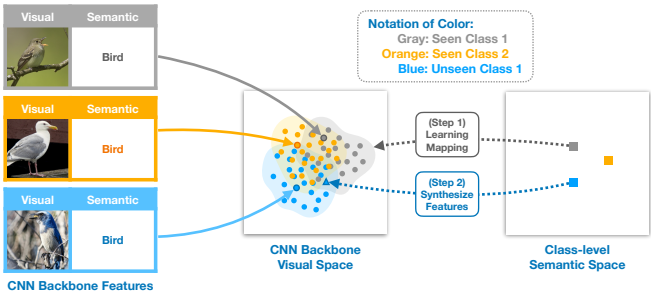
# Semantic Compression Embedding for Generative Zero-Shot Learning

**Ziming Hong**[1*], **Shiming Chen**[1*], **Guo-Sen Xie**[2], **Wenhan Yang**[3], **Jian Zhao**[4], **Yuanjie Shao**[1], **Qinmu Peng**[1] and **Xinge You**[1†]

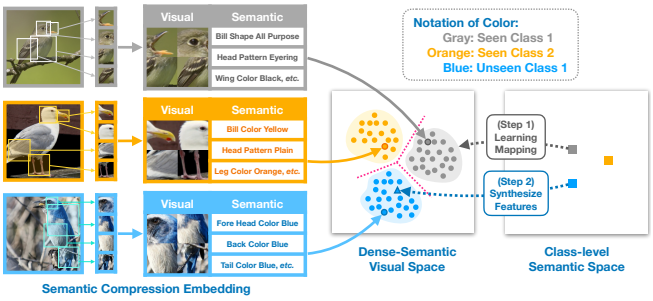
[1]Huazhong University of Science and Technology
[2]Nanjing University of Science and Technology
[3]Nanyang Technological University [4]Institute of North Electronic Equipment
hoongzm@gmail.com, {shimingchen, shaoyuanjie, pengqinmu, youxg}@hust.edu.cn

## Abstract

Generative methods have been successfully applied in zero-shot learning (ZSL) by learning an implicit mapping to alleviate the visual-semantic domain gaps and synthesizing unseen samples to handle the data imbalance between seen and unseen classes. However, existing generative methods simply use visual features extracted from the pre-trained CNN backbone, which lack attribute-level semantic information. Thus, seen classes are indistinguishable and the knowledge transfer from seen to unseen classes is limited. To tackle this issue, we propose a novel Semantic Compression Embedding Guided Generation (SC-EGG) model, which cascades a semantic compression embedding network (SCEN) and an embedding guided generative network (EGGN). The SCEN extracts a group of attribute-level local features, which are further compressed into the new low-dimension visual feature for each sample, thus a dense-semantic visual space is obtained. The EGGN learns a mapping from the class-level semantic space to the dense-semantic visual space, thus improving the discriminability of the synthesized dense-semantic unseen visual features. Extensive experiments on three benchmark datasets, *i.e.*, CUB, SUN and AWA2, demonstrate the significant performance gains of SC-EGG over current state-of-the-art methods and its baselines.




(a) Existing generative ZSLs




(b) Our proposed SC-EGG

Figure 1: Motivation illustration. (a) Existing generative ZSLs learn a mapping from the class-level semantic space to the CNN backbone visual space, which lacks attribute-level semantic information. Thus, the two seen classes are confusing, and the knowledge transfer from seen to unseen classes is also limited. (b) SC-EGG learns a mapping from the class-level semantic space to the more discriminative dense-semantic visual space learned by semantic compression embedding, thus synthesizing the high-quality unseen visual features.

## 1 Introduction

The exciting performance of recent supervised deep learning relies on the massive amount of manually labeled visual samples [Scholkopf and Smola, 2002; He *et al.*, 2016; Cheng *et al.*, 2017]. However, these methods fail to recognize the objects whose classes are not in the training set. Zero-shot learning (ZSL) has been proposed to solve this problem [Palatucci *et al.*, 2009]. Based on the side information shared by seen and unseen classes, ZSL models try to transfer knowledge from seen to unseen classes. Thus, ZSL can recognize the novel unseen classes. According to the classification range, the ZSL task can be grouped into conventional ZSL (CZSL), which merely aims to predict unseen classes, and generalized ZSL (GZSL), which aims at predicting both seen classes and unseen classes.

Recently, generative methods in ZSL (*i.e.*, generative ZSLs) has achieved significant progress [Xian *et al.*, 2018; Xian *et al.*, 2019; Narayan *et al.*, 2020; Chen *et al.*, 2021a; Chen *et al.*, 2021c]. Generative models usually learn an implicit mapping from the class-level semantic space to the visual space using seen data and synthesize unseen visual fea-

*Equal contribution.
†Corresponding author.

tures from the semantic vectors of unseen classes. Thus, ZSL is converted into a standard supervised classification task, which is helpful for tackling the problems of data imbalance between seen and unseen classes, as well as visual-semantic domain gaps.

However, existing generative ZSL methods simply learn a mapping function from the class-level semantic space to the CNN backbone visual space (represented by the visual features directly extracted by a pre-trained CNN backbone, *e.g.*, ResNet101 [He *et al.*, 2016]), which lacks attribute-level semantic information (*e.g.*, "bill color yellow" and "head pattern plain") that is critical for distinguishing seen classes as well as transferring knowledge from seen to unseen classes in ZSL [Xie *et al.*, 2019; Sylvain *et al.*, 2020; Chen *et al.*, 2021c]. For example, as shown in Figure 1(a), the pre-trained CNN backbone may only extract visual features containing semantic information about "Bird" for three samples of different fine-grained bird categories, which are not discriminative enough. As such, the seen classes are confusing, and the knowledge transferred from seen to unseen classes is also limited.

To address the above problems, we propose a novel Semantic Compression Embedding Guided Generation (SC-EGG) model, which cascades a semantic compression embedding network (SCEN) and an embedding guided generative network (EGGN). The SCEN consists of a local embedding network (LEN) and a global embedding network (GEN). The LEN extracts a group of local visual features whose semantics correspond to attributes. Considering the curse of dimension for feature synthesizing, we introduce a new semantic consistent regression loss to enable the GEN to learn the low-dimension global visual feature whose semantic is consistent with the local visual features for each sample. Thus, we compress attribute-level semantic representations into the global visual features and obtain a dense-semantic visual space, as shown in Figure 1(b). The EGGN employs a generative model to learn a mapping from the class-level semantic vectors to the dense-semantic visual space for better knowledge transfer. To alleviate model overfitting to seen classes, we further introduce an embedding guided synthesis loss that takes the trained classifier in GEN to constrain the visual feature synthesizing for both seen and unseen classes in EGGN. Finally, we take the trained EGGN to synthesize an amount of unseen visual features to train a classifier, which is used for ZSL classification.

Our main contributions are summarized as follows:

- We propose a novel ZSL method, termed semantic compression embedding guided generation (SC-EGG) model, to address the problem of attribute-level semantic missing of the CNN backbone visual features, thus further boosting the performance of generative ZSL.
- We propose a new semantic consistent regression loss for semantic compression and an embedding guided synthesis loss for alleviating generative models overfitting to seen classes.
- Extensive experiments on three challenging benchmark datasets, *i.e.*, CUB [Welinder *et al.*, 2010], SUN [Patterson and Hays, 2012] and AWA2 [Xian *et al.*, 2017], demonstrate the significant performance gain of SC-EGG over current state-of-the-art methods and its baseline.

## 2 Related Work

**Embedding-based ZSL.** Embedding-based ZSL methods usually learn a mapping from visual to semantic domains [Lampert *et al.*, 2014; Akata *et al.*, 2016], performing the ZSL classification using nearest-likely strategy according to the distance between itself and class-level semantic descriptors. However, most existing methods are based on global visual features that are not attribute-level semantic-related, resulting in poor discriminative and transferable feature representations. Recently, attention-based local embedding methods [Xie *et al.*, 2019; Huynh and Elhamifar, 2020b; Xu *et al.*, 2020; Han *et al.*, 2021; Chen *et al.*, 2022a; Chen *et al.*, 2022b] were used to enhance the visual feature representations for seen and unseen classes. Unfortunately, these methods learn the ZSL model only on seen classes, inevitably overfitting to seen classes.

**Generative ZSL.** To overcome the limitations of embedding-based ZSL methods, generative ZSL methods employ a generative model to synthesize the unseen visual features for feature augmentation [Xian *et al.*, 2018; Narayan *et al.*, 2020; Chen *et al.*, 2021b]. Felix *et al.* [2018] proposed a conditional Wasserstein GAN (WGAN) that synthesizes visual features by optimizing the Wasserstein distance regularized by a classification loss. In [Xian *et al.*, 2019], the authors introduced an f-VAEGAN framework that combined a WGAN and a VAE to take advantage of the strength of both. Later, lots of generative ZSL methods follow up the f-VAEGAN framework because of its excellent performance [Narayan *et al.*, 2020; Chen *et al.*, 2021a; Yan *et al.*, 2021]. However, these generative methods learn the mapping from semantic vectors to global visual features, which are directly extracted from the pre-trained CNN backbone. Since the global visual features are not semantic-related to the pre-defined attributes in a specific dataset, the learned generative model is poor. As such, the synthesized unseen visual features are not discriminative and transferable.

## 3 Proposed Method

**Problem Definition.** Let $\boldsymbol{x} \in \mathcal{X}$ denotes an input image, $\boldsymbol{y} \in \mathcal{Y}$ denotes the corresponding label. $\mathcal{Y} = \mathcal{Y}^s \cup \mathcal{Y}^u$, where $\mathcal{Y}^s = \{\boldsymbol{y}_1^s, \cdots, \boldsymbol{y}_M^s\}$ denotes the set of $M$ seen classes, $\mathcal{Y}^u = \{\boldsymbol{y}_1^u, \cdots, \boldsymbol{y}_N^u\}$ denotes the set of $N$ unseen classes, and $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$. Both seen and unseen classes have the class-level semantic description vector $\boldsymbol{a}(\boldsymbol{y}_j) = [a_1, \ldots, a_A]^{\mathrm{T}} \in \mathcal{A}, \forall \boldsymbol{y}_j \in \mathcal{Y}^s \cup \mathcal{Y}^u$, which encodes the relationships between all classes, and are available during training. The semantic description vector $\boldsymbol{a}(\boldsymbol{y}_j)$ of class $\boldsymbol{y}_j$ has $A$ elements, each of which corresponds to an attribute. The tasks in CZSL and GZSL are to learn the classifiers $f_{czsl} : \mathcal{X} \rightarrow \mathcal{Y}^u$ and $f_{gzsl} : \mathcal{X} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u$, respectively.

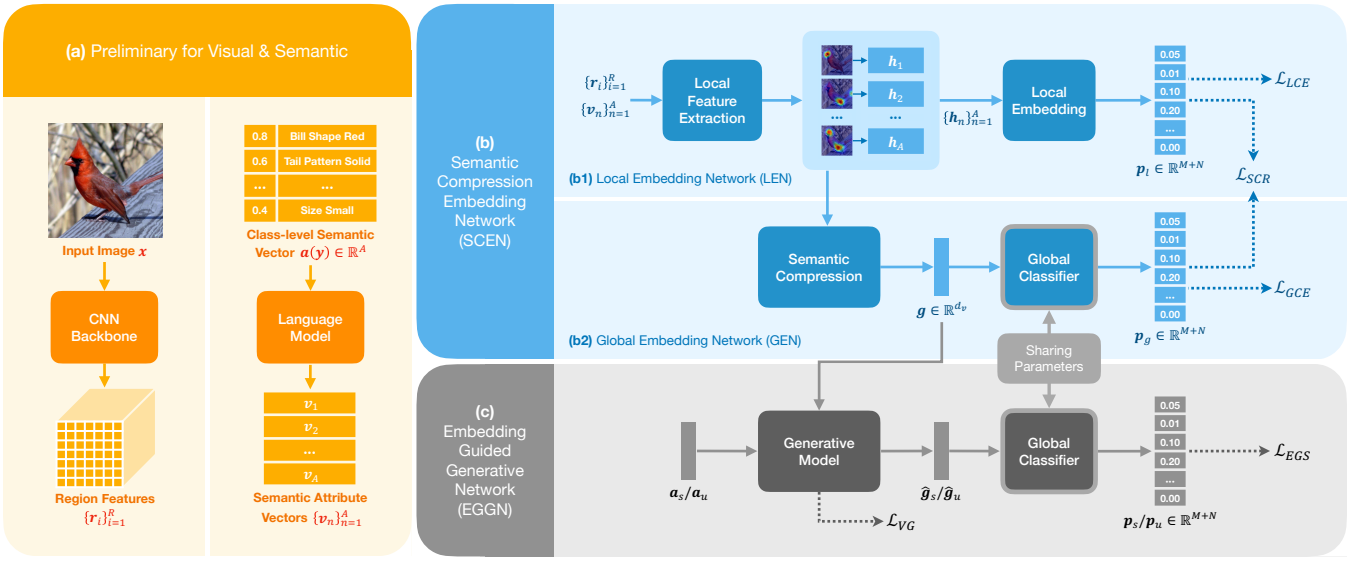



Figure 2: A schematic overview of SC-EGG. (a) Preliminary operations. (b) The semantic compression embedding network (SCEN) contains a local embedding network (b1) that learns local visual features with semantic attribute vectors, and a global embedding network (b2) that learns the dense-semantic global visual feature. (c) The embedding guided generative network (EGGN) contains a generative model that learns a semantic-to-visual mapping, and a classifier (shared with the global classifier in SCEN) that guides feature synthesis.

**Overview.** As shown in Figure 2, the proposed semantic compression embedding guided generation (SC-EGG) model cascades a semantic compression embedding network (SCEN) and an embedding guided generative network (EGGN). Specifically, the SCEN employs a local embedding network (LEN) and global embedding network (GEN) to learn a local-global consistent embedding for semantic compression. The EGGN contains a generative model (TF-VAEGAN [Narayan *et al.*, 2020]) to learn semantic-to-visual mapping guided by a global classifier, which shares parameters with the global classifier in GEN. Preliminarily, we represent the semantic attribute vector of $A$ attributes as $\{\boldsymbol{v}_n\}_{n=1}^A$ using a language model (*e.g.*, GloVe [Pennington *et al.*, 2014]), $\forall \boldsymbol{v}_n \in \mathbb{R}^{d_a}$. In addition, we use the pre-trained CNN backbone to extract a set of region features $\{\boldsymbol{r}_i\}_{i=1}^R$ with $R$ regions for each input image $\boldsymbol{x}$, $\forall \boldsymbol{r}_i \in \mathbb{R}^{d_v}$.

## 3.1 Semantic Compression Embedding Network

Semantic compression embedding network (SCEN) learns a dense-semantic visual space by local-global semantic consistent feature learning, in which the local embedding network (LEN) and the global embedding network (GEN) learn local and global visual features, respectively. It is constrained by a local embedding cross-entropy loss, a global embedding cross-entropy loss, and a semantic consistent regression loss.

### Local Embedding Network

Local embedding network (LEN) firstly locates attribute-level semantic-related image regions to represent a group of local visual features whose semantic is aligned to attributes, using attribute-based attention mechanisms [Huynh and Elhamifar, 2020b]. Thus, the attribute-level semantic-related local features $\{\boldsymbol{h}_n\}_{n=1}^A$ are formulated as:

$$\boldsymbol{h}_n = \sum_{i=1}^{R} \frac{\exp(\boldsymbol{v}_n^{\mathrm{T}} \mathbf{W}_\alpha \boldsymbol{r}_i)}{\sum_{j=1}^{R} \exp(\boldsymbol{v}_n^{\mathrm{T}} \mathbf{W}_\alpha \boldsymbol{r}_j)} \boldsymbol{r}_i, \tag{1}$$

where $\boldsymbol{h}_n \in \mathbb{R}^{d_v}$ denotes the local visual feature whose semantic is aligned with the $n$-th attribute, $\mathbf{W}_\alpha \in \mathbb{R}^{d_a \times d_v}$ is a learnable matrix.

Then, LEN embeds $\{\boldsymbol{h}_n\}_{n=1}^A$ into the semantic space:

$$\begin{aligned} \boldsymbol{e}_l^{\mathrm{T}} &= \mathrm{diag}([\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_A]^{\mathrm{T}}) \mathbf{W}_\beta [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_A] \\ &= \left[\boldsymbol{h}_1^{\mathrm{T}} \mathbf{W}_\beta \boldsymbol{v}_1, \boldsymbol{h}_2^{\mathrm{T}} \mathbf{W}_\beta \boldsymbol{v}_2, \cdots, \boldsymbol{h}_A^{\mathrm{T}} \mathbf{W}_\beta \boldsymbol{v}_A\right], \end{aligned} \tag{2}$$

where $\boldsymbol{e}_l \in \mathbb{R}^A$ denotes the feature embeded in semantic space, $\mathbf{W}_\beta \in \mathbb{R}^{d_v \times d_a}$ is a learnable embedding matrix, and $\mathrm{diag}(\cdot)$ represents the diagonalization operation. After learning the visual-to-semantic embedding, the class prediction result $\boldsymbol{p}_l$ of LEN can be computed based on the similarity between the embeded vector $\boldsymbol{e}_l$ and the class-level semantic vectors $\boldsymbol{a}(\boldsymbol{y}_i)$:

$$\begin{aligned} \boldsymbol{p}_l &= [\boldsymbol{a}(\boldsymbol{y}_1^s), \cdots, \boldsymbol{a}(\boldsymbol{y}_M^s), \boldsymbol{a}(\boldsymbol{y}_1^u), \cdots, \boldsymbol{a}(\boldsymbol{y}_N^u)]^{\mathrm{T}} \boldsymbol{e}_l \\ &= \left[\boldsymbol{e}_l^{\mathrm{T}} \boldsymbol{a}(\boldsymbol{y}_1^s), \cdots, \boldsymbol{e}_l^{\mathrm{T}} \boldsymbol{a}(\boldsymbol{y}_N^u)\right]^{\mathrm{T}}, \end{aligned} \tag{3}$$

where $\boldsymbol{a}(\boldsymbol{y}_i) \in \mathbb{R}^A, \forall \boldsymbol{y}_i \in \mathcal{Y}^s \cup \mathcal{Y}^u$, and $\boldsymbol{p}_l \in \mathbb{R}^{M+N}$ denotes the class prediction score vector.

### Global Embedding Network

Global embedding network (GEN) consists of a semantic compression module $SC$ and a global visual feature classifier $CLS_g$. The $SC$ compresses the attribute-level semantic-related local visual features $\{\boldsymbol{h}_n\}_{n=1}^A$ to the dense-semantic low-dimension global visual feature:

$$\boldsymbol{g} = SC(\{\boldsymbol{h}_n\}_{n=1}^A), \tag{4}$$

where $\boldsymbol{g} \in \mathbb{R}^{d_v}$ denotes the global visual feature. The $SC$, which can be a multilayer perceptron (MLP) with $A$ input units and one output unit, compresses the local features along the dimension of attributes to remove redundant semantic. Then the $CLS_g$ directly embeds the global visual feature to the class-label space:

$$\boldsymbol{p}_g = CLS_g(\boldsymbol{g}), \tag{5}$$

where the $\boldsymbol{p}_g \in \mathbb{R}^{M+N}$ denotes the class prediction score.

**Local Embedding Cross-Entropy Loss**

Aiming at ZSL classification, we take the local embedding cross-entropy loss $\mathcal{L}_{LCE}$ to train the LEN, the $\mathcal{L}_{LCE}$ is formulated as:

$$\mathcal{L}_{LCE} = -\frac{1}{N}\sum_{n=1}^{N}\log\frac{\exp\left(\boldsymbol{p}_l(\boldsymbol{y})\right)}{\sum_{\boldsymbol{y}'\in\mathcal{Y}^s\cup\mathcal{Y}^u}\exp(\boldsymbol{p}_l(\boldsymbol{y}'))}, \tag{6}$$

where $N$ denotes the batch size, $\boldsymbol{p}_l(\boldsymbol{y}')$ is the score predicted by LEN that the $\boldsymbol{x}$ belongs to class $\boldsymbol{y}'$.

**Global Embedding Cross-Entropy Loss**

Simultaneously, we take the global embedding cross-entropy loss $\mathcal{L}_{GCE}$ to train the GEN, the $\mathcal{L}_{GCE}$ can be formulated as:

$$\mathcal{L}_{GCE} = -\frac{1}{N}\sum_{n=1}^{N}\log\frac{\exp\left(\boldsymbol{p}_g(\boldsymbol{y})\right)}{\sum_{\boldsymbol{y}'\in\mathcal{Y}^s\cup\mathcal{Y}^u}\exp(\boldsymbol{p}_g(\boldsymbol{y}'))}, \tag{7}$$

where $N$ denotes the batch size and $\boldsymbol{p}_g(\boldsymbol{y}')$ denotes the score predicted by GEN that the $\boldsymbol{x}$ belongs to class $\boldsymbol{y}'$.

**Semantic Consistent Regression Loss**

In order to guarantee the low-dimension global visual feature $\boldsymbol{g}$ containing attribute-level semantic the same as local visual features $\{\boldsymbol{h}_n\}_{n=1}^{A}$ (*i.e.*, compressing semantic of local features to the global feature), we take the semantic consistent regression loss $\mathcal{L}_{SCR}$ as a constrain for both LEN and GEN:

$$\mathcal{L}_{SCR} = \|\boldsymbol{p}_l - \boldsymbol{p}_g\|_2^2, \tag{8}$$

where $\boldsymbol{p}_l$ and $\boldsymbol{p}_g$ represent the prediction score vector of LEN and GEN, respectively.

## 3.2 Embedding Guided Generative Network

We introduce the embedding guided generative network (EGGN), which learns a mapping from the class-level semantic vector $\boldsymbol{a}_s$ to the dense-semantic visual feature $\boldsymbol{g}_s$ (learned by SCEN), under the optimization of the embedding guided synthesis loss. Based on the semantic compression, the dense-semantic visual feature $\boldsymbol{g}_s$ is low-dimension and attribute-level semantic-related, thus EGGN can synthesize discriminative unseen features $\hat{\boldsymbol{g}}_u$.

The EGGN is based on the TF-VAEGAN [Narayan *et al.*, 2020], including a variational auto-encoder (VAE) and a generative adversarial network (GAN). The VAE consists of an encoder $E$ and a decoder $G$. The GAN consists of a generator $G$, which shares the decoder in VAE, and a discriminator $D$. In addition, a semantic embedding decoder $Dec$ and a feedback module $F$ are introduced to enhance feature synthesizing collectively. The basic loss $\mathcal{L}_{VG}$ of TF-VAEGAN can be formulated as follow:

$$\mathcal{L}_{VG} = \mathcal{L}_{VAE} + \lambda_w\mathcal{L}_{WGAN} + \lambda_r\mathcal{L}_{Rec}, \tag{9}$$

in which $\lambda_w$ and $\lambda_r$ are loss weights and:

$$\begin{aligned}\mathcal{L}_{VAE} =& \mathrm{KL}(E(\boldsymbol{g},\boldsymbol{a})\|p(\boldsymbol{z}|\boldsymbol{a}))- \\ & \mathbb{E}_{E(\boldsymbol{g},\boldsymbol{a})}[\log G(\boldsymbol{z},\boldsymbol{a})],\end{aligned} \tag{10}$$

$$\begin{aligned}\mathcal{L}_{WGAN} =& \mathbb{E}[D(\boldsymbol{g},\boldsymbol{a})] - \mathbb{E}[D(\hat{\boldsymbol{g}},\boldsymbol{a})]- \\ & \lambda\mathbb{E}\left[\left(\|\nabla D(\boldsymbol{g}',\boldsymbol{a})\|_2 - 1\right)^2\right],\end{aligned} \tag{11}$$

$$\mathcal{L}_{Rec} = \mathbb{E}[\|Dec(\boldsymbol{g}) - \boldsymbol{a}\|_1] + \mathbb{E}[\|Dec(\hat{\boldsymbol{g}}) - \boldsymbol{a}\|_1], \tag{12}$$

where KL is the Kullback-Leibler divergence, $p(\boldsymbol{z}|\boldsymbol{a})$ is a prior distribution (assumed to be $\mathcal{N}(0,1)$), $-\log G(\boldsymbol{z},\boldsymbol{a})$ is the visual feature reconstruction loss, $\boldsymbol{g}' = \tau\boldsymbol{g} + (1-\tau)\hat{\boldsymbol{g}}$ with $\tau \sim U(0,1)$ and $\lambda$ is the penalty coefficient. More details of the TF-VAEGAN can be refer to [Narayan *et al.*, 2020].

**Embedding Guided Synthesis Loss**

To encourage EGGN to synthesize the discriminative features and avoid the synthetic unseen features overfitting to seen classes, we introduce an embedding guided synthesis loss based on a classifier, which shares parameters with the global classifier $CLS_g$ in GEN. Specifically, the embedding guided

**Algorithm 1** The algorithm of SC-EGG.

**Input**: the training set $\{\mathcal{X}^s, \mathcal{Y}^s\}$; the testing set $\{\mathcal{X}^u, \mathcal{Y}^u\}$; class-level semantic vectors $\mathcal{A}$; the pre-trained CNN backbone ResNet101; the pre-trained language model GloVe; loss weights (*i.e.*, $\lambda_g$, $\lambda_w$, $\lambda_r$, $\lambda_s$, $\lambda_u$); training epoches for each stage: $E_{S1}$, $E_{S2}$, $E_{S3}$ and $E_C$.
**Output**: the predicted label $c^*$ for the test samples.

Take ResNet101 to extract the region features $\{\boldsymbol{r}_i\}_{i=1}^{R}$ for each sample $\boldsymbol{x}$ in $\mathcal{X}$.
Take GloVe to extract the semantic attribute vectors $\{\boldsymbol{v}_n\}_{n=1}^{A}$ for $A$ attributes.
**while** $iter_1 < E_{S1}$ **do**
  Update parameters of SCEN with Equation (16).
**end while**
**while** $iter_2 < E_{S2}$ **do**
  Update parameters of GEN with Equation (17).
**end while**
**while** $iter_3 < E_{S3}$ **do**
  Update parameters of EGGN with Equation (18).
**end while**
Using the SCEN to extract $\boldsymbol{g}_s$ for seen samples.
Using the EGGN to synthesize $\hat{\boldsymbol{g}}_u$ for unseen classes.
**while** $iter_c < E_C$ **do**
  Update parameters of $f_{czsl}$ (or $f_{gzsl}$) with $\boldsymbol{g}_s$ and $\hat{\boldsymbol{g}}_u$ as inputs.
**end while**
Predict the label $c^*$ of the test samples using the trained classifier $f_{czsl}$ (or $f_{gzsl}$).
**return** the predicted label $c^*$ for the test samples.

| Type | Methods | CUB | | | | SUN | | | | AWA2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CZSL | GZSL | | | CZSL | GZSL | | | CZSL | GZSL | | |
| | | *Acc* | *U* | *S* | *H* | *Acc* | *U* | *S* | *H* | *Acc* | *U* | *S* | *H* |
| Embedding | TCN [Jiang *et al.*, 2019] | 59.5 | 52.6 | 52.0 | 52.3 | 61.5 | 31.2 | 37.3 | 34.0 | 71.2 | 61.2 | 65.8 | 63.4 |
| | AREN [Xie *et al.*, 2019] | 71.8 | 38.9 | **78.7** | 52.1 | 60.6 | 19.0 | 38.8 | 25.5 | 67.9 | 15.6 | **92.9** | 26.7 |
| | DAZLE [Huynh and Elhamifar, 2020b] | 66.0 | 56.7 | 59.6 | 58.1 | 59.4 | **52.3** | 24.3 | 33.2 | 67.9 | 60.3 | 75.7 | 67.1 |
| | RGEN [Xie *et al.*, 2020] | **76.1** | 60.0 | 73.5 | 66.1 | 63.8 | 44.0 | 31.7 | 36.8 | **73.6** | **67.1** | 76.5 | **71.5** |
| | APN [Xu *et al.*, 2020] | 72.0 | **65.3** | 69.3 | **67.2** | 61.6 | 41.9 | 34.0 | 37.6 | 68.4 | 57.1 | 72.4 | 63.9 |
| | CE-GZSL [Han *et al.*, 2021] | **77.5** | 63.9 | 66.8 | 65.3 | 63.3 | 48.8 | 38.6 | 43.1 | 70.4 | 63.1 | 78.6 | 70.0 |
| Generative | f-CLSWGAN [Xian *et al.*, 2018] | 57.3 | 43.7 | 57.7 | 49.7 | 60.8 | 42.6 | 36.6 | 39.4 | 68.2 | 57.9 | 61.4 | 59.6 |
| | f-VAEGAN-D2 [Xian *et al.*, 2019] | 61.0 | 48.4 | 60.1 | 53.6 | 64.7 | 45.1 | 38.0 | 41.3 | 71.1 | 57.6 | 70.6 | 63.5 |
| | LisGAN [Li *et al.*, 2019] | 58.8 | 46.5 | 57.9 | 51.6 | 61.7 | 42.9 | 37.8 | 40.2 | – | – | – | – |
| | TF-VAEGAN [Narayan *et al.*, 2020] | 64.9 | 52.8 | 64.7 | 58.1 | **66.0** | 45.6 | 40.7 | 43.0 | 72.2 | 59.8 | 75.1 | 66.6 |
| | OCD-CVAE [Keshari *et al.*, 2020] | – | 44.8 | 59.9 | 51.3 | – | 44.8 | **42.9** | **43.8** | – | 59.5 | 73.4 | 65.7 |
| | Composer [Huynh and Elhamifar, 2020a] | 69.4 | 56.4 | 63.8 | 59.9 | 62.6 | **55.1** | 22.0 | 31.4 | 71.5 | 62.1 | 77.3 | 68.8 |
| | GCM-CF [Yue *et al.*, 2021] | – | 61.0 | 59.7 | 60.3 | – | 47.9 | 37.8 | 42.2 | – | 60.4 | 75.1 | 60.3 |
| | FREE [Chen *et al.*, 2021a] | – | 55.7 | 59.9 | 57.7 | – | 47.4 | 37.2 | 41.7 | – | 60.4 | 75.4 | 67.1 |
| | HSVA [Chen *et al.*, 2021b] | – | 52.7 | 58.3 | 55.3 | – | 48.6 | 39.0 | 43.3 | – | 56.7 | 79.8 | 66.3 |
| | SDGZSL [Chen *et al.*, 2021c] | 75.5 | 59.9 | 66.4 | 63.0 | – | – | – | – | 72.1 | **64.6** | 73.6 | 68.8 |
| | **SC-EGG (Ours)** | 75.1 | **64.1** | **73.6** | **68.5** | **69.2** | 45.1 | **43.6** | **44.3** | **78.2** | 60.9 | **89.3** | **72.4** |

Table 1: Results (%) of the state-of-the-art CZSL and GZSL on CUB, SUN and AWA2. The best and second-best results are marked in **Red** and **Blue**, respectively. Symbol "–" denotes no results are reported. Methods are categorized into embedding-based ZSLs and generative ZSLs.

synthesis loss $\mathcal{L}_{EGS}$ includes a synthetic seen cross-entropy loss $\mathcal{L}_{SCE}$ and a synthetic unseen cross-entropy loss $\mathcal{L}_{UCE}$, formulated as:

$$\mathcal{L}_{EGS} = \lambda_s \mathcal{L}_{SCE} + \lambda_u \mathcal{L}_{UCE}, \tag{13}$$

where $\lambda_s$ and $\lambda_u$ are loss weights. $\mathcal{L}_{SCE}$ and $\mathcal{L}_{UCE}$ are formulated as:

$$\mathcal{L}_{SCE} = -\frac{1}{N}\sum_{n=1}^{N} \log \frac{\exp\left(\boldsymbol{p}_s(\boldsymbol{y})\right)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^s \cup \mathcal{Y}^u} \exp(\boldsymbol{p}_s(\boldsymbol{y}'))}, \tag{14}$$

$$\mathcal{L}_{UCE} = -\frac{1}{N}\sum_{n=1}^{N} \log \frac{\exp\left(\boldsymbol{p}_u(\boldsymbol{y})\right)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^s \cup \mathcal{Y}^u} \exp(\boldsymbol{p}_u(\boldsymbol{y}'))}, \tag{15}$$

where $N$ denotes the batch size, $\boldsymbol{p}_s = CLS_g(\hat{\boldsymbol{g}}_s)$ and $\boldsymbol{p}_u = CLS_g(\hat{\boldsymbol{g}}_u)$, $\hat{\boldsymbol{g}}_s$ and $\hat{\boldsymbol{g}}_u$ represent the synthetic seen and unseen features, respectively. $\boldsymbol{p}(\boldsymbol{y}')$ denotes the score of the $\boldsymbol{x}$ belonging to class $\boldsymbol{y}'$ predicted by the $CLS_g$.

### 3.3 Model Optimization

We train the SC-EGG in three stages:

**i)** Stage 1: We first train the SCEN by the local and global embedding cross-entropy loss to learn attribute-level semantic-related local features. The loss of stage 1 is formulated as:

$$\mathcal{L}_{S1} = \mathcal{L}_{LCE} + \lambda_g \mathcal{L}_{GCE}. \tag{16}$$

**ii)** Stage 2: At this stage, we train the SCEN by freezing the parameters of LEN and fine-tuning the GEN with the semantic consistent regression loss:

$$\mathcal{L}_{S2} = \mathcal{L}_{SCR}. \tag{17}$$

**iii)** Stage 3: We use the trained SCEN to extract the dense-semantic visual features. Then we train the EGGN to learn a mapping from the class-level semantic space to the dense-semantic visual space. The total loss of stage 3 is:

$$\mathcal{L}_{S3} = \mathcal{L}_{VG} + \mathcal{L}_{EGS}. \tag{18}$$

### 3.4 Classification

After training, we use the SCEN to extract dense-semantic visual features $\boldsymbol{g}_s$ of seen samples and the generator $G$ of EGGN to synthesize unseen features $\hat{\boldsymbol{g}}_u$. Then, we use the $\boldsymbol{g}_s$ and $\hat{\boldsymbol{g}}_u$ to train a CZSL classifier $f_{czsl} : \mathcal{X} \rightarrow \mathcal{Y}^u$ and a GZSL classifier $f_{gzsl} : \mathcal{X} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u$. During inference, test features $\boldsymbol{x}_t$ are processed by SCEN to obtain dense-semantic visual features $\boldsymbol{g}_t$, which are further used as input of classifier $f_{czsl}$ or $f_{gzsl}$ for ZSL prediction. The full procedure of SC-EGG is presented in Algorithm 1.

## 4 Experiments

**Dataset.** We evaluate the proposed SC-EGG on three standard ZSL benchmark datasets: Caltech-UCSD-Birds (CUB) [Welinder *et al.*, 2010], SUN Attribute (SUN) [Patterson and Hays, 2012] and Animals with Attributes2 (AWA2) [Xian *et al.*, 2017]. CUB is a fine-grained dataset which consists of 11,788 images of 200 bird classes. SUN is a fine-grained dataset including 14,340 images from 717 scene classes. AWA2 is a coarse-grained dataset which contains 37,322 images from 50 animal classes.

**Evaluation Protocols.** During testing, we measure the top-1 accuracy both in the CZSL and GZSL tasks following [Xian *et al.*, 2017]. In CZSL, we only predict the unseen classes $\mathcal{Y}_u$ to compute the accuracy of test samples (denoted as $Acc$). In GZSL, we compute the accuracy of the test samples from both seen classes $\mathcal{Y}_s$ (denoted as $S$) and unseen classes $\mathcal{Y}_u$ (denoted as $U$) and their harmonic mean $H = (2 \times S \times U)/(S + U)$.

**Implementation Details.** We use the training splits proposed in [Xian *et al.*, 2018]. ResNet101 [He *et al.*, 2016] (pre-trained on ImageNet) is used for image feature extraction without fine-tuning. We use the Adam as optimizer with $lr = 10^{-4}$ and batchsize $= 64$. We use a single layer FC as the final CZSL or GZSL classifier. Hyperparameters $\lambda_g$, $\lambda_w$, $\lambda_r$, $\lambda_s$ and $\lambda_u$ are respectively set to 1.0, 10.0, 0.01, 0.1 and

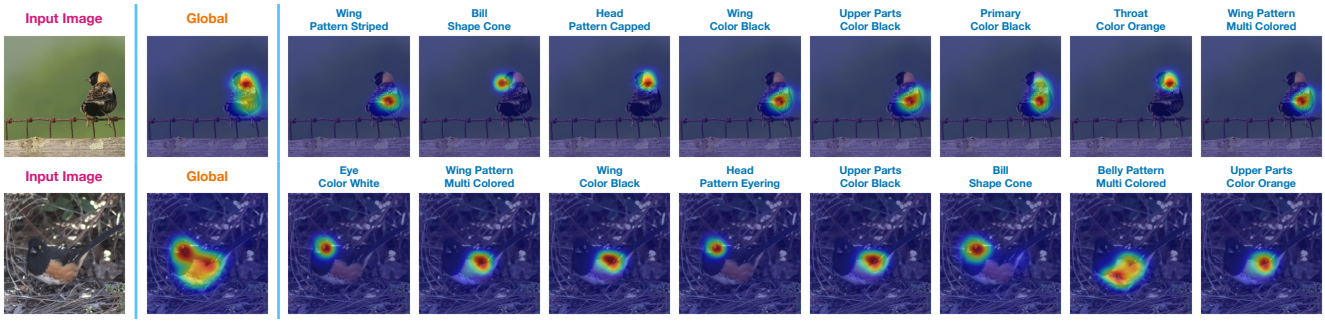



Figure 3: Visualization of attention maps of the SCEN. More results are shown in the online page[1]. (Best viewed in color)

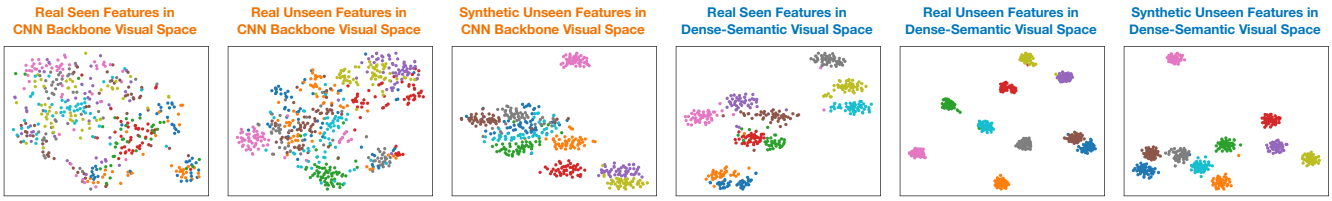



Figure 4: t-SNE visualization on CUB. The results of SUN and AWA2 are presented in the online page[1]. (Best viewed in color)

| Methods | CUB | | | | SUN | | | |
|---|---|---|---|---|---|---|---|---|
| | CZSL | GZSL | | | CZSL | GZSL | | |
| | *Acc* | *U* | *S* | *H* | *Acc* | *U* | *S* | *H* |
| Baseline | 68.3 | 55.2 | 66.4 | 60.3 | 67.2 | 45.5 | 41.1 | 43.2 |
| SC-EGG w/o $\mathcal{L}_{LCE}$ | 71.0 | 59.4 | 69.9 | 64.2 | 67.8 | 46.0 | 39.2 | 42.4 |
| SC-EGG w/o $\mathcal{L}_{GCE}$ | 74.0 | 59.8 | 74.7 | 66.5 | 68.1 | 39.6 | 48.1 | 43.5 |
| SC-EGG w/o $\mathcal{L}_{SCR}$ | 72.9 | 60.1 | 72.2 | 65.6 | 67.3 | 47.2 | 40.1 | 43.4 |
| SC-EGG w/o $\mathcal{L}_{EGS}$ | 72.9 | 60.4 | 72.5 | 65.9 | 68.1 | 47.6 | 40.4 | 43.7 |
| SC-EGG (full) | **75.1** | 64.1 | 73.6 | **68.5** | **69.2** | 45.1 | 43.6 | **44.3** |

Table 2: Results (%) of ablation study on CUB and SUN. The best result of *Acc* and $H$ are marked in **boldface**.

0.1. In addition, the setting of other hyperparameters in TF-VAEGAN follows [Narayan *et al.*, 2020]. We train SC-EGG for 20 epochs in stage 1, 20 epochs in stage 2, and at most 200 epochs in stage 3. The code of SC-EGG is available at the online page[1].

## 4.1 Comparison with State of the Arts

SC-EGG is an inductive method, thus we only compare it with other inductive methods for fair comparisons. Table 1 shows the comparison with state-of-the-arts on CUB, SUN, and AWA2 both in the CZSL and GZSL settings. In the CZSL setting, our SC-EGG achieves the best accuracies of 69.2% and 78.2% on SUN and AWA2, respectively, and obtains significant gains of over 3.2% and 4.6% when compared with all of other methods. On CUB, SC-EGG still obtains competitive performance with a top-1 accuracy of 75.1%. These are benefitted from the transferable representation of the semantic compression embeddings learned by our SC-EGG. In the GZSL setting, the SC-EGG achieves the best performances with harmonic mean ($H$) of 68.5%, 44.3% and 72.4% on CUB, SUN and AWA2, respectively. Compared with other methods that achieve good performance on either seen classes or unseen classes, our SC-EGG achieves a good balance between seen and unseen classes. This is because SC-EGG is capable of learning the discriminative dense-semantic visual representations that are attribute-level semantic-related, thus the quality of seen and unseen visual features are enhanced.

## 4.2 Ablation Study

In this section, we provide further insights into SC-EGG by conducting ablation studies to evaluate the effects of $\mathcal{L}_{LCE}$, $\mathcal{L}_{GCE}$, $\mathcal{L}_{SCR}$ and $\mathcal{L}_{EGS}$. We show the baseline of the generative model (TF-VAEGAN [Narayan *et al.*, 2020]) in our SC-EGG. Results of the ablation study are shown in Table 2. The result of the baseline, which without semantic compression embedding, is significantly worse than the full SC-EGG, with the $Acc/H$ drops by $6.8\%/8.2\%$ and $2.0\%/1.1\%$ on CUB and SUN, respectively. If we incorporate the SCEN without local classification constrain (*i.e.*, SC-EGG w/o $\mathcal{L}_{LCE}$), the model achieves poor ZSL performance against the full SC-EGG with the $Acc/H$ drops by $4.1\%/4.3\%$ and $1.4\%/1.9\%$ on CUB and SUN respectively. When we remove the global classification constrain of SCEN (*i.e.*, SC-EGG w/o $\mathcal{L}_{GCE}$), the performance of SC-EGG will drop slightly. Moreover, removing the semantic consistent regression loss $\mathcal{L}_{SCR}$ results in $2.2\%/2.9\%$ and $1.9\%/0.9\%$ drops of $Acc/H$ on CUB and SUN. If $\mathcal{L}_{EGS}$ is not used, SC-EGG inevitably overfits to seen or unseen classes, leading to inferior results. Our full SC-EGG effectively learns the compressed semantic embeddings for discriminative and transferable feature representations, achieving the best results.

## 4.3 Qualitative Results

**Visualization of Attention Maps.** In Figure 3, we show the visualization of attention maps of the SCEN of two random selected samples. The first column shows the input images, the second column shows the global attention maps of GEN, and the other columns are local attention maps of LEN with top-8 attention scores. As shown in Figure 3, local visual features precisely focus on the regions corresponding to the

[1]https://github.com/HHHZM/SC-EGG

attributes. The global visual feature focuses on the complete instance in the image, avoiding the influence of useless scene information.

**t-SNE Visualizations.** As shown in Figure 4, we present the t-SNE visualizations [Van der Maaten and Hinton, 2008] of real seen/unseen and synthetic unseen visual features in CNN backbone visual space and dense-semantic visual space. Results show that the visual features in the dense-semantic visual space are more discriminative in real seen, real unseen, and synthetic unseen classes than the CNN backbone ones.

## 5 Conclusion

In this paper, we propose a novel semantic compression embedding guided generation model (termed as SC-EGG) for ZSL. SC-EGG cascades a semantic compression embedding network (SCEN) and an embedding guided generative network (EGGN). The SCEN learns a discriminative dense-semantic visual space by semantic compression embedding, and the EGGN learns an implicit mapping from the class-level semantic space to the dense-semantic visual space to synthesize high-quality unseen features, under the guidance of SCEN. Extensive experiments on three popular benchmark datasets demonstrate the superiority of SC-EGG for ZSL.

## Acknowledgements

This work is partially supported by NSFC (61772220, 62172177, 62006244), Special projects for technological innovation in Hubei Province (2018ACA135), Key R&D Plan of Hubei Province (2020BAB027), Natural Science Foundation of Hubei Province (2021CFB332), Young Elite Scientist Sponsorship Program of China Association for Science and Technology (YESS20200140).

## References

[Akata *et al.*, 2016] Zeynep Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:1425–1438, 2016.

[Chen *et al.*, 2021a] Shiming Chen, Wenjie Wang, Beihao Xia, Qinmu Peng, Xinge You, Feng Zheng, and Ling Shao. Free: Feature refinement for generalized zero-shot learning. In *ICCV*, pages 122–131, 2021.

[Chen *et al.*, 2021b] Shiming Chen, Guo-Sen Xie, Qinmu Peng, Yang Liu, Baigui Sun, Hao Li, Xinge You, and Ling Shao. Hsva: Hierarchical semantic-visual adaptation for zero-shot learning. In *NeurIPS*, 2021.

[Chen *et al.*, 2021c] Zhi Chen, Yadan Luo, Ruihong Qiu, Sen Wang, Zi-Yu Huang, Jingjing Li, and Zheng Zhang. Semantics disentangling for generalized zero-shot learning. In *ICCV*, 2021.

[Chen *et al.*, 2022a] Shiming Chen, Ziming Hong, Guo-Sen Xie, Qinmu Peng, Xinge You, Weiping Ding, and Ling Shao. Gndan: Graph navigated dual attention network for zero-shot learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Chen *et al.*, 2022b] Shiming Chen, Ziming Hong, Guo-Sen Xie, Wenhan Yang, Qinmu Peng, Kai Wang, Jian Zhao, and Xinge You. Msdn: Mutually semantic distillation network for zero-shot learning. In *CVPR*, 2022.

[Cheng *et al.*, 2017] Yu Cheng, Jian Zhao, Zhecan Wang, Yan Xu, Karlekar Jayashree, Shengmei Shen, and Jiashi Feng. Know you at one glance: A compact vector representation for low-shot learning. In *ICCV Workshops*, pages 1924–1932, 2017.

[Felix *et al.*, 2018] Rafael Felix, Ian Reid, Gustavo Carneiro, et al. Multi-modal cycle-consistent generalized zero-shot learning. In *ECCV*, pages 21–37, 2018.

[Han *et al.*, 2021] Zongyan Han, Zhenyong Fu, Shuo Chen, and Jian Yang. Contrastive embedding for generalized zero-shot learning. In *CVPR*, pages 2371–2381, 2021.

[He *et al.*, 2016] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Huynh and Elhamifar, 2020a] Dat Huynh and Ehsan Elhamifar. Compositional zero-shot learning via fine-grained dense feature composition. *NeurIPS*, 2020.

[Huynh and Elhamifar, 2020b] Dat Huynh and Ehsan Elhamifar. Fine-grained generalized zero-shot learning via dense attribute-based attention. In *CVPR*, pages 4483–4493, 2020.

[Jiang *et al.*, 2019] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Transferable contrastive network for generalized zero-shot learning. In *ICCV*, pages 9765–9774, 2019.

[Keshari *et al.*, 2020] Rohit Keshari, Richa Singh, and Mayank Vatsa. Generalized zero-shot learning via over-complete distribution. In *CVPR*, pages 13300–13308, 2020.

[Lampert *et al.*, 2014] Christoph H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:453–465, 2014.

[Li *et al.*, 2019] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. In *CVPR*, pages 7402–7411, 2019.

[Narayan *et al.*, 2020] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees GM Snoek, and Ling Shao. Latent embedding feedback and discriminative features for zero-shot classification. In *ECCV*, pages 479–495, 2020.

[Palatucci *et al.*, 2009] Mark Palatucci, D. Pomerleau, Geoffrey E. Hinton, and Tom Michael Mitchell. Zero-shot learning with semantic output codes. In *NeurIPS*, pages 1410–1418, 2009.

[Patterson and Hays, 2012] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, pages 2751–2758, 2012.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Scholkopf and Smola, 2002] B. Scholkopf and A. Smola. Learning with kernels. *MIT Press*, 2002.

[Sylvain *et al.*, 2020] Tristan Sylvain, Linda Petrini, and R. Devon Hjelm. Locality and compositionality in zero-shot learning. In *ICLR*, 2020.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[Welinder *et al.*, 2010] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. *California Institute of Technology*, 2010.

[Xian *et al.*, 2017] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *CVPR*, pages 4582–4591, 2017.

[Xian *et al.*, 2018] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, pages 5542–5551, 2018.

[Xian *et al.*, 2019] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, pages 10275–10284, 2019.

[Xie *et al.*, 2019] Guo-Sen Xie, L. Liu, Xiaobo Jin, F. Zhu, Zheng Zhang, J. Qin, Yazhou Yao, and L. Shao. Attentive region embedding network for zero-shot learning. In *CVPR*, pages 9376–9385, 2019.

[Xie *et al.*, 2020] Guo-Sen Xie, Li Liu, Fan Zhu, Fang Zhao, Zheng Zhang, Yazhou Yao, Jie Qin, and Ling Shao. Region graph embedding network for zero-shot learning. In *ECCV*, pages 562–580, 2020.

[Xu *et al.*, 2020] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. Attribute prototype network for zero-shot learning. In *NeurIPS*, 2020.

[Yan *et al.*, 2021] Caixia Yan, Xiaojun Chang, Zhihui Li, ZongYuan Ge, Weili Guan, Lei Zhu, and Qinghua Zheng. Zeronas: Differentiable generative adversarial networks search for zero-shot learning. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[Yue *et al.*, 2021] Zhongqi Yue, Tan Wang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Counterfactual zero-shot and open-set visual recognition. In *CVPR*, pages 15404–15414, 2021.