Region Graph Embedding Network for Zero-Shot Learning

Guo-Sen Xie[1⋆], Li Liu[1], Fan Zhu[1], Fang Zhao[1], Zheng Zhang[2,3⋆], Yazhou Yao[5], Jie Qin[1], and Ling Shao[1,4]

[1] Inception Institute of Artificial Intelligence, UAE
[2] Harbin Institute of Technology, Shenzhen, China
[3] Peng Cheng Laboratory, Shenzhen, China
[4] Mohamed bin Zayed University of Artificial Intelligence, UAE
[5] Nanjing University of Science and Technology, China

Abstract. Most of the existing Zero-Shot Learning (ZSL) approaches learn direct embeddings from global features or image parts (regions) to the semantic space, which, however, fail to capture the appearance relationships between different local regions within a single image. In this paper, to model the relations among local image regions, we incorporate the region-based relation reasoning into ZSL. Our method, termed as Region Graph Embedding Network (RGEN), is trained end-to-end from raw image data. Specifically, RGEN consists of two branches: the Constrained Part Attention (CPA) branch and the Parts Relation Reasoning (PRR) branch. CPA branch is built upon attention and produces the image regions. To exploit the progressive interactions among these regions, we represent them as a *region graph*, on which the parts relation reasoning is performed with graph convolutions, thus leading to our PRR branch. To train our model, we introduce both a *transfer* loss and a *balance* loss to contrast class similarities and pursue the maximum response consistency among seen and unseen outputs, respectively. Extensive experiments on four datasets well validate the effectiveness of the proposed method under both ZSL and generalized ZSL settings.

Keywords: Zero-shot learning · Parts relation reasoning · Balance loss

1 Introduction

Humans can efficiently recognize instances from unseen categories, by simply exploiting their past knowledge on seen class images as well as descriptions of both seen and unseen classes. This capability of perceiving unseen concepts is dubbed Zero-Shot Learning (ZSL) [35, 24]. However, most of the available deep learning approaches [17, 54, 47, 60] lack such a ZSL-like ability, e.g., the CNN models [46, 68, 70, 75] usually suffer from insufficient (or no) training data. Moreover, annotating large amounts of data is both time consuming and costly [66, 69, 74,

⋆ Corresponding authors (gsxiehm@gmail.com, darrenzz219@gmail.com).

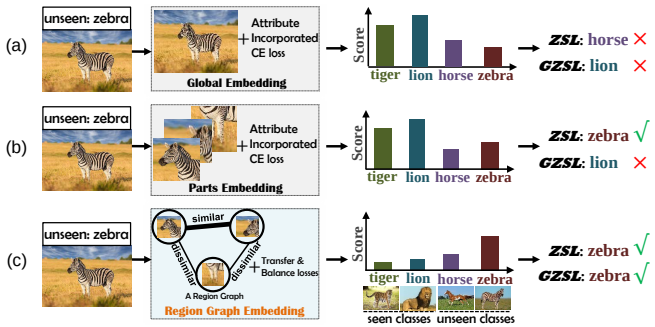

Fig. 1. Three Types of End-to-end ZSL and GZSL Models. (a) Global Embedding, which misclassifies unseen "zebra" to "horse" and "lion" under ZSL and GZSL, respectively. This happens because 1) the global features are not discriminative enough to distinguish these two confused classes, i.e., "zebra" and "horse", and 2) the domain bias under GZSL makes the prediction scores on seen classes ("tiger" and "lion") much higher than those on unseen classes ("horse" and "zebra"). **(b) Parts Embedding**, which correctly classifies "zebra" under ZSL. However, the domain bias still exists, which again results in a misclassification of "zebra" to "lion" under GZSL. **(c) Our RGEN** can distinguish "zebra" under both ZSL and GZSL with a better prediction confidence, which benefits from parts relation reasoning on the region graph and the new training losses (*transfer* and *balance* losses).

76, 61], and novel (unseen) categories are constantly emerging in practical scenarios [58]. As such, ZSL [2, 50] has become an important research topic for its potential to alleviate data annotation costs and handle unseen class recognition.

ZSL typically trains a model by merely leveraging seen class images and then apply it to unseen images, where the label sets of the seen and unseen classes are disjoint. In the ZSL paradigm, the testing label set is unrealistically constrained to only unseen class, hindering its application in the real world. Extending the label set to include both seen and unseen classes during testing leads to Generalized ZSL (GZSL) setting [8, 50, 67, 65]. The semantic descriptions [12] collected for each seen and unseen category ensures efficient knowledge transfer between the two disjoint class sets, making both the ZSL and GZSL tasks feasible.

Semantic descriptions (such as attributes [12], sentences [38] and word vectors [41]) are shared information among seen and unseen categories, through which semantic knowledge is transferred from seen to unseen categories. Attributes are most commonly used and are thus adopted in this paper as well. Seminal works [2, 4, 39, 14, 33] on ZSL rely on seen images and their semantic vectors (quantized attributes) when finding an embedding space, where unseen images are distinguished by nearest neighbor search. Specifically, the embedding space can be categorized into three types: semantic space [23, 38, 48, 41, 9, 42, 37], feature space [5, 29, 40] and latent space [27, 59]. Moreover, thanks to the success of generative models [16, 71], several feature hallucinating approaches [49, 43, 63, 10] have been proposed for converting ZSL into conventional recognition tasks.

Most of the aforementioned methods adopt the following scheme for ZSL: 1) extracting global features from pre-trained [22, 62, 29, 72, 73] or end-to-end trainable nets [33, 42, 26] (Fig.1(a)) and 2) constructing embedding or generative models by associating these features with their semantic vectors. However, these approaches cannot efficiently capture the subtle differences between seen and unseen images [52], thus leading to undesirable semantic transfer. Very recently, attention based end-to-end models [52, 80, 78] have paved the way for discovering more discriminative part (region) [1] features by using semantic vectors as a guidance, showing remarkable improvements under ZSL but not GZSL. However, all these methods focus on direct parts embedding (Fig.1(b)) of these part features and fail to capture appearance relationships among them. Additionally, issues with domain bias [15, 8] still exist, meaning that the learned models merely rely on the seen categories, while ignoring the available unseen attributes.

To tackle the above challenges, in this paper, we first apply the attention method in [52] to generate the attended object regions on each input image. Then, we propose to perform the region-based relation modeling by Graph Convolutional Network (GCN) [21] (§3.3). Specifically, we represent each input image as a **Region Graph** (Fig.1(c)) with each node in the graph representing an attended region in the image. The edges of these region nodes are their pairwise appearance similarities. As such, the updated features after the GCN reasoning can capture the appearance relationships among these local parts, which is a complementary cue for improving the ZSL performance. Furthermore, **Embedding** to the semantic space are conducted for both the original attended region features and the updated ones. On the other hand, to train our model, we first propose a *transfer* loss (detailed in §3.4, Eq.(9)) by transferring the class similarities from seen to unseen classes. The *transfer* loss is designed by extending the seen attributes guided compatibility loss [80] with the collaborative guidance of the contrastive similarity score between seen and unseen attributes. Moreover, to address the domain bias issue (Fig.1(a)-(b)) in the end-to-end GZSL models [52, 80, 79], we propose a *balance* loss by minimizing the maximum response consistency between seen and unseen predictions. To this end, the end-to-end trainable **Network** architecture in Fig. 2 is termed as **Region Graph Embedding Network** (RGEN). Detailly, RGEN consists of the Constrained Part Attention (CPA) branch and the Parts Relation Reasoning (PRR) branch.

To sum up, our contributions are: **(1)** We present a region graph embedding network which incorporates region-based relation reasoning into embedding learning. To the best of our knowledge, this is the first attempt to do this in ZSL domain. **(2)** We propose a novel *region graph* representation capturing relationships between attended parts in a single image; GCN-based parts relation reasoning on this graph is then performed. This leads to the complementary Parts Relation Reasoning (PRR) sub-branch. **(3)** We propose the *transfer* loss and *balance* loss to guide the end-to-end RGEN training. Especially, the novel *balance* loss is capable of tackling the severe domain bias problem in end-to-end GZSL models.

[1] In this paper, part and region are alternatively used.

2 Related Works

(Generalized) Zero-Shot Learning. Early works [24, 18] on ZSL rely on learning attribute classifiers, based on which the class posterior of a test image is deduced. However, associations among these attributes are not well exploited. More recently, a number of embedding based methods [50] have been proposed, which are usually accompanied by a compatibility loss and can effectively address the association issue. Among them, ALE [2] leverages a compatibility hinge loss for learning the association between images and attributes. LATEM [48] is a piecewise extension of ALE. A compatibility based ridge regression is utilized in ESZSL [39]. DEM [66], CMT [41], SJE [4], and DEVICE [14] are also competitive embedding based models. However, these methods usually achieve relatively inferior results, since they adopt global features and/or exploit shallow models. Currently, end-to-end CNN models, such as SCoRe [33], LDF [26], QSFL [42] and LFGAA [28], obtain the best performances. These methods extend the compatibility loss by adding the seen class attributes, and advocate learning more discriminative features. Nevertheless, they struggle to focus on the discriminative parts which are intrinsically accounting for better semantic transfer [11]. Methods designed for ZSL are applicable for GZSL [50], which is more appropriate for real-life applications as it searches the full label space during testing.

Part-based ZSL. Initial works [11, 1, 64] utilized part annotations to discover discriminative part features for tackling fine-grained ZSL. However, part annotations are costly and labor-dependent. More recently, by pursing automatic part discovery [53], attention mechanisms [57, 56, 55, 25] have been applied into ZSL and GZSL [52, 80, 78, 30] for capturing multiple semantic regions, which can facilitate desirable knowledge transfer. These methods achieve remarkable improvements on ZSL, but the performance gains on GZSL are not satisfactory, indicating that they fail at solving the domain bias issue.

In this paper, to solve the realistic inductive ZSL and GZSL tasks (unseen images are inaccessible [27, 19]), we propose a Region Graph Embedding Network (RGEN) with the *transfer* and *balance* losses as supervision. Specifically, the PRR branch is based on GCN [21, 31] for relation reasoning. Although, GCN has been used in ZSL [45, 20] for outputting the visual classifier for each object class, by feeding the word embedding for every object class as inputs; however, we are **the first to explicitly leverage GCN for reasoning about the parts relations within each single image for ZSL**, e.g., the "leg" image is dissimilar with the "head" image in Fig. 1(c). As such, our intuition of using GCN is completely different from [45, 20]. To this end, our RGEN is related yet greatly different from current part- and GCN-based ZSL and GZSL methods.

3 Methodology

Task Definitions. We have N^s training samples from C^s seen classes which are defined as $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$. $\mathcal{X}^{\mathcal{S}}$=$\{x_i^s\}_{i=1}^{N^s}$ and $\mathcal{Y}^{\mathcal{S}}$ are the training data set and its label set, respectively. The seen class label of the ith sample x_i^s is $y_i^s \in \mathcal{Y}^{\mathcal{S}}$.

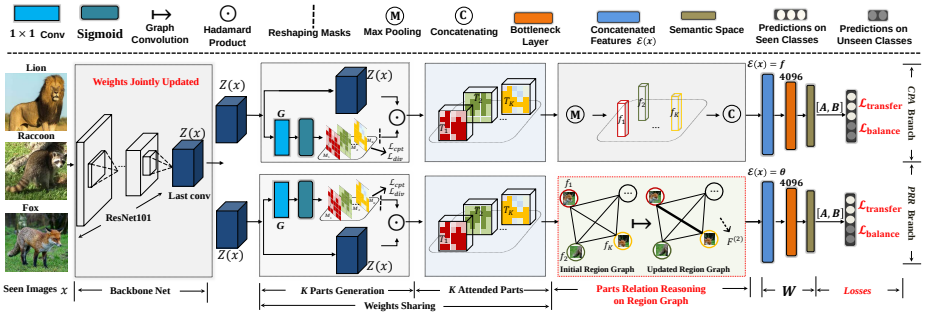

Fig. 2. Architecture. RGEN consists of CPA and PRR branches. For CPA, the input x is first passed through the Backbone Net and K Parts Generation module (constrained by $\mathcal{L}_{\text{cpt}}$ and $\mathcal{L}_{\text{div}}$), thus producing K attended parts: $\{T_i\}_{i=1}^{K}$. Then, max pooling, concatenation ($\mathcal{E}(x)$), bottleneck layer embedding and semantic space embedding are carried out. For PRR, part features $\{f_i\}_{i=1}^{K}$ are taken as input node features of GCN to acquire updated node features $F^{(2)}$. Then, the same operations as CPA is conducted. Finally, $\mathcal{L}_{\text{transfer}}$ and $\mathcal{L}_{\text{balance}}$ (§3.4) are leveraged for training.

$\mathcal{A}^s = \{a_i^s\}_{i=1}^{C^s}$ represents the semantic vector set of seen classes. For ZSL, given an unseen testing set $\mathcal{U} = \{(x_i^u, y_i^u)\}_{i=1}^{N^u}$ with N^u samples, we want to predict the label $y_i^u \in \mathcal{Y}^{\mathcal{U}}$ for each x_i^u. More knowledge for $\mathcal{U}$ is provided by the semantic vector set $\mathcal{A}^u = \{a_i^u\}_{i=1}^{C^u}$ for the C^u unseen classes. The label sets of seen and unseen classes are disjoint, i.e., $\mathcal{Y}^{\mathcal{S}} \cap \mathcal{Y}^{\mathcal{U}} = \emptyset$. Meanwhile, for GZSL, the searched label space is expanded to $\mathcal{Y} = \mathcal{Y}^{\mathcal{S}} \cup \mathcal{Y}^{\mathcal{U}}$ by taking samples from both seen and unseen classes as the testing data. We further denote $a_i^s/a_i^u \in \mathbb{R}^Q$.

3.1 Overview

The Region Graph Embedding Network (RGEN) (Fig. 2) consists of two sub-branches: the Constrained Part Attention (CPA) branch and the Parts Relation Reasoning (PRR) branch. Both branches are jointly trained by the proposed *transfer* and *balance* losses (§3.4). The CPA is capable of automatically discovering more discriminative regions, which applies [52] to generate attended object regions and is different from [52] as follows: 1) unlike [52] without any regularizations on attention masks, compactness and diversity are introduced for learning desirable parts; 2) *transfer* and *balance* losses are leveraged comparing to [52] which uses attribute incorporated cross-entropy loss (Fig.1(b)). Moreover, PRR aims at capturing appearance relationships among the discovered parts by GCN-based [21] graph reasoning. The outputs of such GCNs are updated node features (with each node representing an attended region), which are further used to learn embedding to the semantic space.

We further add a bottleneck layer between the feature space and the low-dimensional semantic space to alleviate the loss of information caused by the extreme reduction in dimensions, e.g., 20,480D→85D on AWA2.

3.2 Constrained Part Attention Branch

Attention Parts Generation. We leverage the soft spatial attention [52] to map image x into a set of K part features. Specifically, suppose the last convolutional feature map w.r.t. x is $Z(x) \in \mathbb{R}^{H\times W\times C}$, with H,W,C being its height, width, and channel number, respectively. Then, K attention masks $\{M_i(x)\}_{i=1}^K$ are obtained by a 1×1 convolution G on $Z(x)$ and a Sigmoid thresholding:

$$\mathcal{M} = \text{Sigmoid}(G(Z(x))) \in \mathbb{R}^{H\times W\times K}, \quad M_i(x) = \mathcal{M}[:,:,i], \tag{1}$$

where $M_i(x) \in \mathbb{R}^{H\times W}$ is the ith attention mask of input x. Based on these masks, we obtain K corresponding attentive feature maps $\{T_i(x)\}_{i=1}^K$ w.r.t. $Z(x)$:

$$T_i(x) = Z(x) \odot R(M_i(x)), \tag{2}$$

where R reshapes the input to be the same shape as $Z(x)$, $\odot$ is an element-wise multiplication and $T_i(x) \in \mathbb{R}^{H\times W\times C}$. Finally, we apply global max-pooling to each $T_i(x)$, and thus get K part features $\{f_i(x)\}_{i=1}^K$ with $f_i(x) \in \mathbb{R}^C$.

$\{f_i(x)\}_{i=1}^K$ have two functions: 1) They are concatenated as a vector $\boldsymbol{f} \in \mathbb{R}^{KC}$ (Fig. 2), which is connected to the bottleneck layer and then the semantic space. Finally the semantic layer output is supervised by the *transfer* and *balance* losses (§3.4). 2) They are taken as nodes and used to construct region graph, which is fed to GCNs [21] in the PRR branch for parts relation reasoning (§3.3).

Constrained Attention Masks. To discover more compact and divergent parts, we follow [78, 80], which constrain the attention masks from the channel clustering. Here, we constrain masks from spatial attention. Specifically, the compact loss and divergent loss for K masks $\{M_i(x)\}_{i=1}^K$ (we drop x for ease of reading) on n_b batch data are:

$$\begin{aligned}
\mathcal{L}_{\text{cpt}} &= \frac{1}{K\times n_b}\sum_{j=1}^{n_b}\sum_{i=1}^{K}\sum_{h,w}\|M_i^{h,w} - \hat{M}_i{}^{h,w}\|_2^2, \\
\mathcal{L}_{\text{div}} &= \frac{1}{K\times n_b}\sum_{j=1}^{n_b}\sum_{i=1}^{K}\sum_{h,w} M_i^{h,w}\tilde{M}_i{}^{h,w},
\end{aligned} \tag{3}$$

where $\hat{M}_i$ is an ideal peaked attention map for the ith part; $\tilde{M}_i{}^{h,w}=\max_{j\neq i} M_j^{h,w}$ is the maximum activation of other masks at coordinate (h, w).

3.3 Parts Relation Reasoning Branch

Each of these acquired K part features $\{f_i(x)\}_{i=1}^K$ represents one attended region. When humans see these image regions (Fig. 1(c)), they can easily tell the appearance relationships among them. To imitate such human behavior in linking image regions, we employ GCN [21] to perform region-based relation modeling, which leads to the PRR branch (together with the afterward operations in bottom stream of Fig. 2). As validated in the experiments of §4.3 and §4.4, parts relation reasoning can help RGEN to achieve an improved performance.

We now construct a region graph $\Gamma \in \mathbb{R}^{K\times K}$ (with K part features as its K nodes) for each input image. In Γ, we have a high confidence edge between similar regions ("head"-"head") and a low confidence edge between dissimilar regions ("head"-"leg") (Fig.1(c)). Specifically, we first conduct l_2-normalization on each $f_i(x)$. Then, the dot-product is leveraged to calculate the pairwise similarity:

$$\Gamma_{ij} =< f_i(x), f_j(x) > . \tag{4}$$

In this case, the dot-product calculation is equal to the cosine similarity metric and the graph has self-connections as well. We further calculate the degree matrix D of Γ with $D_{ii} = \sum_{j=1}^{K} \Gamma_{ij}$.

Given input as the region graph, we leverage GCN to perform reasoning on this graph. Specifically, we use a two-layer GCN propagation that is defined as:

$$F^{(l+1)} = \sigma(D^{-1}\Gamma F^{(l)} W^{(l)}), l = 0, 1, \tag{5}$$

where $F^{(0)} \in \mathbb{R}^{K\times C}$ are the stacked K part features, C is their dimension, $W^{(l)}, l = 0, 1$ are learnable parameters, and σ is the Relu activation function.

Finally, as CPA branch, the updated features $F^{(2)} \in \mathbb{R}^{K\times C}$ by GCNs further undergo a concatenation, a bottleneck layer and an embedding to the semantic space. In this case, the guidance losses are again the *transfer* and *balance* losses.

Here, GCN [21] is desirable due to: 1) It transfers original part features into new ones ($F^{(2)}$) by modeling parts relations automatically. 2) The parameters $W^{(l)}$ are jointly learned with the guidance of attributes. 3) It is entirely different from GCN with word embeddings as inputs [45, 20], which learns visual classifier for each class for ZSL. To the best of our knowledge, this represents the first time GCN-based parts relation reasoning is used to tackle ZSL and GZSL tasks.

3.4 The Transfer and Balance Losses

To make ZSL and GZSL feasible, the achieved features ($\mathcal{E}(x)$ in Fig. 2) should be further embedded into a certain subspace. In this paper, we utilize semantic space as the embedding space. As such, given the ith seen image x_i^s and its ground-truth semantic vector $a_*^s \in \mathcal{A}^s$, suppose its embedded feature is collectively denoted as $\mathcal{E}(x_i^s)$, which equals to the concatenated rows of $F^{(2)}$ in Eq. (5) ($\boldsymbol{\theta}$ in Fig. 2) or the concatenated K part features ($\boldsymbol{f}$ in Fig. 2).

Revisit the ACE Loss. To associate image x_i^s with its true attribute information, the compatibility score τ_i^* is formulated as [2, 26, 42, 33, 52, 80]:

$$\tau_i^* = \mathcal{E}(x_i^s)\boldsymbol{W}a_*^s, \tag{6}$$

where $\boldsymbol{W}$ are the embedding weights that need to be learned jointly, which is a two-layer MLP in our implementation (Fig. 2). Considering τ_i^* as a classification score in the cross-entropy (CE) loss, for seen data from a batch, the Attribute incorporated CE loss (ACE) becomes:

$$\mathcal{L}_{\text{ACE}} = -\frac{1}{n_b}\sum_{i=1}^{n_b} \log \frac{\exp(\tau_i^*)}{\sum_{a_j^s\in\mathcal{A}^s}\exp(\tau_i^j)}, \tag{7}$$

where $\tau_i^j = \mathcal{E}(x_i^s)\boldsymbol{W}a_j^s$, $j = 1, \cdots, C^s$ are the scores on C^s seen semantic vectors.

The Transfer Loss. Eq. (7) introduces $\mathcal{A}^s$ for end-to-end training; however, there are two drawbacks: 1) The learned models are still biased towards seen classes, which is a common issue in ZSL and GZSL; and 2) The performances of these deep models are inferior on GZSL [80, 52]. To alleviate these problems further, we incorporate unseen attributes $\mathcal{A}^u$ into RGEN.

In particular, we first define the l_2-normalized attribute matrix w.r.t. these C^s seen classes and C^u unseen classes as $A \in \mathbb{R}^{Q\times C^s}$ and $B \in \mathbb{R}^{Q\times C^u}$, respectively. Then, we leverage least square regression (LSR) to obtain the reconstruction coefficients $V \in \mathbb{R}^{C^u\times C^s}$ of each seen class attribute w.r.t. all unseen class attributes: $V = (B^\mathsf{T}B + \beta I)^{-1}B^\mathsf{T}A$, which is obtained by solving $\min_V \|A - BV\|_F^2 + \beta\|V\|_F^2$. The ith column of V represents the contrasting class similarity of a_i^s w.r.t. B. To this end, during RGEN training, besides Eq. (7), we propose the following loss:

$$\mathcal{L}_{\text{contra}} = -\frac{1}{n_b}\sum_{i=1}^{n_b}\sum_{j=1}^{C^u} v_{jy_i}\log\widetilde{\zeta_{ij}} + (1 - v_{jy_i})\log(1 - \widetilde{\zeta_{ij}}), \tag{8}$$

where $\zeta_{ij}{=}\mathcal{E}(x_i^s)\boldsymbol{W}a_j^u$, $j{=}1,\cdots,C^u$ are the scores w.r.t. C^u unseen semantic vectors for x_i^s, $\widetilde{\zeta_{ij}}$ is the softmax-layer normalization of ζ_{ij} and y_i is the column location in V w.r.t. ground-truth semantic vector of x_i^s. We formally call the loss combining $\mathcal{L}_{\text{contra}}$ and $\mathcal{L}_{\text{ACE}}$ the *transfer* loss:

$$\mathcal{L}_{\text{transfer}} = \mathcal{L}_{\text{ACE}} + \lambda_1\mathcal{L}_{\text{contra}}. \tag{9}$$

The second term in Eq. (9) is related to [19] but differs from it as follows: 1) Our calculation of the prediction score (ζ_{ij}) is based on an end-to-end trained deep net and the compatibility score. 2) We calculate the contrasting class similarity using LSR regression while in [19] they use sparse coding.

Notably, we implement the *transfer* loss as a fully-connected layer by freezing the weights as $[A, B]{\in}\mathbb{R}^{Q\times(C^u+C^s)}$ during the training phase (Fig. 2). In this way, the seen and unseen attributes can guide the discovery of attention parts, and the relation reasoning among them.

The Balance Loss. To tackle the challenge of extreme domain bias in GZSL, especially encountered in end-to-end models [52, 80], we propose a *balance* loss by pursuing the maximum response consistency, among seen and unseen outputs.

Specifically, given the input seen sample x_i^s, we can get its prediction scores on seen class and unseen class attributes as $P_i^s = A^\mathsf{T}\boldsymbol{W}^\mathsf{T}\mathcal{E}(x_i^s)^\mathsf{T} \in \mathbb{R}^{C^s\times 1}$ and $P_i^u = B^\mathsf{T}\boldsymbol{W}^\mathsf{T}\mathcal{E}(x_i^s)^\mathsf{T} \in \mathbb{R}^{C^u\times 1}$, respectively. To balance these scores from the two sides (seen and unseen), the *balance* loss is proposed for batch data, as follows:

$$\mathcal{L}_{\text{balance}} = \frac{1}{n_b}\sum_{i=1}^{n_b}\|\max P_i^s - \max P_i^u\|_2^2, \tag{10}$$

where $\max P$ outputs the maximum value of the input vector P. The *balance* loss is only utilized for GZSL, and not ZSL, since balancing is not required when only unseen test images are available.

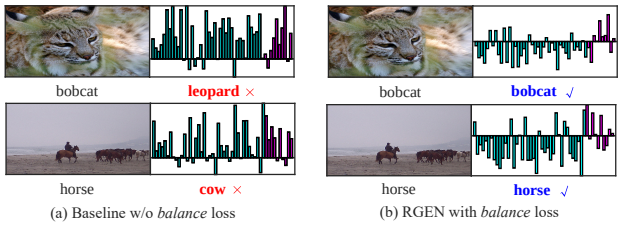

Fig. 3. Cyan and magenta bars are the predicted scores (before the softmax-layer in real-world model) on seen and unseen classes, respectively. Domain bias in (a) has been well addressed by adding our *balance* loss (see (b)).

The intuitions of balancing the predictions between seen (cyan bars in Fig. 3(a)) and unseen (magenta bars in Fig. 3(a)) outputs are two-folds: 1) From the perspective of prediction scaling for end-to-end networks, since no available training data for producing responses on these unseen locations (extreme training data imbalance), we want to balance the numerical scales between seen and unseen predictions. 2) Since some unseen test samples are correctly classified if we only consider their prediction scores on unseen locations (e.g., "zebra" under GZSL in Fig.1(b)), we want to rescue these misclassified samples. Fig. 3 is a real-world example, where we feed unseen images from AWA2 to the RGEN GZSL model (trained with *balance* loss) and its baseline w/o *balance* loss to observe the changing of the predicted scores.

3.5 Training Objective

As we have two branches (the CPA and the PRR), both are guided by our proposed *transfer* and *balance* losses during end-to-end training. However, we have only one stream of data as the input of our net, i.e., the backbone is shared. As such, the final loss for our proposed RGEN is as follows:

$$\mathcal{L}_{\text{RGEN}} = \eta_1 \mathcal{L}_{\text{CPA}} + \eta_2 \mathcal{L}_{\text{PRR}} + \eta_3 \mathcal{L}_{\text{cpt}} + \eta_4 \mathcal{L}_{\text{div}}. \tag{11}$$

The formulations of $\mathcal{L}_{\text{CPA}}$ and $\mathcal{L}_{\text{PRR}}$ are the same as follows:

$$\underbrace{\mathcal{L}_{\text{ACE}} + \lambda_1 \mathcal{L}_{\text{contra}}}_{\mathcal{L}_{\text{transfer}}} + \lambda_2 \mathcal{L}_{\text{balance}}, \tag{12}$$

where λ_1 and λ_2 again takes the same values for the two branches. The difference between $\mathcal{L}_{\text{CPA}}$ and $\mathcal{L}_{\text{PRR}}$ lies in that the concatenated embedding features $\boldsymbol{f}$ and $\boldsymbol{\theta}$ (Fig. 2) come from $\{f_i(x)\}_{i=1}^{K}$ and $F^{(2)}$ (Eq.(5)) for them, respectively. Note that, we take η_1=0.9, η_2=0.1, η_3=1.0, and η_4=1e-4 for all datasets. The selections of λ_1 and λ_2 are further detailed in §4.2.

3.6 Zero-Shot Prediction

In RGEN framework, the unseen test image x^u is predicted in a fused manner. After obtaining the embedding features of x^u in the semantic space w.r.t. CPA

and PRR branches, denoted as, $\psi_{\text{CPA}}(x^u)$ and $\psi_{\text{PRR}}(x^u)$, we calculate their fused result by the same combination coefficients (η_1, η_2) as the training phase, and then predict its label by:

$$y^{u*} = \arg \max_{c \in \mathcal{Y}^{\mathcal{U}}/\mathcal{Y}} (\eta_1 \psi_{\text{CPA}}(x^u) + \eta_2 \psi_{\text{PRR}}(x^u))^{\mathsf{T}} a_c^u. \tag{13}$$

In Eq. (13), $\mathcal{Y}^{\mathcal{U}}/\mathcal{Y}$ corresponds to ZSL/GZSL respectively. In our ablations, we show the performances when setting different combinations of η_1 and η_2 (Fig. 6).

4 Experiments

4.1 Datasets and Settings

We use four standard ZSL and GZSL datasets, i.e., SUN [36], CUB [44], AWA2 [50], and APY [12] to evaluate our RGEN. We use the Proposed Split (PS) [50] for evaluation, as this setting is more strict and does not contain any class overlapping with ImageNet classes [50]. Since images of AWA are not accessible, AWA2 is used instead. The details of these datasets can be found in [50].

RGEN is an end-to-end trainable embedding method. As such, it is fair to compare it with the same types of end-to-end models ([33, 42, 26, 80, 52, 28]). However, to comprehensively review the performance gains of RGEN over other methods, we further compare it with other non end-to-end methods (including the two-stage feature generation methods which are parallel solutions for tackling ZSL and GZSL), and these methods are based on the same ResNet101 features.

4.2 Implementation and Parameters

Almost all compared methods use the 2,048D ResNet101 features. As such, we use ResNet101 in Fig. 2 as our backbone net [17]. The size of input images for the four datasets is 224×224, which makes the size of the last convolutional feature map as $2048 \times 7 \times 7$. For all datasets, the RGEN is trained for a maximum of 40 epochs, with an initial learning rate of 0.001. The architecture for GCN is 2048D-Relu(1024D)-2048D for all datasets. Except for CUB (which has a higher 312D attributes and no bottleneck layer), the datasets all leverage a 4096D bottleneck layer before projecting to semantic space.

The parameters $\eta_1, \eta_2, \eta_3, \eta_4$ are fixed, as stated in §3.5 for all four datasets. The number of parts K is fixed as 10 and β is fixed as 5, for all used datasets. λ_1 is selected from $\{0.001, 0.01, 0.05, 0.07, 0.1\}$ and λ_2 from $\{0.01, 0.05, 0.07, 0.1\}$.

4.3 Zero-Shot Recognition

Mean Class Accuracy (MCA) is adopted as the evaluation metric for ZSL [50]. Table 1 shows the results. As can be seen, **i)** RGEN consistently outperforms most state-of-the-arts by a clear margin and performs best on four datasets among end-to-end models. For instance, RGEN achieves a MCA of 76.1% on

Table 1. ZSL and GZSL results (%) on used datasets. Results with underlines are obtained by further using extra word embeddings, however, our methods only utilize attributes. If there are no results on AWA2, results on AWA are shown. The best, the second best, and the third best results are marked in red, blue, and bold, respectively.

Methods	CUB				SUN				AWA2				APY			
	ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL		
	MCA	ts	tr	H	MCA	ts	tr	H	MCA	ts	tr	H	MCA	ts	tr	H
Non End-to-End																
CONSE(NeurIPS'14) [34]	34.3	1.6	72.2	3.1	38.8	6.8	39.9	11.6	44.5	0.5	90.6	1.0	26.9	0.0	**91.2**	0.0
CMT(NeurIPS'13) [41]	34.6	7.2	49.8	12.6	39.9	8.1	21.8	11.8	37.9	0.5	90.0	1.0	28.0	1.4	**85.2**	2.8
SSE(ICCV'15) [72]	43.9	8.5	46.9	14.4	51.5	2.1	36.4	4.0	61.0	8.1	82.5	14.8	34.0	0.2	78.9	0.4
LATEM(CVPR'16) [48]	49.3	15.2	57.3	24.0	55.3	14.7	28.8	19.5	55.8	11.5	77.3	20.0	35.2	0.1	73.0	0.2
ALE(TPAMI'13) [3]	54.9	23.7	62.8	34.4	58.1	21.8	33.1	26.3	62.5	14.0	81.8	23.9	39.7	4.6	73.7	8.7
DEVISE(NeurIPS'13) [14]	52.0	23.8	53.0	32.8	56.5	16.9	27.4	20.9	59.7	17.1	74.7	27.8	**39.8**	4.9	76.9	9.2
SJE(CVPR'15) [4]	53.9	23.5	59.2	33.6	53.7	14.7	30.5	19.8	61.9	8.0	73.9	14.4	32.9	3.7	55.7	6.9
ESZSL(ICML'15) [39]	53.9	12.6	63.8	21.0	54.5	11.0	27.9	15.8	58.6	5.9	77.8	11.0	38.3	2.4	70.1	4.6
SYNC(CVPR'16) [7]	55.6	11.5	70.9	19.8	56.3	7.9	**43.3**	13.4	46.6	10.0	90.5	18.0	23.9	7.4	66.3	13.3
SAE(CVPR'17) [23]	33.3	7.8	54.0	13.6	40.3	8.8	18.0	11.8	54.1	1.1	82.2	2.2	8.3	0.4	**80.9**	0.9
PSR(CVPR'18) [5]	56.0	24.6	54.3	33.9	61.4	20.8	37.2	26.7	63.8	20.7	73.8	32.3	38.4	13.5	51.4	21.4
DEM(CVPR'17) [66]	51.7	19.6	57.9	29.2	40.3	20.5	34.3	25.6	67.1	30.5	86.4	45.1	35.0	11.1	75.1	19.4
RN(CVPR'18) [58]	55.6	38.1	61.4	47.0	–	–	–	–	64.2	30.0	**93.4**	45.3	–	–	–	–
SP-AEN(CVPR'18) [9]	55.4	34.7	70.6	46.6	59.2	24.9	38.6	30.3	58.5	23.3	90.9	37.1	24.1	13.7	63.4	22.6
IIR(ICCV'19) [6]	63.8	30.4	65.8	41.2	63.5	22.0	34.1	26.7	67.9	17.6	87.0	28.9	–	–	–	–
TCN(ICCV'19) [19]	59.5	52.6	52.0	52.3	61.5	31.2	37.3	34.0	**71.2**	**61.2**	65.8	63.4	38.9	24.1	64.0	**35.1**
Feature Generation Methods																
f-CLSWGAN(CVPR'18) [49]	57.3	43.7	57.7	49.7	60.8	42.6	36.6	39.4	68.2	57.9	61.4	59.6	–	–	–	–
cycle-CLSWGAN(ECCV'18) [13]	58.4	45.7	61.0	52.3	60.0	49.4	33.6	**40.0**	66.3	56.9	64.0	60.2	–	–	–	–
f-VAEGAN-D2 w/o ft(CVPR'19) [51]	61.0	48.4	60.1	53.6	**64.7**	**45.1**	38.0	**41.3**	71.1	57.6	70.6	63.5	–	–	–	–
f-VAEGAN-D2 w ft(CVPR'19) [51]	**72.9**	**63.2**	75.6	**68.9**	**65.6**	**50.1**	37.8	**43.1**	70.3	57.1	76.1	**65.2**	–	–	–	–
End-to-End																
SCoRe(CVPR'17) [33]	62.7	–	–	–	–	–	–	–	61.6	–	–	–	–	–	–	–
QFSL(CVPR'18) [42]	58.8	33.3	48.1	39.4	56.2	30.9	18.5	23.1	63.5	52.1	72.8	60.7	–	–	–	–
LDF(CVPR'18) [26]	67.5	26.4	**81.6**	39.9	–	–	–	–	65.5	9.8	87.4	17.6	–	–	–	–
SGMA(NeurIPS'19) [80]	71.0	36.7	71.3	48.5	–	–	–	–	68.8	37.6	87.1	52.5	–	–	–	–
AREN(CVPR'19) [52]	71.8	38.9	**78.7**	52.1	60.6	19.0	**38.8**	25.5	67.9	15.6	**92.9**	26.7	39.2	9.2	76.9	16.4
LFGAA(ICCV'19) [28]	67.6	36.2	**80.9**	50.0	61.5	18.5	**40.0**	25.3	68.1	27.0	**93.4**	41.9	–	–	–	–
RGEN w/o PRR (Ours)	**75.0**	**61.4**	68.5	**64.7**	63.4	42.7	31.5	36.2	**72.5**	**64.1**	76.4	**69.7**	**43.9**	**29.2**	48.0	**36.3**
RGEN (Ours)	**76.1**	**60.0**	73.5	**66.1**	**63.8**	**44.0**	31.7	36.8	**73.6**	**67.1**	76.5	**71.5**	**44.4**	**30.4**	48.1	**37.2**

CUB, which sets a new state-of-the-art on this dataset by a large margin than the counterparts. However, the performance gains on CUB/AWA2 are better than SUN/APY, this is because: 1) image number per class for SUN is about 20, which limits the RGEN training; 2) the prepared images for APY usually have an extreme aspect ratio, thus hindering the RGEN training. **ii)** The parts relation reasoning branch contributes to the performance improvements, e.g., the performance of RGEN w/o PRR is 75.0% and 72.5% on CUB and AWA2, respectively. This indicates that the parts relation reasoning must discover some underlying information that assists in semantic transfer, though the PRR branch alone does not achieve such a high MCA (Component Analysis in Table 1).

4.4 Generalized Zero-Shot Recognition

For GZSL, the searched label space includes both seen and unseen classes [50]. The evaluation for GZSL is different from ZSL as follows: The MCAs of the unseen/seen test samples are denoted as ts/tr, respectively. Then their Harmonic mean H is defined as $\mathrm{H} = 2 \times \mathrm{tr} \times \mathrm{ts}/(\mathrm{tr} + \mathrm{ts})$. The H score is the key evaluation criterion for GZSL [50], since we want to correctly classify both seen/unseen test images as many as possible (i.e., a higher H score) in the real-world application.

Table 2. ZSL and GZSL results (%) with different GCN structures on CUB.

GCN Layers	ZSL	GZSL			Best GCN Structures	
	MCA	tr	ts	H	ZSL	GZSL
One-layer	75.2	69.7	61.5	65.4	2048-256	2048-128
Two-layer	76.1	60.0	73.5	66.1	2048-1024-2048	2048-1024-2048
Three-layer	75.1	68.5	59.3	63.6	2048-1024-1024-2048	2048-256-256-2048

We conclude from Table 1: **i)** Our RGEN achieves the best Hs compared to its end-to-end counterparts, e.g, we achieve a 71.5 H on AWA2, which represents the current best result. **ii)** The PRR branch can also effectively boost the performances of RGEN under GZSL. **iii)** Our *balance* loss contributes most to the performance improvements (Component Analysis in Table 4). **iv)** Compared with the two-stage feature generation method f-VAEGAN-D2 with fine-tuning [51], which belongs to a parallel solution for ZSL, performances of RGEN (w/o feature generating) are still on par, i.e., on four datasets, we achieve 3/4 better ZSL results, and 2/4 better H under GZSL. Notably, our H score on CUB is worse than [51], however, the latter further uses extra word embeddings [51].

4.5 Ablations

Effects of η_1 and η_2. For RGEN training (Eq. (11)) and testing (Eq. (13)), η_1 and η_2 have the same values in order to keep training and testing consistent. By taking their values from $\{0.0, 0.1, 0.2, \cdots, 0.9, 1.0\}$ and constraining η_1+η_2=1.0, we observe the MCA of RGEN w.r.t. different values of (η_1,η_2) for ZSL (Fig. 6). We find that a small η_2 is better for assisting the RGEN model and, as such, we set $(\eta_1, \eta_2) = (0.9, 0.1)$ for all datasets.

Parts Number. K is fixed to 10 in all our experiments but we vary it from $\{1, 2, 5, 10, 15\}$ to observe the performances of our full RGEN model under both ZSL/GZSL. The results in Fig. 4 show that MCA (of ZSL/tr/ts) and H are stable with a small K, and K=10 is suitable for achieving satisfactory results.

Transfer Loss Coefficient. We show the results of MCA (of ZSL/tr/ts), and H when varying λ_1 over $\{0.0001, 0.001, 0.01, 0.05, 0.07, 0.1\}$ under ZSL/GZSL for RGEN. The results (Fig. 5) are stable for small values from $[0.001, 0.05]$.

Balance Loss Coefficient. The Balance loss is only used under GZSL training; therefore, we vary the value of λ_2 from $\{0.0, 0.01, 0.05, 0.07, 0.1\}$ and observe the MCA (of tr/ts) and H under these values. Fig. 7 shows that a smaller coefficient always achieves better H/ts (with little sacrifice on tr) than the model w/o *balance* regularization (λ_2=0.0) and the overall changing tendency is stable.

GCN Architecture. We fix GCN in PRR branch as a two-layer one (2048-Relu(1024)-2048). We further investigate the influence of one- and three-layer GCN on ZSL/GZSL. Specifically, for one- and three-layer GCN, we vary the node dimension (of the output/middle layer) from $\{128, 256, 512, 1024, 2048\}$ to determine their best results, for fair comparisons with ours. Their best searched architectures are also shown in Table 2, which indicates that a two-layer GCN can better model the parts relation collaboratively with other parameters.

Table 3. Component analysis (MCA) of the best RGEN ZSL model.

ACE Loss	✔	✔			✔			
CD Regularization		✔		✔	✔			✔
Transfer Loss			✔	✔		✔	✔	
PRR Branch	✔				✔	✔	✔	
CUB	69.6	71.3	72.0	74.2	75.0	73.3	75.6	76.1
AWA2	70.1	69.4	69.9	72.3	72.5	70.4	73.0	73.6
SUN	59.8	62.0	62.4	63.1	63.4	63.1	63.3	63.8
APY	37.5	39.5	39.9	43.7	43.9	39.7	43.8	44.4

Table 4. Component analysis (H) of the best RGEN GZSL model.

Transfer Loss	✔	✔	✔	✔	✔	✔	✔	✔
CD Regularization		✔	✔			✔	✔	
PRR Branch			✔	✔			✔	✔
Balance Loss					✔	✔	✔	✔
CUB	37.2	38.6	39.6	38.3	64.6	64.7	66.1	65.1
AWA2	12.5	14.1	14.7	14.9	69.9	69.7	71.5	71.3
SUN	21.2	23.7	24.3	23.6	35.9	36.4	36.8	36.7
APY	15.2	15.6	16.4	15.8	36.8	36.3	37.2	37.7

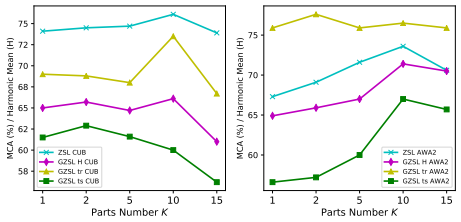

Fig. 4. (MCA,H,tr,ts)-K curves.

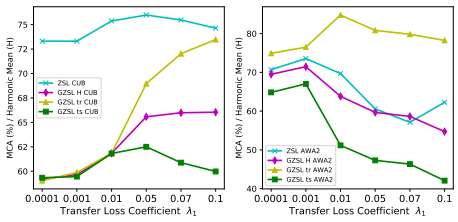

Fig. 5. (MCA,H,tr,ts)-λ_1 curves.

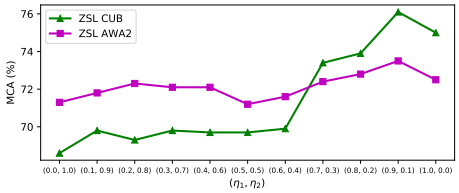

Fig. 6. MCA-(η_1, η_2) curves of RGEN.

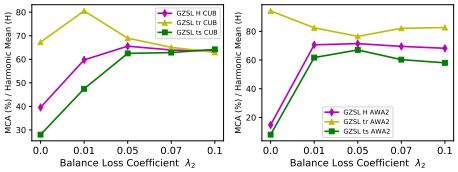

Fig. 7. (H,tr,ts)-λ_2 curves.

Component Analysis. The full RGEN consists of 1) Compact and Divergent (CD) regularizations; 2) Transfer loss; 3) Balance loss (for GZSL); and 4) PRR branch. We assume the PRR branch is trained by 1) and 2), and conduct a component analysis for 1), 2) and 4) under our best ZSL RGEN model (Table 3). As 2) is important for knowledge transfer in ZSL/GZSL, we take this as the indispensable loss for each GZSL model; as such, component analysis for GZSL includes 1), 3), and 4) (Table 4).

4.6 Qualitative Analysis

We use unseen images from CUB under ZSL to visualize the attended parts (Fig. 8). Compared with the baseline (which achieves 71.3% MCA when trained by only the ACE loss, Table 3), RGEN can 1) discover more divergent parts w.r.t. objects; 2) suppress background and redundant foreground regions (maximum mask values in parts #1-4, 9-10 are all small and no similar masks exist among foreground parts #5-8); 3) automatically align the order relationships of different parts (parts #5-8 are consistent w.r.t. different unseen class images). However,

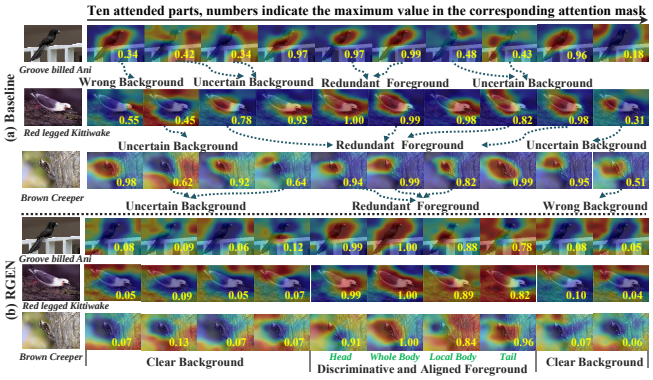

Fig. 8. Visualizations (based on [77]) of unseen images on CUB for (a) baseline and (b) our full RGEN. Three drawbacks exist for baseline, i.e., the attended masks usually contain 1) wrong background; 2) uncertain background; and 3) redundant foreground. By contrary, our RGEN model can address these issues and discover discriminative, divergent and well aligned parts for different unseen class images.

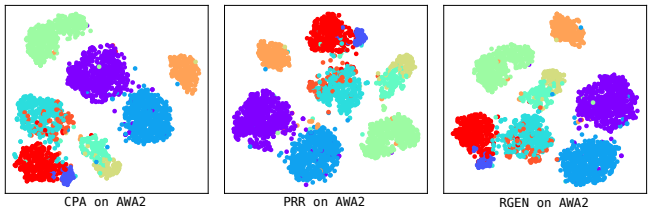

Fig. 9. t-SNE [32] of unseen test images for CPA, PRR, and RGEN.

the baseline has three drawbacks, as shown in Fig. 8. t-SNEs of the unseen test images on AWA2 under GZSL for RGEN and its variants are shown in Fig. 9.

5 Conclusions

The Region Graph Embedding Network (RGEN) is proposed for tackling ZSL and GZSL tasks. RGEN contains the constrained part attention and the parts relation reasoning branches. To guide RGEN training, the *transfer* and *balance* losses are integrated into the framework. The *balance* loss is especially valuable for alleviating the extreme domain bias in the deep GZSL models, providing intrinsic insights for solving GZSL. RGEN sets some new state-of-the-arts for both ZSL and GZSL, on several commonly used benchmarks.

Acknowledgments This work was supported by the National Natural Science Foundation of China (Nos. 61702163 and 61976116), the Fundamental Research Funds for the Central Universities (Nos. 30920021135), and the Key Project of Shenzhen Municipal Technology Research (Nos. JSGG20200103103401723).

References

1. Akata, Z., Malinowski, M., Fritz, M., Schiele, B.: Multi-cue zero-shot learning with strong supervision. In: CVPR (2016)
2. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for attribute-based classification. In: CVPR (2013)
3. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for image classification. In: TPAMI (2016)
4. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: CVPR (2015)
5. Annadani, Y., Biswas, S.: Preserving semantic relations for zero-shot learning. In: CVPR (2018)
6. Cacheux, Y., Borgne, H., Crucianu, M.: Modeling inter and intra-class relations in the triplet loss for zero-shot learning. In: ICCV (2019)
7. Changpinyo, S., Chao, W.L., Gong, B., Sha, F.: Synthesized classifiers for zero-shot learning. In: CVPR (2016)
8. Chao, W.L., Changpinyo, S., Gong, B., Sha, F.: An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In: ECCV (2016)
9. Chen, L., Zhang, H., Xiao, J., Liu, W., Chang, S.F.: Zero-shot visual recognition using semantics-preserving adversarial embedding network. In: CVPR (2018)
10. Elhoseiny, M., Elfeki, M.: Creativity inspired zero-shot learning. In: ICCV (2019)
11. Elhoseiny, M., Zhu, Y., Zhang, H., Elgammal, A.M.: Link the head to the" beak": Zero shot learning from noisy text description at part precision. In: CVPR (2017)
12. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR (2009)
13. Felix, R., Kumar, V.B., Reid, I., Carneiro, G.: Multi-modal cycle-consistent generalized zero-shot learning. In: ECCV (2008)
14. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., T. Mikolov, e.a.: Devise: A deep visual-semantic embedding model. In: NeurIPS (2013)
15. Fu, Y., Hospedales, T.M., Xiang, T., Gong, S.: Transductive multi-view zero-shot learning. In: TPAMI (2015)
16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
18. Jayaraman, D., Grauman, K.: Zero-shot recognition with unreliable attributes. In: NeurIPS (2014)
19. Jiang, H., Wang, R., Shan, S., Chen, X.: Transferable contrastive network for generalized zero-shot learning. In: ICCV (2019)
20. Kampffmeyer, M., Chen, Y., Liang, X., Wang, H., Zhang, Y., Xing, E.: Rethinking knowledge graph propagation for zero-shot learning. In: CVPR (2019)
21. Kipf, T., Welling, M.: Semi-supervised classification with graph convolutional networks. In: arXiv:1609.02907 (2016)
22. Kodirov, E., Xiang, T., Fu, Z., Gong, S.: Unsupervised domain adaptation for zero-shot learning. In: ICCV (2015)
23. Kodirov, E., Xiang, T., Gong, S.: Semantic autoencoder for zero-shot learning. In: CVPR (2017)
24. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009)

25. Li, X., Yang, F., Cheng, H., Liu, W., Shen, D.: Contour knowledge transfer for salient object detection. In: ECCV (2018)
26. Li, Y., Zhang, J., Zhang, J., Huang, K.: Discriminative learning of latent features for zero-shot recognition. In: CVPR (2018)
27. Liu, S., Long, M., Wang, J., Jordan, M.: Generalized zero-shot learning with deep calibration network. In: NeurIPS (2018)
28. Liu, Y., Guo, J., Cai, D., He, X.: Attribute attention for semantic disambiguation in zero-shot learning. In: ICCV (2019)
29. Long, Y., Liu, L., Shen, F., Shao, L., Li, X.: Zero-shot learning using synthesised unseen visual data with diffusion regularisation. In: TPAMI (2017)
30. Lu, X., Wang, W., Ma, C., Shen, J., Shao, L., Porikli, F.: See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In: CVPR (2019)
31. Lu, X., Wang, W., Martin, D., Zhou, T., Shen, J., Luc, V.G.: Video object segmentation with episodic graph memory networks. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
32. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. In: JMLR (2008)
33. Morgado, P., Vasconcelos, N.: Semantically consistent regularization for zero-shot recognition. In: CVPR (2017)
34. Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G.S., Dean, J.: Zero-shot learning by convex combination of semantic embeddings. In: NeurIPS (2014)
35. Palatucci, M., Pomerleau, D., Hinton, G.E., Mitchell, T.M.: Zero-shot learning with semantic output codes. In: NeurIPS (2009)
36. Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: CVPR (2012)
37. Qiao, R., Liu, L., Shen, C., van den Hengel, A.: Less is more: zero-shot learning from online textual documents with noise suppression. In: CVPR (2016)
38. Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: CVPR (2016)
39. Romera-Paredes, B., Torr, P.: An embarrassingly simple approach to zero-shot learning. In: ICML (2015)
40. Shen, Y., Qin, J., Huang, L., Liu, L., Zhu, F., Shao, L.: Invertible zero-shot recognition flows. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
41. Socher, R., Ganjoo, M., Manning, C.D., Ng, A.: Zero-shot learning through cross-modal transfer. In: NeurIPS (2013)
42. Song, J., Shen, C., Yang, Y., Liu, Y., Song, M.: Transductive unbiased embedding for zero-shot learning. In: CVPR (2018)
43. Verma, V.K., Arora, G., Mishra, A., Rai, P.: Generalized zero-shot learning via synthesized examples. In: CVPR (2018)
44. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. In: Technical report (2011)
45. Wang, X., Ye, Y., Gupta, A.: Zero-shot recognition via semantic embeddings and knowledge graphs. In: CVPR (2018)
46. Wu, B., Chen, W., Fan, Y., Zhang, Y., Hou, J., Liu, J., Zhang, T.: Tencent ml-images: A large-scale multi-label image database for visual representation learning. IEEE Access (2019)
47. Wu, B., Jia, F., Liu, W., Ghanem, B., Lyu, S.: Multi-label learning with missing labels using mixed dependency graphs. International Journal of Computer Vision (2018)

48. Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B.: Latent embeddings for zero-shot classification. In: CVPR (2016)
49. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: CVPR (2018)
50. Xian, Y., Schiele, B., Akata, Z.: Zero-shot learning-the good, the bad and the ugly. In: CVPR (2017)
51. Xian, Y., Sharma, S., Saurabh, S., Akata, Z.: f-vaegan-d2: A feature generating framework for any-shot learning. In: CVPR (2019)
52. Xie, G.S., Liu, L., Zhu, F., Zhang, Z., Qin, J., Yao, Y., Shao, L.: Attentive region embedding network for zero-shot learning. In: CVPR (2019)
53. Xie, G.S., Zhang, X.Y., Yang, W., Xu, M., Yan, S., Liu, C.L.: Lg-cnn: From local parts to global discrimination for fine-grained recognition. Pattern Recognition (2017)
54. Xie, G.S., Zhang, Z., Liu, L., Zhu, F., Zhang, X.Y., Shao, L., Li, X.: Srsc: Selective, robust, and supervised constrained feature representation for image classification. IEEE Transactions on Neural Networks and Learning Systems (2019)
55. Xu, H., Saenko, K.: Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In: ECCV (2016)
56. Xu, J., Zhao, R., Zhu, F., Wang, H., Ouyang, W.: Attention-aware compositional network for person re-identification. In: arXiv:1805.03344 (2018)
57. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML (2015)
58. Yang, F.S.Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: CVPR (2018)
59. Yang, G., Liu, J., Xu, J., Li, X.: Dissimilarity representation learning for generalized zero-shot recognition. In: MM (2018)
60. Yao, Y., Shen, F., Xie, G., Liu, L., Zhu, F., Zhang, J., Shen, H.T.: Exploiting web images for multi-output classification: From category to subcategories. IEEE Transactions on Neural Networks and Learning Systems (2020)
61. Yao, Y., Zhang, J., Shen, F., Hua, X., Xu, J., Tang, Z.: Exploiting web images for dataset construction: A domain robust approach. IEEE Transactions on Multimedia (2017)
62. Ye, M., Guo, Y.: Zero-shot classification with discriminative semantic representation learning. In: CVPR (2017)
63. Yu, H., Lee, B.: Zero-shot learning via simultaneous generating and learning. In: NeurIPS (2019)
64. Yu, Y., Ji, Z., Fu, Y., Guo, J., Pang, Y., Zhang, Z.: Stacked semantics-guided attention model for fine-grained zero-shot learning. In: NeurIPS (2018)
65. Yu, Y., Ji, Z., Han, J., Zhang, Z.: Episode-based prototype generating network for zero-shot learning. In: CVPR (2020)
66. Zhang, L., Xiang, T., Gong, S., et al: Learning a deep embedding model for zero-shot learning. In: CVPR (2017)
67. Zhang, L., Wang, P., Liu, L., Shen, C., Wei, W., Zhang, Y., Van Den Hengel, A.: Towards effective deep embedding for zero-shot learning. IEEE Transactions on Circuits and Systems for Video Technology (2020)
68. Zhang, L., Wang, P., Shen, C., Liu, L., Wei, W., Zhang, Y., Van Den Hengel, A.: Adaptive importance learning for improving lightweight image super-resolution network. International Journal of Computer Vision (2020)

69. Zhang, L., Wang, P., Wei, W., Lu, H., Shen, C., van den Hengel, A., Zhang, Y.: Unsupervised domain adaptation using robust class-wise matching. IEEE Transactions on Circuits and Systems for Video Technology (2018)
70. Zhang, L., Wei, W., Bai, C., Gao, Y., Zhang, Y.: Exploiting clustering manifold structure for hyperspectral imagery super-resolution. IEEE Transactions on Image Processing (2018)
71. Zhang, L., Wei, W., Zhang, Y., Shen, C., Van Den Hengel, A., Shi, Q.: Cluster sparsity field: An internal hyperspectral imagery prior for reconstruction. International Journal of Computer Vision (2018)
72. Zhang, Z., Saligrama, V.: Zero-shot learning via semantic similarity embedding. In: ICCV (2015)
73. Zhang, Z., Saligrama, V.: Zero-shot learning via joint latent similarity embedding. In: CVPR (2016)
74. Zhang, Z., Liu, L., Shen, F., Shen, H.T., Shao, L.: Binary multi-view clustering. IEEE transactions on pattern analysis and machine intelligence (2018)
75. Zhao, F., Liao, S., Xie, G.S., Zhao, J., Zhang, K., Shao, L.: Unsupervised domain adaptation with noise resistible mutual-training for person re-identification. In: ECCV (2020)
76. Zhao, F., Zhao, J., Yan, S., Feng, J.: Dynamic conditional networks for few-shot learning. In: ECCV (2018)
77. Zhou, B., Khosla, A., A., L., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR (2016)
78. Zhu, P., Wang, H., Saligrama, V.: Generalized zero-shot recognition based on visually semantic embedding. In: CVPR (2019)
79. Zhu, Y., Elhoseiny, M., Liu, B., Peng, X., Elgammal, A.: A generative adversarial approach for zero-shot learning from noisy texts. In: CVPR (2018)
80. Zhu, Y., Xie, J., Tang, Z., Peng, X., Elgammal, A.: Learning where to look: Semantic-guided multi-attention localization for zero-shot learning. In: NeurIPS (2019)