing of these two layers, the input vector, distribution of attributes for one class, will be filtered without those weakly correlated ones. From Figure 3, the upper one is a heatmap of the attribute distribution of one class in AWA2. We can learn that all of the attributes distribute unevenly. Given the attribute vector, only some reliable correlation attributes are kept, and the rest have been removed, like the lower one in Figure 3, which means all of them are set zero.

Through these convolutional processes, the original semantic feature can be regarded that those unrelated attributes would be filtered, which makes our method focus on those strong-related attribute distribution.

### C. Loss Function

The loss function in our method consists of two parts. The first one is the basic  $\mathcal{L}_T$ , which is calculated by the training image  $\mathbf{y}$  and its corresponding transformed class feature  $\mathbf{v}(x)$ , as shown in Equation 2.

$$\mathcal{L}_T = \text{Loss}(\mathbf{v}(x), \mathbf{y}) \quad (2)$$

$\mathcal{L}_T$  allows the features of a class to be projected to the center of the class of pictures in visual space.

Except for the loss between the image feature and class feature mentioned above, we propose a second loss based on the clustering of the class called superclass loss. We use this

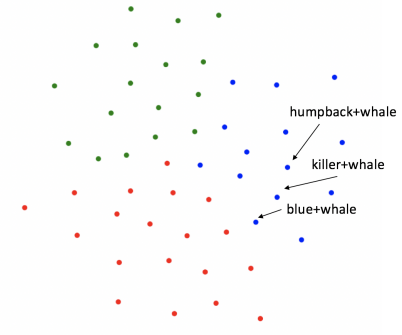


Fig. 4. t-SNE visualization of clustering result with K-means algorithm on AWA2 dataset. Homogeneous classes with similar visual characteristics are divided into one cluster.

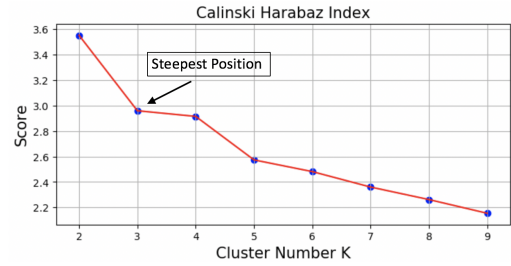


Fig. 5. Evaluation of cluster number with Calinski-Harabaz score on AWA2 dataset. We set the steepest position as our  $K$  number.

hierarchical design to strengthen the training stage so that the learned mapping method can focus more on the region where those homogeneous classes belong, and make the transformed feature space more distinguishable.

We firstly get an embedding for every class name using Word2Vec [30]. Then use K-means [31] algorithm to cluster those embedding of class, naming every new cluster as a superclass. As illustrated in Figure 4, all of 50 classes can be divided into three superclasses, e.g., humpback whale and blue whale are in the same superclass. As for the visual feature, they both are marine organisms, so that the blue background is common in their images. We can get the features of a superclass by calculating the sum and average of all the classes in the cluster.

We use Calinski-Harabaz index score to evaluate the effect in the  $K$  number and then choose the best one.

Supposed we have  $K$  clusters, for cluster  $k = 1, 2, \dots, K$  we define  $N$  - total object number,  $n_k$  - number of objects in cluster  $K$ ,  $c_k$  - centroid,  $C_k$  - indexes of objects.

The covariance matrix within cluster:

$$W = \frac{1}{N - K} \sum_{k=1}^K \sum_{x \in C_k} (x - c_k)(x - c_k)^T \quad (3)$$

And covariance matrix between clusters:

$$B = \frac{1}{K-1} \sum_{k=1}^K n_k (c_k - c) (c_k - c)^T \quad (4)$$

So the Calinski-Harabaz Index score is calculated by:

$$I = \frac{\text{tr}(B)}{\text{tr}(W)} \quad (5)$$

where  $\text{tr}(M)$  denotes the trace of matrix  $M$ .

From Figure 5, we can observe that  $K = 3$  is the steepest position corresponding to the curve. It is natural to find the curve keep descending with an increasing  $K$ , as the score is evaluated on the dispersion degree of classes. However, the steepest place is we need because where gradient closes to zero stands the best trade-off between the cluster number and total quantity.

After we get the clustering result, we can then calculate the feature of one cluster.

$$S_i = \frac{1}{k} \sum_{j=1}^k v(x_{ij}) \quad (6)$$

where  $S_i$  denotes the feature of  $i$ -th superclass and  $x_{ij} \in S_i$ . Supposing  $k$  classes are classified into this superclass, features of this superclass is the the sum and average of all the classes that belongs to it. Then conduct an MSE loss between the initially visual feature and the attained feature.

$$\mathcal{L}_S = \text{Loss}(S_i, y) \quad (7)$$

$\mathcal{L}_S$  could further improve the performance of the transferred feature of classes and effectively alleviates the hubness problem, making the class more inclined to be projected to the center point of its target region and then more distinguishable in visual space.

Finally we get the total loss written as:

$$\mathcal{L} = \mathcal{L}_T + \alpha \mathcal{L}_S \quad (8)$$

where  $\alpha$  is a hyperparameter that controls the weight of our superclass loss.

#### D. Prediction

Through our method, the semantic feature of the unseen class will be projected into the visual space. Then we use the Nearest Neighbor algorithm to search for the most similar class for the test image. To predict the class label  $y^{u*}$ , the index of the maximum compatibility score can be chosen as the predicted label:

$$y^{u*} = \arg \max_c v(x) \quad (9)$$

where  $c \in \mathcal{Y}^U$  in the ZSL setting and  $c \in \mathcal{Y}$  in the GZSL setting.

#### A. Datasets and Settings

We conduct our experiments on 4 widely used datasets for ZSL, AWA2 [32], CUB-200 [33], SUN [34] and aPY [35]. **AWA2**, i.e., Animals with Attributes, consists of 30475 animal images from 50 different classes. Each of them is associating of a continuously 85-D attribute vector. **CUB-200**, i.e., Caltech-UCSD Birds-200-2011, contains a total of 11,788 bird images in 200 categories and 312 binary attributes annotation for every class, which is usually used for image classification and object detection tasks. **SUN**, i.e., Scene UNderstanding research. Like the way how they conducted in [36], we use 645 classes of SUN for training and the other 72 for testing. Each of them has 102 attributes. **aPY**, i.e., Attribute Pascal and Yahoo, is a small-scale dataset, which contains 32 classes with 64 attributes. Twenty of them from Pascal are used for training, and twelve of them from Yahoo for testing.

Considering that the back-end model we use to extract the feature of images is pre-trained based on the ImageNet dataset, and part of them overlap with some unseen classes of the experimental data, we follow the proposed way in [32] to split the training set and test set for all of the four datasets during our experiment to avoid interference with the results in this case.

As for the visual feature, we use ResNet-101 as the back-end model to extract the feature of images. No matter what the dimensions of the original image are, they all are transferred to (3,244,244) format.

Conventional ZSL setting is that only attributes of unseen class can be attained during training, and only unseen class would be predicted during testing. This scene seems not so practical because we do not just classify in categories we have never seen in reality. Therefore generalized ZSL setting was proposed to fit this task into reality, which is both seen and unseen class are used in the testing stage. Except for the conventional ZSL setting, we also conduct our experiments in the GZSL setting to verify the feasibility of our method.

#### B. Hyperparameters and Training Details

After several times of experiments, we found an initial learning rate of 1e-5 and a batch size of 100, only 50 for aPY because it is a small-scale dataset, can get the best performance comparing other hyperparameters.

Another crucial hyperparameter is the number of clusters, which is decided by Calinski-Harabaz score. We use *sklearn* package to draw the score curve on different  $K$  number in Figure 5. Where the slope of the curve changes the most is where the clustering effect is relatively best. Finally, we get  $K = 3$  for AWA2 and aPY,  $K = 4$  for CUB-200, and  $K = 6$  for SUN. Lastly,  $\alpha$  value, which controls the weight of the superclass loss in the total loss, is decided on the performance in our experiments. The specific result is presented in Table II.

We found that most of the four datasets except aPY can achieve better results when  $\alpha$  is about 0.8 and can be improved

TABLE I

PERFORMANCE COMPARISONS IN THE ZSL AND GZSL SETTINGS ON SEVERAL DATASETS, AWA2 [32], CUB-200 [33], SUN AND APY. T1=*Top-1* ACCURACY,  $u$  = Accuracy on  $\mathcal{Y}^U$ ,  $s$  = Accuracy on  $\mathcal{Y}^S$ , AND  $H$  = HARMONIC MEAN. THE BEST NUMBER IS MARKED IN RED. THE SECOND-BEST SCORE IS MARKED IN BLUE. THE RESULTS OF OUR METHOD ARE PRESENTED AT THE LAST TWO ROWS. OURS MEANS THE PROPOSED METHOD, AND **OURS w/S** IS EQUIPPED WITH SUPERCLASS LOSS.

Model	AWA2				CUB-200				SUN				aPY			
	ZSL T1	u	GZSL s	H	ZSL T1	u	GZSL s	H	ZSL T1	u	GZSL s	H	ZSL T1	u	GZSL s	H
DAP	46.1	0.0	84.7	0.0	40.0	1.7	67.9	3.3	39.9	4.2	25.1	7.2	33.8	4.8	78.3	9.0
IAP	35.9	0.9	87.6	1.8	24.0	0.2	<b>72.8</b>	0.4	19.4	1.0	37.8	1.8	36.6	5.7	65.6	10.4
ConSE	44.5	0.5	<b>90.6</b>	1.0	34.3	1.6	<b>72.2</b>	3.1	38.8	6.8	39.9	11.6	26.9	0.0	<b>91.2</b>	0.0
CMT	37.9	8.7	89.0	15.9	34.6	4.7	60.1	8.7	39.9	8.7	28.0	13.3	28.0	10.9	74.2	19.0
SSE	61.0	8.1	82.5	14.8	43.9	8.5	46.9	14.4	51.5	2.1	36.4	4.0	34.0	0.2	78.9	0.4
DeViSE	59.7	17.1	74.7	27.8	52.0	<b>23.8</b>	53.0	32.8	56.5	16.9	27.4	20.9	<b>39.8</b>	4.9	76.9	9.2
SJE	61.9	8.0	73.9	14.4	53.9	23.5	59.2	<b>33.6</b>	53.7	14.7	30.5	19.8	32.9	3.7	55.7	6.9
LATEM	55.8	11.5	77.3	20.0	49.3	15.2	57.3	24.0	55.3	14.7	28.8	19.5	35.2	0.1	73.0	0.2
ESZSL	58.6	5.9	77.8	11.0	53.9	12.6	63.8	21.0	54.5	11.0	27.9	15.8	38.3	2.4	70.1	4.6
ALE	62.5	14.0	81.8	23.9	54.9	<b>23.7</b>	62.8	<b>34.4</b>	58.1	<b>21.8</b>	33.1	<b>26.3</b>	39.7	4.6	73.7	8.7
SYNC	46.6	10.0	<b>90.5</b>	18.0	<b>55.6</b>	11.5	70.9	19.8	56.3	7.9	43.3	13.4	23.9	7.4	66.3	13.3
SAE	54.1	1.1	82.2	2.2	33.3	7.8	54.0	13.6	40.3	8.8	18.0	11.8	8.3	0.4	<b>80.9</b>	0.9
DEM	67.1	<b>30.5</b>	86.4	<b>45.1</b>	51.7	19.6	57.9	29.2	61.9	<b>20.5</b>	34.3	<b>25.6</b>	35.0	11.1	75.1	<b>19.4</b>
<b>Ours</b>	<b>72.6</b>	18.7	88.9	30.9	50.4	18.1	55.7	27.2	<b>62.9</b>	2.2	<b>65.6</b>	4.3	<b>44.1</b>	<b>12.2</b>	60.2	<b>20.2</b>
<b>Ours w/S</b>	<b>72.3</b>	<b>20.7</b>	88.7	<b>33.6</b>	<b>57.8</b>	3.5	71.6	6.6	<b>62.4</b>	7.8	<b>56.6</b>	13.7	39.3	<b>11.0</b>	62.9	18.8

TABLE II

ZSL PERFORMANCE OF THE  $\alpha$  CHOICE. BEST RESULTS ARE MARKED IN BOLD.

$\alpha$	AWA2	CUB	SUN	aPY
0.2	64.4	52.0	61.7	<b>39.3</b>
0.5	64.9	52.4	<b>62.4</b>	27.6
0.8	<b>72.3</b>	<b>57.8</b>	61.5	29.3
1.0	68.3	47.1	62.2	30.9

compared to the case without adding a superclass. On aPY, the addition of superclasses seems only to hurt the performance of our method. As  $\alpha$  increases, that is, the proportion of  $\mathcal{L}_S$  increases, it gradually becomes worse. We think that this is because the categories in aPY are too different. Unlike the other three datasets, there is no visually significant relationship between the categories, which results in poor clustering results. Moreover, in our experiments, only two superclasses were generated from classes of aPY, which may adversely affect the learning performance of our method during training, making the generated class features not distinguishable enough.

### C. Evaluation Metrics

We compare our proposed method with some other recent methods on 4 popular datasets. Top-1 accuracy (ACC) was adopted as the evaluation metric. Besides, to evaluate our method in the GZSL setting, we follow conventional evaluation protocol for GZSL. Suppose that  $ACC_{y^U}$  denotes the ACC for the testing samples only from the unseen classes, and  $ACC_{y^S}$  denotes the ACC for testing samples only from seen classes. Their Harmonic mean  $H$  is calculated as:

$$H = \frac{2 \times ACC_{y^U} \times ACC_{y^S}}{ACC_{y^U} + ACC_{y^S}} \quad (10)$$

### D. The Performance on ZSL

Our experimental results are shown in Table I. We can observe that our method can get the new SOTA result on all of

the four datasets in the conventional ZSL setting. Respectively 5.5% improvement on AWA2, 2.2% improvement on CUB-200, 1.0% on SUN and 4.3% improvement on aPY. This result confirms the feasibility of our method. Comparing the previous method, we assume the reason why our method can get this outstanding result is mainly the reinforcement feature attained by our method. As we can see from Figure 3, our method keeps and enhances those useful attributes and sets those noisy ones as zero, which makes every class’s feature more distinguishable and representative.
















The most significant improvement is on AWA2, 50 classes and 85 attributes in it, and the smallest improvement is on CUB-200, 200 classes and 312 attributes in it. This discrepancy may result from a different number of categories and attributes. The second-largest improvement happens on aPY, another small-scale dataset as same as AWA2. Clearly, under the premise of improvement, our method performs better on small-scale datasets than on large-scale datasets. This situation is evident because more classes mean more difficult to be distinguished in space with the same dimension.

### E. The Performance on GZSL

Unlike ZSL, both seen and unseen classes are included in the search space for GZSL. Our method does not perform as well as conventional ZSL on GZSL but still gets three best results. We use  $u$  and  $s$  to represent the performance of GZSL, which respectively denotes only unseen or seen images used for testing in Table I. Our method performs well in only seen classes included but poorly in only unseen classes included. The result shows that our method has a strong ability to transfer the class feature into visual space accurately. However, on account of no unseen class appearing in the training stage, the projected feature of the unseen class seems not distinguishable and representative. Naturally, the result of the unseen class does not perform well as expected.

TABLE III

TOP FIVE IMAGES CLOSEST TO CLASS FEATURES PROJECTED BY OUR METHOD ON THE AWA2 DATASET. THE NUMBERS BELOW REPRESENT THE EUCLIDEAN DISTANCE BETWEEN FEATURES OF THE PICTURE AND FEATURES OF THE CLASS EMBEDDING. THE CORRECT CATEGORY OF EACH PICTURE IS ALSO MARKED BELOW THE PICTURE. PICTURES NOT IN THIS CATEGORY ARE MARKED IN ITALICS.

Representation of Class	TOP-5 images				
<b>Horse</b>	 19.095 Horse	 20.185 Horse	 20.421 Horse	 20.436 Horse	 20.592 <i>Sheep</i>
<b>Dolphin</b>	 20.597 Dolphin	 21.517 Dolphin	 22.097 Dolphin	 22.111 Dolphin	 22.362 Dolphin
<b>Blue Whale</b>	 18.449 Blue Whale	 19.212 <i>Dolphin</i>	 19.402 <i>Dolphin</i>	 19.435 Blue Whale	 19.727 <i>Dolphin</i>

The hubness problem still exists and affects our results to some extent.

#### F. Superclass Performance

In our setup, we conducted two sets of comparative experiments based on whether or not the superclass loss is added to evaluate its performance.

We got an improvement in six results, three results declined, and others stayed approximately equal ( $\sim 2\%$ ). We think that this design of the superclass gives our method more information to learn and strengthen the ability to induce. A superclass is made up of all the classes classified into it. This further enhances the discrimination of the transferred features. Just like the example we mentioned earlier, humpback whale, blue whale, and dolphin all are marine organisms and belong to the same superclass. Therefore, the characteristics of this superclass will be more generic. Hence the addition of the superclass will make our method tend to map other classes belonging to the class into a specific region to reduce the distance of intraclass and expand the distance of interclass.

To reveal the effect of our method, we randomly select three test classes from the AWA2 dataset and get the mapped features of these classes to search for the closest five images in the test sets, and the results are shown in Table III. We can find that the features mapped by our method are very close to the features of the pictures belonging to this class, that is, they have a very high degree of similarity in visual space. At the same time, we can see that three pictures of dolphins appeared in the top-5 pictures of the blue whale, which also shows that our design for superclasses is feasible and reasonable. It connects these visually consistent class to

enhance the mapping ability of our method, so we can achieve better results when solving the ZSL problem.

#### V. CONCLUSIONS

In this paper, we argue that the visual space has a stronger ability to project class features, so the visual space is used as embedding space in our experiments. Moreover, we replace the conventionally linear projection with a convolutional architecture. The semantic features obtained after mapping can be a more distinguishable representation without the interference of noise attributes. Besides, we propose to use superclasses to represent those classes within the same cluster, and then add a new loss function in the training stage. The hierarchical processing of categories can make the mapped features more distinguishable and the distance between each superclass larger. Our method can get a SOTA result in the ZSL setting and performs well in the GZSL setting. It confirms that our proposed method can indeed enrich the information hidden in semantic attributes, help filter those weakly correlated ones, and alleviate the hubness problem by considering the visual space as the embedding space. Lastly, the superclasses provide a hierarchical architecture to help the output feature be more distinguishable and improve the final performance on several datasets.

In the future, we will continue to move forward based on our present work. We consider adding more modules into our architecture in a positive direction. The popular self-attention mechanism seems to fit the idea of the method proposed in this paper. We consider adding this mechanism to the structure we designed, hoping to achieve better results. Meanwhile, we think that the clustering method applied in our experiments is still too naive. It simply clusters the class

names and does not take into account the more realistic correlation between the classes. At the same time, the feature representation of superclasses is only obtained by summing and averaging, ignoring the correlation between the classes in each superclass. Weighting each class and then obtaining a feature representation for the superclass sounds more reliable. All these will be carried out in our subsequent work.

## REFERENCES

- [1] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 951–958.
- [2] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Transductive multi-view zero-shot learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 11, pp. 2332–2345, 2015.
- [3] A. Lazaridou, G. Dinu, and M. Baroni, "Hubness and pollution: Delving into cross-space mapping for zero-shot learning," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 270–280.
- [4] Y. Shigetou, I. Suzuki, K. Hara, M. Shimbo, and Y. Matsumoto, "Ridge regression, hubness, and zero-shot learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2015, pp. 135–151.
- [5] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Advances in neural information processing systems*, 2009, pp. 1410–1418.
- [6] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International Conference on Machine Learning*, 2015, pp. 2152–2161.
- [7] S. Changpinyo, W.-L. Chao, and F. Sha, "Predicting visual exemplars of unseen classes for zero-shot learning," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3476–3485.
- [8] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2021–2030.
- [9] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1425–1438, 2016.
- [10] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "Devise: A deep visual-semantic embedding model," in *Advances in neural information processing systems*, 2013, pp. 2121–2129.
- [11] Z. Akata, S. E. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2015, pp. 2927–2936.
- [12] Y. Guo, G. Ding, J. Han, and S. Tang, "Zero-shot learning with attribute selection," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018, pp. 6870–6877.
- [13] B. Demirel, R. Gokberk Cinbis, and N. Izkizler-Cinbis, "Attributes2classname: A discriminative model for attribute-based unsupervised zero-shot learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1232–1241.
- [14] O. Russakovsky and F. Li, "Attribute learning in large-scale datasets," in *Trends and Topics in Computer Vision - ECCV 2010 Workshops*, ser. Lecture Notes in Computer Science, K. N. Kutulakos, Ed., vol. 6553. Springer, 2010, pp. 1–14.
- [15] Z. Al-Halah, M. Tapaswi, and R. Stiefelwagen, "Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5975–5984.
- [16] M. Bucher, S. Herbin, and F. Jurie, "Improving semantic embedding consistency by metric learning for zero-shot classification," in *European Conference on Computer Vision*. Springer, 2016, pp. 730–746.
- [17] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3174–3183.
- [18] V. K. Verma and P. Rai, "A simple exponential family framework for zero-shot learning," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2017, pp. 792–808.
- [19] Y. Li and D. Wang, "Zero-shot learning with generative latent prototype model," *arXiv preprint arXiv:1705.09474*, 2017.
- [20] T. Mukherjee and T. Hospedales, "Gaussian visual-linguistic embedding for zero-shot recognition," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 912–918.
- [21] T. Long, X. Xu, Y. Li, F. Shen, J. Song, and H. T. Shen, "Pseudo transfer with marginalized corrupted attribute for zero-shot learning," in *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018*, S. Boll, K. M. Lee, J. Luo, W. Zhu, H. Byun, C. W. Chen, R. Lienhart, and T. Mei, Eds. ACM, 2018, pp. 1802–1810.
- [22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Annual Conference on Neural Information Processing Systems*, 2014, pp. 2672–2680.
- [23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations*, Y. Bengio and Y. LeCun, Eds., 2014.
- [24] V. Kumar Verma, G. Arora, A. Mishra, and P. Rai, "Generalized zero-shot learning via synthesized examples," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4281–4289.
- [25] H. Jiang, R. Wang, S. Shan, and X. Chen, "Learning class prototypes via structure alignment for zero-shot recognition," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 118–134.
- [26] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1757–1772, 2012.
- [27] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *European Conference on Computer Vision*. Springer, 2014, pp. 393–409.
- [28] O. Gune, B. Banerjee, A. More, and S. Chaudhuri, "Generalized zero-shot learning using open set recognition," in *British Machine Vision Conference*, 09 2019.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [31] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl *et al.*, "Constrained k-means clustering with background knowledge," in *Icml*, vol. 1, 2001, pp. 577–584.
- [32] 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*, 2018.
- [33] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.
- [34] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3485–3492.
- [35] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1778–1785.
- [36] 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, 2013.