Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 30(1): 140–167.
DOI: 10.3934/era.2022008
Received: 18 September 2021
Revised: 29 October 2021
Accepted: 17 November 2021
Published: 15 December 2021

Research article

Bi-shifting semantic auto-encoder for zero-shot learning

Yu Wang*

College of Intelligent Systems Science and Engineering, Harbin Engineering University, No.145, Nantong Street, Nangang District, Harbin 150001, China

* **Correspondence:** Email: wangyu1121@hrbeu.edu.cn; Tel: +86-18846126840.

Abstract: Zero-shot learning aims to transfer the model of labeled seen classes in the source domain to the disjoint unseen classes without annotations in the target domain. Most existing approaches generally consider directly adopting the visual-semantic projection function learned in the source domain to the target domain without adaptation. However, due to the distribution discrepancy between the two domains, it remains challenging in dealing with the projection domain shift problem. In this work, we formulate a novel bi-shifting semantic auto-encoder to learn the semantic representations of the target instances and reinforce the generalization ability of the projection function. The encoder aims at mapping the visual features into the semantic space by leveraging the visual features of target instances and is guided by the semantic prototypes of seen classes. While two decoders manage to respectively reconstruct the original visual features in the source and target domains. Thus, our model can capture the generalized semantic characteristics related with the seen and unseen classes to alleviate the projection function problem. Furthermore, we develop an efficient algorithm by the advantage of the linear projection functions. Extensive experiments on the five benchmark datasets demonstrate the competitive performance of our proposed model.

Keywords: zero-shot learning; auto-encoder; projection learning; semantic representation; domain adaptation

1. Introduction

Object recognition aims at detecting the objects and predicting the class label of a given image, which has been widely used in classification [1,2], localization [3–7], segmentation [8,9], retrieval [10–12] and natural language processing [13,14], etc. The significant advances have been reported in a large number of deep learning literatures [15–19]. Despite the exciting success, most methods proposed in those papers are based on supervised learning, which is driven by the availability of manually annotated instances with powerful low-level visual features [7]. However, the frequencies of objects in the wild follow a long-tailed distribution that consist of a few common classes and most rare classes [20]. On one hand, it is difficult for rare classes without sufficient representative labeled instances to train a classifier effectively. Moreover, it is extremely challenging to collect large-scale labeled instances, even if the performance of the model is improved by adding more instances. Taking the large-scale dataset ImageNet [21] as an example, it contains a total of 14M images in 21,841 classes. It is unrealistic to exhaustively annotate hundreds of instances for each class. On the other hand, the labeled instances of certain classes are precious and difficult to obtain significant amount of the corresponding annotated instances, e.g., endangered bird breed in fine-grained datasets, which is hard to annotate images without expert knowledge [22], let alone collecting instances. In addition, new objects emerge over time that are not covered by known classes and have no labeled instances beforehand, e.g., the high-quality radiology images of the patients infected by COVID-19 are not available before 2019. As a result, the conventional approaches cannot tackle above problems. There are increasing efforts to address the problem of insufficient or even no labeled instances, such as one-shot and few-shot learning [23] deal with the classes of few labeled instances; open world recognition performs the tasks: detecting the novelty of the test classes via open set recognition that was initially proposed by [24], progressively labeling instances of novel unseen classes by class-incremental learning, and adapting the model to classify the acquired labeled instances [25]. The above-mentioned techniques reduce the dependence on labeled instances and improve the accuracy, but still require at least some labeled instances for model learning. Unfortunately, the aforementioned strategies fail to determine the class labels of the instances belonging to unseen classes that have no labeled data.

In contrary, humans have the ability to recognize unseen classes by intelligently utilizing the previously learned knowledge extracted from the seen classes. For example, a learner can easily recognize the Persian fallow deer, if he/she has ever seen fallow deer and is aware that it resembles the fallow deer with bigger antlers and white spots around the neck. Therefore, they are capable of distinguishing beyond 30,000 objects [26] as well as varieties of the subordinates. Inspired by the mechanism of human's ability to recognize new objects without seeing all classes in advance, Zero-Shot Learning (ZSL) [27–31]has drawn significant attention and is proposed to recognize the entirely novel classes omitted from training instances by extrapolation from the knowledge contained in the observed classes [32]. More specifically, given labeled training instances of seen classes in the source domain, ZSL aims to establish a model to classify the instances of unseen classes in the target domain, which increasingly reduces the resources in labor and time expenses. In addition to computer vision related to images, the applications of ZSL has been emerging in various fields, such as zero-shot translation [33], bilingual dictionary induction [34] and molecular compound analysis [35].

In the absence of the labeled instances of unseen classes, the key idea underpinning ZSL methods is to explore the knowledge that transfer via shared auxiliary information. Seen classes are associated with unseen classes in a common space, i.e., semantic space, and the high-level semantic representations are considered as auxiliary information among these classes. Thereby, they can act as a bridge to guarantee the feasibility of ZSL. Generally, there are multiple types of semantic information: attributes [36], word vectors [37–39], textual descriptions [40], hierarchical ontology [41, 42], etc. The commonly used semantic space nowadays is attribute space [27]. Each class is endowed with a unique semantic prototype [43] in this space. The prototype is specified by a binary or continuous attribute vector that indicates the class properties manually designed by experts. The relatedness of the classes is represented by the similarity of the semantic prototypes, e.g., the semantic prototype of zebra is closer to that of horse instead of pig, which is agreement to the reality that zebra is semantically related to horse. Therefore, ZSL can learn a model properly with the aid of semantic representations. Most existing ZSL approaches [27, 40, 41, 44–46] exploit a visual-semantic projection to reflect the relationship among the classes. Specifically, the projection is learned to map the low-level visual features of the labeled instances consisting of seen classes only to semantic space during training. At test stage, the learned projection function is applied to map the target instances of unseen classes to the same semantic embedding space where seen and unseen classes reside. Then, the similarities of the predicted semantic presentations and prototypes are measured by certain matric. Employing the nearest neighbor (NN) search, the classification of the target instance is realized by aligning the semantic prototype of unseen class that yields the highest score.

Despite the success of those semantic embedding models, the largest challenge in ZSL is the projection domain shift problem [43] among the disjoint seen and unseen classes and is manifested through the following aspects. On the one hand, the visual feature space is mutually independent of semantic space, and they have distinct distributions. Hence, there is great difficulty in learning an effective and compatible projection function between the two spaces. On the other hand, the visual appearance of the same attributes in seen and unseen classes are fairly different. The discrepancy is analyzed empirically in [43]. It can be seen that shared characteristic "has tail" in target unseen class Pig is visually different from the source seen class Zebra. Thus, there are significant differences in the underlying distributions of the classes that leads to poor performance on novel classes. In other words, if the projection functions learned with the training instances in seen classes are directly adopted to the unseen classes without adaptation, the target instance tends to be shifted far away from the corresponding class prototype, resulting in the unsatisfactory recognition by NN search at test stage.

There is a recent surge of interest in building a better generalizable projection function on the novel classes to be less susceptible to domain shift. Firstly, a large volume of the literatures belongs to inductive setting are published to overcome this problem [27, 32, 47–49]. The most representative one is SAE [48] and it learns the linear projection function from visual feature space to semantic space based on auto-encoder paradigm, in which the decoder is the transpose of encoder and imposed by a reconstruction constraint of the original visual features. However, inductive methods only have access to the seen data, the projection is likely to capture the characteristics of the seen classes rather the unseen ones. As a result, it hinders the effective generalization. Secondly, the generative models are proposed to compensate for the visual features of target unseen data. The two prominent members are Generative Adversarial Networks (GANs) [50–54] and Variational Auto-Encoders (VAEs) [55] that synthesize visual features by utilizing the semantic prototypes of unseen classes. While, it is noticed that the choice of the semantic prototype is essential, as low quality may degrade the effectiveness of the generator. Afterwards, various methods resort to transductive learning [43, 56–62] leverage the unlabeled target instances during training. The existing transductive learning methods are classified into three categories. The first one is label propagation. For example, Fu et al. [43] combines multiple semantic representations with visual features of unseen classes to learn a joint embedding space, in which the target data are aligned with the label embeddings and then the recognition is performed via label propagation. The second one is self-training that progressively improves the classification capacity in an iterative refining process [59]. The last one termed domain adaptation is the most relevant method to our model and has been well-investigated to uncover the common knowledge of the source and target domains [63–65]. Different from the above first two strategies, [65] simultaneously utilize the

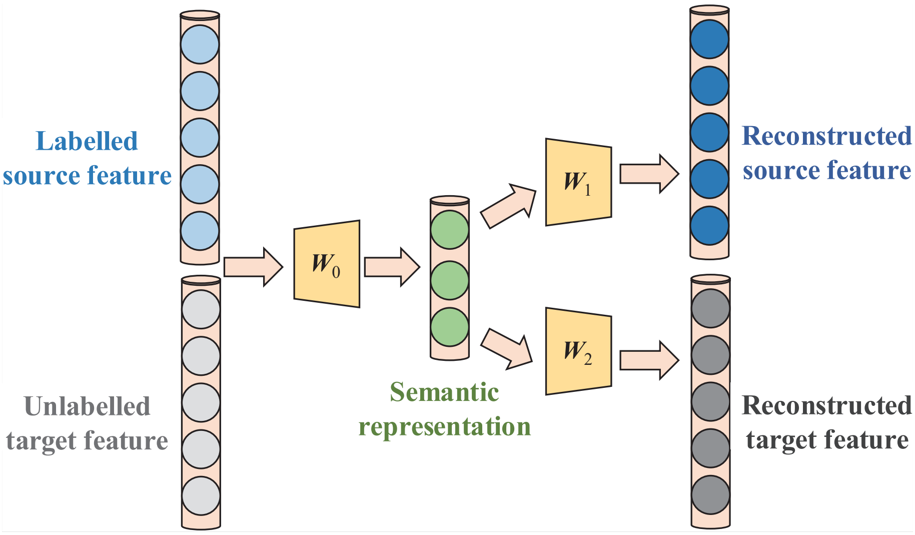

Labelled source feature
Unlabelled target feature
W_0
Semantic representation
W_1
W_2
Reconstructed source feature
Reconstructed target feature

Figure 1. A general view demonstrating the proposed BSAE framework.

visual features and semantic prototypes of unseen classes, our model only utilizes the visual features of the unlabeled instances. Furthermore, our model is not concerned with the distribution alignment of the projected and original domains [63] or that of the features in the immediate space [64], whereas the latent space in our model is semantically meaningful and encourages learning the generalizable semantics containing sufficient information of the visual features through a reconstruction task.

In this paper, we develop a novel model by exploiting the idea of autoencoder framework to solve zero-shot challenges and reveal the relationship between visual features and semantic representations. We assume that the majority semantic properties of the unseen classes are shared with that of seen classes. Following the previous work, we adopt the semantic space as the latent embedding space to preserve the semantic relatedness between the classes. Motivated by [48, 63], our model takes advantage of the bi-shifting linear auto-encoder framework. In specific, the common encoder shared by source and target domains tries to learn the projection from visual feature space to semantic space. Considering the distribution divergence of the disjoint domains, the original features are reconstructed by two different decoders based on the learned semantic representations. It is worth mentioning that there are two regularization terms in our model. Inspired by [48], the first term is designed to inherit the properties of semantic space by incorporating the semantic prototypes of the seen classes and then the projections are constrained to force the learned semantics of unlabeled target instances as close as possible to their class prototypes. Consequently, the semantic mismatch between visual features and semantic representations can be refined. The second regularization term is adopted to enforce that the decoder in target domain is derived from, rather than the same as the decoder supervised learned via semantics of instances in source domain, resulting in truthfully reconstruct the visual features. To that end, we design a novel Bi-shifting Semantic Auto-Encoder (hereafter referred as BSAE) architecture that integrates the merits of both domain adaptation and discriminative ability of class semantics, as shown in Figure 1. However, BSAE is based on linear auto-encoder and quite shallow, experimental results reported in section 4 prove its outstanding performance. For example, the average accuracy on five benchmark datasets under different protocols are whopping 6.9% improvement over the current state-of-the-art. In conclusion, the main contributions are three-fold:

- A simple and effective ZSL model termed BSAE is developed in an auto-encoder framework. Our model not only alleviates the domain shift problem, but also recovers the interaction between visual feature and semantic representation.

- We consider ZSL as the problem of learning the projection functions to explore shared discriminative semantic representations of instances, which are supervised by the semantic prototypes of the seen instances. Meanwhile, the generalizable capability of the learned semantics are enhanced by exploiting the visual features of unlabeled instances.

- An iterative algorithm with high computational efficiency is introduced to solve the problem. Extensive experimental results demonstrate that our approach achieves superior performance on five benchmark datasets, even if the class prototypes of the unseen classes are not available.

The remainder of the paper is organized as follows. In Section 2, we briefly review the related work proposed to overcome the challenges in ZSL. In Section 3, we describe the proposed model and deduce an efficiently iterative algorithm. The results are reported and discussed in Section 4. Finally in Section 5, we present the conclusion and propose several research directions to be investigate.

2. Related work

In this section, we firstly introduce a review about the semantic space exploited in current zero-shot learning. Then, we briefly review of projection learning concerned with our work. Finally, we present the related advances to relive the domain shift problem.

2.1. Semantic space

Semantic representation shared between classes bridges the gap in ZSL and enables the transmission of common knowledge from seen to unseen classes. There are various semantic spaces formed by different class embeddings. Attribute space [27] is the most popular and effective one [46, 66, 67], in which the properties of the classes are described as attributes. However, manually collecting and annotating attributes are heavy reliance on the efforts of experts. The word vector [38, 39] and text description [40] based semantic space are proposed because of relatively less labor intensive. The semantic representations are automatically extracted by embedding models from text corpus (e.g., Wikipedia). In spite of the inconvenience for humans to incorporate the knowledge of the classes into the semantics, as reported in [40], 10 sentence descriptions are collected for each image to construct the semantic space, which is even more expensive than annotating attributes. Moreover. SJE [41] and ESZSL [47] have shown that attribute space is more effective than word vector space. Besides, several ZSL methods take advantage of them via combining the aforementioned semantic spaces [50, 60, 68]. In our work, we consider the attribute space as semantic space.

2.2. Projection learning

The existing ZSL models can be sub-categorized into three groups, depending on how the projection function is established.

2.2.1. Visual-semantic projection

The first group learns a forward projection from visual feature space to semantic space. Lampert et al. [27] proposed two attribute-based classifiers includes direct attribute predictor (DAP) and indirect attribute predictor (IAP) that exploit the attributes to predict the class labels of instances in a two-stage schema. SOC [32] firstly projects the visual features into the semantic space, and then determines the class label through KNN. CONSE [37] exploits a probabilistic model and then predicts the unseen classes via the convex combination of the class-embeddding vectors. DeViSE [38] applies linear corresponding function by combining similarity and hinge ranking loss. ESZSL [47] learns the bilinear compatibility function by optimizing square loss. To optimize the ranking loss, ALE [58] employs a bilinear mapping compatibility function.

2.2.2. Semantic-visual projection

The second group learns a reverse projection from the semantic space to the visual space to rectify the hubness problem [69]. The hubness refers to the phenomenon of some semantic prototypes are nearest neighbors of instances from different classes, which is a curse of demensionality. Zhang et al. [70] proposed a deep end-to-end neural network to embed the class prototypes into the visual feature space that suffer much less from the hubness problem, as discussed in [71]. In addition to the embedding models, generative-based methods are proposed recently to generate instances for unseen classes by leveraging the semantic prototypes, then the ZSL problem is converted into a traditionally supervised problem. f-CLSWGAN [50] and LisGAN [68] explore the conditional generator on semantics to synthesize the visual features. However, it is hard to train generative models because of the min-max optimization. Auto-encoder is an effective framework to extract the representative features in a unsupervised manner and alleviate the domain shift problem. Xu et al. [52] construct visual feature space as latent layer and learns two different regressors for semantic reconstructions. The latent layer of [48] is semantic space and the linear projection between the visual feature and semantic space is learned with the semantic constraint of seen classes to reconstruct the original data. [72] improves the model in [48] by adding a regularization constraint of the projection function, thereby ensuring that the structural risk of the model is minimized.

2.2.3. Immediate projection

In the last group, both visual features and semantic representations are projected into a common space. SYNC [49] learns classifiers of unseen classes by linearly combining base classifiers. Zhang and Saligrama [67] leverage similar class relationships in the common space, which is defined by the seen classes proportions.

Taking full advantage of the first two groups, our model is close to [63] in which a bi-shifting auto-encoder is employed for reconstructing visual features in different domains. Different with [63] that apply nonlinear projections to learn the representations in latent space, our model reinforces the latent space as semantic space with class semantic prototypes and exploits the linear projection functions to fit the distributions of visual and semantic spaces, respectively.

2.3. Domain shift problem

Domain shift problem was firstly reported in [43] and is an open issue in ZSL. It describes that the projection functions learned from the seen classes are biased when exploit them to map instances of unseen classes from visual feature space to semantic space. It is essentially caused by the disjoint seen and unseen classes with different underlying data distributions. The researchers have investigated how to rectify the domain shift problem and obtain competitive results, for instance, SAE [48] imposes an additional reconstruction constraint to the training seen data, resulting in the learned projection function more generalizable across seen and unseen classes. LisGAN [68] refines the domain shift via generating the soul instances related to the semantic representations. However, as the unseen class data are not involved in the model learning, the generalizable ability of the inductive methods is limited. Transductive ZSL is an emerging topic to mitigate the domain shift problem where not only labeled seen class data are available, but also has access to unlabeled unseen class data, which potentially leads to improvements in classification performance. Fu et al. [43] first propose a transductive multi-view embedding framework, and then generate the class labels for unseen class data via label propagation. Kodirov et al. [65] formulate a regularized sparse coding framework to solve the domain shift problem. A measure of inter-class semantic consistency is proposed by [73] to explore the relation between the semantic manifold and visual-semantic projection on seen classes. VCL [56] proposes a visual structure constraint on class centers. Unlike SAE [48] exploits one decoder to reconstruct the features without domain adaptation, our model employs transductive setting and adopts two decoders to reconstruct the visual features in the source and target domains. Additionally, we restrict the similarity constraint of the two different decoders as a regularizer by considering the amount of adaptation from the labeled seen class data rather than being deviated freely. Although we only use the visual features of unseen data rather than the combination of the visual features and semantic representations of target unseen data like others [41,70], our model boosts ZSL performance.

3. Methodology

In this section,we describe the procedures and methods used in this paper. we firstly set up the zero-shot learning problem, then develop our novel model BSAE for this task, and finally derive an efficient algorithm to solve it. Subsequently, the classification of unseen classes can be performed in the original feature space and semantic space.

3.1. Problem definition

We start by introducing some notations and problem definition of our interest. Considering m labeled source instances $\mathcal{S} = \{(x_i^s, y_i^s, s_i^s) | x_i^s \in \mathcal{X}, y_i^s \in C_s\}_{i=1}^m$ are given as training data, where $x_i^s \in \mathbb{R}^d$ denotes the d-dimensional visual feature, y_i^s is the corresponding class label in C_s consisting of τ discrete seen classes, s_i^s is the semantic representation of ith instance. In addition, given p unlabeled target data $\mathcal{T} = \{(x_i^t, y_i^t, s_i^t) | x_i^t \in \mathcal{X}, y_i^t \in C_t\}_{i=1}^p$ of unseen classes, where $x_i^t \in \mathbb{R}^d$ denotes the d-dimensional visual feature, y_i^t is the corresponding label and belongs to the μ unseen classes set C_t. While the seen and unseen classes are disjoint, i.e., $C_s \cap C_t = \emptyset$, the semantic space $\mathcal{A}$ are associated with mitigating this challenge, which is spanned by attribute vector or word vector derived from text for each class. The k-dimensional semantic prototypes of seen and unseen classes are denoted as $A_s = [a_1^s, a_2^s, \ldots, a_\tau^s] \in \mathbb{R}^{k \times \tau}$

and $A_t = [a_1^t, a_2^t, \dots, a_\mu^t] \in \mathbb{R}^{k\times\mu}$. Therefore, $S_{tr} = [s_1^s, s_2^s, \dots, s_m^s] \in \mathbb{R}^{k\times m}$ is given because the source data $X_s = [x_1^s, x_2^s, \dots, x_m^s] \in \mathbb{R}^{d\times m}$ are labeled by either binary or continuous attributes indicating the corresponding class labels $Y_s = \{y_i^s\}_{i=1}^m$. On the contrary, as the target instances $X_t = [x_1^t, x_2^t, \dots, x_p^t] \in \mathbb{R}^{d\times p}$ are unlabeled, $S_{te} = [s_1^t, s_2^t, \dots, s_p^t] \in \mathbb{R}^{k\times p}$ that stands for the semantic prototypes and $Y_t = \{y_i^t\}_{i=1}^p$ that denotes the class labels have to be predicted.

The goal of standard zero-shot learning is to predict the correct class of X_t by learning a classifier $f : \mathcal{T} \to C_t$. The key notations used in this paper are listed in Table 1.

3.2. *Model formulation*

We begin our discussion with auto-encoder (AE) for it being the basis of our model. The simplest form of AE is linear and has one hidden layer [48] that is responsible to truthfully reconstruct the input data as similar as possible. We force the semantic space in hidden layer as [48] so that the latent space is semantically meaningful, e.g., each column of S_{tr} stands for the attribute vector of the corresponding labeled source instance.

Assume that labeled seen-class training set $\mathcal{S}$ and unlabeled target data X_t are available. The proposed BSAE aims to learn a model to estimate the discriminative semantic representations S_{te} and reconstructed features $\hat{X}^t$ of the target instances and then obtain their class labels Y_t in semantic space and visual feature space, respectively. Specifically, considering the seen classes and unseen classes are related in the same class embedding space (e.g., attribute), BSAE consists of three components: (1) It attempts to learn the encoder parameterized by $W_0 \in \mathbb{R}^{k\times d}$ ($k < d$) to project both domains from visual feature space $\mathcal{X}$ to the common semantic space $\mathcal{A}$. In order to guarantee whether the learned semantic representations capture sufficient discriminative information, in terms of the distribution discrepancy between domains, (2) on one hand, the decoder $W_1 \in \mathbb{R}^{d\times k}$ reconstructs the original visual features of source domain exactly. (3) On the other hand, the mapped class embeddings of target domain are projected to the visual features by decoder $W_2 \in \mathbb{R}^{d\times k}$. We simultaneously minimize the reconstruction

Table 1. Key notations

Notations	Descriptions
$X_s \in \mathbb{R}^{d\times m}$	Visual feature of source instances
$X_t \in \mathbb{R}^{d\times p}$	Visual feature of target instances
$Y_s = \{y_i^s\}_{i=1}^m$	Class labels of X_s
$Y_t = \{y_i^t\}_{i=1}^p$	Class labels of X_t
$C_s = \{1, 2, \dots, \tau\}$	Set of τ seen classes
$C_t = \{1, 2, \dots, \mu\}$	Set of μ unseen classes
$A_t \in \mathbb{R}^{k\times\mu}$	Semantic prototypes of C_t
$S_{tr} \in \mathbb{R}^{k\times m}$	Semantic representations of X_s
$S_{te} \in \mathbb{R}^{k\times p}$	Semantic representations of X_t
λ_1, λ_2	Hyper-parameters
$\mathcal{X}$	d-dimensional visual feature space
$\mathcal{A}$	k-dimensional semantic space
m, p	Number of source and target instances
d, k	Dimensionality of visual feature and semantic space

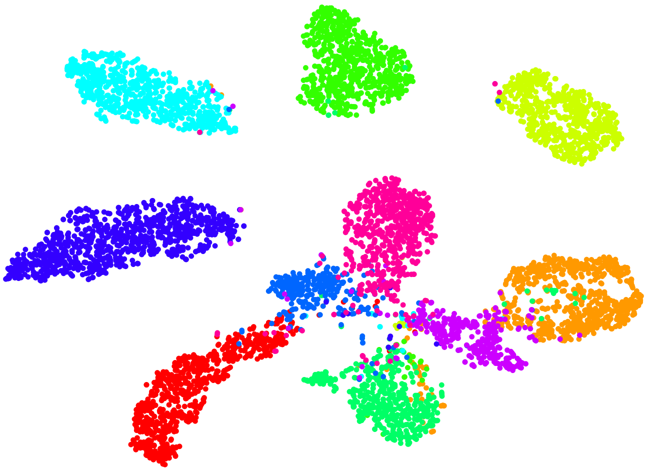


Figure 2. Visualization of AWA1 unseen classes under standard splits (SS) protocol with the learned semantic representations. Our proposed model provides the intra-class cohesion as well as inter-class discrimination.

errors in different domains by utilizing the unlabeled instances from unseen classes to narrow down the domain gap. Therefore, it is applicable to better generalize the learned regression model to unseen classes. As observed from Figure 2, our model preserves enough discriminative information across unseen classes, even in the low dimensional semantic space that exacerbates the hubness problem.

Our model is learned by optimizing the following objective:

$$\begin{aligned} \min_{W_0,W_1,W_2} \quad & J = \|X_s - W_1 W_0 X_s\|_F^2 + \|X_t - W_2 W_0 X_t\|_F^2 \\ s.t. \quad & W_0 X_s = S_{tr}, \end{aligned} \tag{3.1}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix. Eq 3.1 denotes the loss of the autoencoder. It is difficult to solve the objective Eq 3.1 with a hard constraint. To fight off the constraint $W_0 X_s = S_{tr}$ efficiently, we relax the constraint through incorporating a semantic similarity term into Eq 3.1:

$$\min_{W_0,W_1,W_2} \quad J = \|X_s - W_1 S_{tr}\|_F^2 + \|X_t - W_2 W_0 X_t\|_F^2 + \lambda_1 \|W_0 X_s - S_{tr}\|_F^2 \tag{3.2}$$

and λ_1 is the hyper-parameter. The first two terms are regarded as the losses of different decoders. The last term is the loss of encoder. W_2 is unsupervised because of the unknown semantic representation S_{te} of target data and [65] proves that W_2 adapted from W_1 is efficient to this issue. To this end, we adds the regularization term $\|W_2 - W_1\|_F^2$ to Eq 3.2 to restrict the amount of adaptation of the two projections. It is worth noting that W_1 is considered as a basis to ensure W_2 cannot deviate freely from W_1. The full objective of our proposed model then becomes:

$$\min_{W_0,W_1,W_2} \quad J = \|X_s - W_1 S_{tr}\|_F^2 + \|X_t - W_2 W_0 X_t\|_F^2 + \lambda_1 \|W_0 X_s - S_{tr}\|_F^2 + \lambda_2 \|W_2 - W_1\|_F^2, \tag{3.3}$$

where λ_2 is a hyper-parameter used to balance the importance of different terms.

3.3. Optimization

Next, we will formulate our solver as a novel gradient-based algorithm to alternately update projection functions W_0, W_1 and W_2. Note that the conventional iterative algorithms (e.g., Gradient Descent) have been widely exploited to directly solve such problems without computationally efficiency. Whilst our solver depends on the dimension of the features, not the number of instances and hence is more effective than the conventional iterative algorithms. To solve the Eq 3.3, we calculate the partial derivative of it and set it to zero:

$$\frac{\partial J}{W_0} = -\, W_2^T(X_t - W_2W_0X_t)X_t^T + \lambda_1(W_0X_s - S_{tr})X_s^T, \tag{3.4}$$

$$\frac{\partial J}{W_1} = -\, (X_s - W_1S_{tr})S_{tr}^T - \lambda_2(W_2 - W_1), \tag{3.5}$$

$$\frac{\partial J}{W_2} = -\, (X_t - W_2W_0X_t)X_t^T W_0^T + \lambda_2(W_2 - W_1), \tag{3.6}$$

and optimize the following sub-problems through alternative optimization methods.

3.3.1. Fix W_2, update W_0

Setting Eq 3.4 to zero, we obtain:

$$W_2^T W_2 W_0 + \lambda_1 W_0 X_s X_s^T (X_t X_t^T)^{-1} = W_2^T + \lambda_1 S_{tr} X_s^T (X_t X_t^T)^{-1}. \tag{3.7}$$

Let $M_0 = W_2^T W_2, H_0 = \lambda_1 X_s X_s^T (X_t X_t^T)^{-1}, Q_0 = W_2^T + \lambda_1 S_{tr} X_s^T (X_t X_t^T)^{-1}$, we have the Sylvester equation:

$$M_0 W_0 + W_0 H_0 = Q_0, \tag{3.8}$$

where $M_0 \in \mathbb{R}^{k\times k}$ and $H_0 \in \mathbb{R}^{d\times d}$ are square matrices, $Q_0 \in \mathbb{R}^{k\times d}$ is a rectangle matrix. The above matrix function has a unique solution if it satisfies conditions of the Theorem 3.1 quoted in [74]. Obviously, it is easy to meet in practical applications.

Theorem 3.1. *Eq 3.8 has a unique solution if and only if the matrices M_0 and H_0 have distinct eigenvalues, that is, the eigenvalues $\gamma_1, \gamma_2, \ldots, \gamma_k$ of M_0 and $\zeta_1, \zeta_2, \ldots, \zeta_d$ of H_0 satisfy $\gamma_i + \zeta_j \neq 0$ $(i = 1, ..., k; j = 1, ..., d)$.*

As a result, Eq 3.8 can be easily solved by Bartels-Stewart algorithm, which is implemented with *a single line of code*: sylvester in MATLAB:

$$W_0 = \text{sylvester}(M_0, H_0, Q_0). \tag{3.9}$$

3.3.2. Fix W_2, update W_1

Setting Eq 3.5 to zero, then we have the following Sylvester equation:

$$\lambda_2 W_1 + W_1 S_{tr} S_{tr}^T = X_s S_{tr}^T + \lambda_2 W_2. \tag{3.10}$$

Let $M_1 = \lambda_2 I_k$, I_k is the $k \times k$ identity matrix, $H_1 = S_{tr} S_{tr}^T$, $Q_1 = X_s S_{tr}^T + \lambda_2 W_2$. The Eq 3.10 can be efficiently solved in MATLAB:

$$W_1 = \text{sylvester}(M_1, H_1, Q_1). \tag{3.11}$$

3.3.3. Fix W_1, W_0, update W_2

Similarly, we set Eq 3.6 to zero, and have the following formulation:

$$\lambda_2 W_2 + W_2 W_0 X_t X_t^T W_0^T = X_t X_t^T W_0^T + \lambda_2 W_1. \tag{3.12}$$

If we denote $M_2 = M_1, H_2 = W_0 X_t X_t^T W_0^T, Q_2 = X_t X_t^T W_0^T + \lambda_2 W_1$, the above Sylvester equation (3.12) can be solved in MATLAB:

$$W_2 = \text{sylvester}(M_2, H_2, Q_2). \tag{3.13}$$

Algorithm 1 summarizes the implementation of our algorithm. We simply initialize W_2 with all elements of 0.1 for coarse-grained datasets (e.g., AWA1) and all elements of 0.01 for fine-grained datasets (e.g., CUB). The hyper-parameters λ_1 and λ_2 are selected by cross-validations. Details are listed in section 4. The iterations will terminate when the Eq 3.3 converges or reaches a fixed number of iterations.

3.3.4. Complexity and convergence analysis

To this end, we propose to briefly explain the analysis of the time complexity and convergence of our algorithm. As mentioned in **Algorithm 1**, optimizing the objective function Eq 3.3 is actually the process of solving three Sylvester equations. The time complexity of computing each Sylvester equation, e.g., Eq 3.8, given M_0 and H_0, is $O(k^3 + d^3)(d, k \ll \min(m, p))$, which is independent of number of instances. In other words, it can be effectively applied to large-scale datasets. As can be observed from Eq 3.7 to Eq 3.13, due to linear formulations, it is easier to solve three sub-problems with respect to three projection functions W_0, W_1 and W_2 in our proposed model. Concretely, updating each projection function is regarded to solve Sylvester equation. Hence, the objective function Eq 3.3 is non-increasing with a lower bound during the alternative optimization.

Algorithm 1 Bi-shifting Semantic Auto-Encoder

```
Input: Training data $X_s$, $S_{tr}$
       Test data $X_t$
       Hyper-parameters $\lambda_1$, $\lambda_2$
Output: Projection matrices $W_0$, $W_2$
Initialize $W_2$
while not converge do
    Update $W_0$ by Eq 3.9
    Update $W_1$ by Eq 3.11
    Update $W_2$ by Eq 3.13
    Check the converge condition
end while
Return $W_0$, $W_2$
```

3.4. Classification

According to Algorithm 1, we obtain the optimal projection functions W_0 and W_2. We measure the similarity score between the estimated value of target instance and its prototype, and then predict the class label.

In the semantic space, considering a target instance x_i^t, we could firstly calculate the estimated semantic representation with $(S_{te})_i = W_0 x_i^t$, then compare with the prototypes A_t of classes in C_t by calculating the cosine distance between them:

$$l(x_i^t) = \arg\min_j \; d((S_{te})_i, (A_t)_j), \tag{3.14}$$

where $j \in [\mu]$, $(A_t)_j$ is the prototype attribute vector of j–th unseen class and $d(\cdot,\cdot)$ is a distance function. $l(\cdot)$ returns the class label of a target instance.

In the feature space, it is worth mentioning that the predicted visual features $\hat{X}^t$ of unseen classes are easily synthesized by embedding the semantic prototypes of C_t to the visual feature space with $\hat{X}^t = W_2 A_t$. Hence, the ZSL is converted to a conventional classification problem. Empirically, any supervised classifier can be utilized. We simply exploit k-Nearest Neighbor (KNN) to demonstrate the capability of our decoder W_2. Similar to the process in *1)*, the class label of target instance can be inferred by calculating the cosine distance between the prototype projections and the original visual feature x_i^t:

$$l(x_i^t) = \arg\min_j \; d(x_i^t, \hat{X}_j^t), \tag{3.15}$$

where $\hat{X}_j^t$ is the j–th unseen class prototype projected into the visual feature space.

4. Experiment

In this section, we firstly introduce our experimental protocols in detail, then we present our results that are compared with the state-of-the-art approaches on five small-scale benchmark datasets (AWA1, AWA2, CUB, SUN, aP&Y) for conventional zero-shot learning (CZSL) task.

4.1. Experimental setup and metrics

4.1.1. Datasets

Five benchmark datasets are selected from the widely used datasets for ZSL: AwA1 (Animals with Attributes 1) [27], AWA2 (Animals with Attributes 2) [75], aP&Y (Attribute Pascal and Yahoo) [76], CUB (Caltech-UCSD-Birds 200-2011) [22] and SUN (Scene UNderstanding) [77]. We exploit two typical protocols to evaluate the performance of our model: standard splits (SS) [27] and proposed splits (PS) [75]. More concretely, SS is widely used in previous works, but the weakness is that unseen classes are subset of ImageNet during training, resulting in violating the true zero-shot rule. On the contrary, PS ensures that none of the unseen classes used for pre-training the ResNet belong to 1K classes of ImageNet. The fact that PS is much more difficult than SS on account of low correlation between seen and unseen classes. For clarity, the statistics of these datasets are briefly reported in Table 2.

We take advantage of semantic space spanned by continuous attributes like the pioneering works [27, 47, 48]. Each instance is associated with the corresponding continuous class-level attribute. The dimension of the semantic space equals to that of the attributes, e.g., the semantic space of AWA1 is formed by 85-dim attributes. The dimensions of the attributes of all datasets are listed in Table 2.

4.1.2. Visual feature space

Following the general procedure in other literatures, we use visual features extracted by deep convolutional neural networks (CNNs) and GoogleNet features [78] which is the 1024-dim activation of the final pooling layer as in [41]. Furthermore, the latest works adopt the pre-trained 2048-dim ResNet features, which are extracted by 2048-dim top layer pooling units of the 101-layered ResNet, to achieve improved performance [75]. It is worth noting that the ResNet features has two protocols, namely, SS and PS. For GoogleNet features, only SS is provided. For fair comparison, we do not perform any image pre-processing or any other data augmentation techniques, and conduct extensive experiments on the above two types of features.

4.1.3. ZSL settings

To demonstrate the capability of BSAE, we evaluate on the conventional ZSL (CZSL) setting: Assume that the search space is restricted to the unseen classes, the goal is to predict the class labels of C_t at test stage. We use SS and PS protocols in this setting.

4.1.4. Parameter settings

Our BSAE model has two hyper-parameters: λ_1 and λ_2 (see Eq 3.3). We select λ_1 and λ_2 from $\{10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4\}$ for cross-validation. Considering the two split protocols, we propose tuning these parameters in different ways. For SS protocol, the parameters are chosen by means of class-wise cross-validation on C_s as in [67], that is, two seen classes are randomly selected form a validation set in each iteration to choose the best hyper-parameter $\{\lambda_1, \lambda_2\}$ and use them for testing on unseen classes. For PS protocol, we perform hyper-parameter search on a disjoint set of validation set of 13 (AWA1/AWA2), 5 (AP&Y), 50 (CUB) and 65 (SUN) classes respectively [75]. Note that we report the average performance for ensuring the significance of the results.

Table 2. Statistics for five benchmark datasets.

						At Training Time		At Testing Time	
Datasets	Granularity	Size	Attributes	C_s/C_t	Images	SS (C_s)	PS (C_s)	SS (C_t)	PS (C_t)
AWA1	coarse	medium	85	40/10	30,475	24,295	19,832	6180	10,643
AWA2	coarse	medium	85	40/10	37,322	30,337	23,527	6985	13,795
aP&Y	coarse	small	64	20/12	15,339	12,695	5932	2644	9407
CUB	fine	medium	312	150/50	11,788	8855	7057	2933	4731
SUN	fine	medium	102	645/72	14,340	12,900	10,320	1440	4020

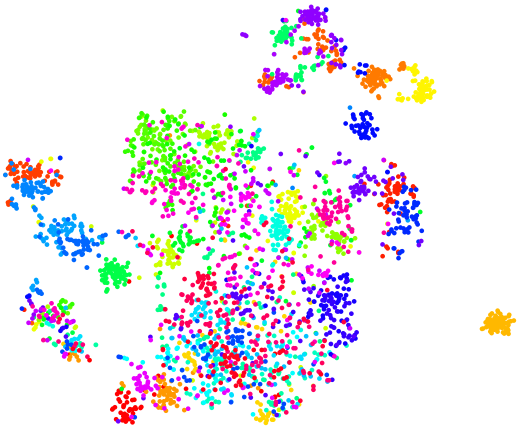


Figure 3. Visualization of ResNet feature distribution of 50 unseen classes on CUB dataset using t-SNE.

4.1.5. Evaluation metric

Most ZSL methods use Top-1 accuracy (e.g., [48]) averaged for all images, where the prediction is correct for the predicted class is coincide with ground-truth. However we are concentrated on high performance of both densely and sparsely populated classes. Therefore, under CZSL setting, we evaluate our method on the benchmark datasets by using per-class top-1 accuracy proposed in [75]. We compute the top-1 accuracy independently for each class, and then average for all unseen classes:

$$acc_{C_t} = \frac{1}{\mu} \sum_{c=1}^{\mu} \frac{\text{\#correct predictions in c}}{\text{\#instances in c}}. \tag{4.1}$$

4.2. Comparative results

We evaluate our proposed framework for zero-shot learning on several benchmark datasets. The competitors are representative, competitive state-of-the-art and recently published that encompass a wide range in zero-shot learning.

4.2.1. Zero-shot learning

In these experiments, the test instances only come from C_t disjoint with the seen classes C_s. We use both SS and PS protocols for more convincing results and the qualitative results are shown in Table 3 and Table 4.

For SS protocol, the SAE [48] is close to our model while lacks of the results of per-class top-1 accuracy with GoogleNet features. For fair comparison, we recreate SAE by following the settings in their original paper and exploit the same classifier to predict the class labels. Leveraging the code available online, we re-implement GFZSL [62] and SYNC [49] to obtain the recognition results. Note that the first 10 methods in Table 3 are cited from [75] and the rest are copied from the original paper. To further verify that our method is not only effective to specific visual features, we implement our model under the SS protocol with 1024-dim GoogleNet features (G) and 2048-dim ResNet features (R).

From comprehensive comparison in Table 3, we witness that: (1) our model achieves the best four of the five evaluations, i.e., AWA1, AWA2, SUN and aP&Y. Specifically, the improvements over the strongest competitor achieve 0.7%, 0.4% and 6.9% on AWA1, AWA2 and aP&Y. For fine-grained dataset SUN that contains more classes and relatively fewer instances per class, while our result of 67.1% is 2.8% higher than the strongest competitor [60]. The accuracy boost can be attributed to the combination of semantic representations and domain adaptation constraints significantly improving the ability for classification. (2) Meanwhile, from Figure 3, we can observe that the overlap between unseen classes of CUB, which is regarded as the well-known complicated dataset, is particularly striking. Moreover, it is hard to learn the visual-semantic projection for the reason that the sparsity of training instances (~ 60 instances per class). However, our model still performs well on this dataset. It is worth pointing out that our model learns a more effective and stable visual-semantic relation from seen data for unseen data analysis. (3) [73] and [75] demonstrate that VggNet and ResNet features lead to

Table 3. Comparative results (%) of CZSL setting on five datasets under the SS protocol. G: GoogleNet; R: ResNet-101; V: VggNet. The best results are highlighted with bold numbers. The second best is in blue. "-" denotes either features or results are provided in the original paper for this dataset.

Method	Feature	AWA1	AWA2	SUN	CUB	aP&Y	Average
DAP [27]	R	57.1	58.7	38.9	37.5	35.2	45.5
IAP [27]	R	48.1	46.9	17.4	27.1	22.4	32.4
CONSE [37]	R	63.6	67.9	44.2	36.7	25.9	47.7
DEVISE [38]	R	72.9	68.6	57.5	53.2	35.4	57.5
CMT [39]	R	58.9	66.3	41.9	37.3	26.9	46.3
SSE [67]	R	68.8	67.5	54.5	43.7	31.1	53.1
SJE [41]	R	76.7	69.5	57.1	55.3	32	58.1
ESZSL [47]	R	74.7	75.6	57.3	55.1	34.4	59.4
LATEM [42]	R	74.8	68.7	56.9	49.4	34.5	56.9
ALE [58]	R	78.6	80.3	59.1	53.2	30.9	60.4
SYNC [49]	R	72.2	71.2	59.1	54.1	39.7	59.3
	G	72.9	-	62.7	54.7	-	-
SAE [48]	R	80.6	80.7	42.4	33.4	8.3	49.1
	G	81.9	-	59.7	53.6	34.5	-
SSZSL [79]	V	88.6	-	-	58.8	49.9	-
DSRL [57]	V	87.2	-	-	57.1	56.3	-
STZSL [80]	V	83.7	-	-	58.7	54.4	-
GFZSL [62]	R	80.5	79.3	62.9	53	51.3	65.4
TSTD [59]	V	90.3	-	-	58.2	-	-
QFSL [61]	R	-	84.8	61.7	**69.7**	-	-
VCL [56]	R	82.0	82.5	63.8	60.1	-	-
DEARF [60]	R	81	81.2	64.3	56.1	-	-
BSAE	R	90.7	**85.2**	60.4	61.8	**63.2**	**72.3**
	G	**91**	-	**67.1**	62.4	57.7	-

improved results in ZSL than GoogleNet features. While our model using GoogleNet features consistently performs favorably against state-of-the-art on the five benchmarks, especially the best result of 91% on AWA1 and 67.1% on SUN. This provide further evidence that our model achieves good performance on coarse-grained and fine-grained datasets even if the features are not the strongest.

For PS protocol, we keep the same setting as in [75] to make sure the unseen classes at test time do not overlap with the 1K training classes of ImageNet. The first 12 reported results are cited from [75] and others copied from their original paper. Generally, the performance is expected degrade under this stricter settings. From the comparative results listed in Table 4, we can make the observation that the average top-1 per-class accuracy of our model performs 4.8% higher than all others and drops least on coarse-to-fine grained datasets among all methods, which illustrates more significant. Due to the similar idea between auto-encoder and GAN-based model, we compare several representative methods, e.g., LisGAN [68] and SRGAN [53]. Our model outperforms these competitors on four of

Table 4. Comparative results (%) of CZSL setting on five datasets under the PS protocol with ResNet-101 features.

Method	AWA1	AWA2	CUB	SUN	aP&Y	Average
DAP [27]	44.1	46.1	40	39.9	33.8	40.8
IAP [27]	35.9	35.9	24	19.4	36.6	30.4
CONSE [37]	45.6	44.5	34.3	38.8	26.9	38
DEVISE [38]	54.2	59.7	52	56.5	39.8	52.4
CMT [39]	39.5	37.9	34.6	33.9	28	34.8
SSE [67]	60.1	61	43.9	51.5	34	50.1
SJE [41]	65.6	61.9	53.9	53.7	32.9	53.6
ESZSL [47]	58.2	58.6	53.9	54.5	38.3	52.7
LATEM [42]	55.1	55.8	49.3	55.3	35.2	50.1
ALE [58]	59.9	62.5	54.9	58.1	39.7	55
SYNC [49]	54	46.6	55.6	56.3	23.9	47.3
SAE [48]	53	54.1	33.3	40.3	8.3	37.8
DEM [70]	68.4	67.1	51.7	61.9	35	56.8
GFZSL [62]	68.3	63.8	49.3	60.6	38.4	56.1
CAVE [55]	71.4	65.8	52.1	61.7	-	-
PSR [81]	-	63.8	56	61.4	38.4	-
TVN [82]	-	68.8	58.1	60.7	-	-
GAZSL [54]	-	68.4	55.8	61.3	41.1	-
f-CLSWGAN [50]	-	68.8	57.3	60.8	40.5	-
GDAN [51]	-	67.7	51	54.8	40.4	-
LESAE [66]	66.1	68.4	53.9	60	40.8	57.8
LisGAN [68]	70.6	-	58.8	61.7	43.1	-
SRGAN [53]	71.9	-	55.4	**62.2**	-	-
DEARF [60]	72.1	69.3	38.5	48.6	-	-
BSR [52]	-	68.4	57.7	61.2	41.3	-
BSAE	**72.3**	**69.4**	**59.7**	58.6	**53**	**62.6**

five datasets, while 3.6% less than SRGAN [53] under the challenging split of SUN. We conjecture that our regression model is over-fitted in terms of scarce training instances of each class.

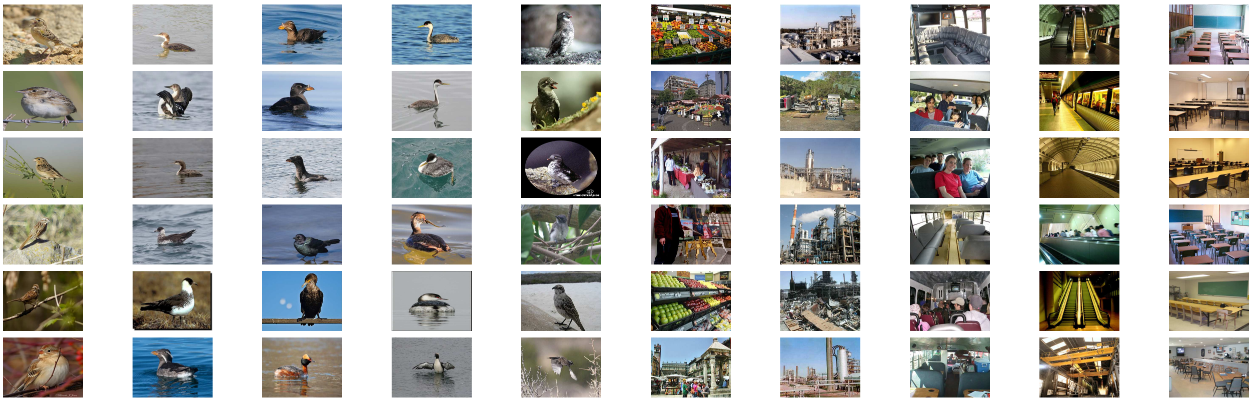
It is noting that there is no single approach claims the best results on all datasets simultaneously [60]. The aforementioned improvements actually create new baselines in the area of ZSL, given that most of the compared models utilize more complicated nonlinear formulations and some of them combine complementary semantic spaces or even generate richer features for unseen classes. In contrast, we apply only one type of semantic space as well as computational fast linear projection functions, but gain a significant performance boost.

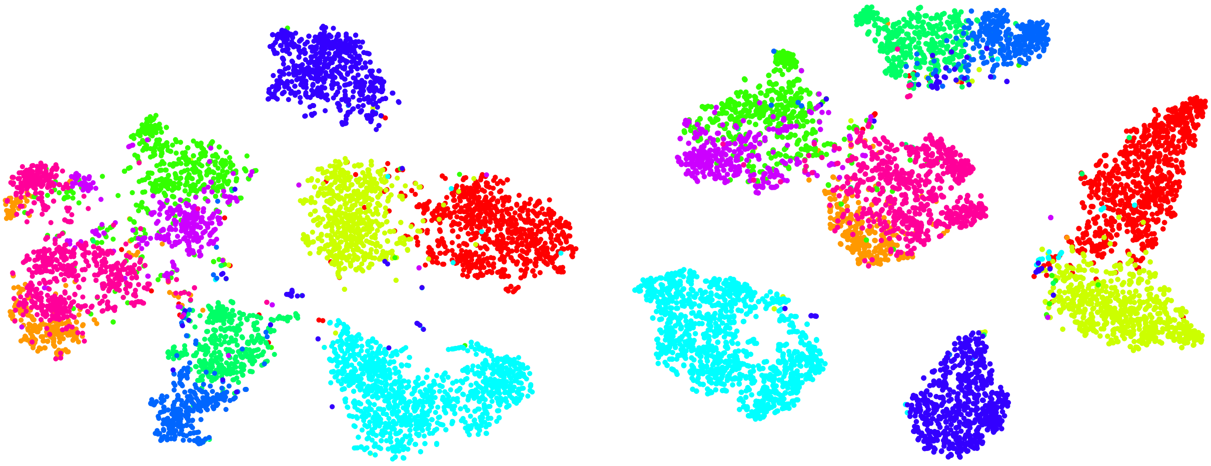
4.2.2. Zero-shot retrieval

We aim to evaluation the effectiveness of the decoder of BSAE via the image retrieval task, which is defined as searching top matched images by taking the provided semantic prototypes of unseen classes as queries. The ratio of the number of accurately retrieved images to that of all retrieved images, namely precision, is regarded as the measurement. Table 5 reports 5 out of 50 classes in CUB and 5 out of 65 classes in SUN and depicts the qualitative results of our designed model with highest anterior and posterior scores for each unseen class. Specifically, each column is a category, with class name and precision are shown at the top. The first three rows in the middle are the top-3 correctly retrieved instances. The following three rows are the top-3 misclassified instances in each unseen class. Observing form the top correct images, BSAE reasonably captures discriminative visual information only using its semantic prototype. It suggests that the adaptation regularization helps make approxiamate inference of unseen instances. Meanwhile, taking the class in the second column as an example, Pomarine Jaeger and Rhinoceros Auklet are visually similar to Pacific Loon, the discriminative ability of the decoder is not enough to distinguish the visual appearances between them. Due to the strong visual similarity and only a few different attributes among the classes, we further notice that it is hardly recognize these classes without expert knowledge, even for humans.

Table 5. Qualitative evaluations of our proposed model on CUB (left) and SUN (right). The first five columns are classes from CUB and the rest from SUN. We report the top-3 instances accurately assigned to each class in the middle and the last rows shows the top-3 misclassified instances.

Grasshopper Sparrow	Pacific Loon	Rhinoceros Auklet	Western Grebe	Least Auklet	market outdoor	recycling plant outdoor	van interior	subway station platform	lecture room
50.5%	88.2%	86.8%	85.7%	84.8%	82.6%	37.8%	44.7%	54.8%	54.5%





(a) Groudtruth visual features

(b) Predicted semantic features

Figure 4. Visualization of the distribution of 10 unseen data points on AWA1 under PS protocol in visual space and semantic space respectively. The left part shows the test features with true labels and the right part shows the learned semantic representations. Different classes are shown in different colors. Better viewed in color.

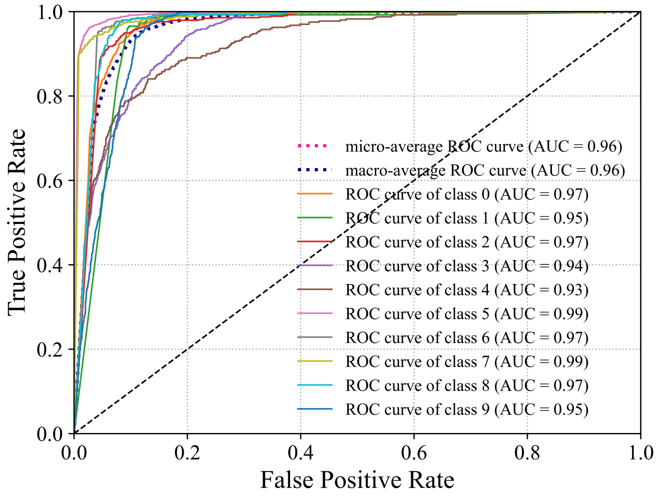
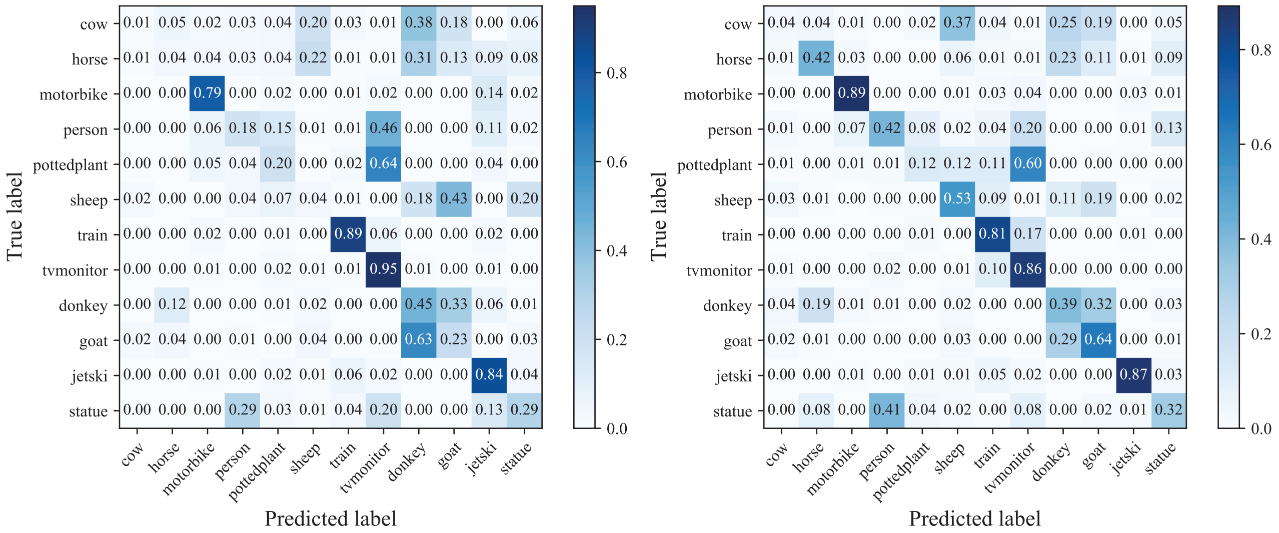


Figure 5. Comparing the ROC curve and AUC value visualization on AWA1 dataset under CZSL setting.

4.3. Further evaluation

4.3.1. Visualization

For straightforward illustration of BSAE in ZSL. We explore t-SNE visualization [83] to compare the visual features with genuine class label (left) and semantic representations with the predicted class labels (right) in Figure 4. Each color represents clustering in the same class and all the features are embedding into two dimensions using t-SNE. It suggests that our model captures the underlying global distribution in the semantic space and performs better on the dataset. It is worth that our model alle-



(a) LisGAN

(b) BSAE

Figure 6. The confusion matrix on the evaluation of aP&Y.

viates the hubness problem in the lower dimensional semantic space. Moreover, the instances of the same class are grouped into one cluster in Figure 4, which confirms that the discriminative semantic representations learned by our model are able to cluster visually similar instances. Therefore, our proposed model preserves the local information of target unseen classes that the closeness are kept in the projected semantic representations.

The ROC curve and AUC value depict the tradeoff between specificity (False Positive Rate) and sensitivity (True Positive Rate) as a metric of the performance of our proposed BSAE. Figure 5 shows the results of the ROC curve and AUC value on AWA1 under PS protocol. We can observe that the ROC curve of the 10 unseen classes are close to the top-left corner of the plot, even though using the simplest KNN classifier.

We observed that almost all methods performed worst on aP&Y compared to other datasets. In order to show our experimental results in a more fine-grained manner, we take the PS protocol of aP&Y dataset as an example, compared with the best competitor LisGAN [68]. Figure 6 shows the confusion matrix of LisGAN and our model i.e., BSAE on the 12 unseen classes. The value in the diagonal of the confusion matrix indicates the ratio of the correctly predicted of each class. The darker color represents the higher class-wise accuracy. It can be seen that BSAE generally performs better on the most classes. Concretely, we boosts 38%, 10%, 24%, 49%, 41%, 3% and 3% on "horse", "motorbike", "person", "sheep", "goat", "jetski" and "statue" against LisGAN respectively. Although the GAN model directly handle zero-shot problem by converting it to a supervised task, we find that our model perform better than GAN-based model, i.e., LisGAN. In addition, it is common that one model does not have the highest accuracy on each unseen class. There will be great improvement in the future.

We measure the inter-class and intra-class distances to investigate BSAE can alleviate domain shift and hubness problem. We follow the two measurements provided by [84]:

$$D^c_{intra} = \frac{1}{n_c} \sum_i D(\varphi(A_c), \psi(s^c_i)), \tag{4.2}$$

$$D^c_{inter} = \frac{1}{C-1} \sum_{j \neq c} D(\varphi(A_c), \varphi(A_j)). \tag{4.3}$$

Where n_c represents the data size of the cth class. C means the number of the classes. $\varphi(\cdot)$ and $\psi(\cdot)$ denote the two dimensional outputs of t-SNE [83]. $D(\cdot)$ is the cosine distance, which reflects the degree of similarity between actual and the compared one [81]. D^c_{intra} stands for the mean distance between the cth class prototype A_c and semantic representations of instances in that class. D^c_{inter} stands for the mean distance between the cth class prototype A_c and all other classes. We compare our proposed model with TSTD [59], which is the best competitor under SS protocol on AWA1 dataset. For consideration of fairness, we re-implement the experiments of the two methods under the same settings in AWA1, i.e., use ResNet-101 features and take continuous class-level attributes as semantic representations. Different from TSTD [59] that applies the attributes of unseen classes during training to improve the performance, BSAE obtains smaller intra-class distances and larger inter-class distances with a large margin as illustrated in Table 6. Thus, BSAE is capable of alleviate domain shift problem as well as hubness problem in lower dimensional semantic space.

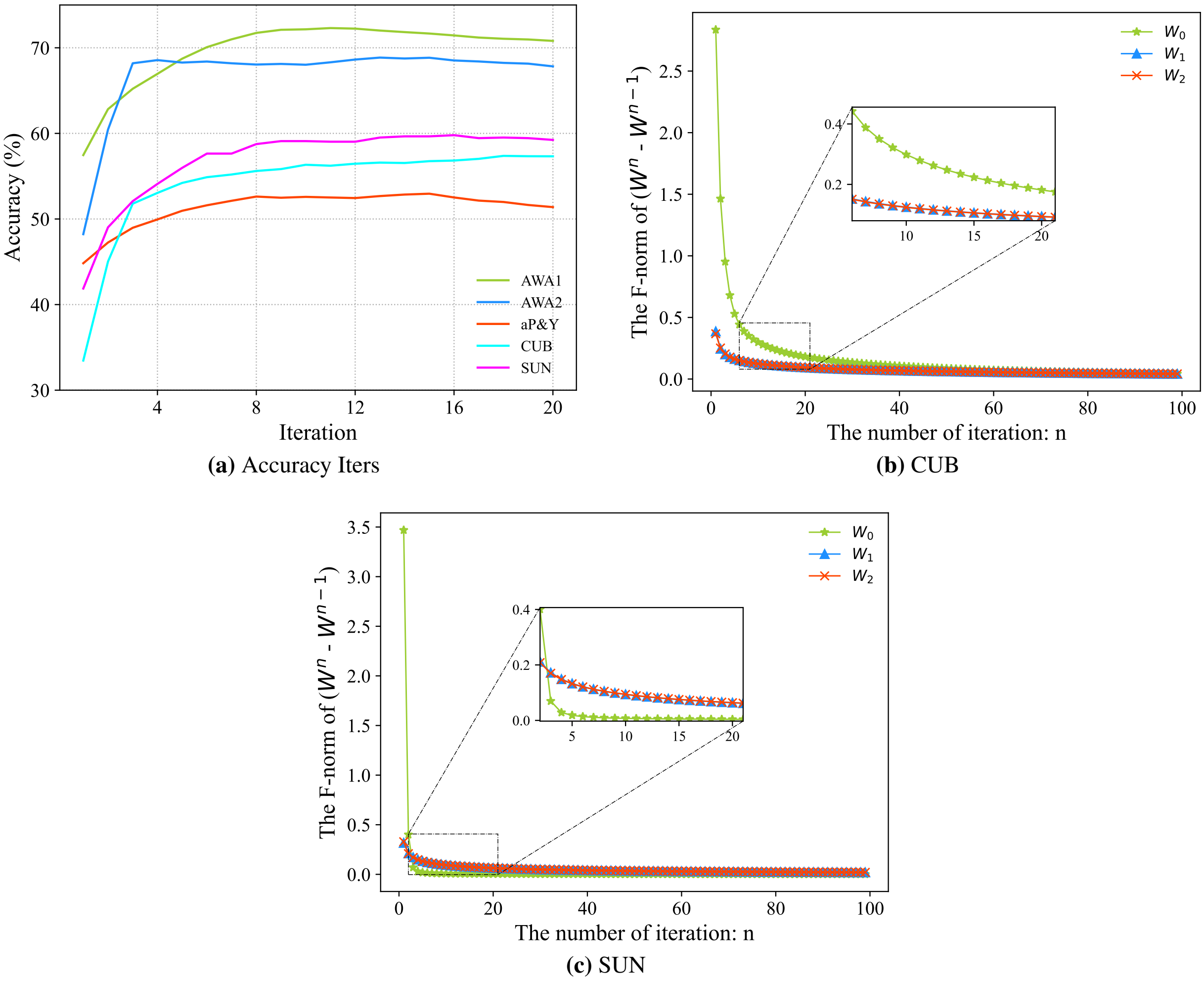
4.3.2. Complexity and convergence analysis

In this section, we analyze the complexity and convergence of BSAE. It is remarkable that the operation of **Algorithm** 1 mostly comes from matrix multiplication. Obviously, it can accelerate the training process greatly. Additionally, we set 200 as the maximum iterations. The F-norm of the parameter variation with respect to the iteration on fine-grained datasets are reported.

From Figure 7a, it is notable that our model reaches 80% of the accuracy within 4 iterations and is close to the highest accuracy around 10 iterations on coarse-grained dataset, e.g., AWA1 and around 20 iterations on fine-grained datasets. These demonstrate that our algorithm has a good practical application for its low complexity and good performance.

Table 6. Comparative intra-class and inter-class distances of unseen classes in AWA1 dataset.

Class Name	D^c_{intra}		D^c_{inter}	
	BSAE	TSTD	BSAE	TSTD
chimpanzee	**0.275**	0.444	**1.663**	1.361
giant+panda	**0.441**	0.442	**1.583**	1.264
leopard	**0.36**	0.39	**1.69**	1.246
persian+cat	**0.077**	0.388	**1.887**	1.501
pig	**0.199**	0.43	**1.868**	1.399
hippopotamus	**0.412**	0.498	**1.594**	1.337
humpback+whale	**0.318**	0.35	**1.28**	1.257
raccoon	**0.224**	0.333	**1.326**	1.192
rat	**0.215**	0.28	**1.28**	1.234
seal	0.257	**0.228**	**1.31**	1.243



(a) Accuracy Iters

(b) CUB

(c) SUN

Figure 7. Accuracy on five datasets and convergence curve on fine-grained datasets: CUB and SUN with iterations.

Figure 7b and 7c shows that the algorithm converges within 40 iterations. It is obvious that the decoder W_2 of the target domain is well restricted by the decoder W_1 of the source domain, which verifies the significance of the adaptation regularization. Moreover, these observations finally support the theoretical analysis of complexity and convergence in Section 3.3.

4.3.3. Ablation study

To provide further insights into the role of the two regularization terms: $||W_0X_s - S_{tr}||_F^2$ and $||W_2 - W_1||_F^2$ in our proposed objective function in helping the model to achieve better performance, we simplify our full model BSAE with various stripped-down versions of the model on the PS protocol of CZSL. Specially, for $\lambda_1 = 0$, when the similarity constraint of he predicted and actual semantic representations are not exploited to encoder, i.e, Eq 3.3 without $||W_0X_s - S_{tr}||_F^2$ term (denoted BSAE-SR), BSAE degrades to only contain adaptation regularization. The encoder does not ensure that the learned semantic representations of each instance is close to its class prototype. For $\lambda_2 = 0$, i.e., BSAE without the adaptation regularization $||W_2-W_1||_F^2$ (denoted BSAE-DR), the decoder in the target domain

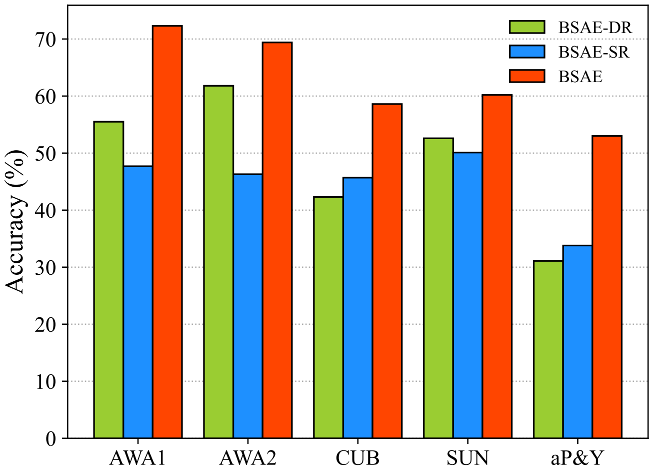


Figure 8. Evaluation of the contributions of each component of our framework on five benchmark datasets (ResNet-101 features).

is not restricted to derive from the decoder in the source domain, which is supervised by the semantic prototypes of the source instances. Figure 8 shows clearly that the two terms contribute to the superior performance of proposed model. We achieved up to around 10% improvements on five datasets. It is reasonable to believe that the learning of semantics will help the learning of domain adaptation among seen and unseen classes.

5. Conclusions

In this paper, we have proposed a novel model called Bi-shifting Auto-Encoder to perform efficient zero-shot recognition in semantic and visual space by taking advantage of autoencoder network. Our model learns the generalizable and computationally fast projection functions in transductive settings, which leverages the labeled source data and the visual features of the unlabeled target data. In particular, to improve the discriminability of the semantic embeddings, the encoder is constrained by aligning the semantic representations of the labeled source instances with their corresponding prototypes of the seen classes. Furthermore, to guarantee the generalizability of the projected semantic representations, two different decoders reconstruct the visual features of the instances in source and target domain simultaneously with the adaptation regularization. Thus, our model recovers the interactions between visual features and semantics, and is able to alleviate the projection shift problem. Extensive experiments are conducted on five benchmark datasets and comparative evaluations demonstrate that our model yields superior performance on zero-shot learning. The major limitation of our model lies in the fact that each class is represented by one attribute prototype in the semantic space, which is insufficient to completely characterize the features of the class, resulting in the semantics of the instances may be misplaced from the class prototype. Therefore, our research work will put effort in exploring different types of semantic representations to investigate the relationships between classes, especially the subtle differences among the classes of fine-grained datasets. An additional limitation of this study is that the full set of unlabeled target instances are utilized, ignoring their distinctive effects on the model learning. A natural processing of this work is to explore the most useful unseen instances that facilitate the zero-shot classification.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. M. Everingham, S. M. Eslami, L. Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: a retrospective, *Int. J. Comput. Vis.*, **111** (2015), 98–136. https://doi.org/10.1007/s11263-014-0733-5
2. S. J. Dickinson, A. Leonardis, B. Schiele, M. J. Tarr, *Object categorization: computer and human vision perspectives*, Cambridge University Press, Cambridge, 2009. https://doi.org/10.1017/cbo9780511635465
3. X. Zhang, Y. H. Yang, Z. Han, H. Wang, C. Gao, Object class detection: a survey, *ACM Comput. Surv.*, **46** (2013), 1–53. https://doi.org/10.1145/2522968.2522978
4. Y. Li, S. Wang, Q. Tian, X. Ding, Feature representation for statistical-learning-based object detection: a review, *Pattern Recognit.*, **48** (2015), 3542–3559. https://doi.org/10.1016/j.patcog.2015.04.018
5. Z. Zhao, P. Zheng, S. Xu, X. Wu, Object detection with deep learning: a review, *IEEE Trans. Neural Netw. Learn. Syst.*, **30** (2019), 3212–3232. https://doi.org/10.1109/tnnls.2018.2876865
6. K. Oksuz, B. C. Cam, S. Kalkan, E. Akbas, Imbalance problems in object detection: a review, *IEEE Trans. Pattern Anal. Mach. Intell.*, **43** (2021), 3388–3415. https://doi.org/10.1109/tpami.2020.2981890
7. L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, et al. Deep learning for generic object detection: a survey, *Int. J. Comput. Vis.*, **128** (2020), 261–318. https://doi.org/10.1007/s11263-019-01247-4
8. S. Ghosh, N. Das, I. Das, U. Maulik, Understanding deep learning techniques for image segmentation, *ACM Comput. Surv.*, **52** (2019), 1–35. https://doi.org/10.1145/3329784
9. S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: a survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, (2021). https://doi.org/10.1109/tpami.2021.3059968
10. R. Datta, D. Joshi, J. Li, J. Z. Wang, Image retrieval: ideas, influences, and trends of the new age, *ACM Comput. Surv.*, **40** (2008), 1–60. https://doi.org/10.1145/1348246.1348248
11. D. Zhang, M. M. Islam, G. Lu, A review on automatic image annotation techniques, *Pattern Recognit.*, **45** (2012), 346–362. https://doi.org/10.1016/j.patcog.2011.05.013
12. X. Li, T. Uricchio, L. Ballan, M. Bertini, C. G. Snoek, A. D. Bimbo, Socializing the semantic gap: a comparative survey on image tag assignment, refinement, and retrieval, *ACM Comput. Surv.*, **49** (2016), 1–39. https://doi.org/10.1145/2906152

13. P. Wiriyathammabhum, D. Summers-Stay, C. Fermüller, Y. Aloimonos, Computer vision and natural language processing: recent approaches in multimedia and robotics, *ACM Comput. Surv.*, **49** (2020), 1–44. https://doi.org/10.1145/3009906

14. Y. Belinkov, J. Glass, Analysis methods in neural language processing: a survey, *Trans. Assoc. Comput. Linguist.*, **7** (2019), 49–72. https://doi.org/10.1162/tacl_a_00254

15. K. Gauman, B. Leibe, Visual object recognition, *Synth. Lect. Artif. Intell. Mach. Learn.*, **5** (2011), 1–181. https://doi.org/10.2200/S00332ED1V01Y201103AIM011

16. Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, *IEEE Trans. Pattern Anal. Mach. Intell.*, **35** (2013), 1798–1828. https://doi.org/10.1109/tpami.2013.50

17. I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, MIT Press, Cambridge, 2016.

18. G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, et al., A survey on deep learning in medical image analysis, *Med. Image Anal.*, **42** (2017), 60–88. https://doi.org/10.1016/j.media.2017.07.005

19. J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, et al., Recent advances in convolutional neural networks, *Pattern Recognit.*, **77** (2018), 354–377. https://doi.org/10.1016/j.patcog.2017.10.013

20. M. G. Kendall, A. Stuart, J. K. Ord, *Kendall's advanced theory of statistics*, 5^{th} edition, Oxford University Press, Oxford, 1987.

21. J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: a large-scale hierarchical image database, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2009), 248–255. https://doi.org/10.1109/cvpr.2009.5206848

22. *California Institute of Technology, The caltech-ucsd birds-200-2011 dataset*, Computation & Neural Systems Technical Report of California Institute of Technology, 2011. Available from: `http://www.vision.caltech.edu/visipedia/CUB-200-2011.html`.

23. Y. Wang, Q. Yao, J. T. Kwok, L. M. Ni, Generalizing from a few examples: a survey on few-shot learning, *ACM Comput. Surv.*, **53** (2020), 1–34. https://doi.org/10.1145/3386252

24. W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, T. E. Boult, Toward open set recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, **35** (2012), 1757–1772. https://doi.org/10.1109/tpami.2012.256

25. C. Geng, S. Huang, S. Chen, Recent advances in open set recognition: a survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, **43** (2021), 3614–3631. https://doi.org/10.1109/tpami.2020.2981604

26. I. Biederman, Recognition-by-components: a theory of human image understanding, *Psychol. Rev.*, **94** (1987), 115–147. https://doi.org/10.1037/0033-295x.94.2.115

27. C. H. Lampert, H. Nickisch, S. Harmeling, Attribute-based classification for zero-shot visual object categorization, *IEEE Trans. Pattern Anal. Mach. Intell.*, **36** (2013), 453–465. https://doi.org/10.1109/tpami.2013.140

28. W. Xu, Y. Xian, J. Wang, B. Schiele, Z. Akata, Attribute prototype network for zero-shot learning, preprint, `arXiv:2008.08290`.

29. S. Changpinyo, W. L. Chao, B. Gong, F. Sha, Classifier and exemplar synthesis for zero-shot learning, *Int. J. Comput. Vis.*, **128** (2020), 166–201. https://doi.org/10.1007/s11263-019-01193-1

30. Z. Ji, H. Wang, Y. Pang, L. Shao, Dual triplet network for image zero-shot learning, *Neurocomputing*, **373** (2020), 90–97. https://doi.org/10.1016/j.neucom.2019.09.062

31. Y. Ma, X. Xu, F. Shen, H. Shen, Similarity preserving feature generating networks for zero-shot learning, *Neurocomputing*, **406** (2020), 333–342. https://doi.org/10.1016/j.neucom.2019.08.111

32. M. M. Palatucci, D. A. Pomerleau, G. E. Hinton, T. Mitchell, Zero-shot learning with semantic output codes, in *Ann. Conf. Neural Inf. Process. Syst.*, MIT Press, **22** (2009), 1410–1418.

33. M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, et al., Google' s multilingual neural machine translation system: enabling zero-shot translation, *Trans. Assoc. Comput. Linguist.*, **5** (2017), 339–351. https://doi.org/10.1162/tacl_a_00065

34. N. Nakashole, R. Flauger, Knowledge distillation for bilingual dictionary induction, in *Conf. Empirical Methods Nat. Language Process.*, ACL, (2017), 2497–2506. https://doi.org/10.18653/v1/d17-1264

35. H. Larochelle, D. Erhan, Y. Bengio, Zero-data learning of new tasks, in *AAAI Conf. Artif. Intell.*, AAAI, **1** (2008), 646–651.

36. C. H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2009), 951–958. https://doi.org/10.1109/cvpr.2009.5206594

37. N. Mohammad, M. Tomas, B. Samy, S. Yoram, S. Jonathon, F. Andrea, et al., Zero-shot learning by convex combination of semantic embeddings, preprint, `arXiv:1312.5650`.

38. A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzat, et al., Devise: a deep visual-semantic embedding model, in *Ann. Conf. Neural Inform. Process. Syst.*, MIT Press, **2** (2013), 2121–2129.

39. R. Socher, M. Ganjoo, H. Sridhar, O. Bastani, C. D. Manning, A. Y. Ng, Zero-shot learning through cross-modal transfer, in *Ann. Conf. Neural Inform. Process. Syst.*, MIT Press, **26** (2013), 935–943.

40. S. Reed, Z. Akata, H. Lee, B. Schiele, Learning deep representations of fine-grained visual descriptions, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2016), 49–58. https://doi.org/10.1109/cvpr.2016.13

41. Z. Akata, S. Reed, D. Walter, H. Lee, B. Schiele, Evaluation of output embeddings for fine-grained image classification, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2015), 2927–2936. https://doi.org/10.1109/cvpr.2015.7298911

42. Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, B. Schiele, Latent embeddings for zero-shot classification, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2016), 69–77. https://doi.org/10.1109/cvpr.2016.15

43. Y. Fu, T. M. Hospedales, T. Xiang, S. Gong, Transductive multi-view zero-shot learning, *IEEE Trans. Pattern Anal. Mach. Intell.*, **37** (2015), 2332–2345. https://doi.org/10.1109/tpami.2015.2408354

44. S. Rahman, S. Khan, F. Porikli, A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning, *IEEE Trans. Image Process.*, **27** (2018), 5652–5667. https://doi.org/10.1109/tip.2018.2861573

45. S. Daghaghi, T. Medini, A. Shrivastava, Sdm-net: a simple and effective model for generalized zero-shot learning, preprint, arXiv:1909.04790.

46. Z. Jia, Z. Zhang, L. Wang, C. Shan, T. Tan, Deep unbiased embedding transfer for zero-shot learning, *IEEE Trans. Image Process.*, **29** (2019), 1958–1971. https://doi.org/10.1109/tip.2019.2947780

47. B. Romera-Paredes, P. H. S. Torr, An embarrassingly simple approach to zero-shot learning, in *Int. Conf. Machine Learn.*, ACM, **37** (2015), 2152–2161. https://doi.org/10.1007/978-3-319-50077-5_2

48. E. Kodirov, T. Xiang, S. Gong, Semantic autoencoder for zero-shot learning, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2017), 3174–3183. https://doi.org/10.1109/cvpr.2017.473

49. S. Changpinyo, W. L. Chao, B. Gong, F. Sha, Synthesized classifiers for zero-shot learning, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2016), 5327–5336. https://doi.org/10.1109/cvpr.2016.575

50. Y. Xian, T. Lorenz, B. Schiele, Z. Akata, Feature generating networks for zero-shot learning, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2018), 5542–5551. https://doi.org/10.1109/cvpr.2018.00581

51. H. Huang, C. Wang, P. S. Yu, C. Wang, Generative dual adversarial network for generalized zero-shot learning, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2019), 801–810. https://doi.org/10.1109/cvpr.2019.00089

52. S. Xu, Z. Gao, G. Xie, Bi-semantic reconstructing generative network for zero-shot learning, preprint, arXiv:1912.03877.

53. Z. Ye, F. Lyu, L. Li, Q. Fu, J. Ren, F. Hu, Sr-gan: semantic rectifying generative adversarial network for zero-shot learning, in *IEEE Int. Conf. Multimedia & Expo*, IEEE, (2019), 85–90. https://doi.org/10.1109/icme.2019.00023

54. Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, A. Elgammal, A generative adversarial approach for zero-shot learning from noisy texts, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2018), 1004–1013. https://doi.org/10.1109/cvpr.2018.00111

55. A. Mishra, S. Krishna Reddy, A. Mittal, H. A. Murthy, A generative model for zero shot learning using conditional variational autoencoders, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2018), 2188–2196. https://doi.org/10.1109/cvprw.2018.00294

56. Z. Wan, D. Chen, Y. Li, X. Yan, J. Zhang, Y. Yu, et al., Transductive zero-shot learning with visual structure constraint, preprint, arXiv:1901.01570.

57. M. Ye, Y. Guo, Zero-shot classification with discriminative semantic representation learning, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2017), 7140–7148. https://doi.org/10.1109/cvpr.2017.542

58. Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for image classification, *IEEE Trans. Pattern Anal. Mach. Intell.*, **38** (2016), 1425–1438. https://doi.org/10.1109/tpami.2015.2487986

59. Y. Yu, Z. Ji, X. Li, J. Guo, Z. Zhang, H. Ling, et al., Transductive zero-shot learning with a self-training dictionary approach, *IEEE T. Cybern.*, **48** (2018), 2908–2919. https://doi.org/10.1109/tcyb.2017.2751741

60. Y. Shi, W. Wei, Discriminative embedding autoencoder with a regressor feedback for zero-shot learning, *IEEE Access*, **8** (2020), 11019–11030. https://doi.org/10.1109/access.2020.2964613

61. J. Song, C. Shen, Y. Yang, Y. Liu, M. Song, Transductive unbiased embedding for zero-shot learning, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2018), 1024–1033. https://doi.org/10.1109/cvpr.2018.00113

62. V. K. Verma, P. Rai, A simple exponential family framework for zero-shot learning, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, (2017), 792–808. https://doi.org/10.1007/978-3-319-71246-8_48

63. M. Kan, S. Shan, X. Chen, Bi-shifting auto-encoder for unsupervised domain adaptation, in *Int. Conf. Comput. Vis.*, IEEE, (2015), 3846–3854. https://doi.org/10.1109/iccv.2015.438

64. J. Zhang, W. Li, P. Ogunbona, Joint geometrical and statistical alignment for visual domain adaptation, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2017), 1859–1867. https://doi.org/10.1109/cvpr.2017.547

65. E. Kodirov, T. Xiang, Z. Fu, S. Gong, Unsupervised domain adaptation for zero-shot learning, in *Int. Conf. Comput. Vis.*, IEEE, (2015), 2452–2460. https://doi.org/10.1109/iccv.2015.282

66. Y. Liu, Q. Gao, J. Li, J. Han, L. Shao, Zero shot learning via low-rank embedded semantic autoencoder, in *International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, (2018), 2490–2496. https://doi.org/10.24963/ijcai.2018/345

67. Z. Zhang, V. Saligrama, Zero-shot learning via semantic similarity embedding, in *Int. Conf. Comput. Vis.*, IEEE, (2015), 4166–4174. https://doi.org/10.1109/iccv.2015.474

68. J. Li, M. Jin, K. Lu, Z. Ding, Z. Huang, Leveraging the invariant side of generative zero-shot learning, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2019), 7394–7403. https://doi.org/10.1109/cvpr.2019.00758

69. M. Radovanović, A. Nanopoulos, M. Ivanović, Hubs in space: popular nearest neighbors in high-dimensional data, *J. Mach. Learn. Res.*, **11** (2010), 2487–2531.

70. L. Zhang, T. Xiang, S. Gong, Learning a deep embedding model for zero-shot learning, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2017), 2021–2030. https://doi.org/10.1109/cvpr.2017.321

71. Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, Y. Matsumoto, Ridge regression, hubness, and zero-shot learning, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, (2015), 135–151. https://doi.org/10.1007/978-3-319-23528-8_9

72. Y. Wu, W. Cao, Y. Liu, Z. Ming, J. Li, B. Lu, Semantic auto-encoder with l2-norm constraint for zero-shot learning, in *Int. Conf. Mach. Learn. Comput.*, ACM, (2021), 101–105. https://doi.org/10.1145/3457682.3457699

73. Y. Li, D. Wang, H. Hu, Y. Lin, Y. Zhuang, Zero-shot recognition using dual visual-semantic mapping paths, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2017), 3279–3287. https://doi.org/10.1109/cvpr.2017.553

74. P. Lancaster, M. Tismenetsky, *The theory of matrices: with applications*, 2^{nd} edition, Academic Press, Amsterdam, 1985.

75. Y. Xian, C. H. Lampert, B. Schiele, Z. Akata, Zero-shot learning – a comprehensive evaluation of the good, the bad and the ugly, *IEEE Trans. Pattern Anal. Mach. Intell.*, **41** (2018), 2251–2265. https://doi.org/10.1109/tpami.2018.2857768

76. A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2009), 1778–1785. https://doi.org/10.1109/cvpr.2009.5206772

77. G. Patterson, C. Xu, H. Su, J. Hays, The sun attribute database: beyond categories for deeper scene understanding, *Int. J. Comput. Vis.*, **108** (2014), 59–81. https://doi.org/10.1007/s11263-013-0695-z

78. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, in *IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, (2015), 1–9. https://doi.org/10.1109/cvpr.2015.7298594

79. S. M. Shojaee, M. S. Baghshah, Semi-supervised zero-shot learning by a clustering-based approach, preprint, `arXiv:1605.09016`.

80. Y. Guo, G. Ding, J. Han, Y. Gao, Zero-shot learning with transferred samples, *IEEE Trans. Image Process.*, **26** (2017), 3277–3290. https://doi.org/10.1109/tip.2017.2696747

81. Y. Annadani, S. Biswas, Preserving semantic relations for zero-shot learning, in *IEEE Conf. Comput. Vis. Pattern Recognition*, IEEE, (2018), 7603–7612. https://doi.org/10.1109/cvpr.2018.00793

82. H. Zhang, Y. Long, Y. Guan, L. Shao, Triple verification network for generalized zero-shot learning, *IEEE Trans. Image Process.*, **28** (2019), 506–517. https://doi.org/10.1109/tip.2018.2869696

83. L. Van der Maaten, G. Hinton, Visualizing data using t-sne, *J. Mach. Learn. Res.*, **9** (2008), 2579–2605.

84. Z. Zhang, Y. Li, J. Yang, Y. Li, M. Gao, Cross-layer autoencoder for zero-shot learning, *IEEE Access*, **7** (2019), 167584–167592. https://doi.org/10.1109/access.2019.2953454

AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)