CrossMark

Zero-Shot Visual Recognition via Bidirectional Latent Embedding

Qian Wang[1] · Ke Chen[1]

Abstract Zero-shot learning for visual recognition, e.g., object and action recognition, has recently attracted a lot of attention. However, it still remains challenging in bridging the semantic gap between visual features and their underlying semantics and transferring knowledge to semantic categories unseen during learning. Unlike most of the existing zero-shot visual recognition methods, we propose a stagewise bidirectional latent embedding framework of two subsequent learning stages for zero-shot visual recognition. In the bottom–up stage, a latent embedding space is first created by exploring the topological and labeling information underlying training data of known classes via a proper supervised subspace learning algorithm and the latent embedding of training data are used to form landmarks that guide embedding semantics underlying unseen classes into this learned latent space. In the top–down stage, semantic representations of unseen-class labels in a given label vocabulary are then embedded to the same latent space to preserve the semantic relatedness between all different classes via our proposed semi-supervised Sammon mapping with the guidance of landmarks. Thus, the resultant latent embedding space allows for predicting the label of a test instance with a simple nearest-neighbor rule. To evaluate the

✉ Ke Chen
   Ke.Chen@manchester.ac.uk

   Qian Wang
   Qian.Wang@manchester.ac.uk

1 School of Computer Science, The University of Manchester,
   Manchester M13 9PL, UK

effectiveness of the proposed framework, we have conducted extensive experiments on four benchmark datasets in object and action recognition, i.e., AwA, CUB-200-2011, UCF101 and HMDB51. The experimental results under comparative studies demonstrate that our proposed approach yields the state-of-the-art performance under inductive and transductive settings.

1 Introduction

Visual recognition refers to various tasks for understanding the content of images or video clips. *Object recognition* and *human action recognition* are two typical visual recognition tasks studied extensively in computer vision community. In the last decade, substantial progresses have been made in object and human action recognition (Andreopoulos and Tsotsos 2013). As a result, we witness a boost of various benchmarks released with more and more classes, which poses greater challenges to computer vision. For example, the number of classes in object recognition benchmarks has increased from 256 in Caltech-256 (Griffin et al. 2007) to 1000 in ImageNet ILSVRC (Russakovsky et al. 2015), while the number of classes in human action recognition has increased from 51 in HMDB51 (Kuehne et al. 2011) to 101 in UCF101 (Soomro et al. 2012). Despite the increasing number of classes in consideration, they are still a small portion of all classes existing in real world. According to Lampert et al. (2014), humans can distinguish approximately 30,000 basic object classes, and much more subordinate ones. Nowadays,

new objects emerge rapidly. Practically, it is impossible to collect and annotate visual data for all the classes to establish a visual recognition system. This leads to a great challenge for visual recognition.

To fight off this challenge, *zero-shot learning* (ZSL) was recently proposed and applied in both object and human action recognition with promising performances, e.g., Akata et al. (2013, 2014, 2015, 2016), Al-Halah and Stiefelhagen (2015), Changpinyo et al. (2016a, b), Fu et al. (2015), Gan et al. (2016), Kodirov et al. (2015), Lampert et al. (2014), Mensink et al. (2014), Norouzi et al. (2014), Romera-Paredes and Torr (2015), Xian et al. (2016), Xu et al. (2015b), Zhang and Saligrama (2015, 2016a, b). Unlike the traditional methods that can only recognize classes appearing in the training data, ZSL is inspired by the learning mechanism of human brain and aims to recognize new classes unseen during learning by exploiting intrinsic semantic relatedness between known and unseen classes. In general, three fundamental elements are required in ZSL; i.e., *visual representation* conveying non-trivial yet informative visual features, *semantic representation* reflecting the relatedness between different classes (especially between known and unseen classes), and *learning model* properly relating visual features to underlying semantics.

Visual representations play an important role in visual recognition. In particular, the visual representations learned with deep Convolutional Neural Networks (CNNs) have improved the performances of object recognition, e.g., Chatfield et al. (2014), He et al. (2016), Simonyan and Zisserman (2015), Szegedy et al. (2015), and human action recognition, e.g., Simonyan and Zisserman (2014), Wang et al. (2016), Wu et al. (2016), Zhao et al. (2015). Benefitting from deep learning, zero-shot visual recognition performances have also been boosted, e.g., Akata et al. (2014), Al-Halah and Stiefelhagen (2015), Reed et al. (2016). In addition, it has been reported that the joint use of multiple visual representations can improve the performances and the robustness of visual recognition, e.g., Fu et al. (2015), Shao et al. (2016).

Semantic representations aim to model the semantic relatedness between different classes. A variety of semantics modelling techniques (Elhoseiny et al. 2015; Frome et al. 2013; Jiang et al. 2014; Lampert et al. 2014; Liu et al. 2011; Mensink et al. 2014; Mikolov et al. 2013) have been developed, e.g., semantic attributes (Jiang et al. 2014; Lampert et al. 2014; Liu et al. 2011) and word vectors (Frome et al. 2013; Mikolov et al. 2013). Semantic attributes are usually manually defined for semantic labels that describe objects and actions contained in images and video streams, while word vectors are automatically learned from unstructured textual data in an unsupervised way.

Given the low-level visual representations of images or video streams and their underlying high-level semantics, the central problem in zero-shot visual recognition is how to transfer knowledge from the visual data of known classes to those of unseen classes. A variety of zero-shot visual recognition methods have been proposed, e.g., Akata et al. (2013, 2014, 2015, 2016), Al-Halah and Stiefelhagen (2015), Changpinyo et al. (2016a, b), Fu et al. (2015), Gan et al. (2016), Kodirov et al. (2015), Lampert et al. (2014), Mensink et al. (2014), Norouzi et al. (2014), Romera-Paredes and Torr (2015), Xian et al. (2016), Xu et al. (2015b), Zhang and Saligrama (2015, 2016a, b). A brief review on zero-shot visual recognition will be described in the next section.

In zero-shot visual recognition, the *semantic gap* is the biggest hurdle; i.e., the distribution of instances in visual space is often distinct from that of their underlying semantics in semantic space as visual features in various forms may convey the same concept. This semantic gap results in a great difficulty in transferring knowledge on known classes to unseen classes. Apart from the semantic gap issue, the *hubness* (Radovanović et al. 2010) is recently identified as a cause that accounts for the poor performance of most existing ZSL models (Dinu et al. 2015; Shigeto et al. 2015; Xu et al. 2015b). "Hubness" refers to the phenomenon that some instances (referred to as *hubs*) in the high-dimensional space appear to be the nearest neighbors of a large number of instances. When nearest-neighbour based algorithms are applied, test instances are likely to be close to those "hubs" regardless of their labels and hence incorrectly labeled as labels of "hubs". In ZSL, the "hubness" phenomenon becomes more severe. Apart from the intrinsic property of high-dimensional space (Radovanović et al. 2010), the hubness is exacerbated by a lack of training instances belonging to unseen classes in visual domain and the *domain shift* problem, where the distribution of training data is different from that of test data, which often occurs in ZSL (Fu et al. 2015; Zhang and Saligrama 2016b).

In this paper, we propose a novel zero-shot visual recognition framework towards bridging the semantic gap and tackling the hubness issue. Unlike most of existing methods, our framework consists of two subsequent stages: bottom–up and top–down stages. In the bottom–up stage, a latent space is learned from a visual representation via supervised subspace learning that preserves intrinsic structures of visual data and promotes the discriminative capability. We expect that the latent space resulting from such subspace learning captures the intrinsic structures underlying visual data and narrows the semantic gap between visual and semantic spaces. After the bottom–up learning, in the latent space, the mean of projected points of training data in the same class forms a *landmark* specified as the embedding point of the corresponding class label. In the top–down stage, the semantic representations of all unseen-class labels in a given vocabulary are then embedded in the same latent space (created in the bottom–up stage) by retaining the semantic relatedness of all different classes in the latent space via the guidance of

the landmarks. By exploring the intrinsic structure of visual data in the bottom–up projection and preserving the semantic relatedness in the top–down projection, we demonstrate that the latent representation works effectively towards bridging the semantic gap and alleviating the adversarial effect of the hubness phenomenon (Shigeto et al. 2015). In addition, the existing transductive post-processing techniques, e.g., Fu et al. (2015), Zhang and Saligrama (2016b), are easily incorporated into our proposed framework to address the domain shift issue. Whenever multiple diversified visual and/or semantic representations are available, our proposed framework can further exploit the synergy among multiple representations seamlessly.

Our main contributions in this paper are summarized as follows: (a) we propose a novel stagewise bidirectional latent embedding framework for zero-shot visual recognition and explore effective and efficient enabling techniques to address the semantic gap issue and to lessen the catastrophic effect of the hubness phenomenon; (b) we extend our framework to scenarios in presence of multiple visual and/or different semantic representations as well as the transductive setting; and (c) we conduct extensive experiments under a comparative study to demonstrate the effectiveness of our proposed framework on several benchmark datasets.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 presents our bidirectional latent embedding framework. Section 4 describes our experimental settings, and Sect. 5 reports experimental results. The last section draws conclusions.

2 Related Work

In this section, we review existing works in zero-shot visual recognition and particularly outline connections and differences between our proposed framework and the related methods. We first provide a taxonomy on zero-shot visual recognition to facilitate our presentation and then briefly review relevant subspace learning methods that could be enabling techniques used to realize our proposed framework.

2.1 Zero-Shot Visual Recognition

There are a number of taxonomies for zero-shot visual recognition. For example, Akata et al. (2016) proposed a taxonomy that highlights two crucial choices in ZSL, i.e., the prior information and the recognition model, while the taxonomy provided by Changpinyo et al. (2016a) is from a perspective of knowledge transfer. To facilitate our presentation in this paper, we would divide the existing zero-shot visual recognition methods into three categories from a perspective on how the existing methods bridge the semantic gap,

namely, *direct mapping, model parameter transfer* and *common space learning*.

Direct mapping is a typical ZSL methodology. Its ultimate goal is learning a mapping function from visual features to semantic representations directly or indirectly (Akata et al. 2014, 2016, 2015; Al-Halah and Stiefelhagen 2015; Gan et al. 2016; Jayaraman and Grauman 2014; Kodirov et al. 2015; Lampert et al. 2009, 2014; Romera-Paredes and Torr 2015; Shigeto et al. 2015; Xian et al. 2016; Xu et al. 2015a, b). Such a mapping is carried out via either a classifier or a regression model depending upon an adopted semantic representation. As the relatedness between any class labels are known in semantic space or its own embedding space, a proper label may be assigned to a test instance in an unseen class by means of semantic relatedness in different manners, e.g., nearest neighbors (Xu et al. 2015a) and probabilistic models (Lampert et al. 2009). However, direct mapping may not be reliable in attribute predictions (Gan et al. 2016; Jayaraman and Grauman 2014). This issue has been addressed by different strategies. Jayaraman and Grauman (2014) use the random forests based post-processing to handle the uncertainties of attribute predictions, while Gan et al. (2016) propose to learn a representation transformation in visual space to enhance the attribute-level discriminative capacity for attribute prediction. Alternatively, Al-Halah and Stiefelhagen (2015) explore the additional underlying attributes by constructing the hierarchy of concepts for reliability. When the semantic representations are continuous, regression models are used to map visual features to semantic representations. A variety of loss functions along with various regularization terms have been employed to establish regression models. For example, Akata et al. (2014, 2015, 2016) and Xian et al. (2016) use structured SVM to maximize the compatibility between estimated and ground-truth semantic representations. Kodirov et al. (2015) formulate the regression as a dictionary learning and sparse coding problem. Romera-Paredes and Torr (2015) make a distinction by minimising the multi-class error rather than the error of the semantic representation prediction and adding further constraints on the model parameters. In direct mapping, however, the generalization of learned mapping models is considerably limited by high intra-class variability. Furthermore, it does not address the domain shift problem well when the training and test data are of different distributions. According to Shigeto et al. (2015), a regression model tends to project the instances closer to the origin than its ground-truth semantic representation, which exacerbates the domain shift problem.

Model parameter transfer is yet another ZSL methodology that estimates model parameters with respect to unseen classes by combining those model parameters learned from known classes via exploiting the inter-class relationship between known and unseen classes in semantic space (Changpinyo et al. 2016a; Gan et al. 2015; Mensink et al.

2014; Norouzi et al. 2014). Unlike direct mapping, the zero-shot visual recognition in model parameter transfer takes place in visual space where the model parameters for unseen classes are usually obtained by a convex combination of base classifiers trained on known classes (Gan et al. 2015; Mensink et al. 2014; Norouzi et al. 2014). More recently, Changpinyo et al. (2016a) proposed a novel approach that gains model parameters for unseen classes by aligning the topology of all the classes in both semantic and model parameter spaces. As a result, model parameter transfer is carried out by exploring base classifiers corresponding to "phantom" classes, which are artificially created and not associated with any real classes, to enhance the flexibility of the model. Since the inter-class relationship among unseen classes is not taken into account, model parameter transfer might be subject to limitation due to a lack of sufficient information for knowledge transfer.

Common space learning is a generic methodology towards bridging the semantic gap and has been applied in ZSL (Changpinyo et al. 2016b; Fu et al. 2015; Zhang and Saligrama 2015, 2016a) as well as other computer vision applications such as image retrieval (Gong et al. 2014) and automatic image description generation (Karpathy and Fei-Fei 2015). This methodology learns a common representation space into which both visual features and semantic representations are projected for effective knowledge transfer. Consequently, zero-shot visual recognition is obtained in this learned common representation space, which is different from direct mapping, where the recognition is obtained in semantic space or its own embedding space that differs from visual embedding space in some direct mapping methods (Akata et al. 2015, 2016), and model parameter transfer, where the recognition takes place in visual space. A learned common space may be either interpretable (Zhang and Saligrama 2015) or latent (Changpinyo et al. 2016b; Fu et al. 2015; Zhang and Saligrama 2016a). Zhang and Saligrama (2015) come up with a semantic similarity embedding method, which leads to semantic space where similarity can be readily measured for zero-shot visual recognition. This method works on viewing any instance in unseen classes as a mixture of those in known classes. More recently, Zhang and Saligrama (2016a) further propose a probabilistic framework for learning joint similarity latent embedding where both visual and semantic embedding along with a class-independent similarity measure are learned simultaneously. As a result, zero-shot visual recognition is obtained via optimization in the joint similarity latent space. Fu et al. (2015) use the *canonical correlation analysis* (CCA) to project multiple views of visual data onto a common latent embedding space to address the domain shift issue. When we prepared this manuscript, one latest zero-shot recognition method (Changpinyo et al. 2016b) emerged, which involves two subsequent learning stages. Nevertheless, the generalization

capability of the aforementioned common space learning models is generally limited as the intra-class variability is not tackled effectively.

Our proposed framework can be viewed as a common space learning approach as zero-shot recognition is obtained in the learned common representation space (c.f. Sect. 3). While all common space learning methods share the same ultimate goal to bridge the semantic gap, their strategies and enabling techniques for attaining this goal may be quite different. To this end, our proposed framework consists of two subsequent learning stages, while most of other common space learning methods fulfil the joint embedding from both visual and semantic spaces simultaneously, e.g., Fu et al. (2015), Zhang and Saligrama (2015, 2016a). Furthermore, our framework tackles the intra-class and inter-class variability in the common space and knowledge transfer explicitly with proper enabling techniques, while other common space learning methods address such issues implicitly, e.g., Zhang and Saligrama (2015, 2016a) or do not take into account intra-class and inter-class variability in the latent space, e.g., Changpinyo et al. (2016b). In terms of enabling techniques, other common space learning methods (Changpinyo et al. 2016b; Fu et al. 2015; Zhang and Saligrama 2015, 2016a) employ different parametric learning models for common space learning with their formulated objectives, while we address this issue by using both parametric (bottom–up) and non-parametric (top–down) learning models. The use of non-parametric model in our proposed framework allows for carrying out knowledge transfer explicitly, which readily distinguishes ours from all the existing common space learning methods that realize knowledge transfer implicitly with a parametric model that relies on the capacity in interpolation and extrapolation for generalization.

2.2 Subspace Learning

Subspace learning aims to find a low-dimensional space for high-dimensional raw data to reside in by preserving and highlighting useful information retained in the data in the high-dimensional space. In ZSL tasks, both the visual and semantic representation spaces could be of a very high dimensionality. To deal with the "curse of dimensionality", subspace learning is often employed to address this issue in ZSL (Akata et al. 2016). In particular, it is essential for common space learning (Fu et al. 2015; Fu and Huang 2010; Zhang and Saligrama 2015, 2016a). In general, subspace learning models are either parametric or non-parametric.

A parametric model learns a projection from a *source* high-dimensional space to a *target* low-dimensional subspace via optimizing certain objectives of interest. For example, *principle component analysis* (PCA) (Jolliffe 2002) learns a projection that maps data points to a set of uncorrelated components accounting for as much of the variability

underlying a data set as possible. *Locality preserving projection* (LPP) (Niyogi 2004) learns a projection for preserving the local neighborhoods in the source space. In a supervised learning scenario, a discriminative subspace can be learned by using label information. For example, *linear discriminant analysis* (LDA) (Cai et al. 2007) leads to a projection that maximizes the separability of projected data points in the LDA subspace. LPP has also been extended to its supervised version by taking the label information into account (Cheng et al. 2005). In our work, we apply the supervised LPP algorithm as an enabling technique for learning a low-dimensional latent space from visual space.

Unlike the aforementioned parametric models, a non-parametric subspace model often learns projecting a set of high-dimensional data points onto a low-dimensional subspace directly to preserve the intrinsic properties in source space. Non-parametric models are suitable especially for a scenario that all the data points in the source space are known or available and the embedding task needs to be undertaken on a given data set without the need of extension to unseen data points during learning. This is a salient characteristic that distinguishes between parametric and non-parametric subspace learning. As a typical non-parametric subspace learning framework, *multi-dimensional scaling* (MDS) (Cox and Cox 2000) refers to a family of algorithms that learn embedding a set of given high-dimensional data points into a low-dimensional subspace by preserving the distance information between data points in the high-dimensional space. Sammon mapping (Sammon 1969) is an effective non-linear MDS algorithm. In our work, we extend the Sammon mapping to a semi-supervised scenario that for a given dataset the embedding of some data points in the subspace is known or fixed in advance and only remaining data points need to be embedded via preserving their distance information to others. To the best of our knowledge, this is a brand new problem that has never been considered in literature but emerges from our proposed framework for knowledge transfer between known and unseen classes.

3 Bidirectional Latent Embedding

In this section, we propose a novel framework for zero-shot visual recognition via *bidirectional latent embedding learning* (BiDiLEL). We first provide an overview on our basic ideas and the problem formulation. Then, we present the bottom–up and the top–down embedding learning with proper enabling techniques, respectively. Finally, we describe the learning model deployment for zero-shot recognition as well as two post-processing techniques for the transductive setting. To facilitate our presentation, Table 1 summarizes the notations used in this paper.

Table 1 Nomenclature

Notation	Description
n_l, n_u	Number of labelled (training) and unlabelled (test) instances
d_x, d_y, d_s	Dimensionality of visual, latent and semantic spaces
$X^l \in \mathbb{R}^{d_x \times n_l}, \boldsymbol{x}_i^l$	Visual representation matrix of all the labelled instances, a column corresponding to an instance
$X^u \in \mathbb{R}^{d_x \times n_u}, \boldsymbol{x}_i^u$	Visual representation matrix of unlabelled instances, a column corresponding to an instance
$Y^l \in \mathbb{R}^{d_y \times n_l}, \boldsymbol{y}_i^l$	Projections of X^l in the latent subspace $\mathcal{Y}$, a column corresponding to an instance
$Y^u \in \mathbb{R}^{d_y \times n_u}, \boldsymbol{y}_i^u$	Projections of X^u in the latent subspace $\mathcal{Y}$, a column corresponding to an instance
$W \in \mathbb{R}^{n_l \times n_l}, L \in \mathbb{R}^{n_l \times n_l}$	Similarity and Laplacian matrices of a given data set of n_l instances
$P \in \mathbb{R}^{d_x \times d_y}$	Projection matrix learned in the bottom–up stage
$\mathcal{C}^l, \mathcal{C}^u, |\mathcal{C}^l|, |\mathcal{C}^u|$	Known and unseen class label sets and the number of known and unseen classes in two sets
$B^l \in \mathbb{R}^{d_y \times |\mathcal{C}^l|}, \boldsymbol{b}_i^l$	Latent embedding for known class labels, a column corresponding to one class
$B^u \in \mathbb{R}^{d_y \times |\mathcal{C}^u|}, \boldsymbol{b}_i^u$	Latent embedding for unseen class labels learned in the top–down stage, a column corresponding to one class

3.1 Overview

The motivation behind our proposed framework is twofold: (a) to narrow the semantic gap, a latent space is learned from visual representations of training data in a supervised manner by preserving intrinsic structures underlying visual data and promoting the discriminative capability simultaneously and (b) for knowledge transfer, the semantic representations of unseen-class labels are then embedded into the learned latent space of favorable properties by taking into account both the embedding of training-class labels and the semantic relatedness between all different classes; i.e., not only the relationships between known and unseen classes but also that between unseen classes. Based on our motivation described above, we propose a framework of a sequential bidirectional learning strategy: the bottom–up learning for creating the latent space from visual data and then the top–down learning for embedding all the unseen-class labels in the learned latent space, as illustrated in Fig. 1.

In the bottom–up stage, the visual representations of training examples are extracted. A proper supervised subspace
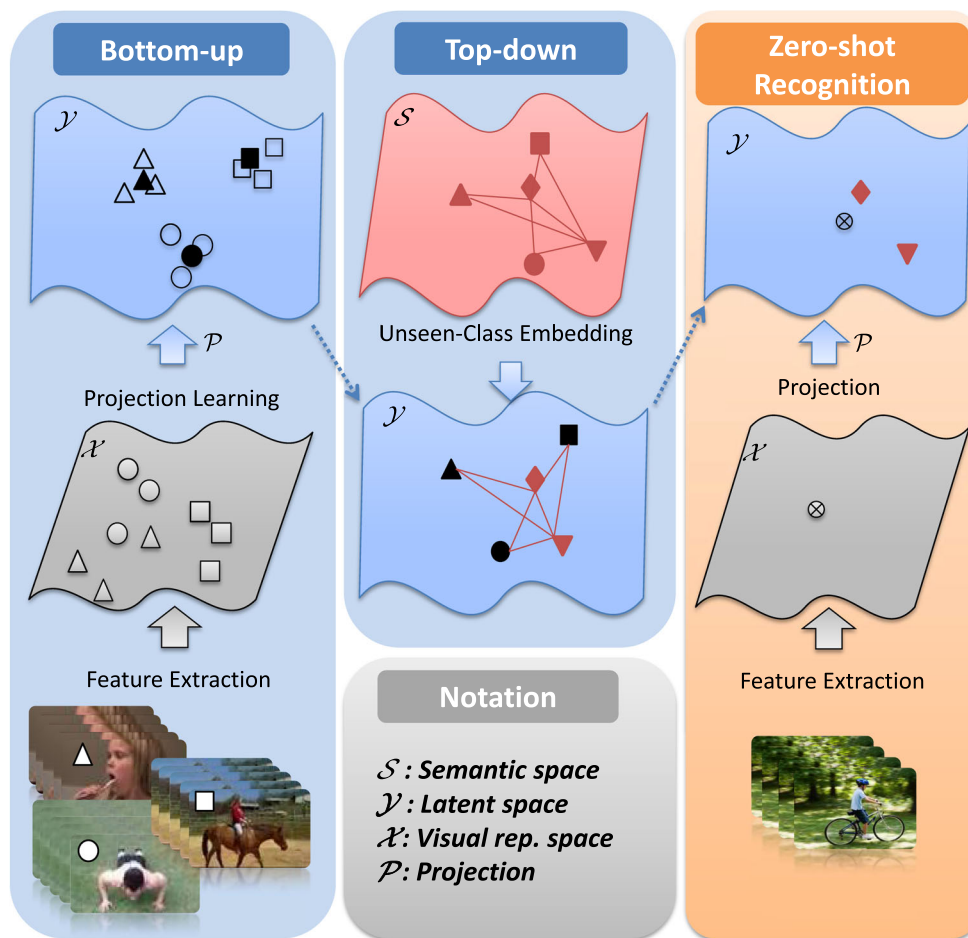
Fig. 1 The proposed bidirectional latent embedding learning (BiDiLEL) framework for zero-shot visual recognition. The BiDiLEL framework consists of two subsequent learning stages. In the bottom–up stage (*left plot*), visual representations in $\mathcal{X}$ are first extracted from the labeled visual data of different training classes marked by *open triangle*, *open circle* and *open square*, respectively. Then a projection $\mathcal{P}$ is learned with a proper supervised subspace learning algorithm to create a latent space $\mathcal{Y}$. The latent embedding of training-class labels are formed by using the mean of the projections of their corresponding training instances in $\mathcal{Y}$, named *landmarks*, marked by *filled triangle*, *filled circle* and *filled square*, respectively. In the top–down stage (*mid-* dle plot), the unseen-class labels in the semantic space $\mathcal{S}$, marked by *filled diamond* and *filled inverted triangle*, are embedded into $\mathcal{Y}$ with a landmark-based learning algorithm in order to preserve the semantic relatedness between all different classes. For zero-shot recognition (*right plot*), the visual representation of a test instance in $\mathcal{X}$, marked by $\otimes$, is projected into the latent space $\mathcal{Y}$ via $\mathcal{P}$ learned in the bottom–up stage. For decision-making, the nearest-neighbor rule is applied by finding out the unseen-class embedding that has the least distance to this instance in $\mathcal{Y}$. That is, the unseen-class label marked by *filled diamond* is assigned to this test instance marked by $\otimes$

learning algorithm is employed to learn a projection $\mathcal{P}$ for preserving the intrinsic locality of instances within the same class and promoting the separability of instances in different classes. As a result, a discriminative latent space $\mathcal{Y}$ is created. Then, we estimate the mean of projections of training instances for every training class. All the estimated means of training classes in $\mathcal{Y}$ are designated for their latent embedding of training-class labels specified in $\mathcal{C}^l$. As a result, we expect that the the bottom–up learning creates the latent embedding of training-class labels that better reflects the semantic relatedness among them and lowers the intra-class variability simultaneously. Thus, we designate all the esti- mated means of training classes as *landmarks* in the latent space and would use them to guide the embedding of unseen-class labels specified in $\mathcal{C}^u$ into the same latent space. The bottom–up latent space learning is carried out by a supervised subspace learning algorithm, *supervised locality preserving projection* (SLPP) (Cheng et al. 2005), which is presented in Sect. 3.2. The motivation behind this choice is to deal with intra-class and inter-class issues along with preserving the intrinsic structure underlying visual data. *Locality preserving projection* (LPP) (Niyogi 2004) is an algorithm that preserves intrinsic structure underlying data, as shown in Niyogi (2004). Its supervised version, SLPP, further exploits

the labeling information to lower the intra-class variability and hence improves the separability between different classes, as shown in Cheng et al. (2005), Zheng et al. (2007); Zhang et al. (2010).

As no training examples in unseen classes are available in ZSL, we have no information on their properties in visual space but clearly know the semantic relatedness between different class labels by means of their semantic representations. In the top–down stage, we thus embed unseen-class labels into the latent space by preserving the semantic relatedness between all different class labels, including training-class to unseen-class as well as unseen-class to unseen-class, guided by the landmarks. Such top–down learning requires a proper enabling technique. To the best of our knowledge, no existing algorithm meets this requirement. Therefore, we propose a semi-supervised MDS algorithm based on the Sammon mapping (Sammon 1969), named *landmark-based Sammon mapping* (LSM), as our enabling technique to learn the latent embedding of unseen-class labels, which is presented in Sect. 3.3.

Once the two subsequent learning tasks are carried out, zero-shot visual recognition is easily obtained in the latent space with a nearest-neighbor rule presented in Sect. 3.4.

Now, we formulate the general problem statement for zero-shot visual recognition. Given a set of labelled instances $X^l = \{x_1^l, x_2^l, \ldots, x_{n_l}^l\} \in \mathcal{X}$, $x_i \in \mathbb{R}^{d_x}$, their labels are denoted by $Z^l = \{z_1^l, z_2^l, \ldots, z_{n_l}^l\}$, $z_i^l \in \mathcal{C}^l$, where $\mathcal{C}^l$ is the set of known class labels. For any given unlabelled instance set $X^u \in \mathbb{R}^{d_x \times n_u}$, the zero-shot visual recognition problem is to predict their labels in $\mathcal{C}^u$ that properly describe the test instances by assuming $\{z_i^u\} \in \mathcal{C}^u$ and $\mathcal{C}^l \cap \mathcal{C}^u = \emptyset$. Here, n_l and n_u are the number of labelled (training) and unlabelled (test) instances, respectively, and d_x is the dimensionality of a visual representation.

3.2 Bottom–Up Latent Space Learning

The bottom–up latent space learning aims to find a projection matrix P that maps instances from their visual space $\mathcal{X}$ to a latent space of a lower dimension $\mathcal{Y}$ to preserve the intrinsic locality of instances within the same class and to promote the separability of instances in different classes. While there are a number of candidate techniques to learn such a latent space, we employ the (SLPP) (Cheng et al. 2005) as the enabling technique since it generally outperforms other candidate techniques, as validated in Sect. 5.

In SLPP, a graph is first constructed with all the training data in X^l to characterize the manifold underlying this data set in the visual representation space $\mathcal{X}$. Following the original settings used in the LPP algorithm (Niyogi 2004), k nearest neighbors (kNN) of a specific data point are used to specify its neighborhood for the graph construction. Train-ing instances $x_i^l \in X^l$ are represented by the nodes in the graph, and an edge is employed to link two nodes when one is in the other's kNN neighborhood. Unlike the unsupervised LPP algorithm, we further take into account the labelling information of the instances when constructing the graph (Cheng et al. 2005). As a result, the edge between two nodes is removed when they do not share the same class label. Therefore, we have a similarity matrix containing all the weights of edges as follows:

$$
W_{ij} = \begin{cases} \exp(-||x_i^l - x_j^l||/2), & x_i^l \in \mathcal{N}_k(x_j^l) \text{ or } x_j^l \in \mathcal{N}_k(x_i^l), \\ & z_i^l = z_j^l \\ 0, & otherwise \end{cases}
\tag{1}
$$

where $\mathcal{N}_k(x)$ denotes the set of k nearest neighbours of x.

In order to preserve the intrinsic local structure, we use the following cost function for learning a projection P:

$$
L(P; W, X^l) = \sum_{i,j} ||P^T x_i^l - P^T x_j^l||_2^2 W_{ij},
\tag{2}
$$

where x_i^l is the i-th column of the input data matrix X^l, corresponding to the feature vector of the i-th training example.

Minimizing the cost function in Eq. (2) enables the nearby instances of the same class label in the visual space to stay as close as possible in the learned latent space. Hence, the intra-class variability is decreased and the inter-class variability is increased reciprocally. For the sake of robustness in numerical computation, the above optimization problem is converted into the following form with the mathematical treatment (Niyogi 2004):

$$
\max_P \frac{Tr\left(P^T X^l D X^{l^T} P\right)}{Tr\left(P^T X^l L X^{l^T} P\right)},
\tag{3}
$$

where $L = D - W$ is the laplacian matrix and D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$.

To penalize the extreme values in the projection matrix P, we further employ a regularization term $Tr(P^T P)$. Thus the cost function in Eq. (2) is now in the following form:

$$
\max_P \frac{Tr\left(P^T X^l D X^{l^T} P\right)}{Tr\left(P^T \left(X^l L X^{l^T} + \alpha I\right) P\right)}
\tag{4}
$$

Finding the optimal projection P is simply boiled down to solving the generalized eigenvalue problem:

$$
X^l D X^{l^T} p = \lambda \left(X^l L X^{l^T} + \alpha I\right) p,
\tag{5}
$$

and the analytic solution is obtained by setting $P = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_d]$ where $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_d$ are those eigenvectors corresponding to the largest d eigenvalues.

Motivated by the treatment proposed by Akata et al. (2013, 2016) for binary label embedding, we further apply two normalization strategies, *centralization* and l_2-*normalization*, to the latent representations of training examples, Y^l, to avoid unfavorable situations in zero-shot recognition. Our motivation behind the treatment is different from theirs (Akata et al. 2013, 2016). For the sake of readability, we have to describe our motivation at the end of Sect. 3.3 as it concerns not only bottom–up but also top–down learning stages. By using the centralization, the latent representations Y^l are centralized to make all the features (i.e., rows) have zero mean. Furthermore, l_2-normalization is applied on each column of Y^l to make all the instances have unit norms, i.e., $\hat{\boldsymbol{y}}_i^l = \boldsymbol{y}_i^l / ||\boldsymbol{y}_i^l||_2$ for $i = 1, 2, \ldots, n_l$. After the centralization and l_2-normalization, the latent embedding of i-th training class, $\boldsymbol{b}_i^l$, is estimated by

$$\boldsymbol{b}_i^l = \frac{1}{n_i} \sum_{z_j^l = i} \hat{\boldsymbol{y}}_j^l, \quad i = 1, \ldots, |\mathcal{C}^l|, \tag{6}$$

where n_i is the number of training instances in the i-th training class, and $|\mathcal{C}^l|$ is the number of training classes. Likewise, all mean points of $|\mathcal{C}^l|$ known classes estimated from training instances, $\boldsymbol{b}_1^l, \ldots, \boldsymbol{b}_{|\mathcal{C}^l|}^l$, are l_2-normalized to have unit norms. We specify all $|\mathcal{C}^l|$ normalized mean points as *landmarks* to provide the guidance for embedding unseen classes into the learned latent space (c.f. Sect. 3.3).

3.3 Top–Down Latent Embedding Learning

The top–down algorithm aims to learn latent embedding of unseen classes. With the guidance of landmarks, i.e., the latent embedding of known classes, all the unseen-class labels are embedded into the same latent space learned in the bottom–up stage via preserving their semantic relatedness pre-defined by an existing semantic representation of class labels (c.f. Sect. 4.3).

Let $B^l = \{\boldsymbol{b}_1^l, \boldsymbol{b}_2^l, \ldots, \boldsymbol{b}_{|\mathcal{C}^l|}^l\} \in \mathbb{R}^{d_y \times |\mathcal{C}^l|}$ collectively denote the latent embedding of all the training classes where d_y is the dimension of the latent space formed in the bottom–up stage. Similarly, the latent embedding of $|\mathcal{C}^u|$ unseen classes are collectively denoted by $B^u = \{\boldsymbol{b}_1^u, \boldsymbol{b}_2^u, \ldots, \boldsymbol{b}_{|\mathcal{C}^u|}^u\} \in \mathbb{R}^{d_y \times |\mathcal{C}^u|}$. In order to preserve the semantic relatedness between all the classes, the distance between two classes in the latent space should be as close to their semantic distance in the semantic space as possible but the embedding of known classes are already settled with Eq. (6) in the bottom–up learning stage. Hence, this leads to a brand new semi-supervised MDS problem. By

---

Algorithm 1 Landmark-based Sammon Mapping (LSM)

Input: The semantic representations for training and unseen classes, S^l and S^u, (or the semantic distance matrix $\Delta = \{\delta_{ij}(\boldsymbol{s}_i, \boldsymbol{s}_j)\}$), the training-class latent embedding B^l, learning rate η.
Output: The latent unseen-class embedding B^{u*}.
1: Initialize B_0^u for $t = 0$ randomly;
2: **repeat**
3:      Calculate gradient $g_t = \nabla_{B_t^u} E(B_t^u)$ (c.f. Appendix A);
4:      Update $B_{t+1}^u := B_t^u + \eta g_t$;
5:      $t := t + 1$;
6: **until** Stopping criteria are satisfied.

---

means of the Sammon mapping (Sammon 1969), we propose a *landmark-based Sammon mapping* (LSM) algorithm to tackle this problem.

By using a proper semantic representation of all class labels, we achieve the semantic representations of training and unseen classes, $S^l \in \mathbb{R}^{d_s \times |\mathcal{C}^l|}$ and $S^u \in \mathbb{R}^{d_s \times |\mathcal{C}^u|}$, where their i-th columns are $\boldsymbol{s}_i^l$ and $\boldsymbol{s}_i^u$, respectively, and d_s is the dimensionality of the semantic space. Then, the LSM cost function is defined by

$$E(B^u) = \frac{1}{|\mathcal{C}^l||\mathcal{C}^u|} \sum_{i=1}^{|\mathcal{C}^l|} \sum_{j=1}^{|\mathcal{C}^u|} \frac{\left(d\left(\boldsymbol{b}_i^l, \boldsymbol{b}_j^u\right) - \delta\left(\boldsymbol{s}_i^l, \boldsymbol{s}_j^u\right) \right)^2}{\delta\left(\boldsymbol{s}_i^l, \boldsymbol{s}_j^u\right)}$$
$$+ \frac{2}{|\mathcal{C}^u|(|\mathcal{C}^u| - 1)} \sum_{i=1}^{|\mathcal{C}^u|} \sum_{j=i+1}^{|\mathcal{C}^u|} \frac{\left(d\left(\boldsymbol{b}_i^u, \boldsymbol{b}_j^u\right) - \delta\left(\boldsymbol{s}_i^u, \boldsymbol{s}_j^u\right) \right)^2}{\delta\left(\boldsymbol{s}_i^u, \boldsymbol{s}_j^u\right)}, \tag{7}$$

where $d(\boldsymbol{x}, \boldsymbol{y})$ and $\delta(\boldsymbol{x}, \boldsymbol{y})$ are the distance metrics in the latent space and the semantic space, respectively. Intuitively, the first term of Eq. (7) concerns the semantic relatedness between known and unseen classes and the second term of Eq. (7) takes into account the semantic relatedness between unseen classes in the top–down learning. Minimizing $E(B^u)$ leads to the solution: $B^{u*} = \arg\min_{B^u} E(B^u)$.

Following Sammon (1969), we derive the LSM algorithm by using the gradient descent optimization procedure. As a result, our LSM algorithm is summarized in Algorithm 1, and the derivation of gradient $\nabla_{B^u} E(B^u)$ used in Algorithm 1 is described in Appendix A. Applying Algorithm 1 to the semantic representations of $|\mathcal{C}^u|$ unseen classes results in their embedding in the latent space: $\boldsymbol{b}_1^u, \ldots, \boldsymbol{b}_{|\mathcal{C}^u|}^u$.

Now we described our motivation underlying two normalization strategies presented at the end of Sect. 3.2. In general, our motivation underlying two normalization strategies aims to facilitate the embedding of unseen-class labels in the top–down stage. As advocated by Akata et al. (2016), the instance-level l_2-normalization of binary attributes of class labels to the unit magnitude and zero-mean centering facilitate zero-shot recognition. For embedding unseen classes in

the latent space, our LSM algorithm has to take into account the distance information between known and unseen classes in both the semantic and the latent spaces. Applying the l_2-normalization to the embedding of training instances thus ensures that the distances measured in two spaces are in the same scale. Applying the centralization is due to the l_2-normalization. All the l_2-normalized training instances in the latent space may concentrate in a small region (on the one surface side of the unit hyper-sphere). This phenomenon may cause no sufficient room or a difficulty to accommodate the embedding of unseen-class labels in the top–down learning. The zero-mean centralization ameliorates the detrimental effect of this phenomenon by scattering training instances in a larger region to facilitate the unseen class label embedding.

3.4 Zero-Shot Recognition in the Latent Space

Once all the class labels are embedded in the latent space by our Algorithm 1, zero-shot visual recognition is gained in the learned latent space. Given a test instance $\boldsymbol{x}_i^u$, its label is predicted in the latent space via the following procedure. First of all, we apply projection P obtained in the bottom–up learning stage to map it into the latent space:

$$\boldsymbol{y}_i^u = P^T \boldsymbol{x}_i^u. \tag{8}$$

After being subtracted by the mean estimated on all the training instances in the latent space, $\boldsymbol{y}_i^u$ is then l_2-normalized in the same manner as done for all training instances. Thus, its label, l^*, is assigned to the class label of which embedding is closest to $\boldsymbol{y}_i^u$; i.e.,

$$l^* = \arg \min_l d\left(\boldsymbol{y}_i^u, \boldsymbol{b}_l^u\right), \tag{9}$$

where $\boldsymbol{b}_l^u$ is the latent embedding of l-th unseen class, and $d(\boldsymbol{x}, \boldsymbol{y})$ is a distance metric in the latent space. In our experiments, the Euclidean distance metric is used for measuring the distance due to the nature of manifold learning in the LPP algorithm (Niyogi 2004).

A recent study Shao et al. (2016) suggests that the use of multiple visual representations can improve the robustness in action recognition. As a result, we have extended our proposed framework to the joint use of multiple complimentary visual representations for robust zero-shot visual recognition, which is presented in Appendix B. To promote robustness, we also come up with a visual representation complementarity measurement, as described in Appendix C.

3.5 Post-processing Techniques

The post-processing in ZSL refers to those techniques that exploit the information conveyed in test instances to improve the ZSL performance. In our work, two existing post-processing techniques, *self-training* (Xu et al. 2015b) and *structured prediction* (Zhang and Saligrama 2016b), are incorporated into our proposed framework.

3.5.1 Self-Training

The self-training (ST) is a post-processing technique proposed by Xu et al. (2015b) in order to alleviate the domain shift problem. The general idea behind the self-training is adjusting the latent embedding of unseen classes according to the distribution of all the test instance projections in the latent space. It is straightforward to incorporate this post-processing technique into our zero-short visual recognition framework. Given the i-th unseen class ($i = 1, 2, \ldots, |\mathcal{C}^u|$), Xu et al. (2015b) adjust the latent embedding $\boldsymbol{b}_i^u$ to $\hat{\boldsymbol{b}}_i^u$, where

$$\hat{\boldsymbol{b}}_i^u := \frac{1}{k} \sum_{\boldsymbol{y}^u \in \mathcal{N}_k(\boldsymbol{b}_i^u)}^{k} \boldsymbol{y}^u. \tag{10}$$

Here, $\mathcal{N}_k(\boldsymbol{b}_i^u)$ is a neighborhood of the latent embedding $\boldsymbol{b}_i^u$ containing the k nearest test instances. In other words, this nearest neighbour search in the self-training is confined to only test instances. As all the test instances have to be used in the self-training, this leads to a *transductive* learning setting. Unlike their treatment in Xu et al. (2015b), in our experiments, we adjust $\boldsymbol{b}_i^u$ to the arithmetic average between $\hat{\boldsymbol{b}}_i^u$ and $\boldsymbol{b}_i^u$, $(\hat{\boldsymbol{b}}_i^u + \boldsymbol{b}_i^u)/2$, for a trade-off between preserving their semantic relatedness and alleviating the domain shift effect.

3.5.2 Structured Prediction

Structured prediction is yet another option for post-processing recently proposed by Zhang and Saligrama (2016b). Similar to self-training, structured prediction also takes advantage of the batch of test instances under the transductive setting. This method was originally proposed for their own zero-shot recognition algorithm (Zhang and Saligrama 2016a). In our work, we adapt it for our proposed framework, which is a simplified version of their structured prediction algorithm (Zhang and Saligrama 2016b) by using only its first step and dropping out the rest steps due to incompatibility to our approach.

In this simplified version, we update the latent embedding of unseen classes B^u by clustering analysis on the batch of test instances. First of all, a number of clusters are generated for all the test instances by the K-means algorithm where the number of clusters is chosen the same as that of unseen

Table 2 Summary of datasets used in our experiments

Number	AwA	CUB-200-2011	UCF101	HMDB51
Attributes	85	312	115	–
Known classes	40	150	51/81	26
Unseen classes	10	50	50/20	25
Instances	30,475	11,788	13,320	6,676

classes $|\mathcal{C}^u|$. In our experiments, we always initialize the cluster centers with the latent embedding of unseen-class labels learned in the top–down stage.[1] After the K-mean clustering, structured prediction needs to establish a one-to-one correspondence between a cluster and a unseen class so that the sum of distances of all possible pairs of cluster center and the unseen-class embedding can be least. Let $A \in \{0, 1\}^{|\mathcal{C}^u| \times |\mathcal{C}^u|}$ denote the one-to-one correspondence matrix where $A_{ij} = 1$ indicates that cluster i corresponds to unseen class j. The correspondence problem is formally formulated as follows:

$$\min_A \sum_{c=1}^{|\mathcal{C}^u|} \sum_{k=1}^{|\mathcal{C}^u|} A_{kc} \cdot d\left(\boldsymbol{m}_k, \boldsymbol{b}_c^u\right)$$
$$s.t. \quad \forall k, \forall c, \quad \sum_k A_{kc} = 1, \sum_c A_{kc} = 1, \quad (11)$$

where $\boldsymbol{m}_k$ is the center of k-th cluster, $\boldsymbol{b}_c^u$ is the c-th unseen-class latent embedding and $d(\cdot, \cdot)$ is Euclidean distance metric. This optimization problem in Eq. (11) can be solved by linear programming (Zhang and Saligrama 2016b).

For zero-shot recognition, a test instance falling into a specific cluster is assigned to the label of its corresponding unseen class based on the correspondence matrix A.

4 Experimental Settings

In this section, we describe our experimental settings including the information of benchmark datasets, the visual and the semantic representations used in our experiments, the investigation of different factors that may affect the zero-shot visual recognition accuracy and our comparative study.

4.1 Dataset

In our experiments, we employ four publicly accessible datasets to evaluate our proposed framework. The first two are benchmarks for zero-shot object recognition, namely animal with attributes (AwA) (Lampert et al. 2014) and Caltech-UCSD Birds-200-2011 (CUB-200-2011) (Wah et al.

2011). As both are among those most commonly used datasets used to evaluate ZSL algorithms in literature, we can directly compare the performance of our approach to that of those state-of-the-art zero-shot visual recognition methods. Other two datasets are UCF101 (Soomro et al. 2012) and HMDB51 (Kuehne et al. 2011), which are benchmarks widely used to evaluate the performance of a human action recognition algorithm in presence of a large number of classes. To evaluate the performance in zero-shot human action recognition, we use the same class-wise data splits on UCF101 and HMDB51 as suggested by Xu et al. (2015a, b) in our experiments, which allows us to compare ours to theirs explicitly.

Table 2 summarizes the main information of four datasets used in our experiments. The specific setting for zero-shot visual recognition is highlighted as follows:

- *AwA* there are 30,475 animal images belonging to 50 classes. The 40/10 (known/unseen) class-wise data split has been originally set by the dataset collectors (Lampert et al. 2014).
- *CUB-200-2011* this is a fine-grained dataset of 11,788 images regarding 200 different bird species, collected by Wah et al. (2011). The class-wise data split is often 150/50 (known/unseen) on this dataset in previous works. In our experiments, we follow the same 100/50/50 class-wise data split for training/validation/test used in Akata et al. (2015); Reed et al. (2016); Xian et al. (2016).
- *UCF101* it is a human action recognition dataset collected from YouTube by Soomro et al. (2012). There are 13,320 real action video clips falling into 101 action categories. In our experiments, we use 51/50 and 81/20 (known/unseen) class-wise data splits. We use the same 30 independent 51/50 splits[2] randomly generated by Xu et al. (2015a). Regarding 81/20 splits, we randomly generate 30 independent splits as this setting does not appear in their work Xu et al. (2015a).
- *HMDB51* it contains 6,766 video clips from 51 human action classes, collected by Kuehne et al. (2011). Once again, we use the same 30 independent 26/25 splits randomly generated by Xu et al. (2015a).

---

[1] Our empirical study suggests that the random initialization in the K-mean clustering may lead to better performance but causes structured prediction to be unstable.

[2] The dataset of all 30 splits are available online: http://staff.cs.manchester.ac.uk/~kechen/BiDiLEL.

4.2 Visual Representation

The latest progresses in computer vision suggest that features learned by using deep *convolutional neural networks* (CNNs) significantly outperform any of hand-crafted counterparts in object recognition (Simonyan and Zisserman 2015; Szegedy et al. 2015). Features learned by deep CNNs have also been applied in zero-shot visual recognition (Akata et al. 2014; Al-Halah and Stiefelhagen 2015; Fu et al. 2015). In our experiments, we use two different pre-trained deep CNN models to generate visual representations of images in AwA and CUB-200-2011. For a direct comparison with state-of-the-art methods, we follow their settings by using the top fully connected layer of GoogLeNet of 1024 dimensions (Szegedy et al. 2015) and the top pooling layer of VGG19 of 4096 dimensions (Simonyan and Zisserman 2015) to generate feature vectors of images. In particular, MatConvNet (Vedaldi and Lenc 2015) has been employed to extract the aforementioned deep features.

There are many different visual representations that characterize video streams regarding human actions. After investigating the existing visual representations for human action video streams, we employ two kinds of state-of-the-art visual representations for human action video streams in our experiments, i.e. the *improved dense trajectory* (IDT) (Wang and Schmid 2013) and the *convolutional 3D* (C3D) (Tran et al. 2015). Our empirical studies described in Appendix C along with those reported in literature suggest that two selected visual representations not only outperform a number of candidate representations but also are highly complementary to each other. The IDT is a class of state-of-the-art hand-crafted visual representations proposed by Wang and Schmid (2013) for human action recognition. Four different types of visual descriptors, HOG, HOF, MBHx and MBHy, are extracted from each spatio-temporal volume, and their dimensions are reduced by a factor of two with PCA. Then the representations of a video stream are generated by the Fisher vector derived from a Gaussian mixture model of 256 components. Thus, the video representations have 24,576 features for HOG, MBHx, MBHy and 27,648 for HOF (Peng et al. 2016; Wang and Schmid 2013), respectively. For computational efficiency, we further apply PCA on those video representations to reduce their dimensions down to 3,000 in our experiments. Note that the visual representation, IDT(MBH), in our experiments refers to a feature vector formed by concatenating MBHx and MBHy. C3D (Tran et al. 2015) is an effective approach that uses deep CNNs for spatio-temporal video representation learning. In our experiments, we use the model provided by Tran et al. (2015). This model was pre-trained on the Sports-1M dataset. Following the settings in Tran et al. (2015), we divide a video stream into segments in length of 16 frames and there is an overlap of eight frames on two consecutive segments. As a result, the fc6 activations are first extracted for all the segments and then averaged to form a 4096-dimensional video representation.

In our experiments for multiple visual representations, different visual representations described above are jointly used via our proposed combination approach described in Appendix B.

4.3 Semantic Representation

To evaluate our proposed framework thoroughly, we employ two widely used semantic representations, *attributes* and *word vectors*, in our experiments.

As shown in Table 2, AwA and CUB-200-2011 self-contain 85 and 312 class-level continuous attributes that characterize each class label, respectively. UCF101 class labels have been manually annotated with 115 binary attributes by Jiang et al. (2014). To our knowledge, however, there are no attributes for those class labels appearing in HMDB51. Hence, we cannot report attribute-based results on this dataset. Table 3 exemplifies some typical attributes used in different datasets. Following the suggestion made by Akata et al. (2016), Changpinyo et al. (2016a) and Zhang and Saligrama (2015), we also apply l_2-normalization to each of attributes vectors to facilitate their latent embedding. In our experiments, we use Euclidean distance metric to measure the semantic distance between attributes of two class labels during the top–down latent embedding learning.

Unlike attribute-based semantic representations, Mikolov et al. (2013) propose a continuous skip-gram model to learn a distributed semantic representation, *word vectors*, in an unsupervised way. In our experiments, we employ the skip-gram model (well known as *Word2Vec*) (Mikolov et al. 2013), trained on the Google News dataset containing about 100 billion words for AwA, UCF101 and HMDB51, where the word embedding space is of 300 dimensions. However, there are a number of out-of-vocabulary words in CUB-200-2011.

Table 3 Exemplification of typical attributes used in different datasets

Dataset	Attribute
AwA	colours(black, brown, red, etc.), stripes, furry, hairless, big, small, paws, longneck, tail, chewteeth, fast, smelly, bipedal, jungle, water, cave, group, grazer, insects
CUB-200-2011	bill_shape(curved, dagger, hooked, needle, etc.), wing_color(blue, yellow, etc.), upperparts_color, tail_shape(forked, rounded, pointed, squared, etc.)
UCF101	object(ball_like, rope_like, animal, sharp, etc.), bodyparts_visible(face, fullbody, onehand, etc.), body_motion(flipping, walking, diving, bending, etc.)

As a result, we employ 400-dimensional word vectors trained on English-language Wikipedia (Akata et al. 2015; Xian et al. 2016) for CUB-200-2011. Following the existing works, we use the "cosine" distance metric to measure the semantic distance between two class labels in a word embedding space during the top–down latent embedding learning.

4.4 On Hyper-Parameters

It is well known that hyper-parameters in a learning model may critically determine its performance. Thus, we investigate the impact of different hyper-parameters involved in our proposed framework to search for "optimal" hyper-parameter values. In general, there are four hyper-parameters; i.e., the number of nearest neighbors (k_G) for the graph construction in SLPP, the trade-off factor (α) applied to the regularization in SLPP and the dimensionality of a learned latent space (d_y) during the bottom–up latent embedding learning as well as the number of nearest neighbors (k_{ST}) when the self-training (Xu et al. 2015b) is used.

In our experiments, we use the *classwise* cross-validation to seek the optimal hyper-parameter values and investigate how each hyper-parameter affects the performance. We strictly follow the procedure suggested by Akata et al. (2016), Zhang and Saligrama (2016a) to do the cross-validation on all the datasets apart from CUB-200-2011 that has a standard training/validataion/test split. In a trial, we randomly reserve 20% training classes as validation data and the rest of training classes are used as training data. In our experiments, we repeat such a cross-validation experiment for multiple trials and report the averaging performance on validation data. For AwA, five trials were conducted in our cross-validation based on its default training/test split. For two human action datasets, UCF101 and HMDB51, each has 30 different training/test splits provided by Xu et al. (2015a). For each of 30 splits, we conducted three-trial cross-validation to achieve the optimal hyper-parameter values for this split only. Hence, our cross-validation experiment on a human action dataset had to be repeated for 30 times on all the splits respectively.

Without considering the post-processing of self-training, our approach has three hyper-parameters, α, d_y and k_G. It would be extremely expensive computationally if an exhausted grid search is conducted. In our experiments, we adopt a two-stage procedure to find out optimal hyper-parameters for different visual representations respectively. We first conducted a coarse grid search with $\alpha = 0.1, 10$, $d_y = 10, 100, 500$, and $k_G = 1, 10, 50$. Then, we further fine-tune each of hyper-parameters sequentially by fixing the remaining two hyper-parameters.

In our fine-tuning stage, we conduct the cross-validation experiments for each of four hyper-parameters sequentially based on the information (on how sensitive a hyper-parameter is to the performance) obtained from the coarse grid search. Thus, our fine-tuning stage performs in the following order:

- **α** First of all, we investigate the impact of α in Eq. (4). In our experiment, we fix the initial optimal value of d_y and k_G resulting from the grid search to look into the impact of α by setting it to 0.001, 0.01, 0.1, 1, 10, 100 and 1000.
- **d_y** As training class labels are used in the bottom–up latent embedding learning, the proper value of d_y may depend on the number of training classes that varies across different datasets. To investigate the zero-shot recognition accuracy with different d_y values in a large range, we use the optimal values of α found in the previous step and fix the initial optimal value k_G resulting from the grid search. In our experiment, we look into $d_y = 50, 100, 150, 200, 250$ and 300.
- **k_G** By making use of the optimal α and d_y values achieved from two previous steps, we look into the impact of k_G defined in Eq. (1) for each dataset in the same manner by fixing other hyper-parameters and allowing only k_G to change in a large range: $k_G = 5, 10, 15, 20, 25$ and 30, respectively, to see how k_G affects the zero-shot recognition accuracy on different datasets.
- **k_{ST}** For this post-processing, we fix the optimal values of three hyper-parameters found as described above and evaluate the zero-shot recognition accuracy with a large range of k_{ST} in Eq. (10) from 20 to 200 with an interval of 20 on each dataset, as suggested in Xu et al. (2015b).

As a result, the set of hyper-parameter values leading to the *best* accuracy in the above fine-tuning process are treated as *"optimal"* and used in test to yield the performance for unseen classes.

4.5 On Enabling Techniques

This experimental setting aims to explore the proper enabling techniques for our proposed framework and investigate the role played by two subsequent learning stages. As stated in Sect. 3.1, there are a number of candidate subspace learning techniques that could be used in the bottom–up learning as reviewed in Sect. 2.2. To the best of our knowledge, however, none of the existing non-parametric subspace learning model can be directly applied to the top–down learning where the task emerges from our proposed framework (c.f. Sect. 3.3). Motivated by the work Changpinyo et al. (2016b), we employ a parametric learning model as a baseline for the top–down learning. In all the experiments described below, the nearest-neighbor rule described in Sect. 3.4 is used for zero-shot recognition.

For the bottom–up latent space learning, we conduct a comparative study on four candidate techniques (c.f. Sect. 2.2): two unsupervised algorithms, PCA and LPP, and

two supervised algorithms, LDA and SLPP.[3] For fairness, we apply the same cross-validation procedure described in Sect. 4.4 to find out the optimal hyper-parameter values, i.e., d_y for PCA, α, d_y and k_G for LPP. For LDA, however, the dimension of the latent space is intrinsically determined by the number of training classes. Hence, the dimension of its latent space is set to the number of training classes subtracted by one. Furthermore, we apply our LSM algorithm directly to visual representations without the bottom–up learning. This experiment yields a baseline that clearly exhibits the role played by each of two subsequent learning stages in our framework.

In addition, some existing ZSL methods could be enabling techniques applied to our bottom–up latent space learning,[4] e.g., SJE (Akata et al. 2015), LatEm (Xian et al. 2016) and CCA (Fu et al. 2015). Unlike the aforementioned subspace learning where no semantic representations of labels are considered, those ZSL algorithms take into account semantic representations during projection learning. For example, SJE (Akata et al. 2015) learns a projection matrix W such that given a pair of visual and semantic representations, $\boldsymbol{x}$ and $\boldsymbol{y}$, similarity score $\boldsymbol{x}^T W \boldsymbol{y}$ is maximized if $\boldsymbol{x}$ has a label represented by $\boldsymbol{y}$. LatEm extends SJE to a nonlinear model with multiple piecewise linear models by learning different projection matrices such that different instances can select the most appropriate projection matrices. CCA is an algorithm used to learn a common space from two multidimensional variables such that the correlation between the projections of the two variables in the common space can be maximized. Furthermore, the canonical correlation problem may be converted into a distance minimization problem: $min_{W,W'} ||XW - YW'||_F$ (Hardoon et al. 2004), where $||\cdot||_F$ is the Frobenius norm and W and W' are projection matrices for source and target embedding (to the common space). In our experiments, we strictly follow the experimental setting described in the original literature and the learned projections from visual to target space are used to form the latent space. As a result, the dimensionality of the latent space is equal to the dimensionality of semantic representations for SJE and LatEm, and the dimension of latent space learned by CCA is found by the same cross-validation procedure described in Sect. 4.4. It is worth mentioning that LatEm yields multiple projection matrices, which results in multiple "latent" spaces. Hence, zero-shot recognition has to take into account all of such "latent" spaces. There are two manners for the nearest-neighbor based decision-making: minimum distance and averaging distance to a label embedding in multiple

"latent" spaces. As the averaging distance always outperforms the minimum distance, we only report the results based on the averaging distance.

Our LSM algorithm described in Sect. 3.3 is always employed for the top–down embedding learning in all the aforementioned experiments regarding the bottom–up learning. We further conduct an experiment by employing the *support vector regression* (SVR) (Smola and Vapnik 1997) to replace the LSM for the top–down learning. This experiment is based on SLPP used in the bottom–up stage. When SVR is used, the top–down learning is formulated as a regression task (Changpinyo et al. 2016b) and the regressor is trained based on training data where the landmarks are targets used for learning. As our LSM and the SVR work in a quite different manner for the top–down learning, it is possible to combine their results to improve the zero-shot recognition performance as well as to understand their behavior. To this end, we further use a simple ensemble strategy to combine the two methods. Let $\boldsymbol{b}^u_{lsm}$ and $\boldsymbol{b}^u_{svr}$ $(u = 1, \ldots, |\mathcal{C}^u|)$ denote the latent embedding for unseen classes resulting from two different top–down techniques, respectively. Thus, the combined embedding of unseen classes is defined by $(\boldsymbol{b}^u_{lsm} + \boldsymbol{b}^u_{svr})/2$ $(u = 1, \ldots, |\mathcal{C}^u|)$ to be used in zero-shot recognition.

It is worth mentioning that the optimal hyper-parameter values in various candidate techniques are also achieved via the same classwise cross-validation protocol suggested by Akata et al. (2016), Zhang and Saligrama (2016a).

4.6 On the Joint Use of Multiple Semantic Representations

The joint use of multiple semantic representations can also improve the robustness in zero-shot visual recognition (Akata et al. 2014, 2015; Changpinyo et al. 2016a; Xian et al. 2016). Our framework allows for jointly using multiple semantic representations easily. Since our recognition process described in Algorithm 1 requires only between-class semantic distances as inputs, we use a convex combination of semantic distance matrices to exploit the information conveyed in multiple semantic spaces.

Given attributes and word vectors used in our experiments, let Δ^{Att} and Δ^{WV} denote the corresponding semantic distance matrices achieved by using attributes and word vectors, respectively. The fused distance matrix is achieved by $\Delta = \gamma \Delta^{WV} + (1 - \gamma)\Delta^{Att}$, where γ is in the range of (0.0, 1.0) and used to trade-off the contributions of two different types of semantic representations. In our experiments, we investigate the optimal value of γ via a grid search by setting $\gamma = 0.1, 0.2, \ldots, 0.9$ with the classwise cross-validation.

As the aforementioned strategy for the simultaneous use of two semantic representations affects both the top–down and the bottom–up learning, we have to apply the same cross-

---

[3] The implementation of PCA and LDA used in our experiments is based on the open source available online: http://www.cad.zju.edu.cn/home/dengcai/Data/DimensionReduction.html.

[4] An anonymous reviewer pointed out this fact and suggested this experiment.

validation protocol described in Sect. 4.4 first to find the optimal values of all other hyper-parameters, α, d_y, k_G and k_{ST}, especially for the scenario that two semantic representations are jointly used. In our experiments, we exploited experimental results on a single semantic representation to achieve those optimal hyper-parameter values. As a result, we chose the set of hyper-parameter values leading to the best *averaging* accuracy regarding two semantic representations (when used individually on a visual representation) as the optimal values. Thus, this set of optimal hyper-parameter values are fixed to be used in the subsequent classwise cross-validation that decides the optimal value of γ.

4.7 On the Comparative Study

To evaluate our proposed framework thoroughly, we conduct a comparative study by comparing ours to most of state-of-the-art zero-shot visual recognition methods on four benchmark datasets described in Sect. 4.1. For a fair comparison, we adopt the same experimental settings and use the optimal hyper-parameter values reported in literature so that one can clearly see the results yielded by different methods under the same conditions.

Below, we briefly describe the state-of-the-art zero-shot visual recognition methods used in our comparative study.

- *Direct attribute prediction (DAP)* DAP proposed by Lampert et al. (2009) is among those earliest methods for ZSL, which is often used as a baseline in zero-shot visual recognition (Al-Halah and Stiefelhagen 2015; Gan et al. 2016; Xu et al. 2015b). It learns a direct mapping from visual representation to attributes of their corresponding class labels. In deployment, the attributes associated with a test instance are predicted by the learned mapping functions. Then the label of this test instance is inferred with a probabilistic model.
- *Indirect attribute prediction (IAP)* IAP (Lampert et al. 2009) is yet another baseline ZSL method (Al-Halah and Stiefelhagen 2015; Gan et al. 2016; Xu et al. 2015b). Unlike DAP, in deployment, IAP first predicts the probability scores of all the known classes for the test instance and then apply the known class-attribute relationship in semantic space to estimate the probability scores of attributes. With the prediction of attributes, the label of this test instance is predicted in the same way as DAP.
- *Structured joint embedding (SJE)* SJE (Akata et al. 2014) learns a joint embedding space by maximizing the compatibility of visual and semantic representations $\mathbf{x}^T W \mathbf{s}$. The objective used for learning W in SJE is similar to that proposed for the structured SVM parameter learning (Tsochantaridis et al. 2005).
- *Synthesized classifiers (Syn-Classifier)* Syn-Classifier (Changpinyo et al. 2016a) is a recent zero-shot object

recognition method that exploits the relations between known and unseen classes in the semantic space. As a result, the so-called "phantom" classes are explored to model the relations between known and unseen classes for ZSL.
- *Exemplar prediction [EXEM(SynC)]* Changpinyo et al. (2016b) is yet another bidirectional latent space learning method similar to ours where PCA and SVR are used to learn the latent space and to predict the exemplars for unseen classes. Once the exemplars of unseen classes are predicted, they are treated as ideal semantic representations and Syn-Classifier (Changpinyo et al. 2016a) is used for zero-shot recognition.
- *Latent embedding (LatEm)* LatEm (Xian et al. 2016) is a non-trivial extension of SJE. Instead of learning a single mapping transformation in SJE, it learns a piecewise linear compatibility function of K parameter matrices W_i ($i = 1, \ldots, K$). Given a test instance $\mathbf{x}$, it will be labelled as the class whose semantic representation maximises $\max_{1 \le i \le K} \mathbf{x}^T W_i \mathbf{s}$.
- *Hierarchical attribute transfer (HAT)* HAT (Al-Halah and Stiefelhagen 2015) explores the hierarchical structures underlying the set of attributes. Based on the relations of the original attributes, additional high-level attributes are exploited to enhance the knowledge transfer.
- *Kernel-alignment domain-invariant component analysis (KDICA)* KDICA (Gan et al. 2016) learns a feature transformation of the visual representations to eliminate the mismatches between different classes in terms of their marginal distributions over the input. Once the transformation is learned, the representation yielded by this transformation is used for its attribute prediction.
- *Semantic similarity embedding (SSE)* SSE (Zhang and Saligrama 2015) learns a model that decomposes the visual and semantic representations into a mixture of known classes. Thus, all the unseen classes can be represented by such "mixture patterns". Given a test instance, its visual representation is first decomposed into the mixture of known classes, and its "mixture pattern" is used against all the unseen classes. A label of the class with the most similar mixture pattern is assigned to this test instance.
- *Joint latent similarity embedding (JLSE)* JLSE (Zhang and Saligrama 2016a) is one of the latest zero-shot recognition methods. It formulates zero-shot recognition as a binary prediction problem by assigning a binary label to a pair of source and target domain instances. The visual and semantic representations are mapped to their corresponding latent spaces via dictionary learning and the joint latent similarity embedding is learnt with a probabilistic model via a joint optimization on two latent spaces so that

a pair of matched source and target domain instances can be found.

- *Unsupervised domain adaptation (UDA)* UDA (Kodirov et al. 2015) is proposed to tackle the domain shift problem in zero-shot recognition by regularizing the projection learning for unseen instances with the projection learned with training data in known classes. Due to using test instances in projection learning, it is a typical transductive ZSL algorithm.
- *Transductive multiview-hypergraph label propagation (TMV-HLP)* TMV-HLP (Fu et al. 2015) employs multiple visual and semantic representations to learn a common space. Heterogeneous hyper-graphs are constructed for multiple views and label propagation in zero-shot object recognition. This method is proposed especially for transductive ZSL.
- *Ridge regression + nearest-neighbor (RR + NN)* RR+NN (Xu et al. 2015b) is one of latest methods proposed for zero-shot human action recognition. In Xu et al. (2015b), a ridge regression from visual to semantic representations is learned with the training data. Then the learned regression model is first used to map a test instance from visual to semantic spaces. Then a nearest neighbour algorithm is employed to assign a class label to this test instance in the semantic space.
- *Manifold regression + self-training + normalized nearest-neighbor (MR + ST + NRM)* MR+ST+NRM (Xu et al. 2015b) is one of latest methods proposed for zero-shot human action recognition. Similar to ours, the manifold of visual space is considered to learn a smooth regression model towards enhancing the generalisation to unseen classes. The self-training (ST) and the normalized nearest neighbour (NRM) (Dinu et al. 2015) techniques are further employed towards further improving the zero-shot recognition accuracy.

5 Experimental Results

In this section, we report our experimental results[5] corresponding to our settings described in Sects. 4.4–4.7, where the per-class accuracy is used in evaluation.

5.1 Results on Hyper-Parameters

By using the cross-validation protocol described in Sect. 4.4, we report experimental results via the mean and the standard error of per-class recognition accuracy over multiple cross-validation trials for all the datasets unless a dataset

has a standard classwise split. The initial grid search suggests that the initial optimal values of d_y and k_G are 100 and 10, respectively, regardless of different visual representations and are hence used in the hyper-parameter fine-tuning stage described in Sect. 4.4.

Figure 2 shows the detailed cross-validation results in terms of statistics (mean and standard error) obtained in the fine-tuning stage for two object recognition datasets. It is evident from Fig. 2 that different values of α affect the recognition accuracy significantly, while k_G has the least effects on performance. Based on results illustrated in Fig. 2, we choose the set of hyper-parameter values leading to the best accuracy in each case when specific visual and semantic representations work together as "optimal" for such a case. For clarity, we explicitly list all the optimal hyper-parameter values for different scenarios on two object recognition datasets in Table 4. It is worth stating that the optimal hyper-parameter values for the scenario that two semantic representations are jointly used are easily achieved with the results shown in Fig. 2; i.e., for a specific visual representation, the averaging accuracy on two semantic representations can be immediately achieved at each grid point of a hyper-parameter and the optimal value can hence be found easily for this combination scenario.

As there are 30 different training/test splits (Xu et al. 2015a) for each of two human action datasets, UCF101 and HMDB51, we have 30 sets of optimal hyper-parameter values on a dataset for each of scenarios that combine specific visual and semantic representations. As we used four different visual representations and up to two semantic representations in our experiments, there are totally up to eight different scenarios. Due to the limited space, it is impossible to include all the details in this paper but we have made all the experimental results on two human action datasets available on our project website.

The optimal hyper-parameter values achieved via the aforementioned classwise cross-validation experiments are used in the comparative study reported in Sect. 5.4.

5.2 Results on Enabling Techniques

By using the settings described in Sect. 4.5, we conduct the experiments to explore proper enabling techniques. Table 5 shows the zero-shot recognition performance resulting from the baseline without the bottom–up learning and the use of different enabling techniques, where a bold-font figure indicates the best performance of statistical significance in a specific setting, and a italic-font figure suggests that the performance has been improved due to the combination of different embedding of unseen-class labels resulting from our LSM and SVR.

Regarding those enabling techniques for the bottom–up learning, it is evident from Table 5 that SLPP generally per-

---

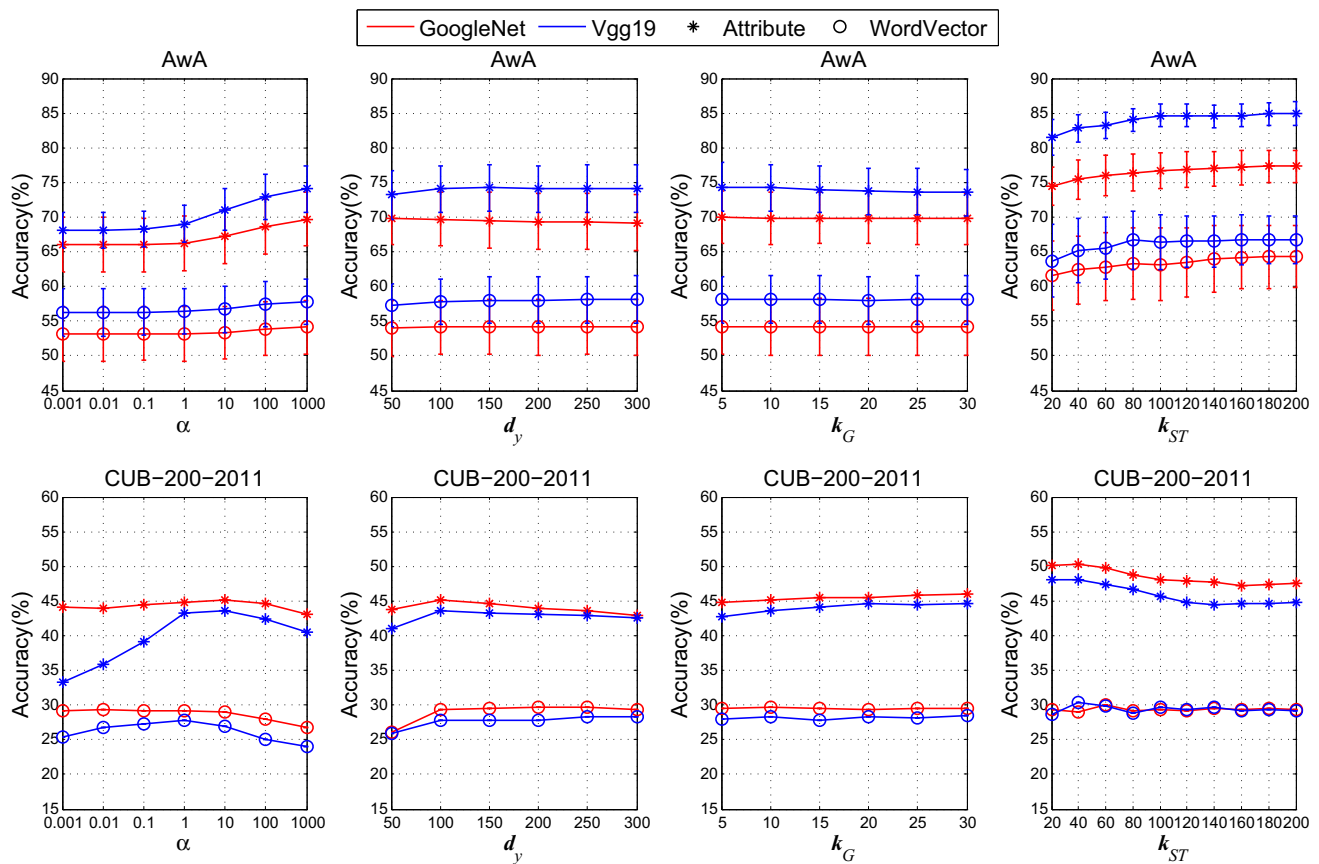[5] The source code used in our experiments as well as more experimental results not reported in this paper are available on our project website: http://staff.cs.manchester.ac.uk/~kechen/BiDiLEL.

Fig. 2 The classwise cross-validation results on AwA and CUB-200-2011 used to determine the optimal hyper-parameter values

Table 4 Optimal hyper-parameter values in our approach on two object recognition datasets, corresponding to different visual and semantic representations, obtained with the cross-validation protocol described in Sect. 4.4

Dataset	Vis. Rep.	Sem. Rep.	Hyper-parameter			
			α	d_y	k_G	k_{ST}
AwA	GoogLeNet	WV	1000	300	15	200
		Att	1000	50	5	180
		Comb	1000	50	5	200
	Vgg19	WV	1000	300	10	160
		Att	1000	150	5	180
		Comb	1000	150	5	200
CUB-200-2011	GoogLeNet	WV	0.01	250	10	60
		Att	10	100	30	40
		Comb	10	100	30	60
	Vgg19	WV	1	250	30	40
		Att	10	100	20	20
		Comb	1	100	30	40

Vis. Rep., visual representation; Sem. Rep., semantic representation; Att, attributes; WV, word vectors; Comb, the combination of attributes and word vectors

forms the best regardless of datasets and representations. By a closer look at Table 5, we observe that the performance of PCA and LPP is comparable to that of SLPP when deep representations, e.g., GoogLeNet, Vgg19 and C3D, are used. This suggests that the additional use of labeling information in SLPP does not improve the generalization performance substantially. It is also evident from Table 5 that the aggressive use of labeling information in LDA usually results in poor generalization. Such performance is attributed to the fact that, to some extent, the visual features generated by

Table 5 Zero-shot visual recognition performance (mean ± standard error)% of our approach resulting from the baseline without the bottom–up learning and the use of different enabling techniques in the bottom–up and the top–down learning stages

Dataset	Vis. Rep.	Sem. Rep.	LSM					SVR	LSM & SVR
			Vis. Rep.	PCA	LPP	LDA	SLPP	SLPP	SLPP
AwA	GoogLeNet	WV	**57.0**	56.4	56.2	51.1	56.1	55.9	57.7
		Att	**74.2**	73.3	72.1	72.6	72.4	**74.1**	74.5
	Vgg19	WV	**57.3**	56.2	56.4	51.0	56.7	**57.7**	59.6
		Att	**79.8**	78.9	79.0	73.9	79.1	75.7	78.6
CUB-200-2011	GoogLeNet	WV	29.5	29.3	32.7	**36.7**	34.5	30.4	32.7
		Att	43.5	43.9	45.9	42.0	49.7	**50.7**	52.4
	Vgg19	WV	29.5	28.9	34.9	**36.7**	**37.0**	33.2	34.9
		Att	42.7	42.8	45.0	42.8	47.6	**49.7**	49.5
UCF101 (81/20)	C3D	WV	36.6 ± 1.1	37.6 ± 1.1	$\mathbf{38.1 \pm\pm 1.2}$	31.9 ± 0.9	$\mathbf{38.3 \pm 1.2}$	35.1 ± 0.8	36.5 ± 0.9
		Att	35.3 ± 1.1	38.3 ± 1.0	38.7 ± 1.2	34.5 ± 1.2	39.2 ± 1.0	$\mathbf{43.3 \pm 1.0}$	43.7 ± 1.1
	MBH	WV	21.6 ± 0.8	23.8 ± 0.9	27.3 ± 0.9	24.0 ± 0.9	$\mathbf{29.9 \pm 1.1}$	26.6 ± 0.9	27.7 ± 0.8
		Att	21.1 ± 0.9	24.6 ± 0.9	26.5 ± 0.8	27.5 ± 0.8	$\mathbf{31.4 \pm 0.8}$	30.6 ± 0.8	32.2 ± 0.8
	IDT	WV	18.4 ± 0.5	20.5 ± 0.6	28.4 ± 0.9	31.3 ± 1.1	$\mathbf{32.6 \pm 1.1}$	29.4 ± 0.9	31.3 ± 1.1
		Att	21.2 ± 0.7	22.9 ± 0.8	28.4 ± 0.9	$\mathbf{34.5 \pm 0.9}$	34.2 ± 0.8	33.7 ± 0.7	35.0 ± 0.7
UCF101 (51/50)	C3D	WV	17.8 ± 0.4	$\mathbf{18.5 \pm 0.4}$	$\mathbf{18.6 \pm 0.4}$	16.3 ± 0.4	$\mathbf{18.9 \pm 0.4}$	17.9 ± 0.5	18.9 ± 0.5
		Att	18.4 ± 0.4	20.2 ± 0.4	20.5 ± 0.4	19.2 ± 0.4	20.5 ± 0.5	$\mathbf{23.8 \pm 0.6}$	24.2 ± 0.5
	MBH	WV	9.7 ± 0.3	10.7 ± 0.2	12.5 ± 0.3	11.7 ± 0.3	$\mathbf{14.0 \pm 0.3}$	12.8 ± 0.3	13.5 ± 0.3
		Att	10.0 ± 0.3	11.6 ± 0.3	12.8 ± 0.3	14.5 ± 0.3	$\mathbf{15.2 \pm 0.3}$	$\mathbf{15.2 \pm 0.4}$	16.0 ± 0.3
	IDT	WV	8.5 ± 0.2	9.2 ± 0.2	13.5 ± 0.4	14.4 ± 0.4	$\mathbf{15.4 \pm 0.4}$	14.3 ± 0.2	14.9 ± 0.3
		Att	9.7 ± 0.3	10.6 ± 0.3	13.3 ± 0.4	$\mathbf{17.3 \pm 0.4}$	16.6 ± 0.3	16.5 ± 0.4	16.9 ± 0.4
HMDB51	C3D	WV	18.8 ± 0.7	18.5 ± 0.7	18.3 ± 0.7	15.1 ± 0.6	18.6 ± 0.7	$\mathbf{19.3 \pm 0.7}$	19.5 ± 0.6
	MBH	WV	10.6 ± 0.4	11.7 ± 0.4	12.5 ± 0.5	12.0 ± 0.4	$\mathbf{14.0 \pm 0.6}$	12.9 ± 0.4	13.3 ± 0.5
	IDT	WV	11.3 ± 0.4	10.7 ± 0.4	12.7 ± 0.7	15.4 ± 0.5	$\mathbf{16.4 \pm 0.6}$	15.8 ± 0.6	16.0 ± 0.6

Vis. Rep., visual representation; Sem. Rep., semantic representation; Att, attributes; WV, word vectors

Table 6 Results on SJE, LatEm and CCA used as the enabling techniques for the bottom–up learning while the LSM is used for the top–down learning

Dataset	Vis. Rep.	Sem. Rep.	SJE	LatEm	CCA
AwA	GoogLeNet	WV	47.8	53.1	48.9
		Att	70.0	73.2	72.7
	Vgg19	WV	48.2	57.4	51.9
		Att	75.7	76.5	75.5
CUB-200-2011	GoogLeNet	WV	26.8	26.6	37.1
		Att	39.2	34.8	49.7
	Vgg19	WV	26.7	25.1	37.9
		Att	37.2	36.0	49.2

deep CNNs via supervised learning on a much larger dataset characterize the intrinsic structure of visual data and discriminative aspects of images or video streams belonging to different classes. Further supervised learning on such visual representations may lead to overfitting to training classes. It is particularly true on AwA where the deep features of visual data sufficiently capture the intrinsic "cluster" structure; it is observed from Table 5 that without the bottom–up learning, our LSM algorithm yields the better performance than that of itself working on four candidate subspace learning algorithms used in the bottom–up learning. This suggests that the bottom–up learning might be redundant for a dataset such as AwA. As clearly shown in Table 5, however, the bottom–up learning on other three datasets leads to a performance gain regardless of different visual and semantic representations used. On the other hand, we observe that the performance of LDA is also comparable to that of SLPP when a kernel representation space is used by the joint use of multiple visual representations, e.g., IDT on UCF101. This suggests that after being mapped onto a kernel representation space, the instances in different classes are not separated well, and the use of labeling information improves the discriminative aspects in the latent space. Based on the baseline performance, we conclude that the proper bottom–up learning is required by taking into account preserving intrinsic structure underlying visual data and promoting the discriminative capability simultaneously unless a visual representation has already captured the intrinsic "cluster" structure of a visual data set.

Regarding the enabling top–down learning techniques, the results shown in Table 5 reveal that LSM generally performs better than SVR, although its performance is inferior to that of SVR in some occasions for specific visual and semantic representations used on different datasets: GoogLeNet + Att and Vgg19 + WV on AwA, Att on CUB-200-2011 and C3D + Att on UCF101. Furthermore, an interesting phenomenon is observed from Table 5 that the combination of LSM and SVR in unseen-class embedding always improves the performance of SVR whenever SVR outperforms LSM but the further improvement does not always happen when our LSM

outperforms SVR. The experimental results exhibit the difference between the SVR, a parametric model, and our LSM, a non-parametric model in knowledge transfer.

Regarding the use of existing ZSL methods for bottom–up learning, we have only done the experiments on two object recognition benchmark datasets since results on these two datasets are only reported in the literature regarding three candidate methods, SJE, LatEm and CCA. It is evident from Table 6 that SLPP generally outperforms three methods on AwA although the performance of LatEm is better than that of using specific visual and semantic representation combinations, GoogleNet + Att and Vgg19 + WV. However, CCA outperforms SLPP on CUB-200-2011 for those visual and semantic representation combinations: GoogLeNet + WV, Vgg19 + WV and Vgg19 + Att. This suggests that a proper enabling technique for the bottom–up learning may be dependent of a specific dataset. Fortunately, different enabling techniques can be easily and flexibly applied in our framework.

In summary, the above experimental results suggest that SLPP can preserve intrinsic structure underlying visual data and facilitate discriminating different classes in the latent space. Thus, SLPP provides a proper enabling technique for the bottom–up learning. On the other hand, our proposed LSM works effectively in comparison to SVR and is hence a proper enabling technique for the top–down learning.

5.3 Results on the Joint Use of Multiple Semantic Representations

By using the settings described in Sect. 4.6, we conduct experiments to seek the optimal value of γ used in combining two semantic representations: attributes and word vectors. As there are many candidate visual representations, we adopt only those that lead to the state-of-the-art performance in our experiments. As there are no attributes available in HMDB51, our experiments are done on AwA, CUB-200-2011 and UCF101. While different values of γ in its permissible range are used in the experiments, $\gamma = 0.0$ cor-
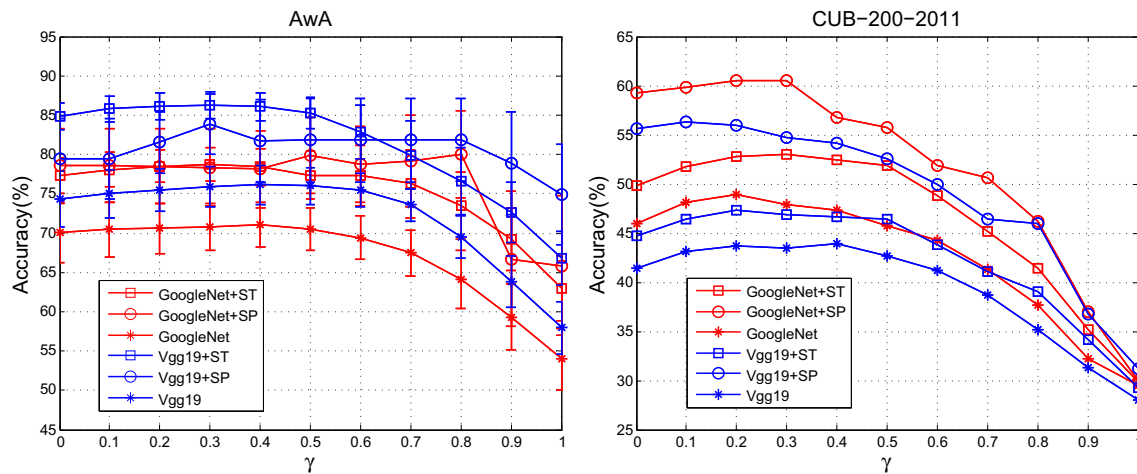
Fig. 3 The classwise cross-validation results on AwA and CUB-200-2011 when two semantic representations are jointly used

responds to the situation that attributes are only used and $\gamma = 1.0$ indicates that word vectors are only used.

Figure 3 illustrates the classwise cross-validation results for different values of γ in the joint use of two semantic representations on two object recognition datasets. From Fig. 3, we see the optimal hyper-parameter values for different visual representations in different settings, which are used in the comparative study reported in Sect. 5.4. Under the inductive setting, $\gamma = 0.4$ for AwA regardless of visual representations and $\gamma = 0.2$, 0.4 for CUB-200-2011 when GoogleNet and Vgg19 are used, respectively. When the self-teaching is used in the transductive setting, $\gamma = 0.3$ for AwA regardless of visual representations and $\gamma = 0.3$, 0.2 for CUB-200-2011 when GoogleNet and Vgg19 are used, respectively. When the structure prediction is used in the transductive setting, $\gamma = 0.8$, 0.3 for AwA and $\gamma = 0.3$, 0.1 for CUB-200-2011 when GoogleNet and Vgg19 are used, respectively.

Likewise, the classwise cross-validation was done on 30 training/test splits for different scenarios on each of two human action datasets, respectively, as same as described in Sect. 5.1. Consequently, those optimal γ values on 30 splits, which are also available on our project website, are used in the comparative study reported in Sect. 5.4.

5.4 Results on Comparative Study

By using the settings described in Sect. 4.7, we conduct experiments to compare ours to a number of state-of-the-art zero-shot visual recognition methods. By using the identical experimental protocol as suggested in literature, we can directly compare the performance to that reported in literature. For our approach, we report the mean and standard deviation resulting from five random initial conditions used in the top–down learning on AwA and CUB-200-2011 as well as the mean and standard error of the mean resulting

from 30 training/test splits on UCF101 and HMDB51 while the detailed experimental results can be found on our project website. To facilitate our presentation, we group the experimental results in terms of zero-shot object and human action recognition.

5.4.1 Results on Zero-Shot Object Recognition

Table 7 shows the performance of different approaches in zero-shot object recognition where the best performance is highlighted with bold font and the results from the inductive and the transductive settings are separated with a delimiter.

For AwA, it is evident from Table 7 that in the attribute-based inductive setting our approach based on Vgg19 visual features outperforms all other state-of-the-art approaches with a high accuracy of 79.1% in terms of *per-class* accuracy except JLSE that reports the *per-image* accuracy of 80.5%. In its corresponding transductive setting, the use of *self-training* (ST) in our approach based on GoogLeNet and Vgg19 visual features lifts the accuracy to 86.2 and 88.5%, respectively, and the use of *structured prediction* (SP) further improves the accuracy to 92.6 and 95.0%, respectively. In the word-vector based inductive setting, our approach based on Vgg19 visual features and 300-dimensional word vectors[6] yields an accuracy of 56.1%, which is lower than that of SJE but higher than that of LatEm where 400-dimensional word vectors are used in their experiments. In the transductive setting, we observe that both ST and SP lead to a higher accuracy. Especially, the use of SP dramatically improves the accuracy from 56.1% to 76.0% based on GoogleNet features. Our results suggest that SP is constantly superior to ST under the transductive setting. While the combination of two semantic representations sig-

---

Table 7 Zero-shot object recognition per-class accuracy (mean ± standard deviation) % of different approaches on AwA and CUB-200-2011 datasets

Method	Vis. Rep.	AwA			CUB-200-2011		
		Att	WV	Comb	Att	WV	Comb
DAP (Al-Halah and Stiefelhagen 2015)	GoogLeNet	59.9	–	–	36.7	–	–
SJE (Akata et al. 2015)	GoogLeNet	66.7	**60.1**	73.9	50.1	28.4	**51.0**
SynC (Changpinyo et al. 2016a)	GoogLeNet	72.9	–	76.3	54.7[b]	–	–
EXEM(SynC) (Changpinyo et al. 2016b)	GoogLeNet	77.2	–	–	**59.8**[b]	–	–
LatEm (Xian et al. 2016)	GoogLeNet	72.5	52.3	76.1	45.6	33.1	47.4
HAT (Al-Halah and Stiefelhagen 2015)	GoogLeNet	74.9	–	–	51.8[†]	–	–
BiDiLEL(Ours)	GoogLeNet	72.4 ± 0.0	56.1 ± 0.0	73.5 ± 0.0	49.7 ± 0.0	34.5 ± 0.0	50.9 ± 0.2
KDICA (Gan et al. 2016)	Vgg19	73.8	–	–	43.7	–	–
SSE (Zhang and Saligrama 2015)	Vgg19	76.3 ± 0.8	–	–	30.4 ± 0.2	–	–
JLSE (Zhang and Saligrama 2016a)	Vgg19	**80.5 ± 0.5**[c]	–	–	42.1 ± 0.6	–	–
BiDiLEL(Ours)	Vgg19	79.1 ± 0.0	56.7 ± 0.0	**78.8 ± 0.0**	47.6 ± 0.0	**37.0 ± 0.0**	48.4 ± 0.1
UDA(Kodirov et al. 2015)[a]	OverFeat	73.2	–	75.6	39.5	–	40.6
TMV-HLP (Fu et al. 2015)[a]	OverFeat+Decaf	–	–	80.5	–	–	47.9
BiDiLEL+ST (Ours)[a]	GoogLeNet	86.2 ± 0.0	59.5 ± 0.0	85.6 ± 0.0	53.5 ± 0.0	38.0 ± 0.0	56.6 ± 0.0
BiDiLEL+SP (Ours)[a]	GoogLeNet	92.6 ± 0.0	**76.0 ± 0.0**	92.5 ± 0.0	**62.8 ± 0.0**	37.7 ± 0.0	**61.1 ± 0.0**
JLSE+SP (Zhang and Saligrama 2016b)[a]	Vgg19	92.1 ± 0.1	–	–	55.3 ± 0.8	–	–
BiDiLEL+ST(Ours)[a]	Vgg19	88.5 ± 0.0	57.3 ± 0.0	89.7 ± 0.0	52.8 ± 0.0	**40.9 ± 0.0**	53.0 ± 0.0
BiDiLEL+SP (Ours)[a]	Vgg19	**95.0 ± 0.0**	68.9 ± 0.0	**94.9 ± 0.0**	59.3 ± 0.1	40.6 ± 0.0	57.4 ± 0.0

Vis. Rep., visual representation; Sem. Rep., semantic representation; Att, attributes; WV, word vectors; Comb, combination of two semantic representations

[a] This method uses unlabelled test instances during learning under a transductive setting

[b] Fact that the result is generated based on their specific splits publicly unavailable

[c] Results on per-image accuracy

'–' refers to no result reported for this setting

nificantly improves the performance of some methods, e.g., SJE, it is not a case for our approach on this dataset. It is observed that the combination of attributes and word vectors generally does not improve the performance on AwA regardless of visual representations.

For CUB-200-2011, EXEM(SynC) yields the best accuracy of 59.8% in the attribute-based inductive setting but their classwise data split protocol is unavailable publicly. In contrast, the best performance of our approach is 49.7% with GoogleNet features, which is better than that of DAP, LatEM, SSE, JLSE and KDICA but worse than that of SJE, HAT and SynC. The use of SP in the attribute-based transductive setting leads our approach to an accuracy of 62.8%. In the word-vector based settings, it is evident from Table 7 that our approach outperforms all others; 37% accuracy is achieved with Vgg19 features under the inductive setting and the use of ST and SP under the transductive setting lifts the the accuracy to 40.9 and 40.6%, respectively. Similar to other methods, e.g., SJE and LatEm, the joint use of two semantic representations further improves the performance of our approach on CUB-200-2011 in the inductive setting. Nevertheless, the combination of semantic representations under the transductive setting leads to limited improvement only when ST is used but does not work when SP is applied in our approach.

It is worth pointing out that the cost function used in our LSM algorithm is non-convex and the gradient-based local search only leads to a local optimum. However, our experimental results shown in Table 7 suggest that the LSM learning on two benchmark object recognition datasets is insensitive to different unseen-class embedding initialization and almost always converges to the same solution.

5.4.2 Results on Zero-Shot Human Action Recognition

For zero-shot human action recognition, to the best of our knowledge, there are much fewer studies than zero-shot object recognition in literature. Hence, we compare ours to all the existing approaches (Gan et al. 2016; Kodirov et al. 2015; Xu et al. 2015b). It is worth clarifying that our experiments concern only zero-shot human action recognition while the previous work Xu et al. (2015b) addresses other issues, e.g., action detection, which is not studied in our work. In addition, Xu et al. (2015b) come up with the data augmentation technique to improve the performance. However, we notice that in their experiments, some classes from auxiliary data used for training are re-used in test, which violates the fundamental assumption of ZSL that training and test classes must be mutually excluded. Thus, we do not compare ours to theirs (Xu et al. 2015b) in terms of the data augmentation. Since SP almost always outperforms ST for the post-processing, we only report the results yielded by SP under the transductive setting in Table 8.

Table 8 shows the zero-shot recognition results of different methods on UCF101 and HMDB51. In the inductive setting, our approach yields the best performance on two different UCF101 classwise splits, 51/50 and 81/20. It is clearly seen from Table 8 that our approach leads to the highest accuracy of 22.2 and 19.6% on average for the 51/50 split and the highest accuracy of 43.3 and 40.8% on average for the 81/20 split by using attributes and word vectors, respectively, along with appropriate visual representations. Despite the use of the same visual representations, our approach outperforms all the others regardless of semantic representations. Moreover, it is evident from Table 8 that the exactly same conclusion on the results achieved in the inductive setting can be drawn in the transductive setting, where our approach results in the highest accuracy of 29.8 and 23.0% on average for the 51/50 split and the highest accuracy of 57.1 and 49.3% on average for the 81/20 split by using attributes and word vectors, respectively, along with appropriate visual representations. Furthermore, the results shown in Table 8 suggest that the joint use of two semantic representations always improve the performance of our approach substantially regardless of visual representations and classwise splits; for the 51/50 and the 81/20 splits, the highest accuracy is 26.4 and 51.1% on average, respectively, in the inductive setting and the highest accuracy is 35.1 and 66.9% on average, respectively, in the transductive setting. For HMDB51, the behavior of our approach is identical to that on the 51/50 split of UCF101 in both inductive and transductive settings when word vectors are used. Ours yields the highest averaging accuracy of 20.6% in the inductive setting and 22.3% with SP along with C3D+IDT features in the transductive setting, respectively, although our approach underperforms MR+ST+NRM when IDT(HOG,HOF,MBH) features are used. Here, it is worth pointing out that neither of the optimal hyper-parameter search methods were described nor the detailed experimental results on each of 30 training/test splits were reported in Gan et al. (2016), Kodirov et al. (2015), Xu et al. (2015b). In general, we summarize the main results shown in Table 8 as follows: (a) the use of attributes always outperforms that of word vectors when the same visual representations are employed, which is consistent with Akata et al. (2016); (b) the deep representation C3D outperforms the state-of-the-art hand-crafted visual representations significantly in all the settings; (c) the joint use of two semantic representations substantially improves the performance of our approach; and (d) under the transductive setting, SP does not always improve the zero-shot recognition performance probably due to the highly complex intrinsic structure underlying visual data.

In summary, the experimental results achieved from our comparative study suggest that our proposed framework yields the favorable performance and is generally comparable

Table 8 Zero-shot human action recognition performance (mean ± standard error)% of different approaches on UCF101 and HMDB51 datasets

Method	Vis. Rep.	UCF101 (51/50)			UCF101 (81/20)			HMDB51
		Att	WV	Comb	Att	WV	Comb	WV
DAP (Xu et al. 2015b)	IDT(HOG,HOF,MBH)	15.2 ± 0.3	–	–	–	–	–	–
IAP (Xu et al. 2015b)	IDT(HOG,HOF,MBH)	15.6 ± 0.3	–	–	–	–	–	–
RR + NN (Xu et al. 2015b)	IDT(HOG,HOF,MBH)	–	11.7 ± 0.2	–	–	–	–	14.5 ± 0.1
DAP (Gan et al. 2016)	C3D	–	–	–	26.8 ± 1.1	–	–	–
KDICA (Gan et al. 2016)	C3D	–	–	–	31.1 ± 0.8	–	–	–
BiDiLEL (Ours)	IDT(MBH)	15.2 ± 0.3	14.0 ± 0.3	17.1 ± 0.3	31.4 ± 0.8	29.9 ± 1.1	36.3 ± 1.0	14.0 ± 0.6
BiDiLEL (Ours)	IDT(HOG,HOF,MBH)	16.6 ± 0.3	15.4 ± 0.4	19.5 ± 0.4	34.2 ± 0.8	32.6 ± 1.1	39.6 ± 1.0	16.4 ± 0.6
BiDiLEL (Ours)	C3D	20.5 ± 0.5	18.9 ± 0.4	24.4 ± 0.6	39.2 ± 1.0	38.3 ± 1.2	47.5 ± 1.3	18.6 ± 0.7
BiDiLEL (Ours)	C3D + IDT	22.2 ± 0.5	19.6 ± 0.5	26.4 ± 0.6	43.3 ± 1.2	40.8 ± 1.2	51.1 ± 1.2	20.6 ± 0.8
UDA (Kodirov et al. 2015)[a]	IDT(MBH)[b]	13.2 ± 0.6	–	–	20.1 ± 1.0	–	–	–
MR + ST + NRM (Xu et al. 2015b)[a]	IDT(HOG,HOF,MBH)	–	18.0 ± 0.4	–	–	–	–	19.1 ± 0.5
BiDiLEL + SP (Ours)[a]	IDT(MBH)	17.6 ± 0.6	15.2 ± 0.6	19.1 ± 0.9	41.1 ± 1.4	36.6 ± 1.9	44.3 ± 1.8	13.5 ± 0.6
BiDiLEL + SP (Ours)[a]	IDT(HOG,HOF,MBH)	21.8 ± 0.7	17.0 ± 0.6	23.3 ± 0.8	48.3 ± 1.6	40.3 ± 1.6	51.0 ± 2.0	15.9 ± 0.7
BiDiLEL + SP (Ours)[a]	C3D	28.3 ± 1.0	21.4 ± 0.8	31.6 ± 1.2	50.1 ± 2.0	45.6 ± 2.0	58.3 ± 1.8	18.9 ± 1.1
BiDiLEL + SP (Ours)[a]	C3D + IDT	29.8 ± 1.0	23.0 ± 0.9	35.1 ± 1.1	57.1 ± 1.7	49.3 ± 2.0	66.9 ± 1.9	22.3 ± 1.1

Vis. Rep., visual representation; Sem. Rep., semantic representation; Att, attributes; WV, word vectors; Comb, combination of two semantic representations

[a] This method uses unlabelled test instances during learning under a transductive setting

[b] Visual representation is encoded with bag-of-features

'–' refers to no result reported for this setting

to all the existing state-of-the-art zero-shot visual recognition methods described in Sect. 4.7.

6 Concluding Remarks

In this paper, we have proposed a novel bidirectional latent embedding learning framework for zero-shot visual recognition. Unlike the existing ZSL approaches, our framework works in two subsequent learning stages. The bottom–up learning first creates a latent space by exploring intrinsic structures underlying visual data and the labeling information contained in training data. Thus, the means of projected training instances of the same class labels form the embedding of known class labels and are treated as landmarks. The top–down learning subsequently adopts a semi-supervised manner to embed all the unseen-class labels in the latent space with the guidance of landmarks in order to preserve the semantic relatedness between all different classes in the latent space. Thanks to the favorable properties of this latent space, the label of a test instance is easily predicted with a nearest-neighbor rule. Our thorough evaluation under comparative studies suggests that our framework works effectively and its performance is competitive with most of state-of-the-art zero-shot visual recognition approaches on four benchmark datasets.

In our ongoing research, we would further explore potential enabling techniques to improve the performance and extend our proposed framework to other kinds of ZSL problems in computer vision, e.g., multi-label zero-shot visual recognition. Despite being proposed for zero-shot visual recognition, we expect that our proposed framework also works on ZSL problems in different domains, e.g., zero-shot audio classification, zero-shot music genre recognition and and zero-shot multimedia information retrieval.

Appendix A: Derivation of Gradient on the LSM Cost Function

In this appendix, we derive the gradient of $E(B^u)$ defined in Eq. (7). To facilitate our presentation, we simplified our notation as follows: d_{ij}^{lu}, d_{ij}^{uu}, δ_{ij}^{lu} and δ_{ij}^{uu} denote $d(\boldsymbol{b}_i^l, \boldsymbol{b}_j^u)$, $d(\boldsymbol{b}_i^u, \boldsymbol{b}_j^u)$, $\delta(\boldsymbol{s}_i^l, \boldsymbol{s}_j^u)$ and $\delta(\boldsymbol{s}_i^u, \boldsymbol{s}_j^u)$, respectively, where $d(\cdot, \cdot)$ and $\delta(\cdot, \cdot)$ are distance metrics used in the latent and semantic spaces.

Based on the simplified notation, Eq. (7) is re-written as follows:

$$E(B^u) = \frac{1}{|\mathcal{C}^l||\mathcal{C}^u|} \sum_{i=1}^{|\mathcal{C}^l|} \frac{\left(d_{ij}^{lu} - \delta_{ij}^{lu}\right)^2}{\delta_{ij}^{lu}} + \frac{2}{|\mathcal{C}^u|(|\mathcal{C}^u| - 1)} \sum_{i=j+1}^{|\mathcal{C}^u|} \frac{\left(d_{ij}^{uu} - \delta_{ij}^{uu}\right)^2}{\delta_{ij}^{uu}}. \tag{A.1}$$

Let $\boldsymbol{b}_j^u = (b_{j1}^u, \ldots, b_{jd_y}^u)$ denote the embedding of unseen class j in the latent space, where b_{jk}^u is its k-th element. By applying the chain rule, we achieve

$$\frac{\partial E(B^u)}{\partial b_{jk}^u} = \frac{\partial E(B^u)}{\partial d_{ij}^{lu}} \frac{\partial d_{ij}^{lu}}{\partial b_{jk}^u} + \frac{\partial E(B^u)}{\partial d_{ij}^{uu}} \frac{\partial d_{ij}^{uu}}{\partial b_{jk}^u}. \tag{A.2}$$

For the first term in Eq. (A.2), we have

$$\frac{\partial E(B^u)}{\partial d_{ij}^{lu}} = \frac{2}{|\mathcal{C}^l||\mathcal{C}^u|} \sum_{i=1}^{|\mathcal{C}^l|} \frac{\left(d_{ij}^{lu} - \delta_{ij}^{lu}\right)}{\delta_{ij}^{lu}}, \tag{A.3}$$

and

$$\frac{\partial d_{ij}^{lu}}{\partial b_{jk}^u} = \frac{-2\left(b_{ik}^l - b_{jk}^u\right)}{2\sqrt{\sum_k \left(b_{ik}^l - b_{jk}^u\right)^2}} = \frac{b_{jk}^u - b_{ik}^l}{d_{ij}^{lu}}. \tag{A.4}$$

Likewise, for the second term in Eq. (A.2), we have

$$\frac{\partial E(B^u)}{\partial d_{ij}^{uu}} = \frac{4}{|\mathcal{C}^u|(|\mathcal{C}^u| - 1)} \sum_{i=1}^{|\mathcal{C}^u|} \frac{\left(d_{ij}^{uu} - \delta_{ij}^{uu}\right)}{\delta_{ij}^{uu}}, \tag{A.5}$$

and

$$\frac{\partial d_{ij}^{uu}}{\partial b_{jk}^u} = \frac{-2\left(b_{ik}^u - b_{jk}^u\right)}{2\sqrt{\sum_k \left(b_{ik}^u - b_{jk}^u\right)^2}} = \frac{b_{jk}^u - b_{ik}^u}{d_{ij}^{uu}}. \tag{A.6}$$

Inserting Eqs. (A.3)–(A.6) into Eq. (A.2) leads to

$$\frac{\partial E(B^u)}{\partial b_{jk}^u} = \frac{2}{|\mathcal{C}^l||\mathcal{C}^u|} \sum_{i=1}^{|\mathcal{C}^l|} \frac{d_{ij}^{lu} - \delta_{ij}^{lu}}{\delta_{ij}^{lu} d_{ij}^{lu}} \left(b_{jk}^u - b_{ik}^l\right) + \frac{4}{|\mathcal{C}^u|(|\mathcal{C}^u| - 1)} \sum_{i=j+1}^{|\mathcal{C}^u|} \frac{d_{ij}^{uu} - \delta_{ij}^{uu}}{\delta_{ij}^{uu} d_{ij}^{uu}} \left(b_{jk}^u - b_{ik}^u\right). \tag{A.7}$$

Thus, we obtain the gradient of $E(B^u)$ with respect to B^u used in Algorithm 1: $\nabla_{B^u} E(B^u) = \left(\frac{\partial E(B^u)}{\partial b_{jk}^u}\right)_{|\mathcal{C}^u| \times d_y}$.

Appendix B: Extension to the Joint Use of Multiple Visual Representations

In this appendix, we present the extension of our bidirectional latent embedding framework in the presence of multiple visual representations.

In general, different visual representations are often of various dimensionality. To tackle this problem, we apply the kernel-based methodology (Cristianini and Shawe-Taylor 2000) by mapping the original visual space $\mathcal{X}$ to a pre-specified kernel space $\mathcal{K}$. For the visual representations X^l, the mapping leads to the corresponding kernel representations $K^l \in \mathbb{R}^{n_l \times n_l}$ where K_i^l is the i-th column of the kernel matrix K^l and $K_{ij}^l = k(\boldsymbol{x}_i^l, \boldsymbol{x}_j^l)$. $k(\boldsymbol{x}_i^l, \boldsymbol{x}_j^l)$ stands for a kernel function of certain favorable properties, e.g., the linear kernel function used in our experiments is $k(\boldsymbol{x}_i^l, \boldsymbol{x}_j^l) = \boldsymbol{x}_i^{l\,T} \boldsymbol{x}_j^l$. As there is the same dimensionality in the kernel space, the latent embedding can be learned via a joint use of the kernel representations of different visual representations regardless of their various dimensionality.

Given M different visual representations $X^{(1)}, X^{(2)}, \ldots, X^{(M)}$, we estimate their similarity matrices $W^{(1)}, W^{(2)}, \ldots, W^{(M)}$ with Eq. (1), respectively, and generate their respective kernel matrices $K^{(1)}, K^{(2)}, \ldots, K^{(M)}$ as described above. Then, we combine similarity and kernel matrices with their arithmetic averages:

$$\widetilde{W} = \frac{1}{M} \sum_{m=1}^{M} W^{(m)}, \tag{A.8}$$

and

$$\widetilde{K} = \frac{1}{M} \sum_{m=1}^{M} K^{(m)}. \tag{A.9}$$

Here we assume different visual representations contribute equally. Otherwise, any weighted fusion schemes in Yu et al. (2015) may directly replace our simple averaging-based fusion scheme from a computational perspective. However, the use of different weighted fusion algorithms may lead to considerably different performance. How to select a proper weighted fusion algorithm is non-trivial but not addressed in this paper.

By substituting W and X^l in Eq. (2) with $\widetilde{W}$ in Eq. (A.8) and $\widetilde{K}$ in Eq. (A.9), the projection P can be learned from multiple visual representations with the same bottom–up learning algorithm (c.f. Eqs. (2–5)). Applying the projection P to the kernel representation of any instance leads to its embedding in the latent space. Thus, we can embed all the training instances in X^l into the learned latent space by

$$Y^l = P^T \widetilde{K}^l, \tag{A.10}$$

where $\widetilde{K}^l$ is the combined kernel representation of training data X^l. For the same reason, the centralization and the l_2-normalization need to be applied to Y^l prior to the landmark generation and the top–down learning as presented in Sects. 3.2 and 3.3. As the joint use of multiple visual representations merely affects learning the projection P, the landmark generation and the top–down learning in our proposed framework keep unchanged in this circumstance.

After the bidirectional latent embedding learning, however, zero-shot recognition described in Sect. 3.4 has to be adapted for multiple visual representations accordingly. Given a test instance $\boldsymbol{x}_i^u$, its label is predicted in the latent space via the following procedure. First of all, its representation in the kernel space $\mathcal{K}$ is achieved by

$$\widetilde{K}_i^u = \left\{ \widetilde{k}\left(\boldsymbol{x}_i^u, \boldsymbol{x}_1^l\right), \widetilde{k}\left(\boldsymbol{x}_i^u, \boldsymbol{x}_2^l\right), \ldots, \widetilde{k}\left(\boldsymbol{x}_i^u, \boldsymbol{x}_{n_l}^l\right) \right\}^T, \tag{A.11}$$

where $\widetilde{k}(\cdot, \cdot)$ is the combined kernel function via the arithmetic averages of M kernel representations of this instance arising from its M different visual representations. Then we apply projection P to map it into the learned latent space:

$$\boldsymbol{y}_i^u = P^T \widetilde{K}_i^u. \tag{A.12}$$

After $\boldsymbol{y}_i^u$ is centralized and normalized in the same manner as done for all the training instances, its label, l^*, is assigned to the class label of which embedding is closest to $\boldsymbol{y}_i^u$; i.e.,

$$l^* = \arg\min_l d\left(\boldsymbol{y}_i^u, \boldsymbol{b}_l^u\right), \tag{A.13}$$

where $\boldsymbol{b}_l^u$ is the latent embedding of l-th unseen class, and $d(\boldsymbol{x}, \boldsymbol{y})$ is a distance metric in the latent space.

Appendix C: Visual Representation Complementarity Measurement and Selection

For the success in the joint use of multiple visual representations, diversity yet complementarity of multiple visual representations play a crucial role in zero-shot visual recognition. In this appendix, we describe our approach to measuring the complementarity between different visual representations and a complementarity-based algorithm used in finding complementary visual representations to maximize the performance, which has been used in our experiments.

C.1: The Complementarity Measurement

The complementarity of multiple visual representations have been exploited in previous works. Although those empiri-

cal studies, e.g., the results reported by Shao et al. (2016), strongly suggest that the better performance can be obtained by combining multiple visual representations in human action classification, little has been done on a quantitative complementarity measurement. To this end, we propose an approach to measuring the complementarity of visual representations based on the diversity of local distribution in a representation space.

First of all, we define the complementarity measurement of two visual representations $X^{(1)} \in \mathbb{R}^{d_1 \times n}$ and $X^{(2)} \in \mathbb{R}^{d_2 \times n}$, where d_1 and d_2 are the dimensionality of the two visual representations, respectively, and n is the number of instances. For each instance $\mathbf{x}_i$, $i = 1, 2, \ldots, n$, we denote its k nearest neighbours (kNN) in space $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$ by $\mathcal{N}_k^{(1)}(i)$ and $\mathcal{N}_k^{(2)}(i)$, respectively. To facilitate our presentation, we simplify our notation of $\mathcal{N}_k^{(m)}(i)$ to be $\mathcal{N}_i^{(m)}$. According to the labels of the instances in the kNN neighborhood, the set $\mathcal{N}_i^{(m)}$ can be divided into two disjoint subsets:

$$\mathcal{N}_i^{(m)} = \mathcal{I}_i^{(m)} \cup \mathcal{E}_i^{(m)}, \quad m = 1, 2, \quad i = 1, 2, \ldots, n$$

where $\mathcal{I}_i^{(m)}$ and $\mathcal{E}_i^{(m)}$ are the subsets that contain nearest neighbours of the same label as that of $\mathbf{x}_i$ and of different labels, respectively. Thus, we define the complementarity between representations $X^{(1)}$ and $X^{(2)}$ as follows:

$$c\left(X^{(1)}, X^{(2)}\right) = \frac{min\left(|\mathcal{I}^{(1)}|, |\mathcal{I}^{(2)}|\right) - |\mathcal{I}^{(1)} \cap \mathcal{I}^{(2)}|}{|\mathcal{I}^{(1)}| + |\mathcal{I}^{(2)}| - |\mathcal{I}^{(1)} \cap \mathcal{I}^{(2)}|},$$

(A.14)

where $\mathcal{I}^{(m)} = \cup_{i=1}^{n} \mathcal{I}_i^{(m)}$ for $m = 1, 2$, and $|\cdot|$ denotes the cardinality of a set. The value of c ranges from 0 to 0.5. Intuitively, the greater the value of c is, the higher complementarity between two representations is.

In the presence of more than two visual representations, we have to measure the complementarity between one and the remaining representations instead of another single one as treated in Eq. (A.14). Fortunately, we can extend the measurement defined in Eq. (A.14) to this general scenario. Without loss of generality, we define the complementarity between representation $X^{(1)}$ and a set of representations $S = \{X^{(2)}, \ldots, X^{(M)}\}$ as follows:

$$c(X^{(1)}, S) = \frac{min\left(|\mathcal{I}^{(1)}|, |\mathcal{I}^{2,\ldots,M}|\right) - |\mathcal{I}^{(1)} \cap \mathcal{I}^{2,\ldots,M}|}{|\mathcal{I}^{(1)}| + |\mathcal{I}^{2,\ldots,M}| - |\mathcal{I}^{(1)} \cap \mathcal{I}^{2,\ldots,M}|},$$

(A.15)

where $|\mathcal{I}^{2,\ldots,M}| = |\mathcal{I}^{(2)} \cup \mathcal{I}^{(3)} \cdots \cup \mathcal{I}^{(M)}|$. Thus, Eq. (A.15) forms a generic complementarity measurement for multiple visual representations.

C.2: Finding Complementary Visual Representations

Given a set of representations $\{X^{(1)}, X^{(2)}, \ldots, X^{(M)}\}$, we aim to select a subset of representations $S_{selected}$ where the complementarity between each element and another is as high as possible. Assume we already have a set $S_{selected}$ containing m complementary representations, and a set $S_{candidate}$ containing $M - m$ candidate representations, we can decide which representation in $S_{candidate}$ should be selected to join $S_{selected}$ by using the complementarity measurement defined in Eq. (A.15). In particular, we estimate the complementarity between each candidate representation and the set of all the representations in $S_{selected}$, and the one of highest complementarity is selected. The selection procedure terminates when a pre-defined condition is satisfied. For example, a pre-defined condition may be a maximum number of representations to be allowed in $S_{selected}$ or a threshold specified by a minimal value of complementarity measurement. The complementary representation selection procedure is summarized in Algorithm A.1.

Algorithm A.1 Finding Complementary Representations.

Input: $S_{candidate}$ and $S_{selected} = \emptyset$.
Output: $S_{selected}$.
Initialize: Compute the classification performance of each representation in $S_{candidate}$, and move the one with best performance from $S_{candidate}$ to $S_{selected}$.
1: **while** Termination condition is not satisfied **do**
2:    **for** Each candidate representation $X^m \in S_{candidate}$ **do**
3:       Compute $c(X^{(m)}, S_{selected})$.
4:    **end for**
5:    Select the $X^{(m)}$, with highest $c(X^{(m)}, S_{selected})$.
6:    Move the $X^{(m)}$ from $S_{candidate}$ to $S_{selected}$.
7: **end while**

C.3: Application in Zero-shot Human Action Recognition

Here, we demonstrate the effectiveness of our proposed approach to finding complementary visual representations for zero-shot human action recognition. We apply Algorithm A.1 to candidate visual representations ranging from handcrafted to deep visual representations on UCF101 and HMDB51. For the hand-crafted candidates, we choose the state-of-the-art *improved dense trajectory* (IDT) based representations. To distill the video-level representations, two different encoding methods, bag-of-features and Fisher vector, are employed to generate four different descriptors, HOG, HOF, MBHx and MBHy (Wang and Schmid 2013). Thus, there are a total of eight different IDT-based local representations. Besides, two global video-level representations, GIST3D (Solmaz et al. 2013) and STLPC (Shao et al. 2014), are also taken into account. For deep representations, we use

the C3D (Tran et al. 2015) representation. Thus, all the 11 different visual representations constitute the candidate set, $S_{candidate}$.

On UCF101 and HMDB51, we set the termination condition to be five visual representations at maximum in $S_{selected}$ in Algorithm A.1. Applying Algorithm A.1 to 11 candidate representations on two datasets leads to the same $S_{selected}$ consisting of C3D and four FV-based IDT representations. To verify this measured result, we use our bidirectional latent embedding framework working on incrementally added representations with the same settings described in Sect. 4. As illustrated in Figs. 4, 5 and 6, the performance of zero-shot human action recognition achieved in 30 trials is constantly improved as more and more selected representations are used, which suggests those selected representations are indeed
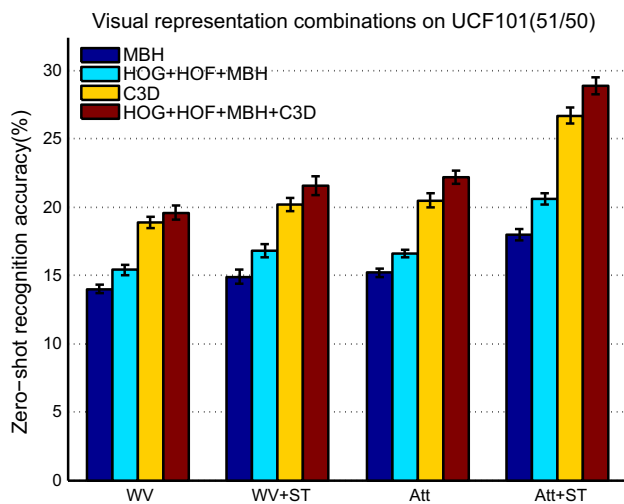

Fig. 6 Results regarding the joint use of multiple visual representations (mean and standard error) on HMDB51

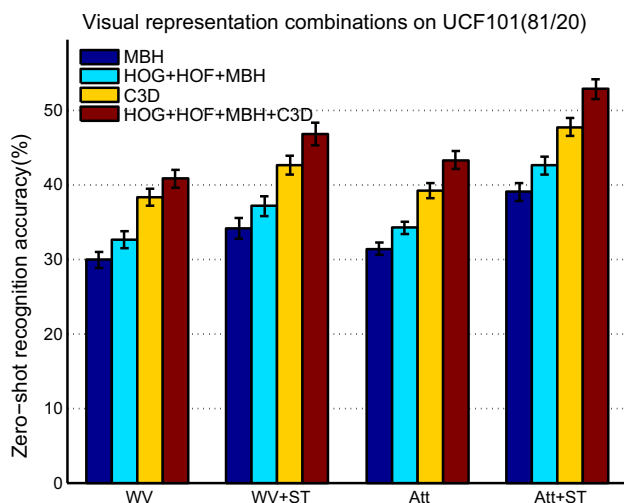**Fig. 4** Results regarding the joint use of multiple visual representations (mean and standard error) on UCF101 (51/50 split)

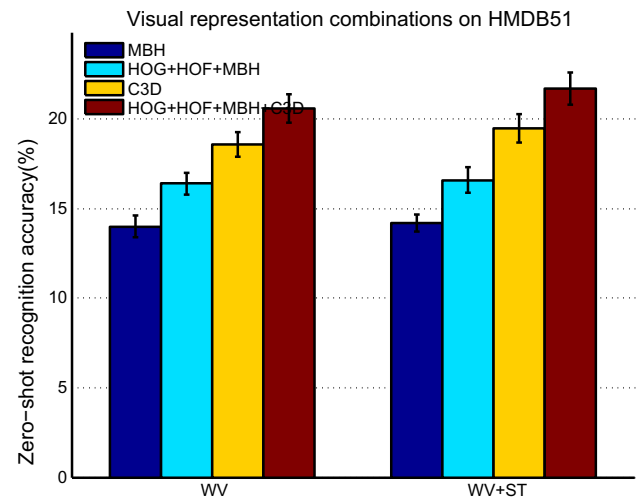complementary. In particular, the combination of the deep C3D representation and four IDT-based hand-crafted representations yields the best performance that is significantly better than that of using any single visual representations.

In conclusion, we anticipate that the technique presented in this appendix would facilitate the use of multiple visual representations in not only visual recognition but also other pattern recognition applications.

References

Akata, Z., Lee, H., & Schiele, B. (2014). *Zero-shot learning with structured embeddings*. arXiv:1409.8403.

Akata, Z., Perronnin, F., Harchaoui, Z., & Schmid, C. (2013). Label-embedding for attribute-based classification. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 819–826).

Akata, Z., Perronnin, F., Harchaoui, Z., & Schmid, C. (2016). Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*, 1425–1438.

Akata, Z., Reed, S., Walter, D., Lee, H., & Schiele, B. (2015). Evaluation of output embeddings for fine-grained image classification. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2927–2936).

Al-Halah, Z., & Stiefelhagen, R. (2015). How to transfer? Zero-shot object recognition via hierarchical transfer of semantic attributes. In *IEEE winter conference on applications of computer vision (WACV)* (pp. 837–843). IEEE.

Andreopoulos, A., & Tsotsos, J. K. (2013). 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding*, *117*, 827–891.

Cai, D., He, X., & Han, J. (2007). Semi-supervised discriminant analysis. In *International conference on computer vision* (pp. 1–7). IEEE.

Changpinyo, S., Chao, W.-L., Gong, B., & Sha, F. (2016a). Synthesized classifiers for zero-shot learning. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Changpinyo, S., Chao, W.-L., & Sha, F. (2016b). *Predicting visual exemplars of unseen classes for zero-shot learning*. arXiv:1605.08151.

Fig. 5 Results regarding the joint use of multiple visual representations (mean and standard error) on UCF101 (81/20 split)

Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. In *British machine vision conference (BMVC)*.

Cheng, J., Liu, Q., Lu, H., & Chen, Y.-W. (2005). Supervised kernel locality preserving projections for face recognition. *Neurocomputing*, *67*, 443–449.

Cox, T. F., & Cox, M. A. (2000). *Multidimensional scaling*. Boca Raton: CRC press.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press.

Dinu, G., Lazaridou, A., & Baroni, M. (2015). Improving zero-shot learning by mitigating the hubness problem. In *International conference on learning representations workshop*.

Elhoseiny, M., Elgammal, A., & Saleh, B. (2015). Tell and predict: Kernel classifier prediction for unseen visual classes from unstructured text descriptions. In *IEEE conference on computer vision and pattern recognition (CVPR) workshop on language and vision*.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Mikolov, T. et al. (2013). Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems* (pp. 2121–2129).

Fu, Y., Hospedales, T. M., Xiang, T., & Gong, S. (2015). Transductive multi-view zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*, 2332–2345.

Fu, Y., & Huang, T. (2010). Manifold and subspace learning for pattern recognition. *Pattern Recognition and Machine Vision*, *6*, 215.

Gan, C., Lin, M., Yang, Y., Zhuang, Y., & Hauptmann, A. G. (2015). Exploring semantic inter-class relationships (SIR) for zero-shot action recognition. In *Twenty-ninth AAAI conference on artificial intelligence*.

Gan, C., Yang, T., & Gong, B. (2016). Learning attributes equals multi-source domain generalization. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Gong, Y., Ke, Q., Isard, M., & Lazebnik, S. (2014). A multi-view embedding space for modeling internet images, tags, and their semantics. *International Journal of Computer Vision*, *106*, 210–233.

Griffin, G., Holub, A., & Perona, P. (2007). Caltech-256 object category dataset. Technical report 7694. California Institute of Technology. http://www.vision.caltech.edu/Image_Datasets/Caltech256/.

Hardoon, D. R., Szedmak, S., & Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, *16*, 2639–2664.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778).

Jayaraman, D., & Grauman, K. (2014). Zero-shot recognition with unreliable attributes. In *Advances in neural information processing systems* (pp. 3464–3472).

Jiang, Y.-G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., & Sukthankar, R. (2014). *THUMOS challenge: Action recognition with a large number of classes*. http://crcv.ucf.edu/THUMOS14/.

Jolliffe, I. (2002). *Principal component analysis*. Hoboken: Wiley Online Library.

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3128–3137).

Kodirov, E., Xiang, T., Fu, Z., & Gong, S. (2015). Unsupervised domain adaptation for zero-shot learning. In *IEEE international conference on computer vision (ICCV)* (pp. 2452–2460).

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). HMDB: a large video database for human motion recognition. In *IEEE international conference on computer vision (ICCV)* (pp. 2556–2563). IEEE.

Lampert, C. H., Nickisch, H., & Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer.

In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 951–958). IEEE.

Lampert, C. H., Nickisch, H., & Harmeling, S. (2014). Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*, 453–465.

Liu, J., Kuipers, B., & Savarese, S. (2011). Recognizing human actions by attributes. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3337–3344). IEEE.

Mensink, T., Gavves, E., & Snoek, C. (2014). COSTA: Co-occurrence statistics for zero-shot classification. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2441–2448).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Niyogi, X. (2004). Locality preserving projections. In *Neural information processing systems* (Vol. 16, p. 153). MIT.

Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G. S., & Dean, J. (2014). Zero-shot learning by convex combination of semantic embeddings. In *International conference on learning representations (ICLR)*.

Peng, X., Wang, L., Wang, X., & Qiao, Y. (2016). Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, *150*, 109–125.

Radovanović, M., Nanopoulos, A., & Ivanović, M. (2010). Hubs in space: Popular nearest neighbors in high-dimensional data. *The Journal of Machine Learning Research*, *11*, 2487–2531.

Reed, S., Akata, Z., Schiele, B., & Lee, H. (2016). Learning deep representations of fine-grained visual descriptions. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Romera-Paredes, B., & Torr, P. (2015). An embarrassingly simple approach to zero-shot learning. In *International conference on machine learning (ICML)* (pp. 2152–2161).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211–252.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, *18*, 401–409.

Shao, L., Liu, L., & Yu, M. (2016). Kernelized multiview projection for robust action recognition. *International Journal of Computer Vision*, *118*, 115–129.

Shao, L., Zhen, X., Tao, D., & Li, X. (2014). Spatio-temporal laplacian pyramid coding for action recognition. *IEEE Transactions on Cybernetics*, *44*, 817–827.

Shigeto, Y., Suzuki, I., Hara, K., Shimbo, M., and Matsumoto, Y. (2015). Ridge regression, hubness, and zero-shot learning. In *Machine learning and knowledge discovery in databases* (pp. 135–151). Springer.

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems* (pp. 568–576).

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International conference on learning representations*.

Smola, A., & Vapnik, V. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems*, *9*, 155–161.

Solmaz, B., Assari, S. M., & Shah, M. (2013). Classifying web videos using a global video descriptor. *Machine Vision and Applications*, *24*, 1473–1485.

Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper

with convolutions. In *IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In *International conference on computer vision (ICCV)* (pp. 4489–4497).

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*, 1453–1484.

Vedaldi, A., & Lenc, K. (2015). Matconvnet—Convolutional neural networks for matlab. In *ACM international conference on multimedia*.

Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset. Technical report CNS-TR-2010-001. California Institute of Technology. http://www.vision.caltech.edu/visipedia/CUB-200-2011.html.

Wang, H., & Schmid, C. (2013). Action recognition with improved trajectories. In *IEEE international conference on computer vision (ICCV)* (pp. 3551–3558). IEEE.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., & Van Gool, L. (2016). Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision (ECCV)*.

Wu, Z., Jiang, Y.-G., Wang, X., Ye, H., Xue, X., & Wang, J. (2016). Multi-stream multi-class fusion of deep networks for video classification. In *ACM multimedia (ACM MM)*.

Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., & Schiele, B. (2016). Latent embeddings for zero-shot classification. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Xu, X., Hospedales, T., & Gong, S. (2015a). Semantic embedding space for zero-shot action recognition. In *IEEE international conference on image processing (ICIP)* (pp. 63–67). IEEE.

Xu, X., Hospedales, T., & Gong, S. (2015b). *Zero-shot action recognition by word-vector embedding*. arXiv:1511.04458.

Yu, M., Liu, L., & Shao, L. (2015). *Kernelized multiview projection*. arXiv:1508.00430.

Zhang, H., Deng, W., Guo, J., & Yang, J. (2010). Locality preserving and global discriminant projection with prior information. *Machine Vision and Applications*, *21*, 577–585.

Zhang, Z., & Saligrama, V. (2015). Zero-shot learning via semantic similarity embedding. In *IEEE international conference on computer vision (ICCV)* (pp. 4166–4174).

Zhang, Z., & Saligrama, V. (2016a). Zero-shot learning via joint latent similarity embedding. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 6034–6042).

Zhang, Z., & Saligrama, V. (2016b). Zero-shot recognition via structured prediction. In *European conference on computer vision* (pp. 533–548). Springer.

Zhao, S., Liu, Y., Han, Y., & Hong, R. (2015). *Pooling the convolutional layers in deep convnets for action recognition*. arXiv:1511.02126.

Zheng, Z., Yang, F., Tan, W., Jia, J., & Yang, J. (2007). Gabor feature-based face recognition using supervised locality preserving projection. *Signal Processing*, *87*, 2473–2483.