Pattern Recognition Letters 159 (2022) 125–131

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

# Cross-modal propagation network for generalized zero-shot learning

Ting Guo[a], Jianqing Liang[a,*], Jiye Liang[a], Guo-Sen Xie[b]

[a] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi, China
[b] Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates

ARTICLE INFO

*Article history:*
Received 11 March 2022
Revised 5 May 2022
Accepted 9 May 2022
Available online 11 May 2022

Edited by: Jiwen Lu

*Keywords:*
Zero-shot learning
Generative adversarial network
Meta-learning
Label propagation

ABSTRACT

Zero-shot learning (ZSL) aims to recognize unseen classes by transferring semantic knowledge from seen classes to unseen ones. Since only seen classes are available during training, the domain bias issue, i.e., the trained model is biased toward seen classes, is the key issue for ZSL. To alleviate the bias problem, generation-based approaches are proposed to build generative models that can generate fake visual features of unseen classes by utilizing semantic vectors. However, most of the existing generative methods still suffer some degree of domain bias caused by the ambiguous generation of fake features. In this paper, we propose a cross-modal propagation network (CMPN), which adopts an episode-based meta-learning strategy. CMPN incorporates the adaptive graph construction and label propagation into the generative ZSL framework for guaranteeing an unambiguous and discriminative fake feature generating. By further leveraging the manifold structure of different modalities in the latent space, CMPN can implicitly ensure intra-class compactness and inter-class separation through label propagation classification in latent space. Extensive experiments on four datasets validate the effectiveness of CMPN under both ZSL and generalized ZSL (GZSL) settings.



## 1. Introduction

The success of deep learning [1], which has gained considerable development for various tasks [2,3], relies on the availability of sufficient training data that is annotated by humans. However, in realistic scenarios, some classes have a considerable number of training samples, while others have few or even no training data. Since the collection of labeled data for some classes is labor-intensive and sometimes impossible, zero-shot learning (ZSL) [4] is developed to address the limitations in deep learning. ZSL identifies samples of classes that do not appear in the training set with the aid of semantic information, such as attribute vectors defined manually [5] or word embedding vectors [6]. Semantic information serves as a bridge to connect the seen classes and unseen ones, and the ZSL model aims to recognize unseen classes by transferring semantic knowledge from seen classes to unseen ones.

In recent years, ZSL has received more attention in the field of vision. Most of the ZSL methods [7–9] learn a visual-semantic embedding function that aims to obtain a latent representation space for both visual and semantic. However, generalized ZSL (GZSL), being a more challenging task, tests both seen and unseen class images in the testing stage. Since only seen classes are available in the training stage, unseen test images will have a high probability of being misclassified as seen classes, which we call the domain bias problem. Some methods [10–19] have been proposed to alleviate this problem in GZSL, of which generation-based methods [10–15] have received considerable attention. Generation-based methods mitigate the domain bias problem by training generators to synthesize sufficient fake visual features for unseen classes using semantic vectors. SE-GZSL [10] proposes a feedback-driven mechanism to get an improved generator based on the framework of the conditional variational autoencoder. f-CLSWGAN [11] trains wasserstein generative adversarial networks (WGAN) based on class-level semantic information. CADA-VAE [12] trains variational autoencoder (VAE) to encode and decode visual and semantic features respectively, it also uses a cross-alignment (CA) loss and a distribution aligned (DA) loss to train the generator. f-VAEGAN-D2 [13] combines the advantages of VAE and generative adversarial networks (GAN), learning the marginal feature distribution of unlabeled images by utilizing unconditional discriminator. Considering that previous methods optimize the model based only on seen classes and neglect to explicitly learn to generate fake visual features of unseen ones in the training stage. The idea of episode-based meta-learning [20,21] has inspired the community to solve the domain bias problem in GZSL. ZSML [14] has been proposed to deal with ZSL by exploiting a learning paradigm of meta-learning.

* Corresponding author.
*E-mail address:* liangjq@sxu.edu.cn (J. Liang).

https://doi.org/10.1016/j.patrec.2022.05.009
0167-8655/

It designs episode learning based on model-agnostic meta-learning [21]. The ZSL is simulated in each episode to learn to generate high-quality features with the given semantic vectors. After multiple episodes, the model can gradually accumulate the ability of generating unambiguous fake visual features.

However, previous generative ZSL/GZSL approaches have not considered whether the synthesized fake visual features of unseen classes can guide the classification of real visual features, that is, those methods still fail to guarantee intra-class compactness and inter-class separation in latent space. Thus, the adaptive graph construction and label propagation are incorporated into the generative ZSL model, which is termed as cross-modal propagation network (CMPN). In CMPN, motivated by the assumption that samples with different modalities and the same semantics satisfy the manifold assumption in the latent space, we construct a graph of visual and semantic samples in latent space and implement cross-modal label propagation for classification. In this way, the fake visual features generated by semantic vectors can propagate label information to visual features, achieving cross-modal label propagation. Specifically, we adopt a meta-learning strategy for training. Our main contributions are summarized as follows:

1) We propose a cross-modal propagation network (CMPN) based on meta-learning for advancing the previous generative ZSL/GZSL methods. CMPN guarantees intra-class compactness and inter-class separation in the latent space, which is a common representation space for both visual and semantic modal.
2) CMPN incorporates adaptive graph construction and label propagation into the generative ZSL/GZSL model in the latent space for generating unambiguous and discriminative fake features.
3) CMPN is evaluated using conventional benchmark datasets, i.e. AWA1, AWA2, CUB, and aPY under ZSL and GZSL settings. Extensive experiments on these benchmarks validate the effectiveness of CMPN.

## 2. Related work

### 2.1. Zero-shot learning

Zero-shot learning (ZSL) [4] aims to identify the image of unseen classes, which do not appear in the training stage. The key of ZSL is transferring semantic knowledge from seen classes to unseen ones. Some works [7–9] are dedicated to obtaining a shared embedding space for visual and semantic modal utilizing seen classes. ZSL focuses only on the classification of unseen classes, however, it is more realistic to include both seen and unseen classes in the testing set. Thus generalized ZSL (GZSL) considers both the classification of test seen and test unseen images during the testing. The methods of GZSL are dedicated to solving the bias problem, where unseen test images are often misclassified into seen classes. These works can be divided into generation-based methods [10–15,19] and attribute-based methods [16–18]. Generation-based GZSL trains the generative model based on the seen classes, then generates unseen classes samples utilizing the given semantic vectors, merging the seen class samples to construct a full-observed training dataset. After that, we can train a supervised classification model (e.g., SVM or softmax classifier) to achieve the classification of GZSL. Attribute-based GZSL utilizes class attribute vectors to guide the transformation from visual space to semantic space, increasing the separability of visual features in semantic space. After that, the classifier searches for the class attribute vector with the highest compatibility.

### 2.2. Meta-learning

To address the drawbacks of traditional machine learning where data is rare or expensive unavailable, meta-learning [22] has been proposed to improve learning performance by accumulating experience. The meta-learning model uses a training model of multiple episodes. Meta-learning has advanced the development of deep learning and thus led to the explosion of research on meta-learning recently. Meta-learning has been successfully applied in several domains, which include few shot image recognition [20], reinforcement learning [23], and hyper-parametric optimization [24]. Hospedales et al. [25] mention that meta-learning can improve the generalization of models for a given problem.

### 2.3. Label propagation

The label propagation (LP) [26], a graph-based semi-supervised learning method, which is proposed to predict the label of unlabeled samples. It is an iterative algorithm that builds a graph model by the relationship between samples, each node updates its label guided by the neighboring samples. LP has two assumptions under the semi-supervised learning setting: (i) The smoothness assumption, where neighboring samples have the same label. (ii) The manifold assumption, where points on the same manifold structure have the same labels. As such, when the data satisfies these two assumptions, we can use LP to solve the semi-supervised learning problem. LP has attracted tremendous attention because of its simple and easy implementation, short algorithm execution time, and good classification effect, and has been widely applied to multimedia information classification [27], community mining [28] and other fields. In this paper, we argue that samples with different modalities but the same semantics satisfy the smoothness assumption and manifold assumption in the latent space.

## 3. Methodology

In this section, we introduce a cross-modal propagation network (CMPN). Firstly, we give the problem definitions of ZSL and GZSL. Then, we present the proposed CMPN and illustrate different parts of it.

### 3.1. Problem definition

We first formalize the ZSL and GZSL tasks. Given a training set $\mathcal{D} = \{(x_i, y_i, a(y_i))|x_i \in \mathcal{X}, y_i \in \mathcal{Y}^{\mathcal{S}}, a(y_i) \in \mathcal{A}\}$, $x_i \in \mathcal{R}^{d_1 \times 1}$ is the $d_1$-dimensional visual feature vector and $y_i$ is the label of $x_i$ in the training set. $a(y_i) \in \mathcal{A}$ is the class semantic embedding. In addition, we have a disjoint class label set $\mathcal{U} = \{(y_j, a(y_j))|y_j \in \mathcal{Y}^{\mathcal{U}}, a(y_j) \in \mathcal{A}\}$ of unseen classes, where visual features are missing. It is well known that, $\mathcal{Y}^{\mathcal{S}} \bigcap \mathcal{Y}^{\mathcal{U}} = \emptyset$. The goal of ZSL is to learn a prediction: $\mathcal{X} \rightarrow \mathcal{Y}^{\mathcal{U}}$. In a generalizd setting, the test images come from both seen and unseen classes. With $\mathcal{D}$ and $\mathcal{U}$, we learn a prediction: $\mathcal{X} \rightarrow \mathcal{Y}^{\mathcal{S}} \cup \mathcal{Y}^{\mathcal{U}}$.

### 3.2. Cross-modal propagation network (CMPN)

CMPN adopts a meta-learning strategy, as shown in Fig. 1. In the meta-training stage, we randomly select $2C$ classes from the seen classes to construct an episode, where $C$ classes are treated as seen classes, and other $C$ classes are treated as fake unseen classes. In an episode, the zero-shot task is simulated to classify images of $C$ fake unseen classes, provided that $C$ seen classes are present. After multiple episodes, the base model gradually accumulates the ability to handle zero-shot problems. In the meta-testing stage, the features are generated utilizing the semantic vector of unseen classes.

For the base model training, we integrate the generative model, feature embedding, adaptive graph construction, and label propagation into a unified framework. The framework diagram of CMPN is shown in Fig. 2. The goal of CMPN is to ensure that

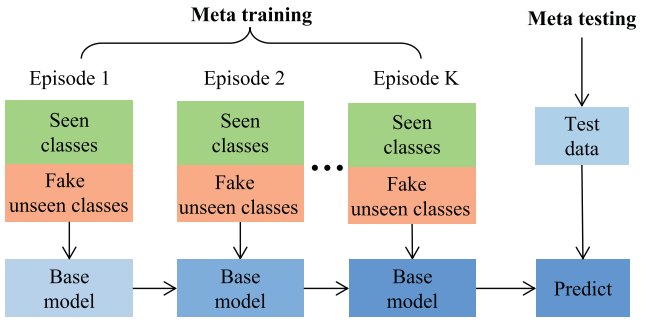



**Fig. 1.** The architecture of meta ZSL.

the fake visual features generated by the semantic vectors have intra-class compactness and inter-class separation. The generative model, which is used to generate fake visual features, ensures that the visual and semantic information is aligned in the visual space. The feature embedding is a neural network that maps features in the visual space to a new latent space. In that space, we obtain the nearest neighbor graph via adaptive graph construction and perform label propagation. Adaptive graph construction computes the parameters of distance metric to explore the manifold structure between different modalities and label propagation aims to use the label information of semantic features to predict the label of visual features.

### 3.2.1. Feature generation

For each episode $S^i = \{S_{tr}, S_{te}\}$, $S_{tr}$ is the samples of seen classes including visual samples $\mathcal{X}_s$ and their corresponding semantic vectors $\mathcal{A}_s$, $S_{te}$ is the samples of fake unseen classes including visual samples $\mathcal{X}_u$ and the corresponding semantic vectors $\mathcal{A}_u$. In CMPN, we use $S_{tr}$ and $S_{te}$ for transductive training under one episode. For the generator $G$: $\mathcal{Z} \times \mathcal{A} \to \mathcal{X}$, random Gaussian noise $z \in \mathcal{Z}$ and semantic vector $a \in \mathcal{A}$ are the inputs of the generator to generate features $\hat{x} \in \mathcal{X}$. The discriminator $D$ aims to discriminate whether the sample is a fake visual feature $\hat{x}$ or a real visual feature $x$. We use the classical GAN [29] as the generator. The generator tries to generate fake visual features that can fool the discriminator $D$. With $G$ and $D$, we can construct the relationship between the visual features and the semantic features. The discriminator $D$ can be learned by optimizing the adversarial objective. The losses of $G$ and $D$ are defined as: $\mathcal{L}_D = \mathbf{E}[D(x,a)] - \mathbf{E}[D(\hat{x},a)]$, $\mathcal{L}_G = -\mathbf{E}[D(\hat{x},a)]$.

### 3.2.2. Feature embedding

Feature generation mainly generates fake visual features utilizing semantic vectors, but we argue that the original visual space may lack discriminative ability. Therefore we map the real visual features and fake visual features to a new latent space $E$. To bootstrap the space $E$ to be more discriminative, we perform adaptive graph construction and label propagation in this space.

### 3.2.3. Adaptive graph construction

Manifold learning is to map the high-dimensional data to low-dimensional space so that this low-dimensional data can reflect some essential structural properties of the high-dimensional data. The most important factor in manifold learning is the construction of graph that reflects the true manifold of the data. Therefore, the choice of distance metric is particularly important when constructing the graph. In this paper, we use Gaussian similarity function as distance metric: $w_{ij} = \exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$, where $\sigma$ is the length scale parameter. The nearest neighbor graph structure changes with different values of $\sigma$. For label propagation, the value of $\sigma$ is worth considering. In this paper, we use the information of the sample itself to learn $\sigma$, so the $P$ network is utilized to get the $\sigma$ corresponding to each sample in space $E$ : $\sigma_i = P(E(x_i))$.

$$w_{ij} = exp(-\frac{1}{2}||\frac{E(x_i)}{\sigma_i}, \frac{E(x_j)}{\sigma_j}||^2). \tag{1}$$

The adaptive graph construction utilizes the network to learn the length scale parameter $\sigma$. The similarity between nodes is calculated utilizing the metric function and the graph is constructed using k-nearest neighbors. As shown in Fig. 2, the graph of an episode is constructed where the nodes are $E(x_i)$ and the edges are the relationship between two points.

### 3.2.4. Label propagation

The goal of label propagation is to use the label information of $\mathcal{A}_u$ to infer the label of samples in $\mathcal{X}_u$. It can be deduced that the design of the base model in CMPN is a transductive-style manner. The cross-modal label propagation learning aims to obtain a more discriminative latent space. A symmetric adjacency matrix $W$ has been constructed, $w_{ij}$ is the similarity of the sample pair. We then perform a graph Laplacian on $W$, $R = D^{-1/2}WD^{-1/2}$, where $D$ is a diagonal matrix, the value of $D_{ii}$ is the sum of $i_{th}$ row of $W$. The label matrix $Y$ is further defined as follows:

$$Y_{ij} = \begin{cases} 1 & x_i \in S_{tr} \bigwedge y_i = j \text{ or } x_i \in \hat{\mathcal{X}} \bigwedge y_i = j \\ 0 & x_i \in \mathcal{X}_u. \end{cases} \tag{2}$$

Label propagation identifies the label iteratively: $F = (I - \gamma R)^{-1}Y$, where $I$ is the identity matrix, $\gamma \in (0,1)$ is a parameter. The classification loss is calculated by measuring the prediction of label and ground truth of $S_{tr}$ and $S_{te}$. $F$ is converted to probabilistic score by softmax: $P(\tilde{y}_i = j) = \frac{exp(F_{ij})}{\sum_{j=1} exp(F_{ij})}$, where $\tilde{y}_i$

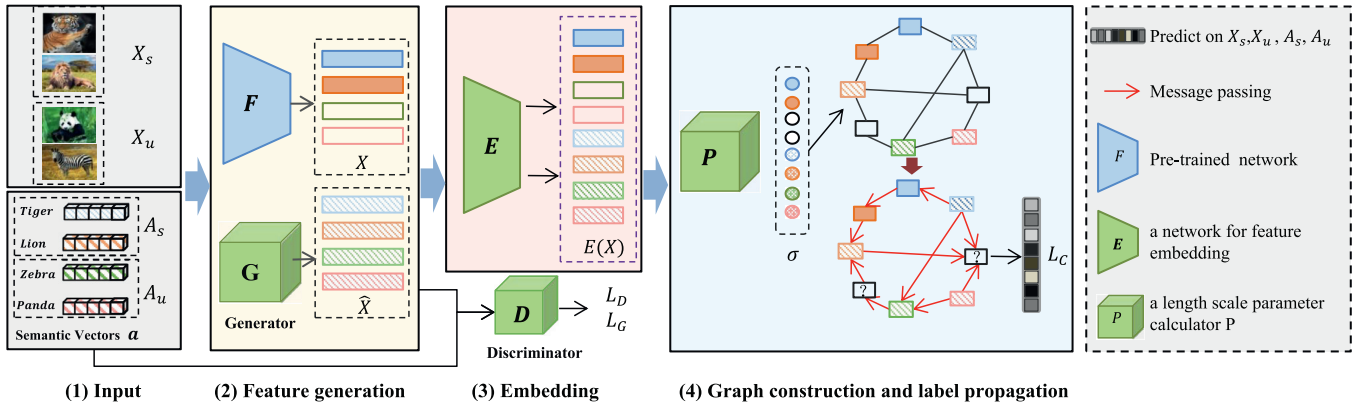



**Fig. 2.** The architecture of CMPN for one episode. Different colors are used to distinguish the classes of the vectors, i.e. "blue", "orange", "green", and "pink" represent "Tiger", "Lion", "Zebra" and "Panda", respectively.

**Table 1**
The detail of the datasets. "Att" is the dimension of the attribute. "S/U" is the number of seen classes and unseen classes. "Img (Tr)" is the number of images of the train. "Img (Te-s)" is the number of images that belong to seen classes of the test set. "Img (Te-u)" is the number of images that belong to unseen classes of the test set.

| Dataset | Att | S/U | Img (Tr) | Img (Te-s) | Img (Te-u) |
|---|---|---|---|---|---|
| AWA1 | 85 | 40/10 | 19,832 | 4958 | 5685 |
| AWA2 | 85 | 40/10 | 23,527 | 5882 | 7913 |
| CUB | 1024 | 150/50 | 7057 | 1764 | 2967 |
| aPY | 64 | 20/12 | 5932 | 1483 | 7924 |

is the predicted label of the $i_{th}$ sample. The classification loss is computed as:

$$\mathcal{L}_C = \sum_{i=1}^{C\times(N_1+N_2)} \sum_{j=1}^{2C} -\mathbb{I}(y_i == j) log(P(\tilde{y}_i = j|x_i)), \tag{3}$$

where $N_1$ and $N_2$ are the number of samples for each class in $S_{tr}$ and $S_{ts}$ respectively, and $y_i$ is the ground truth of $x_i$. Indicator function $\mathbb{I}(b)$, $\mathbb{I}(b) == 1$ if $b$ is true, else $\mathbb{I}(b) == 0$. The objective $L_{GC}$ is given as: $\mathcal{L}_{GC} = \alpha\mathcal{L}_G + \beta\mathcal{L}_C$, where $\alpha$, $\beta$ are hyperparameters. We train an episode in an end-to-end style. After several episodes of training, we believe that the model has accumulated experience in handling zero-shot problems. Therefore, the total loss of our model in an episode takes the form of: $\mathcal{L} = \mathcal{L}_D + \mathcal{L}_{GC}$.

### 3.3. Testing procedure

After training, we get the trained generator and network for feature embedding, so we can generate the labeled samples of unseen classes. For ZSL, we use the synthesized samples to train a classification model, such as SVM. For GZSL, the synthesized samples of unseen classes merge the samples of the seen class to train a classification model. Finally, we make predictions for the input test visual features $v$. $f(E(v)) = \arg\max_{y\in\tilde{\mathcal{Y}}} M(y|E(v);\theta')$. Where $\theta'$ is the parameter of the classification model $M$. For ZSL, $\tilde{\mathcal{Y}} = \mathcal{Y}^{\mathcal{U}}$, for GZSL, $\tilde{\mathcal{Y}} = \mathcal{Y}^S \cup \mathcal{Y}^{\mathcal{U}}$.

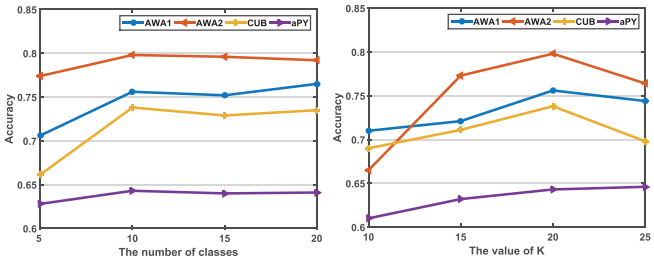
## 4. Experiments

### 4.1. Datasets and settings

We evaluate our model on four benchmark datasets, as shown in Table 1, including Animals with Attributes 1 (AWA1) [4], Animals with Attributes 2 (AWA2) [30], Caltech-UCSD Birds (CUB) [31], and Attribute Pascal and Yahoo (aPY) [32], where the criteria of dataset partition follows the [30]. Each dataset consists of visual features that are extracted from ResNet-101 pre-trained on ImageNet and semantic vectors are designed by humans (Table 1).

The whole framework consists of a generator ($G$), a discriminator ($D$), a network for feature embedding ($E$), and a length scale parameter calculator ($P$). For all the data we set the output dimension of generator $G$ to 2048, which is consistent with the visual dimension, the output dimension of network for feature embedding $E$ to 512, and the output of parameter calculator $P$ to 1 dimension. The generator $G$ and discriminator $D$ contain three fully-connected layers with LeakyReLU activation. The network $E$ contains one fully-connected layer with LeakyReLU activation. The length scale parameter calculator $P$ consists of two convolution blocks, each containing a $3\times3$ convolution, batch normalization, ReLU activation, and $2\times2$ max-pooling.

We use the episode-training paradigm of meta-learning, in an episode, $S_{tr}$ includes 10 classes with 5 samples each, $S_{te}$ includes 10 classes with 3 samples each. For parameters $\alpha$ and $\beta$, we take $\alpha = 0.1$, $\beta = 1$ for ZSL task and $\alpha = 1$, $\beta = 10$ for GZSL task. The value of $\gamma$ of label propagation is set to 0.99. For ZSL, the AWA1 dataset needs to generate 100 samples per class, 100 samples per class for AWA2 dataset, 50 samples per class for CUB dataset, and 100 samples per class for aPY dataset. For GZSL, the AWA1 dataset needs to generate 500 samples per class, 600 samples per class for AWA2 dataset, 50 samples per class for CUB dataset, and 600 samples per class for aPY dataset. We train the model for 5000 iterations for the CUB dataset and 500 iterations for the aPY dataset. The AWA1 and AWA2 datasets converge after 20,000 iterations. For ZSL, the evaluation criterion is classification accuracy, and for GZSL, the evaluation criterion is a harmonic mean which is defined as $H = \frac{2\times As\times Au}{As+Au}$, $H$ is determined by the class average accuracy of seen classes $As$ and class average accuracy of unseen classes $Au$. The higher $H$ is, the better the GZSL algorithm is.




(a) The impact of different number of classes. (b) The impact of the different value of $K$.

**Fig. 3.** Results of the hyper-parameters analysis.

### 4.2. Comparisons with state-of-the-art methods

Table 2 shows the results of CMPN compared to the state-of-the-art GZSL methods. Our CMPN achieves significant improvements of at least 1.1%, 0.9%, 1.9% and 2.9% of the harmonic mean on AWA1, AWA2, CUB and aPY, respectively. Specifically, our approach achieves competitive results compared to the current meta ZSL methods [14,36]. These results demonstrate the superiority and great potential of CMPN for feature generation.

Table 3 shows the comparison of CMPN with the existing state-of-the-art methods for ZSL. Our CMPN achieves significant improvements of at least 2.1%, 6.5%, 3.0% and 0.3% of the classification accuracy on AWA1, AWA2, CUB and aPY, respectively. Compared with relation net [36] and ZSML [14], CMPN considers the cross-modal label information propagation during training to ensure the generation of discriminative features.

### 4.3. Analysis of hyper-parameters

We evaluate the impact of the different number of selected seen and fake unseen classes in an episode. We restrict the number of classes of seen and fake unseen classes to be the same. The number of classes ranged from 5 to 20 with an interval of 5. As can be seen in Fig. 3(a), the choice of the number of classes greatly affects the classification performance. We think it is most appropriate to choose 10 seen classes and 10 fake unseen classes form an episode.

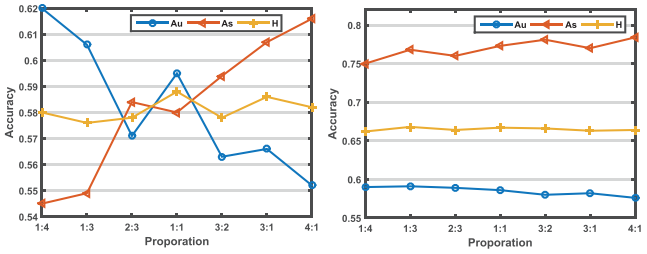
By constraining the total number of seen classes and fake unseen classes to be constant, we investigate the performances with different proportion of seen class number in $S_{tr}$ versus fake unseen class number in $S_{te}$. Particularly, proportions from $[1:4, 1:3, 2:3, 1:1, 3:2, 3:1, 4:1]$ are utilized, e.g. $1:4$ stands for selecting 4 seen classes and 16 fake unseen classes with a constant total class number of 20. As shown in Fig. 4, in the generalized setting, $As$ tends to increase and $Au$ tends to decrease while the proportions increase for CUB and AWA2. The changing tendency of

**Table 2**
GZSL results on AWA1, AWA2, CUB, and aPY. *As*/*Au*: the class average accuracy of seen/unseen classes (%). *H*: the harmonic mean accuracy (%). Best and second best results are marked with bold and underlined.

| | AWA1 | | | AWA2 | | | CUB | | | aPY | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Au | As | H | Au | As | H | Au | As | H | Au | As | H |
| ESZSL [7] | 6.6 | 75.6 | 12.1 | 5.9 | 77.8 | 11.0 | 12.6 | 63.8 | 21.0 | 2.4 | 70.1 | 4.6 |
| SJE [33] | 11.3 | 74.6 | 19.6 | 8.0 | 73.9 | 14.4 | 23.5 | 59.2 | 33.6 | – | – | – |
| LATEM [8] | 7.3 | 71.7 | 13.3 | 11.5 | 77.3 | 20.0 | 15.5 | 57.3 | 24.0 | 0.1 | 73.0 | 0.2 |
| ALE [34] | 16.8 | 76.1 | 27.5 | 14.0 | 81.8 | 23.9 | 23.7 | 62.8 | 34.4 | – | – | – |
| SYNC [9] | 8.9 | 87.3 | 16.2 | 10.0 | 90.5 | 18.0 | 11.5 | 70.9 | 19.8 | 7.4 | 66.3 | 13.3 |
| DEM [35] | 32.8 | 84.7 | 47.3 | 30.5 | 86.4 | 45.1 | 19.6 | 57.9 | 29.2 | 11.1 | 75.1 | 19.4 |
| RN [36] | 31.4 | 91.3 | 46.7 | 30.0 | 93.4 | 45.3 | 38.1 | 61.1 | 47.0 | – | – | – |
| DCN [37] | – | – | – | 25.5 | 84.2 | 39.1 | 28.4 | 60.7 | 38.7 | 14.2 | 75.0 | 23.9 |
| TCN[38] | 49.4 | 76.5 | 60.0 | 61.2 | 65.8 | 63.4 | 52.6 | 52.0 | 52.3 | – | – | – |
| EDE [39] | 36.9 | 90.6 | 52.4 | 35.2 | 93.0 | 51.1 | 21.0 | 66.0 | 31.9 | 7.8 | 75.3 | 14.1 |
| GAZSL [40] | 29.6 | 84.2 | 43.8 | 35.4 | 86.9 | 50.3 | 31.7 | 61.3 | 41.8 | – | – | – |
| SE-GZSL [10] | 56.3 | 67.8 | 61.5 | 68.1 | 58.3 | 62.8 | 41.5 | 53.3 | 46.7 | – | – | – |
| $f$-CLSWGAN [11] | 61.4 | 57.9 | 59.6 | 57.9 | 61.4 | 59.6 | 43.7 | 57.7 | 49.7 | – | – | – |
| cycle-CLSGAN [41] | 56.9 | 64.0 | 60.2 | – | – | – | 45.7 | 61.0 | 52.3 | – | – | – |
| ABP [42] | 57.3 | 67.1 | 61.8 | 55.3 | 72.6 | 62.6 | 47.0 | 54.8 | 50.6 | – | – | – |
| CADA-VAE [12] | 57.3 | 72.8 | 64.1 | 55.8 | 75.0 | 63.9 | 47.2 | 35.7 | 40.6 | 14.7 | 30.5 | 19.8 |
| F-VAEGAN-D2 [13] | – | – | – | 57.6 | 70.6 | 63.5 | 48.4 | 60.1 | 53.6 | – | – | – |
| Zero-VAE-GAN [15] | 58.2 | 66.8 | 62.3 | 57.1 | 70.9 | 62.5 | 43.6 | 47.9 | 45.5 | 32.0 | 52.2 | 39.7 |
| AMAZ [43] | 64.4 | 63.6 | 64.1 | 60.1 | 69.2 | 64.3 | 58.2 | 55.7 | 56.9 | – | – | – |
| DUET [44] | – | – | – | 48.2 | 90.2 | 63.4 | 39.7 | 80.1 | 53.1 | 21.8 | 55.6 | 31.3 |
| ZSML [14] | 57.4 | 71.1 | 63.5 | 58.9 | 74.6 | 65.8 | 60.0 | 52.1 | 55.7 | 36.3 | 46.6 | 40.9 |
| CMPN | 61.0 | 70.1 | **65.2** | 58.6 | 77.3 | **66.7** | 59.5 | 58.0 | **58.8** | 31.3 | 68.5 | **42.9** |

**Table 3**
ZSL results on AWA1, AWA2, CUB, and aPY (%). Best and second best results are marked with bold and underlined.

| Method | AWA1 | AWA2 | CUB | aPY |
|---|---|---|---|---|
| ESZSL [7] | 58.2 | 58.6 | 53.9 | 38.3 |
| LATEM [8] | 55.1 | 55.8 | 49.3 | 35.2 |
| SYNC [9] | 54.0 | 46.6 | 55.6 | 23.9 |
| DEM [35] | 68.4 | 67.1 | 51.7 | 35.0 |
| DCN [37] | – | 65.2 | 56.2 | 43.6 |
| SE-GZSL [10] | 69.5 | 69.2 | 59.6 | – |
| $f$-CLSWGAN [11] | – | 68.2 | 57.3 | – |
| F-VAEGAN-D2 [13] | 71.1 | – | 61.0 | – |
| CADA-VAE [12] | 62.3 | 64 | 60.4 | – |
| Zero-VAE-GAN [15] | 71.4 | 69.3 | 54.8 | 37.4 |
| LisGAN [45] | 70.6 | 70.4 | 58.8 | – |
| APNet [17] | 68.0 | 68.0 | 57.7 | 41.3 |
| RN [36] | 68.2 | 64.2 | 55.6 | – |
| EDE [39] | 70.1 | 66.5 | 57.1 | 20.4 |
| EXEM (1NNs) [46] | 68.1 | 64.6 | 58.0 | – |
| ZSML [14] | 73.5 | 76.1 | 69.7 | 64.0 |
| CMPN | **75.6** | **82.6** | **72.7** | **64.3** |




(a) The impact of different proportion of the class number on CUB.

(b) The impact of different proportion of the class number on AWA2.

**Fig. 4.** Effect of the proportion of the seen class number versus fake unseen class number in an episode.

*H*, overall, is stable. $S_{te}$ ensures that the fake visual features generated by semantic vectors can propagate label information to visual features, and as the amount of $S_{te}$ classes decreases, this ability decreases, leading to a decrease of *Au*. To compromise between *As* and *Au*, we chose a proportion of 1 : 1, i.e. 10 classes for both $S_{tr}$ and $S_{te}$ to ensure the discriminability of the generated samples.

**Table 4**
Experimental results of ZSL of four datasets with different dimensions of the latent space *E*.

| Dimensions of *E* | AWA1 | AWA2 | CUB | aPY |
|---|---|---|---|---|
| 512 | **75.6** | **82.6** | **72.7** | 64.3 |
| 1024 | 72.9 | 77.5 | 72.6 | **64.5** |
| 2048 | 73.7 | 75.5 | 74.5 | 63.6 |

For graph construction, we use *K* nearest neighbors and the choice of *K* affects the graph structure, different graph structures affect the results of label propagation. We choose different *K*, from 10 to 25, with an interval of 5. As can be seen in Fig. 3(b), the choice of *K* greatly affects the classification performance. When *K* takes the value of 20, the *H* results of all four datasets reach the best.

To obtain a more discriminative space, we map the real and fake visual features to a new space *E*. We show the effect of different dimensions of *E* on the CMPN model. We analyze the ZSL results with 512, 1024, 2048 dimensions in *E*. From Table 4, we observe that for ZSL, setting the dimension of *E* to 512 is desirable, which can achieve a trade-off between performance and computational cost.

### 4.4. Ablation study

CMPN consists of four components, the generative model GAN, the network for feature embedding *E*, a length scale parameter calculator *P*, and the label propagation LP. As shown in Fig. 6, we remove one of the components to measure the impact of that component on the results. Besides, we also validate the effectiveness of the episode-training paradigm.

*Effectiveness of episode-training paradigm* To verify the effectiveness of this learning approach, we select 0 fake unseen classes when constructing episodes. *C* classes are randomly selected and $N_1$ samples from each class are chosen as $S_{tr}$. From *C* classes, which are consistent with the $S_{tr}$, $N_2$ samples from each class are chosen as $S_{te}$. As shown in Fig. 5, we can conclude that the *Au*

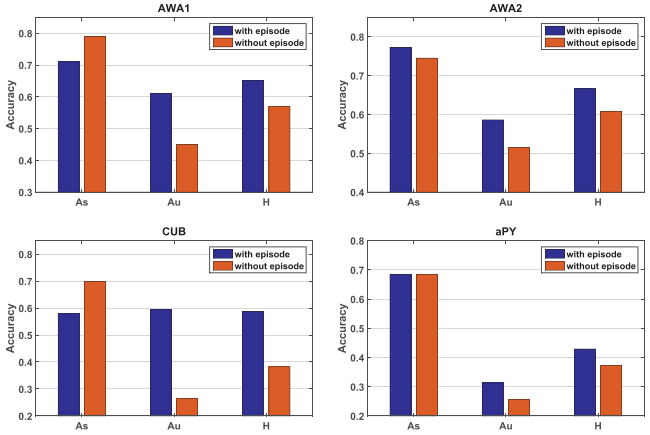



**Fig. 5.** Effectiveness of episode-training paradigm.

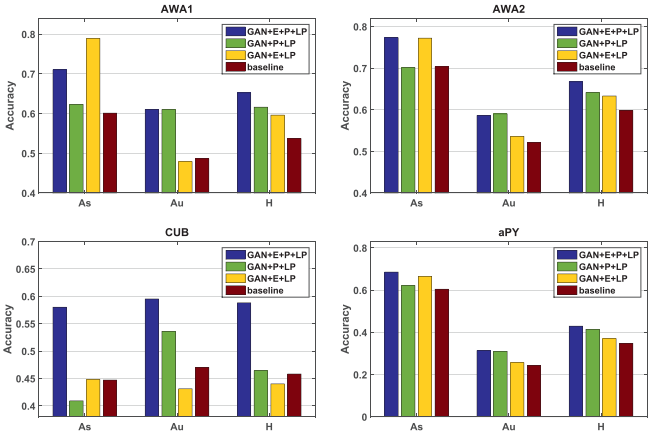



**Fig. 6.** Effects of different components on four datasets with GZSL setting.

and *H* of all datasets decrease significantly in the case without the episode-training paradigm. Indicating that the episode-training paradigm plays a role in CMPN, increasing the generalization performance of the model by learning from multiple simulated ZSL tasks.

*Effectiveness of feature embedding* We argue that the obtained space *E* has stronger discriminative power compared to the original visual space. To prove the conjecture, we remove the network *E* from the CMPN model and perform label propagation in the original visual space. The *As* metric of model GAN+P+LP is less than CMPN, and the *H* is also less than CMPN, indicating that the feature mapping has improved the accuracy of the seen class to some extent. Space *E* is more discriminative than the original visual space.

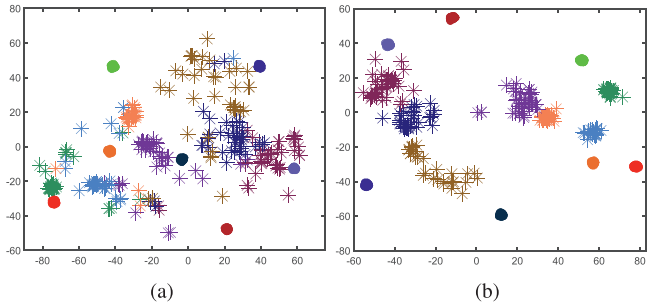



**Fig. 8.** t-SNE visualizations of several randomly selected seen and unseen classes on CUB. $*$ indicates unseen classes, $\bullet$ indicates seen classes.

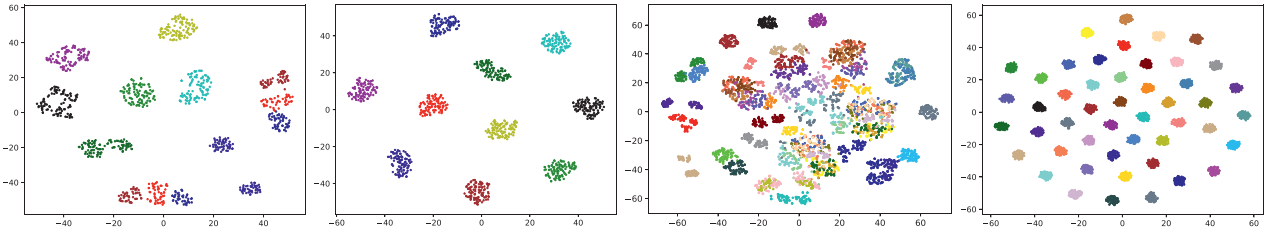
*Effectiveness of learning the length scale parameter* To verify the effectiveness of this graph construction method, we set the length scale parameter $\sigma$ to 1. For datasets AWA1, aPY, the *Au*, *H* metric of model GAN + E + LP is less than CMPN. For datasets AWA2, CUB, the *Au*, *As*, and *H* metric of model GAN + E + LP is less than CMPN. This shows that learning the length scale parameter $\sigma$ through the network is practical and effective for adaptive graph construction.

*Effectiveness of label propagation* As for baseline model, we only consider GAN loss. The results of the baseline model in metric *As*, *Au*, and *H* are less than those of CMPN, which indicates that it is crucial to consider the high quality of the synthesized samples and ensure that the generated samples can guide the classification of real visual samples.

### 4.5. Visualizations

We show t-SNE [47] visualizations of synthesized unseen samples on AWA2/CUB under ZSL for ZSML and CMPN. From Fig. 7 it can be concluded that: (1) CMPN is able to generate features with more inter-class discriminability and intra-class compactness; (2) CMPN consistently outperforms baseline ZSML for both datasets AWA2 and CUB.

To further demonstrate that CMPN can the manifold structure to ensure intra-class compactness and inter-class separation in the latent space, we show the t-SNE visualization of several randomly selected seen and unseen class samples on CUB. Fig. 8(a) visualizes the classes in the latent space without considering the manifold structure (the design of the classification loss does not consider the manifold structure and only focuses on the classification of the generated samples), and Fig. 8(b) shows the visualization of the classes under the CMPN model. We can obtain that: (1) Since the unseen classes are generated, the unseen classes in (a) and (b) have good properties. (2) Compared to Fig. 8(a), the visible classes in Fig. 8(b) are separable between classes and more compact within classes. (3) Compared to Fig. 8(a), there are more diversity between the seen and unseen classes in Fig. 8(b).



(a) AWA2: ZSML. (b) AWA2: CMPN. (c) CUB: ZSML. (d) CUB: CMPN.

**Fig. 7.** t-SNE visualizations of synthesized unseen samples on AWA2/CUB under ZSL for ZSML and CMPN, respectively.

## 5. Conclusion and future work

In this paper, we propose a new meta ZSL model, the cross-modal propagation network model (CMPN), which uses an episode-training paradigm. To ensure intra-class compactness and inter-class separability in the latent space, the CMPN integrates adaptive graph construction and label propagation into the generative model. CMPN also can guarantee the unambiguous and discriminative fake feature generating. Extensive experiments on these benchmarks validate the effectiveness of CMPN. The effectiveness of the proposed CMPN is further demonstrated by ablation experiments. In the future, we will further explore the structural information of multimodal to pursue better performance of zero-shot learning.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work was supported by the National Key Research and Development Program of China (2020AAA0106100), the National Natural Science Foundation of China (No. 62006147).

## References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: NeurIPS, 2012, pp. 1106–1114.
[2] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.
[3] G.S. Xie, X.Y. Zhang, Y. Yao, Z. Zhang, F. Zhao, L. Shao, Vman: a virtual mainstay alignment network for transductive zero-shot learning, IEEE Trans. Image Process. 30 (2021) 4316–4329.
[4] C.H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: CVPR, 2009, pp. 951–958.
[5] C. Lampert, H. Nickisch, S. Harmeling, Attribute-based classification for zero-shot visual object categorization, IEEE Trans. Pattern Anal. Mach. Intell. 36 (3) (2014) 453–465.
[6] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: NeurIPS, 2013, pp. 3111–3119.
[7] B. Romera-Paredes, P.H.S. Torr, An embarrassingly simple approach to zero-shot learning, in: ICML, 2015, pp. 2152–2161.
[8] Y. Xian, Z. Akata, G. Sharma, Q.N. Nguyen, M. Hein, B. Schiele, Latent embeddings for zero-shot classification, in: CVPR, 2016, pp. 69–77.
[9] S. Changpinyo, W.-L. Chao, B. Gong, F. Sha, Synthesized classifiers for zero-shot learning, in: CVPR, 2016, pp. 5327–5336.
[10] V.K. Verma, G. Arora, A. Mishra, P. Rai, Generalized zero-shot learning via synthesized examples, in: CVPR, 2018, pp. 4281–4289.
[11] Y. Xian, T. Lorenz, B. Schiele, Z. Akata, Feature generating networks for zero-shot learning, in: CVPR, 2018, pp. 5542–5551.
[12] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, Z. Akata, Generalized zero-and few-shot learning via aligned variational autoencoders, CVPR, 2019. 8274–8255
[13] Y. Xian, S. Sharma, B. Schiele, Z. Akata, F-VAEGAN-D2: a feature generating framework for any-shot learning, in: CVPR, 2019, pp. 10275–10284.
[14] V.K. Verma, D. Brahma, P. Rai, Meta-learning for generalized zero-shot learning, in: AAAI, 2020, pp. 6062–6069.
[15] R. Gao, X. Hou, J. Qin, J. Chen, L. Liu, F. Zhu, Z. Zhang, L. Shao, Zero-VAE-GAN: generating unseen features for generalized and transductive zero-shot learning, IEEE Trans. Image Process. 29 (2020) 3665–3680.
[16] G. Xie, L. Liu, X. Jin, F. Zhu, Z. Zhang, J. Qin, Y. Yao, L. Shao, Attentive region embedding network for zero-shot learning, in: CVPR, 2019, pp. 9384–9393.
[17] W. Xu, Y. Xian, J. Wang, B. Schiele, Z. Akata, Attribute prototype network for zero-shot learning, in: NeurIPS, 2020, pp. 21969–21980.
[18] G. Xie, L. Liu, F. Zhu, F. Zhao, Z. Zhang, Y. Yao, J. Qin, L. Shao, Region graph embedding network for zero-shot learning, in: ECCV, 2020, pp. 562–580.
[19] G.-S. Xie, Z. Zhang, G. Liu, F. Zhu, L. Liu, L. Shao, X. Li, Generalized zero-shot learning with multiple graph adaptive generative networks, IEEE Trans. Neural Netw. Learn. Syst. (2021) 1–13.
[20] J. Snell, K. Swersky, R.S. Zemel, Prototypical networks for few-shot learning, in: NeurIPS, 2017, pp. 4077–4087.
[21] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: ICML, 2017, pp. 1126–1135.
[22] S. Thrun, L.Y. Pratt, Learning to learn: introduction and overview, Learn. Learn (1998) 3–17.
[23] F. Alet, M.F. Schneider, T. Lozano-Pérez, L.P. Kaelbling, Meta-learning curiosity algorithms, ICLR, 2020.
[24] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, M. Pontil, Bilevel programming for hyperparameter optimization and meta-learning, in: ICML, 2018, pp. 1563–1572.
[25] T.M. Hospedales, A. Antoniou, P. Micaelli, A.J. Storkey, Meta-learning in neural networks: a survey, IEEE Trans. Pattern Anal. Mach. Intell. (2021), doi:10.1109/TPAMI.2021.3079209.
[26] X. Zhu, Z. Ghahramain, Learning from labels and unlabeled data with label propagation, Tech. Rep. 3175 (2004) (2002) 237–244.
[27] V. Badrinarayanan, F. Galasso, R. Cipolla, Label propagation in video sequences, in: CVPR, 2010, pp. 3265–3272.
[28] N. Chen, Y. Liu, H. Chen, J. Cheng, Detecting communities in social networks using label propagation with information entropy, Phys. A 471 (2017) 788–798.
[29] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A.C. Courville, Y. Bengio, Generative adversarial nets, in: NeurIPS, 2014, pp. 2672–2680.
[30] Y. Xian, C.H. Lampert, S. Bernt, A. Zeynep, Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly, IEEE Trans. Pattern Anal. Mach. Intell. 41 (9) (2019) 2251–2262.
[31] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 Dataset, Technical Report, 2011.
[32] A. Farhadi, I. Endres, D. Hoiem, D.A. Forsyth, Describing objects by their attributes, in: CVPR, 2009, pp. 1778–1785.
[33] Z. Akata, S.E. Reed, D. Walter, H. Lee, B. Schiele, Evaluation of output embeddings for fine-grained image classification, in: CVPR, 2015, pp. 2927–2936.
[34] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for image classification, IEEE Trans. Pattern Anal. Mach. Intell. 38 (7) (2016) 1425–1438.
[35] L. Zhang, T. Xiang, S. Gong, Learning a deep embedding model for zero-shot learning, in: CVPR, 2017, pp. 3010–3019.
[36] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: Relation network for few-shot learning, in: CVPR, 2018, pp. 1199–1208.
[37] S. Liu, M. Long, J. Wang, M.I. Jordan, Generalized zero-shot learning with deep calibration network, in: NeurIPS, 2018, pp. 2009–2019.
[38] H. Jiang, R. Wang, S. Shan, X. Chen, Transferable contrastive network for generalized zero-shot learning, in: ICCV, 2019, pp. 9764–9773.
[39] L. Zhang, P. Wang, L. Liu, C. Shen, W. Wei, Y. Zhang, A. van den Hengel, Towards effective deep embedding for zero-shot learning, IEEE Trans. Circuits Syst. Video Technol. 30 (9) (2020) 2843–2852.
[40] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, A. Elgammal, A generative adversarial approach for zero-shot learning from noisy texts, in: CVPR, 2018, pp. 1004–1013.
[41] R. Felix, B.G.V. Kumar, I.D. Reid, G. Carneiro, Multi-modal cycle-consistent generalized zero-shot learning, in: ECCV, 2018, pp. 21–37.
[42] Y. Zhu, J. Xie, B. Liu, A. Elgammal, Learning feature-to-feature translator by alternating back-propagation for generative zero-shot learning, in: ICCV, 2019, pp. 9843–9853.
[43] Y. Li, Z. Liu, L. Yao, X. Wang, C. Wang, Attribute-modulated generative meta learning for zero-shot classification, IEEE Trans. Multimed. (2021).
[44] Z. Jia, Z. Zhang, L. Wang, C. Shan, T. Tan, Deep unbiased embedding transfer for zero-shot learning, IEEE Trans. Image Process. 29 (2020) 1958–1971.
[45] J. Li, M. Jing, K. Lu, Z. Ding, L. Zhu, Z. Huang, Leveraging the invariant side of generative zero-shot learning, in: CVPR, 2019, pp. 7402–7411.
[46] S. Changpinyo, W. Chao, B. Gong, F. Sha, Classifier and exemplar synthesis for zero-shot learning, Int. J. Comput. Vis. 128 (1) (2020) 166–201.
[47] V.D.M. Laurens, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2605) (2008) 2579–2605.