Zero-shot Learning via Recurrent Knowledge Transfer

Bo Zhao[1,2], Xinwei Sun[3], Xiaopeng Hong[4], Yuan Yao[5] and Yizhou Wang[1,2]
[1]Nat'l Engineering Laboratory for Video Technology,
Cooperative Medianet Innovation Center,
Computer Science Dept., Peking University
[2]Deepwise AI Lab
[3]School of Mathematical Science, Peking University
[4]Center for Machine Vision and Signal Analysis, University of Oulu
[5]Department of Mathematics, Hong Kong University of Science and Technology
bozhao@pku.edu.cn, sxwxiaoxiaohehe@pku.edu.cn
xiaopeng.hong@oulu.fi, yuany@ust.hk, Yizhou.Wang@pku.edu.cn

Abstract

Zero-shot learning (ZSL) which aims to learn new concepts without any labeled training data is a promising solution to large-scale concept learning. Recently, many works implement zero-shot learning by transferring structural knowledge from the semantic embedding space to the image feature space. However, we observe that such direct knowledge transfer may suffer from the space shift problem *in the form of the inconsistency of geometric structures in the training and testing spaces. To alleviate this problem, we propose a novel method which actualizes recurrent knowledge transfer (RecKT) between the two spaces. Specifically, we unite the two spaces into the joint embedding space in which unseen image data are missing. The proposed method provides a* synthesis-refinement *mechanism to learn the shared subspace structure (SSS) and synthesize missing data simultaneously in the joint embedding space. The synthesized unseen image data are utilized to construct the classifier for unseen classes. Experimental results show that our method outperforms the state-of-the-art on three popular datasets. The ablation experiment and visualization of the learning process illustrate how our method can alleviate the space shift problem. By product, our method provides a perspective to interpret the ZSL performance by implementing subspace clustering on the learned SSS.*

1. Introduction

Currently, supervised-learning frameworks only focus on a small fraction of concepts in the real world. For instance, ImageNet Large Scale Visual Recognition Challenge (ILSVRC) contains 1,000 popular categories in ImageNet [9] for training and testing. This number is far away from human beings' learning ability that ordinary people can distinguish more than 30,000 basic-level concepts [3]. In addition, to learn a particular concept, popular deep networks [23, 20, 36, 39, 16, 34] require hundreds to thousands of labeled training data. However, it may be impossible to collect enough number of labeled training data for all categories, such as wild animals, rare plants, and industrial products. Hence, it is unrealistic to extend the recognition ability of machines only relying on collecting more training data. Inspired by human beings' ability that people can learn from descriptions without visual samples, zero-shot learning (ZSL) [33] aims to learn new concepts without any training data. In this paper, we focus on zero-shot object recognition [22].

In ZSL, labeled images from seen classes (source domain) are given for training, while no training images from unseen classes (target domain) are provided. The goal is to recognize testing images from unseen classes by leveraging auxiliary knowledge to enable knowledge transfer. Usually, images are embedded in the image feature space using hand-crafted [32, 27, 8] or deep [20, 36, 39, 16] feature extractors, and labels are embedded in the semantic (label) embedding space [13] using auxiliary knowledge, e.g., attributes [22], word vectors of labels [38]. Recently, [46] proposed a new evaluation protocol and data splits for ZSL to avoid the overlap between some unseen classes and ImageNet-2012 dataset which is frequently used for pre-training feature extractors. The new setting has been evaluated in papers such as [50, 45]. In this paper, we still follow the classical ZSL setting used in [22, 43, 5, 53, 15, 25].

According to the different modes of knowledge transfer, we categorize existing ZSL methods into two types, namely,

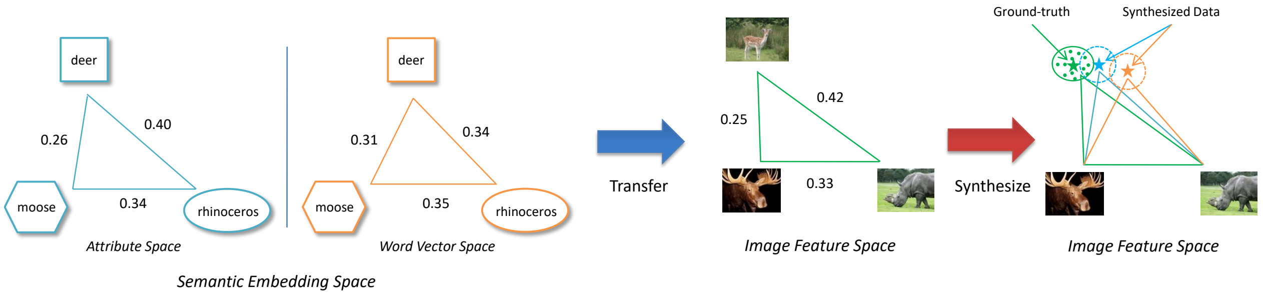


Figure 1: Illustration of the space shift problem. The weight on each edge is the normalized Euclidean distance (dissimilarity) between two classes. The star is the center of data (points), which represents the data distribution. Clearly, the geometric structures significantly differ from each other in the three embedding spaces. Hence, the synthesized image data (blue and yellow stars) based on the transferred structure from other spaces deviate from the ground-truth data distribution (the green star). Therefore, the classifier learned based on the synthesized data may not fit real testing data.

the *mapping-transfer framework* and the *structure-transfer framework*. The mapping-transfer framework [33, 22] aims to learn the mapping function from the image feature space (F) to the semantic embedding space (E), i.e. F→E, using seen class data. A testing unseen image is first mapped to the semantic embedding space (using the learned mapping function) then classified in this space [22, 19]. In other words, the learned mapping function is learned from seen classes then tested on unseen classes. This framework mainly suffers from the "domain shift problem" [13] which indicates the shift of training (seen classes) and testing (unseen classes) domains.

Recently structure-transfer methods [30, 5, 43, 54, 15] become popular. They try to learn the structural knowledge in the semantic embedding space and then transfer it to the image feature space for synthesizing unseen image data [43] or classification models [5]. In the testing phase, images of unseen classes are classified based on these synthesized data or models. The underlying assumption of this framework is that the structures (e.g., manifold structure) among classes are, to some extent, similar in the two spaces. Hence, the structural knowledge is transferable between the two spaces. Compared to the mapping-transfer framework, the structure-transfer framework avoids the domain shift problem, as it does not transfer the mapping function between **training and testing classes**. However, we find that such structure-transfer framework suffers from another problem caused by the shift of **training and testing spaces**, which will be described in detail in Section 1.1.

1.1. Space Shift Problem

Different embedding spaces are built using inherently different data and methods. Specifically, the image feature space is constructed by image features [32, 27, 8, 20, 36, 39, 16]. The attribute space and word vector space are built based on human knowledge [22] and the co-occurrence between words [29] respectively. The geometric structures among classes differ in these embedding spaces. We illustrate this geometric difference in Fig. 1. As shown in the figure, although "moose" and "deer" are always the closest in different embedding spaces, the geometric structures formed by "moose", "deer" and "rhinoceros" significantly change. In existing methods, the structure among classes learned in the semantic embedded space is directly transferred to the image feature space for synthesizing virtual image data. However, the learned structural knowledge is biased towards the training space. Hence, the synthesized image data (the blue and yellow stars in the figure) deviate from the ground-truth data distribution (the green star). Furthermore, the classifier trained on such biased synthesized data will not fit the real testing data. Therefore, the direct knowledge transfer between the image feature and semantic embedding spaces suffers from the problem of the inconsistency of geometric structures in the training and testing spaces. We name it the "*space shift problem*" due to the shift of training and testing spaces.

The essential reason why traditional structure-transfer methods suffer from the space shift problem is that the structural knowledge is learned and exploited in different spaces asynchronously. We assume that the image feature space and the semantic embedding space share part of the structure, however, they both have their own private parts. When the structural knowledge is learned in one space, both the shared and private parts are learned [4]. Hence, it will be biased to the training space.

To solve this problem, this paper focuses on reducing the bias (or private knowledge) during the learning process. We propose a novel method (illustrated in Fig. 2) to learn the shared subspace structure (SSS) in the two embedding spaces. Specifically, we joint the image feature and semantic embedding spaces by concatenating the corresponding datapoints in the two spaces. An alternating optimization algorithm is proposed to learn the SSS and synthesize missing data (unseen image data) simultaneously. During the learning process, the knowledge is recurrently transferred between the image feature and semantic embedding spaces. Hence, the learned SSS and synthesized missing data adapt to both two spaces. In this way, the space shift problem is alleviated. The synthesized unseen image data are further utilized to learn the classifier for unseen classes.

In comparison experiments (Sec. 3.2), our method outperforms the state-of-the-art ZSL methods on three popular datasets, namely Animals with Attributes [21], Caltech-UCSD Birds-200-2011 [42] and ImageNet [9]. In Sec. 3.3, we implement spectral clustering on the learned SSS. Classes in the dataset are divided into many meaningful clusters. The clustering results interpret the effectiveness of subspace structure in knowledge transfer. Then, in Sec. 3.4, we verify that our method can alleviate the space shift problem by both the ablation experiment and visualization of the learning process.

The main contributions of this paper include: 1) The space shift problem in zero-shot learning is first identified. 2) We propose a novel method based on the shared subspace structure to alleviate the space shift problem by implementing recurrent knowledge transfer (RecKT). 3) Many meaningful clusters in a dataset can be discovered using our method, which helps interpret ZSL performance.

2. Methodology

In zero-shot learning, images and labels of training seen classes are provided, i.e., $(\mathbf{X}^s, \mathbf{Y}^s) = \{(\mathbf{x}_1^s, y_1^s), ..., (\mathbf{x}_{N^s}^s, y_{N^s}^s)\}$. N^s denotes the number of all images of seen classes. For unseen classes, only the list of candidate labels $\mathbf{Y}^u = \{y_1^u, ..., y_{K^u}^u\}$ are known. K^s and K^u mean the numbers of seen and unseen classes respectively, while $K = K^s + K^u$ denotes the total number of all classes. Each seen or unseen image datum, $\mathbf{x}_i^s$ or $\mathbf{x}_i^u \in \Re^d$, is a d-dimensional feature vector in the image feature space. The label sets of the seen and unseen classes are disjoint, i.e. $\mathbf{Y}^s \cap \mathbf{Y}^u = \emptyset$. Auxiliary knowledge, e.g., attributes or/and word vectors, are provided for embedding all classes into the semantic (label) embedding space. $\mathbf{E}^s = [\mathbf{e}_1^s, ..., \mathbf{e}_{K^s}^s]$ and $\mathbf{E}^u = [\mathbf{e}_1^u, ..., \mathbf{e}_{K^u}^u]$ denote semantic embeddings of seen and unseen classes respectively. $\mathbf{e}_k^s$ and $\mathbf{e}_k^u \in \Re^p$ correspond to the labels y_k^s and y_k^u. In this way, seen and unseen classes are semantically connected, and knowledge transfer is enabled. Using the seen data pairs $(\mathbf{x}_i^s, y_i^s)$, ZSL aims to predict the label y_i^u for each testing unseen image $\mathbf{x}_i^u$ by leveraging the auxiliary knowledge $\mathbf{E}^s$ and $\mathbf{E}^u$ for knowledge transfer.

2.1. Class Prototype

We learn and transfer the class-level rather than instance-level structural knowledge between different embedding spaces. The reason is that most datasets provide only one attribute and/or one word vector per class and only the class-level structure can be learned. We use one datum, namely prototype [37, 13], to represent each class in the image feature space and semantic embedding space respectively. Similar to existing works [53, 43], we assume that data from each class form a tight cluster and are linearly separable from other classes in the image feature space, e.g., deep feature spaces. The class prototype in the image feature space (namely image prototype), which is denoted as $\mathbf{f}_k$, can be calculated by averaging all instances in the class. For seen classes, $\mathbf{f}_k^s = \frac{1}{N_k} \sum \mathbf{x}_i^s,\ s.t.\ y_i = k,\ k \in \{1, ..., K^s\}$. N_k denotes the instance number in the kth class. Note that only image prototypes of K^s seen classes can be calculated in the training phase, as we do not have any training images from unseen classes. The class prototype in the semantic embedding space (namely semantic prototype) is defined as class-level semantic embeddings $\mathbf{e}_k$ as provided in many datasets or calculated by averaging all instance-level semantic embeddings in the class. In this way, we have datum pair $(\mathbf{f}_k^s, \mathbf{e}_k^s)$ of each seen class. For each unseen class $(\mathbf{f}_k^u, \mathbf{e}_k^u)$, the image prototype $\mathbf{f}_k^u$ is missing, which is illustrated in Fig. 2. Then we learn and transfer knowledge based on these datum pairs.

2.2. Shared Subspace Structure

Joint Embedding Space. Although the structures among classes in the image feature space and semantic embedding space differ, we aim to learn and transfer shared structural knowledge which is compatible with both two spaces. Hence, we unite the two spaces and form the joint embedding space by concatenating corresponding image prototypes and semantic prototypes (vectors). This operation is illustrated in Fig. 2. Instead of direct concatenation of two vectors [24], we use weighted concatenation, i.e. $\begin{pmatrix} \mathbf{e}_k \\ \gamma \mathbf{f}_k \end{pmatrix}$, where $\gamma < 1$ is the weight. The structure learned on the joint embeddings will be transferable, because it is learned based on data in both two spaces. As the image prototypes of unseen classes are unknown, there exist missing values in these joint embeddings.

Shared Subspace Structure. We assume that the shared structure between the image feature and semantic embedding spaces is a subspace structure. Intuitively, when implements subspace clustering [11] on AwA dataset, “tiger”, “lion”, “bobcat” and “leopard” are divided into one subspace, while “spider monkey”, “gorilla” and “chimpanzee” are in another subspace. Such subspace structure (partition) should be compatible in both image feature and semantic embedding spaces. Therefore, we propose to learn the shared subspace structure (SSS) in the joint embedding space. By joint learning, instead of traditional learning and transferring steps, the shared knowledge between two spaces can be captured, while the private parts are discarded.

Specifically, we reconstruct every (both seen and unseen)

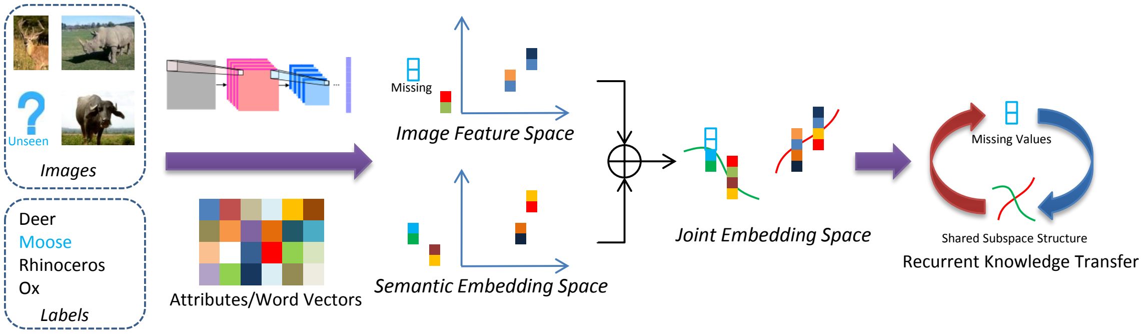


Figure 2: Overview of the proposed method. The images and labels are embedded into the image feature space and semantic embedding space respectively. Next, the image features and semantic embeddings are united in the joint embedding space with missing values (unseen image prototypes). Then, we alternatively learn the shared subspace structure (SSS) and missing values. During the learning process, structural knowledge is recurrently transferred between the two spaces.

class prototype by linearly regressing on others in the joint embedding space. The objective function is

$$L = \left\| \begin{pmatrix} \mathbf{E} \\ \gamma \mathbf{F} \end{pmatrix} - \begin{pmatrix} \mathbf{E} \\ \gamma \mathbf{F} \end{pmatrix} \mathbf{A} \right\|_F^2 + \lambda \Omega(\mathbf{A}), \quad s.t. \;\; \alpha_{k,k} \equiv 0. \tag{1}$$

In this equation, $\mathbf{F} = [\mathbf{F}^s, \mathbf{F}^u]$ contains both seen and unseen image prototypes, while $\mathbf{F}^u$ is missing (unknown). $\mathbf{F}^s = [\mathbf{f}_1^s, ..., \mathbf{f}_{K^s}^s]$ and $\mathbf{F}^u = [\mathbf{f}_1^u, ..., \mathbf{f}_{K^u}^u]$ are the sets of seen and unseen image prototypes respectively. The set $\mathbf{E} = [\mathbf{E}^s, \mathbf{E}^u]$ has the similar meaning, where $\mathbf{E}^s = [\mathbf{e}_1^s, ..., \mathbf{e}_{K^s}^s]$ and $\mathbf{E}^u = [\mathbf{e}_1^u, ..., \mathbf{e}_{K^u}^u]$. The vectors in set $\mathbf{E}$ and $\mathbf{F}$ are one-to-one corresponding. $\mathbf{A} = [\alpha_1, ..., \alpha_K]$ contains reconstruction coefficients (i.e. shared subspace structure) for all classes, where $K = K^s + K^u$. Here, $\alpha_k \in \Re^K$ is a column vector. The regularization term $\Omega(\mathbf{A})$ is designed to be sparse for discovering the subspace structure, which will be discussed later.

In Eq. 1, there are two unknown variables, namely, unseen image prototypes $\mathbf{F}^u$ and the shared subspace structure $\mathbf{A}$. We learn the two variables by minimizing the objective function,

$$\mathbf{F}^u, \mathbf{A} = \operatorname*{argmin}_{\mathbf{F}^u, \mathbf{A}} L. \tag{2}$$

Sparsity and Locality Regularization. Generally speaking, for each class, there are only a few classes that have strong correlation with it in a dataset [49, 7]. For example, in AwA dataset, "chimpanzee" has only two close relatives, namely, "spider monkey" and "gorilla", which are both visually and semantically close. Hence, we choose generalized Lasso regularization [41], i.e.

$$\Omega(\mathbf{A}) = \sum_{k=1}^{K} \|\mathbf{D}_k \alpha_k\|_1, \tag{3}$$

to regularize the learned SSS to be sparse. Each $\mathbf{D}_k$ is a diagonal matrix.

For small-scale datasets, we simply set each $\mathbf{D}_k$ to be an identity matrix. In this way, the regularization term $\Omega(\mathbf{A})$ is simplified to be $\sum_{k=1}^{K} \|\alpha_k\|_1$.

For large-scale datasets (e.g., ImageNet), many classes are too far away from each other. For instance, the animal "tiger" has little relationship with the industrial product "umbrella". However, the sparsity regularization cannot guarantee the locality constraint. Hence, we further introduce the locality regularization to discourage the learning of reconstruction coefficients between two distant classes. Specifically, the ith element ($\mathbf{D}_k^{i,i}$) on the diagonal of $\mathbf{D}_k$ is set to be the locality penalty based on the distance (or dissimilarity) between the kth and ith classes,

$$\mathbf{D}_k^{i,i} = \begin{cases} g(\mathbf{e}_i, \mathbf{e}_k) & if \;\; i \neq k, \\ 1, & else. \end{cases} \tag{4}$$

The function $g(\cdot, \cdot)$ is an increasing function of the distance, and $g(\cdot, \cdot) > 0$. $\mathbf{D}_k^{k,k}$ is set to be 1 for simplifying calculation. This regularization implies that the classes which are close to the target class are encouraged to be selected (with non-zero $\alpha_{i,i}$).

2.3. Recurrent Knowledge Transfer

In this section, we discuss how to solve the Eq. 2, in other words, how to learn unseen image prototypes $\mathbf{F}^u$ and the shared subspace structure $\mathbf{A}$. Although the objective function (Eq. 1) is not jointly convex for $\mathbf{F}^u$ and $\mathbf{A}$, it is convex for each variable respectively. Hence, we present an alternating optimization algorithm to solve it and realize recurrent knowledge transfer between the image feature and semantic embedding spaces.

First, we fix $\mathbf{F}^u$ and optimize Eq. 2 over $\mathbf{A}$ by

$$\min_{\mathbf{A}} L = \min_{\mathbf{A}} \sum_{k}^{K} \left\| \begin{matrix} \mathbf{e}_k - \mathbf{E}\alpha_k \\ \gamma(\mathbf{f}_k - \mathbf{F}\alpha_k) \end{matrix} \right\|_F^2 \tag{5}$$
$$+ \lambda \|\mathbf{D}_k \alpha_k\|_1, \quad s.t. \;\; \alpha_{k,k} \equiv 0.$$

With the notation $\beta_k = \mathbf{D}_k\alpha_k$, we can transform Eq. 5 into

$$\min_{\mathbf{A}} L = \min_{\mathbf{A}} \sum_k^K \left\| \begin{pmatrix} \mathbf{e}_k \\ \gamma \mathbf{f}_k \end{pmatrix} - \begin{pmatrix} \mathbf{E}(\mathbf{D}_k)^{-1} \\ \gamma \mathbf{F}(\mathbf{D}_k)^{-1} \end{pmatrix} \beta_k \right\|_F^2 + \lambda\|\beta_k\|_1, \quad s.t.\ \ \alpha_{k,k} \equiv 0. \tag{6}$$

Note that Eq. 6 is a typical LASSO problem, and β_k can be easily solved by many available solvers, e.g., LeastR [26]. Then α_k can be obtained by $\alpha_k = (\mathbf{D}_k)^{-1}\beta_k$. As the $\mathbf{F}^u$ is unknown at the beginning, we set $\gamma = 0$ to disable it during the initialization of $\mathbf{A}$. It means that we learn $\mathbf{A}$ only using semantic embeddings in the first step.

Then, we fix $\mathbf{A}$ and optimize Eq. 2 over $\mathbf{F}^u$ by

$$\begin{aligned} \min_{\mathbf{F}^u} L &= \min_{\mathbf{F}^u} \sum_k^K \gamma \|\mathbf{f}_k - \mathbf{F}\alpha_k\|_F^2 \\ &= \min_{\mathbf{F}^u} \gamma \|\mathbf{F}(\mathbf{I} - \mathbf{A})\|_F^2, \end{aligned} \tag{7}$$

where $\mathbf{I}$ is the identity matrix. With the notation $\theta = (\mathbf{I} - \mathbf{A})$, Eq. 7 is simplified as

$$\begin{aligned} \min_{\mathbf{F}^u} \gamma \|\mathbf{F}\theta\|_F^2 &= \min_{\mathbf{F}^u} \gamma \left\| (\mathbf{F}^s \ \ \mathbf{F}^u) \begin{pmatrix} \theta^s \\ \theta^u \end{pmatrix} \right\|_F^2 \\ &= \min_{\mathbf{F}^u} \gamma \|\mathbf{F}^s\theta^s + \mathbf{F}^u\theta^u\|_F^2. \end{aligned} \tag{8}$$

θ is split into seen part θ^s and unseen part θ^u which correspond to $\mathbf{F}^s$ and $\mathbf{F}^u$ respectively. Then, the solution $\mathbf{F}^u$ can be obtained by

$$\mathbf{F}^u = -\mathbf{F}^s\theta^s(\theta^u)^{-1}. \tag{9}$$

We iterate the above two steps until both of them converge. Finally, the unseen image prototypes $\mathbf{F}^u$ and the shared subspace structure $\mathbf{A}$ are both learned. This alternating optimization algorithm can be viewed as a kind of block coordinate descent algorithm with two blocks, hence the convergence of the proposed algorithm is guaranteed by [47]. The whole alternating optimization algorithm is summarized in Alg. 1.

Here, we explain how our method can alleviate the space shift problem in detail. In the first iteration of the learning process, structural knowledge ($\mathbf{A}$) is learned in the semantic embedding space then transferred to the image feature space to synthesize $\mathbf{F}^u$. This step is similar to existing structure-transfer methods [43, 54]. The learned $\mathbf{A}$, however, is biased towards the semantic embedding space. Hence, $\mathbf{F}^u$ which is synthesized based on such biased $\mathbf{A}$ deviates from the ground-truth data distribution. In the following iterations, $\mathbf{A}$ is first updated according to the current $\mathbf{F}^u$. In this way, the structural knowledge in the image feature space is transferred to the semantic embedding space. Then, $\mathbf{F}^u$ is updated based on the current $\mathbf{A}$. Therefore, the structural knowledge is transferred from the semantic embedding space to the image feature space. Finally, this recurrent knowledge transfer converges, meanwhile $\mathbf{A}$ and $\mathbf{F}^u$ are refined.

Algorithm 1 Recurrent Knowledge Transfer

```
Input: Seen image prototypes $\mathbf{F}^s$, all semantic prototypes
    $\mathbf{E}^s$ and $\mathbf{E}^u$;
Output: Synthesized unseen image prototypes $\mathbf{F}^u$ and re-
    construction coefficients $\mathbf{A}$;
Initialize: Set $\gamma$ and $\delta$;
Construct $\mathbf{D}_k$ for each class;
while not converge do
    Update $\mathbf{A}$ by solving Eq. 6;
    Update $\mathbf{F}^u$ using Eq. 9;
end while
```

2.4. Zero-shot Classification

To classify testing instances from unseen classes, we adopt the Nearest Neighbor classifier, which is applied in many existing works [33, 19]. Specifically, with synthesized unseen image prototypes $\mathbf{F}^u$, each test instance can be classified based on distance to $\mathbf{F}^u$, and its label is predicted to be the one with the minimum distance, i.e.

$$y_i^u = \operatorname*{argmin}_k \|\mathbf{x}_i^u - \mathbf{f}_k^u\|_F, \tag{10}$$

where $\mathbf{f}_k^u$ means each synthesized unseen image prototype.

3. Experiments

In this section, we implement experiments to verify the effectiveness of our method and the importance of the proposed space shift problem. The datasets and experimental settings are presented in Sec. 3.1. We compare our method with state-of-the-art ZSL methods in Sec. 3.2. Experiments in Sec. 3.3 justify that the shared subspace structure is reasonable. The subspace clustering result on the dataset is also illustrated and analyzed, which helps interpret ZSL performance. In Sec. 3.4, the existence of the space shift problem and how our method can relieve it are explained.

3.1. Datasets & Settings

Datasets. We evaluate our method on three popular datasets, namely Animals with Attributes (AwA) [21], Caltech-UCSD Birds-200-2011 (CUB) [42] and ImageNet [9]. AwA is a coarse-grained dataset which contains images of 50 kinds of common animals. 10 classes are selected as the unseen classes, and the rest are the seen classes. 85-dim attributes are provided. CUB is a fine-grained dataset that contains 200 kinds of birds. 50 classes are used as the

Method	AwA		CUB		ImageNet		
	Fea.	Acc.	Fea.	Acc.	Fea.	Top1	Top5
DAP	V	57.5		-			
SJE	G	66.7	G	50.1			
ZSKL	R	71.0	G	51.7			
LatEm	G	76.1	G	47.4			
SP-AEN		-	R	55.4			
PSR		-	R	56.0			
LEESD	G	76.6	G	56.2			
SS-Voc	O	78.3†		-	O	**9.5**†	16.8†
ConSE		-		-	O	7.8	15.5
DeViSE		-		-	O	5.2	12.8
JLSE	V	80.46	V	42.11			
SC	V	70.49	G+R	50.81			
ESZSL	V	79.53	G+R	51.90			
RKT	V	81.41	G+R	55.59			
Ours	V	**83.62**	G+R	**58.10**	V	8.14	**18.26**

Table 1: Comparison to the state-of-the-art (%). † means that extra vocabulary knowledge (nearly 310k word vectors) is utilized. *ESZSL*, *RKT* and *SC* are re-implemented using the same features and semantics. For image features, O: OverFeat, V: VGG, G: GoogLeNet, R: ResNet. Some results are vacant due to the lack of released code and parameters.

unseen classes. The rest 150 classes are the seen classes. 312-dim attributes are provided. We follow the seen/unseen splits of AwA and CUB used in [43]. ImageNet 2012/2010 is a large-scale dataset. No attributes are provided in this dataset. Following [14], we use 1,000 classes in ImageNet 2012 as seen classes. 360 classes in ImageNet 2010 which do not exist in ImageNet 2012 serve as unseen classes. We compare to state-of-the-art methods under the inductive setting, i.e., no images from unseen classes are available in training phase.

Image & Semantic Embedding. For coarse-grained datasets, AwA and ImageNet, we use image features extracted by VGG-19 [36]. Attributes and 500-dim word vectors are used as semantic embeddings. For fine-grained dataset (CUB), we use GoogLeNet [40] + ResNet [17] features. Attributes and 1024-dim word vectors are used as semantic embeddings. Similar as many existing works [13, 43, 52], we use official models (VGG, GoogLeNet, and ResNet) pre-trained on ImageNet to extract features.

Parameter Selection. There are only two free parameters, namely λ and γ, in the loss function. We select these parameters by Cross-Validation. Specifically, we split the seen classes into 5 folds for keeping the same seen/unseen ratio. One fold is used as new "unseen" classes, and the rest are "seen" classes. The parameters are selected based on the average performance on these folds. The searching range of λ and γ are $10^{[-2:2]}$ and $10^{[-2:0)}$ respectively. For large-scare dataset, we choose $g(\mathbf{e}_i, \mathbf{e}_k) = \log(1 + \|\mathbf{e}_i, \mathbf{e}_k\|_F)$ to calculate the penalty value.

3.2. Comparison to the State-of-the-Art

3.2.1 Small-scale Datasets

There are many works evaluated on AwA and CUB datasets. We compare to 11 state-of-the-art methods which are representative of different frameworks. They are DAP [22], SJE [1], SC [5], ZSKL [50], LatEm [44], SP-AEN [6], PSR [2], LEESD [10], SS-Voc [14], JLSE [52] and ESZSL [35] and RKT [43]. We re-implement experiments of *SC*, *ESZSL* and *RKT* using the same features and semantic embeddings. For the rest, we report their best performance in corresponding papers.

Tab.1 shows the comparison of classification accuracies (%) of different methods. It is clear that our method achieves the best performances (namely **83.62%** and **58.10%**) on both AwA and CUB. Our method outperforms the runner-up methods (RKT on AwA and LEESD on CUB) by 2.21% and 1.90% on the two datasets respectively. Because CUB has more classes while fewer images, the overall classification accuracy is much lower than AwA.

3.2.2 Large-scale Dataset

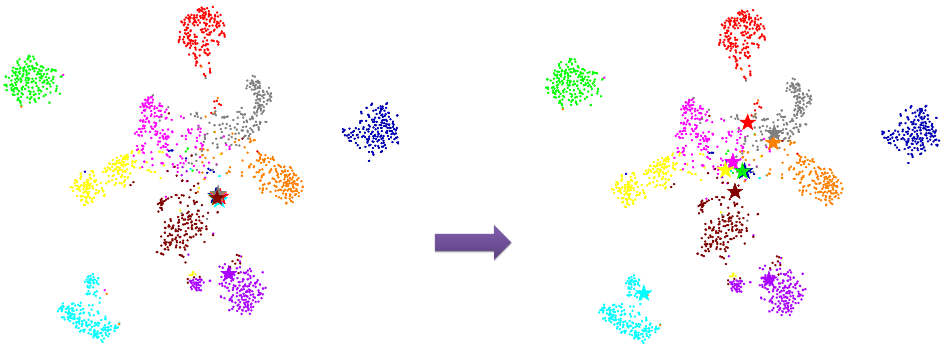
The large number of categories makes zero-shot learning more difficult on ImageNet dataset. Only a few ZSL methods are evaluated on ImageNet dataset. We compare to state-of-the-art methods, namely, SS-Voc [14], ConSE [31], DeViSE [12]. Notice that extra vocabulary knowledge is utilized in SS-Voc [14]. The result is shown in Tab. 1. Although extra vocabulary knowledge is utilized in SS-Voc, our method still outperforms it and achieves the best performance (**18.26%**) on ImageNet measured on Top-5 classification accuracy. When measured on Top-1 classification accuracy, our method obtains 8.14% classification accuracy, which is better than ConSE and DeViSE. The experimental results on ImageNet show that our method works well on large-scale datasets.

3.3. Shared Subspace Structure

3.3.1 Sparsity and Locality Regularization

First, we analyze the effectiveness of sparsity and locality regularization in the learning of shared subspace structure (Eq. 1). We calculate the classification accuracies on three datasets with different regularization terms, namely "sparsity + locality", "only sparsity" and "no regularization". For "only sparsity" experiment, each $\boldsymbol{D}_k$ is defined as an identity matrix. For "no regularization" experiment, we set λ to be zero, so that $\boldsymbol{D}_k$ is disabled. We apply the locality regularization only on the large-scale dataset, i.e., ImageNet.

As shown in Tab. 2, there is steady performance improvement on all three datasets when the sparsity regularization is introduced. When only a single type of semantic embeddings is used, namely no fusion of attributes and



Before adaptation Acc. = 63.05%

After adaptation Acc. = 72.95%

Figure 3: Illustration of how our method alleviates the space shift problem. We visualize the distribution of the synthesized unseen image prototypes before/after adaptation to the image feature space. Acc. means the classification accuracy on the testing classes. Points with the same color are data belonging to the same class. Stars represent the synthesized image prototypes of corresponding classes (the same color). The left part (before adaptation) can be considered as the traditional method (direct learning and transferring). After the adaptation (the right part), synthesized image prototypes are more separable and close to the real data distribution. Remarkable improvement can be found in both synthesized image prototypes and classification accuracy (9.90%).

	ImageNet	
Locality	W - Top 1	W - Top 5
With	**8.14**	**18.26**
Without	7.72	17.79

Table 3: Analysis of the locality regularization. W : word vector. We find that the locality regularization can improve the classification accuracy (%) on the large-scale dataset (e.g., ImageNet).

word vectors, the improvement brought by the sparsity regularization is remarkable. On AwA, the improvement is 15.68% and 8.52% using attributes and word vectors respectively. On CUB, the improvement is 7.77% using attributes as the semantic embeddings. The Top-1 and Top-5 classification accuracies on ImageNet both nearly double due to the introduction of sparsity regularization. Hence, the sparsity regularization is important for both small-scale and large-scale datasets.

When the locality regularization is introduced to the large-scale dataset, ImageNet, the Top-1 and Top-5 classification accuracies rise from 7.72% to 8.14% and 17.79% to 18.26% respectively (shown in Tab. 3). This improvement proves the effectiveness of the locality regularization for large-scale datasets.

3.3.2 Discovering Latent Clusters

After learning the coefficients matrix $\mathbf{A}$, we do spectral clustering on the similarity graph $\tilde{\mathbf{G}} = (\mathbf{V}, \tilde{\mathbf{A}})$, where $\mathbf{V}$ refers to the set of class prototypes and $\tilde{\mathbf{A}}$ is defined as $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{A}^T$. When each $\mathbf{D}_k$ in Eq. 3 is an identity matrix, the loss function is similar to that in in Sparse Subspace Clustering (SSC) [11, 48] but with missing values. In SSC, it assumes that each datum is drawn from a linear subspace $\mathcal{S}$ with a basis $\mathcal{U}$, which is learned via singular value decomposition. Different from SSC, we don't enforce affine combination constraint in our method. Instead, we consider the linear subspace here. By implementing the spectral clustering on $\tilde{\mathbf{G}}$, we divide all classes into many clusters.

We illustrate the clustering result on AwA dataset. The reason we choose AwA is that there is a moderate number of classes in AwA, while there are too many classes in CUB and ImageNet for an intuitive illustration. In this experiment, attributes serve as the semantic embeddings. We can set different cluster numbers in K-means algorithm and discover different kinds of clustering results. Tab. 4 shows the clustering result when the cluster number is set to 11. It is clear that these clusters are meaningful. Classes in each cluster are close to each other, while different clusters are separable from each other. For example, classes in Cluster_1 are all aquatic animals. Cluster_2 are the whales (dolphin belongs to whales). Cluster_4 contains four fierce Felidae species. Cluster_11 includes three kinds of primates. Clearly, these clusters are separable subsets of animals.

This clustering result also verifies the good ZSL performance on AwA, because all unseen classes (bold ones in Tab. 4) have close seen classes in the same cluster, which enables effective knowledge transfer. In addition, these unseen classes are equably in separable clusters. This is also an important reason why high ZSL accuracy can be achieved on AwA dataset.

3.4. Space Shift Problem

3.4.1 Ablation Experiment

We verify the space shift problem by implementing ablation experiments. Specifically, we compare the performance of our method with/without the adaptation to the image feature

	AwA			CUB			ImageNet	
Sparsity	A	W	AW	A	W	AW	W - Top 1	W - Top 5
With	**80.61**	**72.95**	**83.62**	**54.32**	**47.88**	**58.10**	**7.72**	**17.79**
Without	64.93	64.43	81.22	46.55	47.33	56.76	3.38	9.23

Table 2: Analysis of the sparsity regularization. A/W means attribute/word vector. Clearly, the sparsity regularization is important for improving ZSL performance (%).

No.	Classes in Each Cluster						
1	beaver	walrus	otter	**seal**			
2	killer whale	blue whale	dolphin	**humpback whale**			
3	antelope	moose	deer	giraffe	zebra	horse	
4	lion	bobcat	tiger	**leopard**			
5	mouse	hamster	squirrel	mole	rabbit	sheep	
6	elephant	rhinoceros	**hippopotamus**				
7	buffalo	cow	ox	**pig**			
8	skunk	**raccoon**	**rat**				
9	collie	dalmatian	German shepherd	chihuahua	Siamese cat	**Persian cat**	**giant panda**
10	fox	weasel	wolf	grizzly bear	polar bear	bat	
11	spider monkey	gorilla	**chimpanzee**				

Table 4: Clustering result on AwA. Bold ones are unseen classes. Obviously, these clusters are meaningful and separable from each other.

	AwA		CUB	
Embedding	A	W	A	W
with Adapt.	**80.61**	**72.95**	**54.32**	**47.88**
without Adapt.	79.70	63.05	52.87	34.12

Table 5: The ablation experiment for verifying the space shift problem. Adapt. means adaptation to the image feature space.

space. To disable the adaptation, we simply set the parameter $\gamma = 0$. Only one iteration will be implemented and the loss will converge after the first iteration. In other words, $\mathbf{A}$ is learned only based on semantic embeddings then transferred to the image feature space directly for synthesizing unseen image prototypes. Like traditional structure-transfer methods, it suffers from the space shift problem. The performance comparison is shown in Tab. 5. We only compare the performance with the attributes (A) and word vectors (W) as semantic embeddings respectively. The fusion (A+W) of the two kinds of semantic embeddings [43] may make it difficult to tell whether the performance promotion is caused by the adaptation or the fusion.

From Tab. 5, we can find that the adaptation brings improvement of classification accuracy on both two datasets and two kinds of semantic embeddings. Specifically, with attributes as the semantic embeddings, the adaptation to the image feature space brings 0.91% and 1.45% improvement of classification accuracies on AwA and CUB respectively. When word vectors are used as the semantic embeddings, remarkable improvement is shown on both AwA and CUB, namely, 9.90% and 13.76%. The reason is that there exists larger divergence between the word vector space and image feature space. It is also verified in many existing works [51, 44, 18] that attributes work better than word vectors.

3.4.2 Visualization of the Learning Process

We visualize the how our method alleviates space shift problem in the Fig. 3. The figure is drawn by implementing t-SNE [28] on testing unseen image features for dimensionality reduction. The experiment is performed on AwA with word vectors as semantic embeddings. Before adaptation (in the 1st iteration), the learned structure is directly transferred to the image feature space. Hence, it suffers from the space shift problem. As shown in Fig. 3, the synthesized image prototypes are not well separated before adaptation. The classification accuracy is only 63.05%. After the adaptation (by recurrent knowledge transfer), synthesized image prototypes are more separable and close to the real data distribution, compared to the 1st iteration. In addition, the classification accuracy increases to 72.95%. Overall, remarkable improvement can be found in both synthesized image prototypes and classification accuracy (9.90%) when the adaptation to the image feature space is realized by recurrent knowledge transfer. This experiment verifies that the space shift problem exists and significantly influences ZSL performance. Our recurrent knowledge transfer can relieve the space shift problem and improve classification accuracy.

Acknowledgement

This work was supported in part by NSFC-61527804, NSFC-61625201, NSFC-61650202 and NSFC-61572205. This work was also supported by the Academy of Finland and Infotech Oulu.

References

[1] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015.

[2] Y. Annadani and S. Biswas. Preserving semantic relations for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7603–7612, 2018.

[3] I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

[4] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351, 2016.

[5] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, pages 5327–5336, 2016.

[6] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2018.

[7] J. Choi, S. J. Hwang, L. Sigal, and L. S. Davis. Knowledge transfer with interactive learning of semantic relationships. In *AAAI*, pages 1505–1511, 2016.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[10] Z. Ding, M. Shao, and Y. Fu. Low-rank embedded ensemble semantic dictionary for zero-shot learning. *CVPR*, 2017.

[11] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, pages 2790–2797, 2009.

[12] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[13] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Transductive multi-view zero-shot learning. *TPAMI*, 37(11):2332–2345, 2015.

[14] Y. Fu and L. Sigal. Semi-supervised vocabulary-informed learning. In *CVPR*, pages 5337–5346, 2016.

[15] Y. Guo, G. Ding, J. Han, and Y. Gao. Synthesizing samples fro zero-shot learning. IJCAI, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2016.

[18] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, pages 2452–2460, 2015.

[19] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. *CVPR*, 2017.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[21] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958, 2009.

[22] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 36(3):453–465, 2014.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. 86(11):2278–2324, 1998.

[24] X. Li, Y. Guo, and D. Schuurmans. Semi-supervised zero-shot classification with label representation learning. In *ICCV*, pages 4211–4219, 2015.

[25] Y. Li, J. Zhang, J. Zhang, and K. Huang. Discriminative learning of latent features for zero-shot recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7463–7471, 2018.

[26] J. Liu and J. Ye. Efficient euclidean projections in linear time. In *26th Annual ICML*. ACM, 2009.

[27] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[28] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 2008.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[30] S. Naha and Y. Wang. Zero-shot object recognition using semantic label vectors. In *CRV*, 2015.

[31] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *NIPS*, 2013.

[32] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[33] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *NIPS*, pages 1410–1418, 2009.

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[35] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015.

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[38] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, pages 935–943, 2013.

[39] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[41] R. J. Tibshirani. *The solution path of the generalized lasso*. Stanford University, 2011.

[42] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

[43] D. Wang, Y. Li, Y. Lin, and Y. Zhuang. Relational knowledge transfer for zero-shot learning. In *Thirtieth AAAI*, 2016.

[44] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *CVPR*, pages 69–77, 2016.

[45] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[46] Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning-the good, the bad and the ugly. *CVPR*, 2017.

[47] Y. Xu and W. Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on imaging sciences*, 6(3):1758–1789, 2013.

[48] C. Yang, D. P. Robinson, and R. Vidal. Sparse subspace clustering with missing entries. In *ICLM*, pages 2643–2672, 2015.

[49] J. Yoon and S. J. Hwang. Combined group and exclusive sparsity for deep neural networks. In *ICML*, 2017.

[50] H. Zhang and P. Koniusz. Zero-shot kernel learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7670–7679, 2018.

[51] L. Zhang, T. Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. *CVPR*, 2017.

[52] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In *CVPR*, 2016.

[53] Z. Zhang and V. Saligrama. Zero-shot recognition via structured prediction. In *ECCV*, 2016.

[54] B. Zhao, B. Wu, T. Wu, and Y. Wang. Zero-shot learning posed as a missing data problem. In *Proceedings of ICCV Workshops*, pages 2616–2622, 2017.