



Transferrable Feature and Projection Learning with Class Hierarchy for Zero-Shot Learning

Aoxue Li¹ · Zhiwu Lu² · Jiechao Guan² · Tao Xiang³ · Liwei Wang¹ · Ji-Rong Wen²

Received: 12 October 2018 / Accepted: 12 May 2020 / Published online: 3 June 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Zero-shot learning (ZSL) aims to transfer knowledge from seen classes to unseen ones so that the latter can be recognised without any training samples. This is made possible by learning a projection function between a feature space and a semantic space (e.g. attribute space). Considering the seen and unseen classes as two domains, a big domain gap often exists which challenges ZSL. In this work, we propose a novel inductive ZSL model that leverages superclasses as the bridge between seen and unseen classes to narrow the domain gap. Specifically, we first build a class hierarchy of multiple superclass layers and a single class layer, where the superclasses are automatically generated by data-driven clustering over the semantic representations of all seen and unseen class names. We then exploit the superclasses from the class hierarchy to tackle the domain gap challenge in two aspects: deep feature learning and projection function learning. First, to narrow the domain gap in the feature space, we define a recurrent neural network over superclasses and then plug it into a convolutional neural network for enforcing the superclass hierarchy. Second, to further learn a transferrable projection function for ZSL, a novel projection function learning method is proposed by exploiting the superclasses to align the two domains. Importantly, our transferrable feature and projection learning methods can be easily extended to a closely related task—few-shot learning (FSL). Extensive experiments show that the proposed model outperforms the state-of-the-art alternatives in both ZSL and FSL tasks.

Keywords Zero-shot learning · Class hierarchy · Recurrent neural network · Deep feature learning · Projection function learning · Few-shot learning

1 Introduction

In the past 5 years, deep neural network (DNN) based models (Huang et al. 2017; Donahue et al. 2014) have achieved

super-human performance on the ILSVRC 1K recognition task. However, most existing object recognition models, particularly those DNN-based ones, require hundreds of image samples to be collected for each object class; many of the object classes are rare and it is thus extremely hard, sometimes impossible to collect sufficient training samples, even with social media. One way to bypass the difficulty in collecting sufficient training data for object recognition is zero-shot learning (ZSL) (Frome et al. 2013; Rohrbach et al. 2013; Lampert et al. 2014; Akata et al. 2015; Zhang and Saligrama 2016b; Chao et al. 2016). The goal of ZSL is to recognise a new/unseen class without any training samples from the class. A ZSL model typically assumes that each class name is embedded in a semantic space. With this semantic space and a visual feature space representing the appearance of an object in an image, it chooses a joint embedding space (which is often defined directly with the semantic space) and learns a projection function so that both the visual features and the semantic vectors are embedded in the same space. Under the inductive ZSL setting, the projection function is learned

Communicated by Cristian Sminchisescu.

✉ Zhiwu Lu
zhiwu.lu@gmail.com

Aoxue Li
lax@pku.edu.cn

Tao Xiang
t.xiang@surrey.ac.uk

- ¹ The Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
- ² The Beijing Key Laboratory of Big Data Management and Analysis Methods, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China
- ³ The Department of Electrical and Electronic Engineering, University of Surrey, Guildford, Surrey GU2 7XH, UK

only with the seen class training samples and then directly used to project the unseen class samples to the joint embedding space. The class label of a test unseen-class sample is assigned to the nearest unseen class prototype.

One of the biggest challenges in ZSL is the domain gap between the seen and unseen classes. As mentioned above, the projection functions learned from the seen classes are applied to the unseen class data. However, the unseen classes are often visually very different from the seen ones; therefore the domain gap between the seen and unseen class domains can be big, meaning that the same projection function may not be able to project an unseen class image to be close to its corresponding class name in the joint embedding space for correct recognition.

To tackle the domain gap challenge, most previous works focus on learning more transferrable projection functions. Specifically, many ZSL models resort to transductive learning (Ye and Guo 2017; Guo et al. 2016; Kodirov et al. 2015), where the unlabeled samples from unseen classes are used to learn the projection functions. However, such an approach assumes the access to all unlabelled test samples beforehand, and thus deviates from the motivation of ZSL: the unseen class samples are rare. These transductive ZSL models are thus limited in their practicality in real-world scenarios. Moreover, other than projection function learning, deep feature learning has been largely overlooked in ZSL. Most previous works, if not all, simply use visual features extracted by convolutional neural network (CNN) models pretrained on ImageNet (Russakovsky et al. 2015). We argue that deep feature learning and projection function learning should be equally important in ZSL.

In this paper, we propose a novel inductive ZSL model that leverages the superclasses as the bridge between seen and unseen classes to narrow the domain gap. The idea is simple: if an unseen class falls into a superclass that contains one or more seen classes, the task of recognising it becomes easier because we now have training samples at superclass-level. Exploiting the shared superclass among seen and unseen classes thus provides a means for narrowing the domain gap. In this work, we generate the superclasses by a data-driven approach, without the need of a human-annotated taxonomy. Specifically, we construct a tree-structured class hierarchy that consists of multiple superclass layers and a single class layer. In this tree-structured hierarchy, we take both seen classes and unseen classes as the leaves, and generate the superclasses by clustering over the semantic representations of all seen and unseen class names (see the orange box in Fig. 1). Moreover, by exploiting the superclasses from the class hierarchy, the proposed model aims to narrow the domain gap in two aspects: deep feature learning and projection function learning.

To learn transferrable visual features, we propose a novel deep feature learning model based on the superclasses from

the class hierarchy. Specifically, a CNN model is first designed for both class and superclass classification. Since the seen and unseen classes may have the same superclasses, directly training this CNN model enables us to learn transferrable deep features. To further strengthen the feature transferability, we explicitly encode the hierarchical structure among classes/superclasses into the CNN model, by plugging recurrent neural network (RNN) (Graves et al. 2013; Zheng et al. 2015; Liu et al. 2017) components into the network. Overall, a novel CNN-RNN architecture is designed for transferrable deep feature learning, as illustrated in the green box in Fig. 1.

To learn a transferrable projection function for inductive ZSL, we propose a novel projection function learning method by utilising the superclasses to align the two domains, as shown in the blue box in Fig. 1. Specifically, we formulate the projection function learning on each superclass layer as a graph regularised self-reconstruction problem. An efficient iterative algorithm is developed as the solver. The results of multiple superclass layers are combined to boost the performance of projection function learning on the single class layer.

Importantly, our model can be easily extended to a closely related problem—few-shot learning (FSL) (Fei-Fei et al. 2006; Lake et al. 2013; Vinyals et al. 2016; Snell et al. 2017; Finn et al. 2017). Specifically, the above transferrable feature and projection learning methods involved in our model are employed to solve the FSL problem without any modification. Experiments on the widely-used mini-ImageNet dataset (Snell et al. 2017) show that our model achieves the state-of-the-art results. This suggests that the class hierarchy is also important for narrowing down the domain gap in the FSL problem.

Our contributions are: (1) a novel inductive ZSL model is proposed to align the seen and unseen class domains by utilising the superclasses shared across the two domains. To our best knowledge, this is the first time that the superclasses generated by data-driven clustering have been leveraged in both feature learning and projection learning to narrow the domain gap for inductive ZSL. (2) Due to the domain alignment using the superclasses from the class hierarchy, we have created the new state-of-the-art for ZSL and FSL.

2 Related Work

2.1 Semantic Space

Three semantic spaces are typically used for ZSL: the attribute space (Zhang and Saligrama 2015; Guo et al. 2016), the word vector space (Frome et al. 2013; Norouzi et al. 2014), and the textual description space (Reed et al. 2016; Fu and Sigal 2016). The attribute space is the most widely-

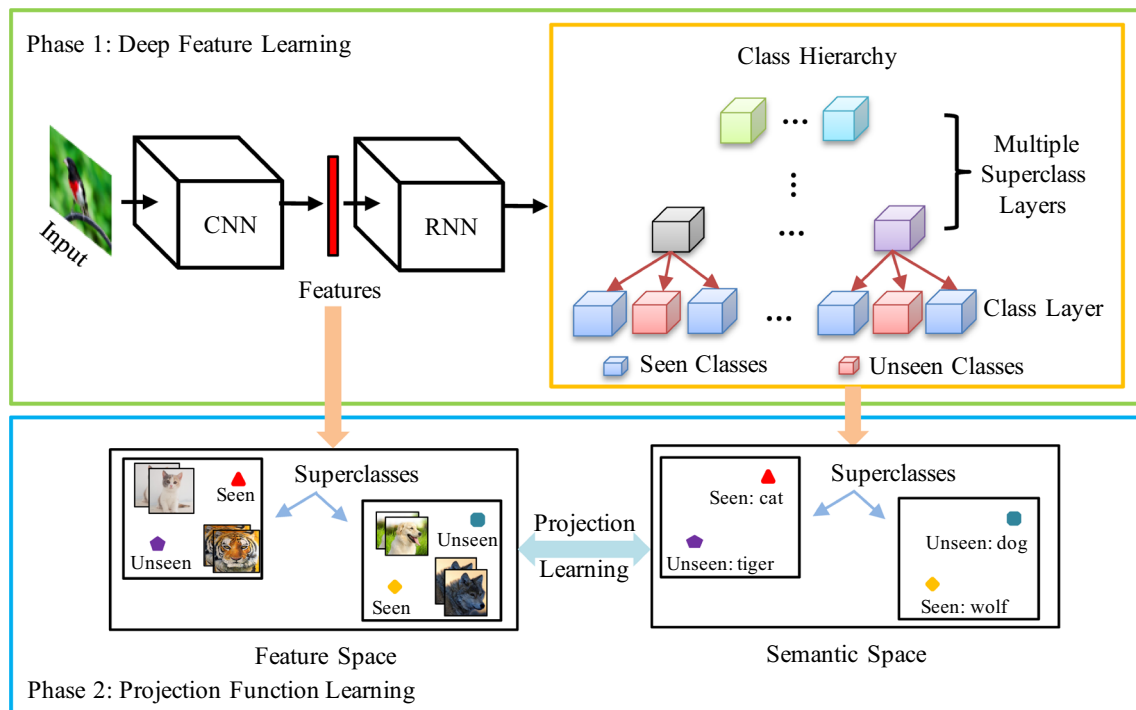


Fig. 1 Overview of the proposed model for inductive ZSL with class hierarchy (including deep feature learning and projection function learning) (Color figure online)

used semantic space. However, for large-scale problems, annotating attributes for each class becomes very difficult. Recently, the semantic word-vector space has begun to be popular especially in large-scale problems (Fu et al. 2015b; Kodirov et al. 2015), since no manually defined ontology is required and any class name can be represented as a word vector for free. In addition, the semantic space can also be created by directly learning from textual descriptions of categories such as Wikipedia articles (Fu and Sigal 2016) and sentence descriptions (Reed et al. 2016).

2.2 Inductive ZSL

Existing inductive ZSL models can be grouped into two categories depending on whether the unseen class semantic representations are used implicitly or explicitly during model training: (1) inductive ZSL with implicit exploitation of unseen class labels: no unseen class samples are used at the training stage; the side information about the unseen classes is semantic (i.e., seen and unseen class names are represented in the same semantic space), but unseen class labels represented in the semantic space (i.e. class prototypes) are not used directly during training. If attributes are used to form the semantic space for ZSL (Kankuekul et al. 2012; Lampert et al. 2014), it is assumed that the unseen classes are defined by the same set of attributes. If word vectors are used (Frome et al. 2013; Socher et al. 2013), the word embed-

ding is typically learned using human knowledge bases about both classes, e.g., Wikipedia pages of the unseen classes are also used for word embedding learning. (2) Inductive ZSL with explicit exploitation of unseen class labels: similarly as before—no unseen class samples and assuming the sharing of same semantic spaces. But this time, the unseen class labels are used explicitly in training the ZSL model. ZSL with semantic class prototype graph (Fu et al. 2018; Li et al. 2017) and ZSL with unseen class data synthesis (Bucher et al. 2017; Xian et al. 2018; Zhu et al. 2018) fall into this category. In this work, we focus on the second setting, i.e., inductive ZSL with explicit unseen class label exploitation.

2.3 Projection Learning

ZSL is typically made possible by learning a projection function that aligns the visual feature and semantic spaces. Existing projection learning models fall into three groups: (1) the first group of models learn a projection function from a visual feature space to a semantic space (i.e. in a forward projection direction) by employing conventional regression/ranking models (Lampert et al. 2014; Akata et al. 2015) or deep neural network regression/ranking models (Socher et al. 2013; Frome et al. 2013; Lei Ba et al. 2015). (2) The second group of models choose the reverse projection direction (Shigeto et al. 2015; Kodirov et al. 2015; Shojaei and Baghshah 2016; Zhang et al. 2017), i.e., from

the semantic space to the feature space, to alleviate the hubness problem suffered by nearest neighbour search in a high dimensional space (Radovanović et al. 2010). (3) The third group of models learn an intermediate space as the embedding space, where both the feature space and the semantic space are projected to (Lu 2015; Zhang and Saligrama 2016a; Changpinyo et al. 2016). As a combination of the first and second groups, the proposed model integrates both forward and reverse projections for ZSL, which is similar to Kodirov et al. (2017). However, it differs from Kodirov et al. (2017) in two aspects (see Sect. 3.4): (1) the extra use of superclass-level semantics based on the class hierarchy; (2) the extra update of the image-level semantic representations via Laplacian regularization.

2.4 Projection Domain Shift

To overcome the projection domain shift (Fu et al. 2015a) caused by the domain gap in ZSL, transductive ZSL has been proposed to learn the projection function with not only labelled data from seen classes but also unlabelled data from unseen classes. According to whether the predicted labels of the test images are iteratively used for model learning, existing transductive ZSL models can be divided into two groups: (1) the first group (Fu et al. 2015a; Rohrbach et al. 2013; Ye and Guo 2017) first constructs a graph and then the domain gap is reduced by label propagation (Wang and Zhang 2008). A variant is the structured prediction model (Zhang and Saligrama 2016b) which utilises a Gaussian parameterisation of the unseen class domain label predictions. (2) The second group (Guo et al. 2016; Kodirov et al. 2015; Li et al. 2015; Shojaei and Baghshah 2016; Wang and Chen 2017; Yu et al. 2017) involves using the predicted labels of the unseen class data in an iterative model update/adaptation process as in self-training (Xu et al. 2015, 2017). However, these transductive ZSL models assume the access to all unlabelled test samples, which is often invalid in the context of ZSL because new classes typically appear dynamically and unavailable before model learning. Instead of assuming the access to all test unseen-class data for transductive learning, our model is developed based on inductive learning, and it resorts to learning visual features and projection function only with the seen class data to counter the projection domain shift.

2.5 ZSL with Superclasses

Although hierarchical detection and classification have been widely studied (Bo et al. 2011; Redmon and Farhadi 2017), little attention has been paid to ZSL with superclasses. There are only two exceptions. One is Hwang and Sigal (2014), which learns the relation between attributes and superclasses for semantic embedding, and the other is Akata et al. (2015) which utilises superclasses to define a seman-

tic space for ZSL. However, the superclasses used in these works are obtained from the manually-defined hierarchical taxonomy, which needs additional cost to collect. In this paper, our model is more flexible by generating superclasses automatically with data-driven clustering over semantic representations of all seen/unseen class names. In addition, the superclasses are not induced into the feature learning in Akata et al. (2015); Hwang and Sigal (2014).

2.6 ZSL with Feature Learning

Most existing ZSL models extract visual features by using CNNs pretrained on ImageNet. This is to assume that the feature extraction model will generalise equally well for seen and unseen classes. This assumption is clearly invalid, particularly under the recently proposed ‘pure’ ZSL setting (Xian et al. 2017), where the unseen classes have no overlapping with the ImageNet 1K classes used to train the feature extraction model. The only notable exception is Long et al. (2018) which uses the central loss and fine-tunes ImageNet pretrained deep model on seen classes. However, without any transferrable learning module, the feature extraction model proposed in Long et al. (2018) has no guarantee of generalising well to unseen classes.

2.7 Label Correlation Modeling

Although no existing ZSL work has considered modeling label correlations to address the domain gap issue, the idea of using the relationship of class labels to improve multi-label classification has been exploited (Deng et al. 2014; Zheng et al. 2015; Liu et al. 2017). The label relationship has been modeled as hierarchy and exclusion graph (HEX) (Deng et al. 2014), conditional random field (CRF) (Zheng et al. 2015), or RNN (Liu et al. 2017). Our model differs in two aspects: (1) it is designed to learn transferrable deep features and projection functions for ZSL, rather than making more accurate label prediction; and (2) the label correlation is guided by a class hierarchy rather than exhaustive as in CRF (Zheng et al. 2015) and defined by the label frequency as in RNN (Liu et al. 2017).

2.8 Few-Shot Learning

Few-shot learning (FSL) (Fei-Fei et al. 2006; Lake et al. 2013; Vinyals et al. 2016; Snell et al. 2017; Finn et al. 2017; Sung et al. 2018) aims to recognise novel classes from very few labelled examples. Such label scarcity issue challenges the standard fine-tuning strategy used in deep learning. Data augmentation can alleviate the label scarcity issue, but cannot solve it. The latest FSL approaches thus choose to transform the deep network training process to meta learning where the transferrable knowledge is learned in the form of

good initial conditions, embeddings, or optimisation strategies (Finn et al. 2017; Sung et al. 2018). In this paper, we directly employ our transferrable feature learning and projection learning methods to solve FSL task. Experimental results in Sect. 4.2 demonstrate that, similar to the ZSL task, our transferrable feature learning and projection learning methods can also achieve state-of-the-art results in the FSL task.

3 Methodology

3.1 Problem Definition

We first formally define the ZSL problem as follows. Let $S = \{s_1, \dots, s_p\}$ denote the set of seen classes and $U = \{u_1, \dots, u_q\}$ denote the set of unseen classes, where p and q are the total numbers of seen classes and unseen classes, respectively. These two sets of classes are disjoint, i.e. $S \cap U = \emptyset$. Similarly, $\bar{Z}_s = [\bar{z}_1^{(s)}, \dots, \bar{z}_p^{(s)}]_{d_z \times p}$ and $\bar{Z}_u = [\bar{z}_1^{(u)}, \dots, \bar{z}_q^{(u)}]_{d_z \times q}$ denote the corresponding semantic representations (e.g. d_z -dimensional attribute vector) of seen and unseen classes. The training set is denoted as $D_s = \{(I_i, y_i, z_i) : i = 1, \dots, N_s\}$, where I_i denotes the i -th image in the training set, y_i denotes its corresponding label w.r.t. S , $z_i = \bar{z}_{y_i}^{(s)}$ denotes its d_z -dimensional semantic representation, and N_s denotes the total number of training images. The test set is denoted as $D_u = \{(I_j, y_j, z_j) : j = 1, \dots, N_u\}$, where I_j denotes the j -th image in the test set, y_j denotes its corresponding unknown label w.r.t. U , $z_j = \bar{z}_{y_j}^{(u)}$ denotes its d_z -dimensional semantic representation, and N_u denotes the total number of test images. We thus have the training seen-class semantic matrix $Z_s = [z_i]_{N_s \times d_z}$. The goal of inductive ZSL is to predict the unseen-class sample labels $\{y_j : j = 1, \dots, N_u\}$ using a model trained only with D_s . In a generalised ZSL setting, the test samples come from both seen and unseen classes, and the ZSL problem becomes more realistic yet more challenging.

3.2 Class Hierarchy Construction

In this section, we describe how to construct a tree-structured class hierarchy using a data-driven approach based on k-means clustering. The class hierarchy consists of multiple superclass layers and one single class layer. Starting from the leaf class nodes (i.e. both seen and unseen classes), we obtain the nodes in the upper layer by clustering the nodes in the lower layer, forming a tree-structured class hierarchy (see the red dashed box in Fig. 2). We thus obtain a set $R = \{r_l : l = 1, \dots, n_r\}$ that collects the number of clusters at each superclass layer, where r_l denotes the number of clusters in the l -th superclass layer, and n_r denotes the total number of superclass layers in the class hierarchy.

l ($l = 1, \dots, n_r$) denotes the index-number of each superclass layer, and $l = 0$ denotes the index number of the class layer. Each class label y_i can be mapped to its corresponding superclasses $V_i = \{v_i^l : l = 1, \dots, n_r\}$, where v_i^l denotes the superclass label of y_i in the l -th superclass layer. This results in multiple labels at different class/superclass levels for the image I_i . More concretely, at each layer of the tree-structured hierarchy, we cluster the nodes by their semantic representations. Each cluster then forms a parent node (i.e. a superclass) in the upper layer of the tree. In this way, the semantic representation of the superclass is, in a sense, a mixing of the semantics of its children classes. Since the superclass labels are shared across both seen and unseen class domains, the proposed model can help to overcome the domain gap challenge for ZSL. Note that the performance of the proposed model is shown to be robust (when multiple trials are repeated), although k-means is used for class hierarchy construction.

3.3 Deep Feature Learning

In this section, we describe our deep feature learning model, which is trained only using the seen class data D_s (along with the class hierarchy constructed using semantic representations of all seen and unseen names) but expected to represent well the unseen class data D_u . In this paper, we propose a CNN-RNN model defined with superclasses from the aforementioned class hierarchy to learn transferrable visual features for unseen class samples. This deep feature learning model takes an image I_i as the input and then outputs a d_f -dimensional feature vector. Concretely, we extend a CNN model by two steps for predicting the superclass-level labels, using the shared CNN generated features. The first step is to predict the labels at different class/superclass levels (see the yellow dashed box in Fig. 2), so that the shared superclasses at those layers can make the learned features suitable for representing the unseen classes. The second step is to encode the hierarchical structure of class/superclass layers into superclass label prediction. That is, we infer each superclass label by considering the prediction results of the same or lower class/superclass layers (see the blue dashed box in Fig. 2). We are essentially learning a hierarchical classifier—the features learned need to be useful for not only recognising the leaf-level classes, but also the higher level classes/superclasses, which are shared with the unseen classes. This makes sure that the learned features are relevant to the unseen classes, even without using any samples from those classes. In this work, we propose to exploit the hierarchical structure using an RNN model. When the RNN is combined with the CNN, we obtain a CNN-RNN model for deep feature learning. We will give its technical details in the following.

For the first class/superclass label prediction step, we add $n_r + 1$ parallel fully-connected (FC) layers along with soft-

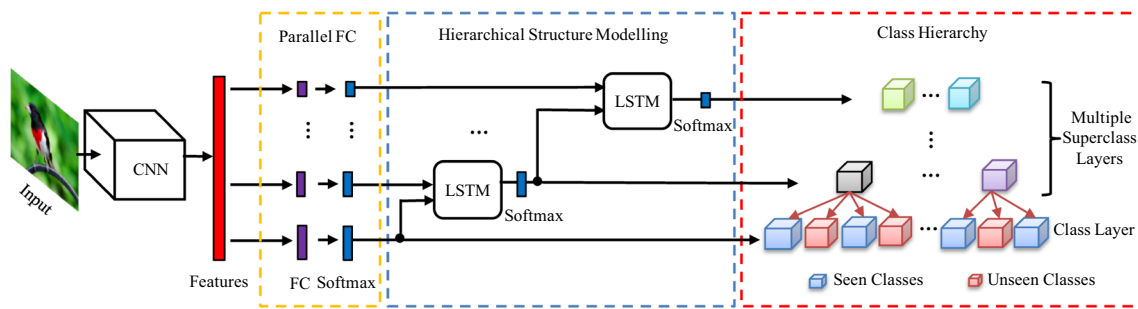


Fig. 2 Overview of the proposed CNN-RNN model for deep feature learning with three-layer class hierarchy (Color figure online)

max layers on top of the CNN model for feature extraction, as shown in the yellow dashed box in Fig. 2. Given an object sample, each fully-connected layer with softmax thus predicts the probability distribution of the corresponding-level classes (i.e. class/superclass-level). Further, we introduce an RNN to model the hierarchical structure between layers and integrate this label-relation encoding module with the label prediction steps. More specifically, we model the hierarchical structure among multiple class/superclass layers by n_r Long-Short-Term-Memory (LSTM) networks. Note that the ‘bottom-up’ (or ‘top-down’) strategy used for modeling the hierarchical structure is up to whether the label inference becomes more difficult in the bottom-up (or top-down) direction. It feels intuitive that the label inference becomes more difficult in the top-down direction, with more superclasses/classes to be recognised. However, this is not the truth for our class hierarchy, because it is constructed (layer by layer) with k-means clustering and the imbalance problem becomes more serious in the bottom-up direction. By focusing on the imbalance problem in label inference, we thus choose the ‘bottom-up’ strategy in the following.

In this paper, to model the hierarchical structure among classes/superclasses in the leaf class layer and the first l ($l = 1, \dots, n_r$) superclass layers (from bottom, near the leaf class layer), we also employ a two-time-step LSTM network which predicts the labels corresponding to the l -th superclass layer. Concretely, in the first time step, we feed the output of the previous LSTM as input, since the previous LSTM has fused all information of the first $l - 1$ superclass layers and the leaf class layer. In the next time step, the output of the current superclass-level fully-connected layer (with softmax) is used as input and the hidden cell predicts the probability distribution of the current superclass-level labels (see the blue dashed box in Fig. 2). We thus provide the formal formulation of the LSTM network as follows:

$$\begin{aligned} [\tilde{h}_i^{l-1}, \tilde{c}_i^{l-1}] &= \begin{cases} \text{LSTM}^l(p_i^0, h_i^0, c_i^0), & l = 1 \\ \text{LSTM}^l(\hat{p}_i^{l-1}, h_i^{l-1}, c_i^{l-1}), & l = 2, \dots, n_r \end{cases} \\ [h_i^l, c_i^l] &= \text{LSTM}^l(p_i^l, \tilde{h}_i^{l-1}, \tilde{c}_i^{l-1}) \end{aligned} \quad (1)$$

where $\text{LSTM}^l(\cdot, \cdot, \cdot)$ is a forward step of the LSTM unit corresponding to the l -th superclass level. The other notations are defined as follows. \hat{p}_i^l ($l = 1, \dots, n_r$) denotes the superclass probability distribution of the i -th training image predicted by the LSTM unit corresponding to the l -th superclass level: $\hat{p}_i^l = \text{softmax}(\hat{W}^l \cdot h_i^l + \hat{b}^l)$, with \hat{W}^l and \hat{b}^l being the weight and bias of the output layer. p_i^l ($l = 1, \dots, n_r$) denotes the output of the fully-connected layer corresponding to the l -th superclass level for the i -th training image, and p_i^0 denotes the output of the fully-connected layer corresponding to the class level for the i -th training image. h_i^{l-1} and c_i^{l-1} are the outputted hidden state and cell state of the previous LSTM network, respectively. h_i^0 and c_i^0 are the initial hidden state and cell state, respectively. \tilde{h}_i^{l-1} (or \tilde{h}_i^l) and \tilde{c}_i^{l-1} (or \tilde{c}_i^l) are the intermediate (or outputted) hidden state and cell state of the current LSTM network. We can obtain a total of n_r two-time-step LSTM networks to model the hierarchical structure among the superclasses/classes in the hierarchy. Each two-time-step LSTM and its single time step are illustrated in Fig. 3.

Although the Multi-Layer Perception (MLP) modules can be used instead of the LSTM modules, LSTM has one advantage in hierarchical structure modelling over MLP: the long-term context across multiple layers can be captured by LSTM, while this important information is hard for MLP to obtain. More specifically, each LSTM module is two-time-step, but its first time step takes the output of the last LSTM module as input and it thus integrates the outputs from all the lower layers. In this way, the whole of all LSTM modules can be regarded as a \tilde{t} -time-step LSTM model (where $\tilde{t} > 2$ denotes the total number of superclass layers and class layer), and thus the long short-term memory can be realized in our model (i.e., the long-term context across multiple layers can be exploited for deep feature learning).

Finally, by merging the hierarchical structure modelling with the original class label prediction, we define the loss function for the i -th training image I_i as follows:

$$\mathcal{L} = \mathcal{L}_{cls}(y_i, p_i^0) + \sum_{l=1}^{n_r} \lambda_l \mathcal{L}_{cls}(v_i^l, \hat{p}_i^l), \quad (2)$$

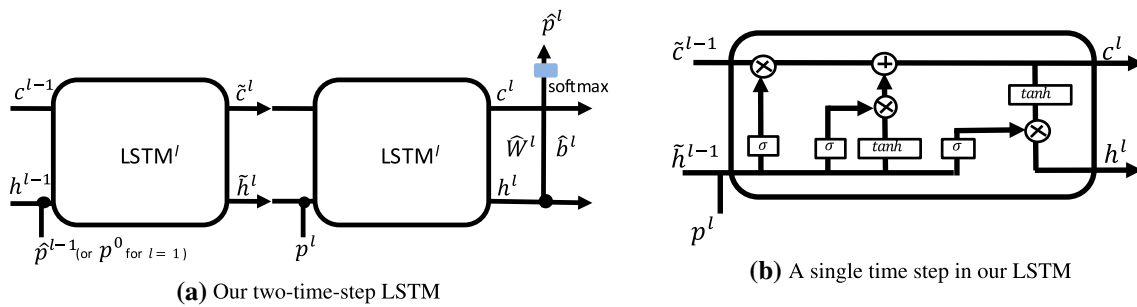


Fig. 3 **a** Illustration of our two-time-step LSTM. **b** Illustration of a single time step in our two-time-step LSTM. We omit the index of image to make the figures easy to understand. The σ and \tanh operators denote

the sigmoid and tanh activation functions, respectively. The \otimes and \oplus operators denote element-wise multiplication and addition, respectively

where \mathcal{L}_{cls} is the cross-entropy loss. The other notations are defined as follows. y_i denotes the ground-truth label of the image I_i at class level. v_i^l ($l = 1, \dots, n_r$) denotes the ground-truth label of the image I_i w.r.t. the l -th superclass. λ_l ($l = 1, \dots, n_r$) denotes the weight for the loss w.r.t. the l -th superclass. In this work, we empirically set $\lambda_l = 1$. Note that the LSTM modules in our model are only trained during model learning (but not for label prediction at test time), and thus they are only used for feature learning.

3.4 Projection Function Learning

Once the feature extraction model is learned using the training seen-class data, it can be used to extract visual features from both training and test images and then compute the two feature matrices F_s and F_u for the seen and unseen class domains, respectively. Each row of F_s (or F_u) is the learned visual features of a training sample (or test sample). With the seen-class feature matrix F_s , we propose a novel projection learning method which exploits the superclasses from the class hierarchy to align the seen and unseen class domains. Since the superclasses are shared across the two domains, the proposed method enables us to learn a transferrable projection function for inductive ZSL. We give the details of our method below.

3.4.1 Projection Learning over Superclasses

As mentioned in Sect. 3.2, each class label y_i can be mapped to its corresponding superclasses $V_i = \{v_i^l : l = 1, \dots, n_r\}$, where v_i^l denotes the superclass label of y_i in the l -th superclass layer of the class hierarchy. Note that each superclass label is just an index-number, represented with the real-valued cluster centroid vector (i.e. semantic vector). Let e_i^l denote the d_z -dimensional semantic vector of the superclass label v_i^l . We thus obtain n_r training superclass semantic matrices $\{E_s^l = [e_i^l]_{N_s \times d_z}, l = 1, \dots, n_r\}$, where each matrix

collects the semantic vectors of superclasses in the corresponding layer for all seen class samples.

Given the training seen-class feature matrix $F_s \in \mathbb{R}^{N_s \times d_f}$ and the initial training superclass-semantic matrix $E_s^l \in \mathbb{R}^{N_s \times d_z}$, we model the set of seen-class images as a graph $\mathcal{G} = \{\mathcal{A}, V\}$ with its vertex set $V = F_s$ and weight matrix $\mathcal{A} = \{a_{uv}\}$, where a_{uv} denotes the similarity between visual features of the u -th and v -th seen class images (i.e. f_u and f_v). It should be noted that the weight matrix \mathcal{A} is usually assumed to be nonnegative and symmetric. We thus compute the normalised Laplacian matrix \mathcal{L} of the graph \mathcal{G} by the following equation:

$$\mathcal{L} = I - D^{-1/2} \mathcal{A} D^{-1/2} \quad (3)$$

where I is an $N_s \times N_s$ identity matrix, and D is an $N_s \times N_s$ diagonal matrix with its u -th diagonal entry $\sum_v a_{uv}$.

Based on the well-known Laplacian regularisation and superclass-semantic representations, our projection learning method solves the following optimisation problem with respect to the l -th superclass layer:

$$\begin{aligned} \min_{W^l, \tilde{E}_s^l} L(W^l, \tilde{E}_s^l) \\ = \|F_s W^l - \tilde{E}_s^l\|_F^2 + \mu^l \|F_s - \tilde{E}_s^l W^{lT}\|_F^2 \\ + \epsilon^l \text{tr}(\tilde{E}_s^{lT} \mathcal{L} \tilde{E}_s^l) + \nu^l \|\tilde{E}_s^l - E_s^l\|_F^2 + \eta^l \|W^l\|_F^2 \end{aligned} \quad (4)$$

where $W^l \in \mathbb{R}^{d_f \times d_z}$ is the projection matrix between the superclass-semantic representation w.r.t. the l -th superclass layer and the visual feature representation, and $\tilde{E}_s^l \in \mathbb{R}^{N_s \times d_z}$ is the updated superclass-semantic matrix during model learning. Note that the image-level semantic representations are the same per superclass in E_s^l . Our motivation of introducing the ‘updated’ semantic representations in \tilde{E}_s^l is that the image-level semantic representations per superclass can be learned (now not the same mean vector) and the obtained better semantic representations are expected to improve the performance of projection learning.

In the above objective function, the first two terms $\|F_s W^l - \tilde{E}_s^l\|_F^2$ and $\|F_s - \tilde{E}_s^l W^{lT}\|_F^2$ aim to learn bidirectional linear projections between F_s and \tilde{E}_s^l , which in fact is a self-reconstruction task and has been verified to have good generalisation ability. Moreover, the third term $\text{tr}(\tilde{E}_s^{lT} \mathcal{L} \tilde{E}_s^l)$ denotes the well-known graph regularisation, which can enforce the superclass-semantic representations \tilde{E}_s^l to preserve the graph locality of image features F_s and thus benefit the feature self-reconstruction. The fourth term $\|\tilde{E}_s^l - E_s^l\|_F^2$ is a fitting constraint between \tilde{E}_s^l and E_s^l , meaning the image-level semantic representations should be close to their superclass prototype. The last term $\|W^l\|_F^2$ is the Frobenius norm used to regularise W^l . These terms are weighed by the four regularisation parameters $\mu^l, \epsilon^l, \nu^l, \eta^l$. Although self-reconstruction is also considered in Kodirov et al. (2017), the proposed model has two main differences: (1) the extra use of superclass-level semantics based on the class hierarchy; (2) the extra update of the image-level semantic representations via Laplacian regularization.

We then develop an efficient approach to tackle the graph regularised self-reconstruction problem in Eq. (4). Specifically, we solve Eq. (4) by alternately optimising the following two subproblems:

$$W^{l*} = \arg \min_{W^l} L(W^l, \tilde{E}_s^{l*}) \quad (5)$$

$$\tilde{E}_s^{l*} = \arg \min_{\tilde{E}_s^l} L(W^{l*}, \tilde{E}_s^l) \quad (6)$$

Here, \tilde{E}_s^{l*} is initialised with E_s^l . Taking a convex quadratic formulation, each of the above two subproblems has a global optimal solution. The two solvers are given below.

For the first subproblem, with $\tilde{E}_s^l = \tilde{E}_s^{l*}$ fixed, the solution of $\arg \min_{W^l} L(W^l, \tilde{E}_s^{l*})$ can be found by setting $\frac{\partial L(W^l, \tilde{E}_s^{l*})}{\partial W^l} = 0$. We thus obtain a linear equation:

$$(F_s^T F_s + \eta^l I) W^l + \mu^l W^l (E_s^{l*})^T E_s^{l*} = (1 + \mu^l) F_s^T E_s^{l*} \quad (7)$$

Let $\alpha^l = \mu^l / (1 + \mu^l) \in (0, 1)$ and $\gamma^l = \eta^l / (1 + \mu^l)$. In this paper, we empirically set $\gamma^l = 0.01$. We have:

$$[(1 - \alpha^l) F_s^T F_s + \gamma^l I] W^l + W^l (\alpha^l (E_s^{l*})^T E_s^{l*}) = F_s^T E_s^{l*} \quad (8)$$

which is a Sylvester equation. Since $(1 - \alpha^l) F_s^T F_s + \gamma^l I \in \mathbb{R}^{d_f \times d_f}$ and $\alpha^l (E_s^{l*})^T E_s^{l*} \in \mathbb{R}^{d_z \times d_z}$ ($d_f, d_z \ll N_s$), this equation can be solved efficiently by the Bartels–Stewart algorithm (Bartels and Stewart 1972). In MATLAB, it is simply implemented with the function ‘sylvester’.

For the second subproblem, with $W^l = W^{l*}$ fixed, the solution of $\arg \min_{\tilde{E}_s^l} L(W^{l*}, \tilde{E}_s^l)$ can be found by setting

Algorithm 1 Projection Learning over Superclasses

Input: Training seen-class feature matrix F_s
Initial training superclass-semantic matrix E_s^l
Parameters $\alpha^l, \beta^l, \epsilon^l$
Output: W^{l*}
1. Set $\tilde{E}_s^{l*} = E_s^l$;
2. Construct a graph \mathcal{G} and compute \mathcal{L} with Eq. (3);
repeat
3. Find the best solution W^{l*} by solving Eq. (8);
4. Updating \tilde{E}_s^l with W^{l*} by solving Eq. (10);
until a stopping criterion is met

$\frac{\partial L(W^{l*}, \tilde{E}_s^l)}{\partial \tilde{E}_s^l} = 0$. We thus have:

$$\begin{aligned} \tilde{E}_s^l (\mu^l (W^{l*})^T W^{l*} + (1 + \nu^l) I) + \epsilon^l \mathcal{L} \tilde{E}_s^l \\ = (1 + \mu^l) F_s W^{l*} + \nu^l E_s^l \end{aligned} \quad (9)$$

Let $\beta^l = 1 / (1 + \nu^l) \in (0, 1)$. We obtain:

$$\begin{aligned} \tilde{E}_s^l [\alpha^l \beta^l (W^{l*})^T W^{l*} + (1 - \alpha^l) I] + \epsilon^l \mathcal{L} \tilde{E}_s^l \\ = \beta^l F_s W^{l*} + (1 - \alpha^l) (1 - \beta^l) E_s^l \end{aligned} \quad (10)$$

which is a Sylvester equation. Since $\alpha^l \beta^l (W^{l*})^T W^{l*} + (1 - \alpha^l) I \in \mathbb{R}^{d_z \times d_z}$ ($d_z \ll N_s$), this equation can be solved efficiently by the Bartels–Stewart algorithm. Importantly, the time complexity of solving Eqs. (8) and (10) is linear with respect to the number of samples. Given that the proposed algorithm is shown to converge very quickly, it is efficient even for large-scale problems.

To sum up, we outline the proposed projection function learning method in Algorithm 1. As illustrated in Fig. 4, once the best projection function is learned, we first project the superclass prototypes corresponding to the l -th superclass layer into the feature space. We then perform nearest neighbour search over the likely superclasses (obtained from the higher-level superclass layer as described in Sect. 3.4.2) in the feature space to predict the superclass labels corresponding to the l -th superclass layer for test samples. Finally, we generate the set of the most likely unseen class labels for each test sample (see Algorithm 2) using the superclass labels obtained from the first superclass layer. This candidate set can improve the final zero-shot recognition over unseen classes.

Note that we do not choose to utilise the CNN-RNN model proposed in Sect. 3.3 to replace the above method for predicting the superclass labels of test samples, because *this model would be not trained¹ and thus fail² over some superclasses*

¹ Keeping in mind that each superclass has inherited training image samples only from seen classes, we find that our CNN-RNN model is not trained over some superclasses that only contain unseen classes.

² Our CNN-RNN model can still extract transferrable features for these test unseen-class samples because the corresponding unseen classes share higher-level superclasses with some seen classes.

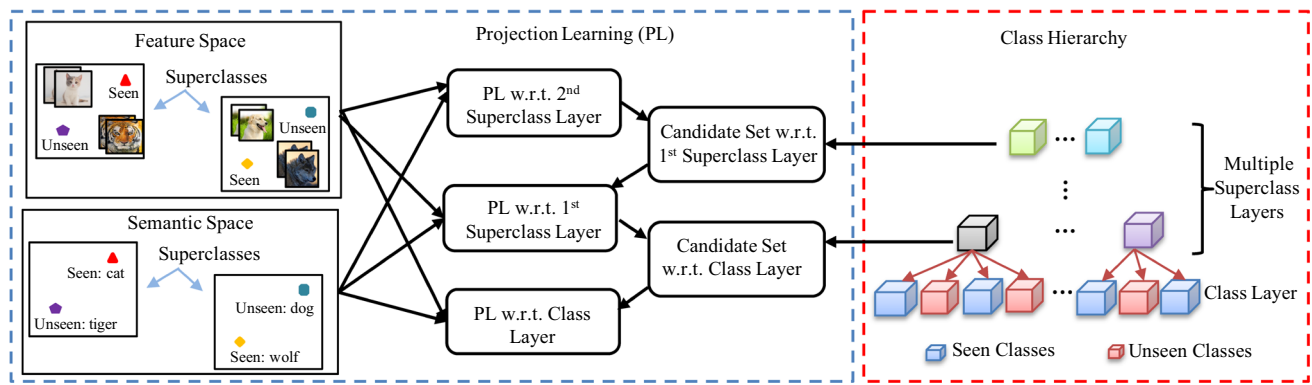


Fig. 4 Overview of the label inference model by our proposed projection learning (PL) with three-layer class hierarchy

Algorithm 2 Full ZSL Algorithm

Input: Training set D_s (excluding features)

Test set D_u (excluding features and labels)

Semantic class prototypes \tilde{Z}_s and \tilde{Z}_u

Parameters $\lambda_l, \alpha^l, \beta^l, \epsilon^l, \alpha^c, \beta^c, \epsilon^c$

series Output: Labels of test samples

series Class Hierarchy Construction:

1. Construct the tree-structured class hierarchy as in Sect. 3.2;

series Deep Feature Learning:

2. Train a CNN-RNN model according to Eq. (2);

3. Compute feature matrices F_s for the seen class domain;

series Projection Function Learning:

4. Solve Eq. (4) for predicting the superclass labels using Algorithm 1;

5. Generate the set of the most likely unseen class labels for each test sample;

6. Solve Eq. (11) with an algorithm similar to Algorithm 1;

7. Predict the unseen class label of each test sample.

that only contain unseen classes. In contrast, our projection learning method addresses this issue by projecting the superclass semantic representations to the feature space and then performing nearest neighbor search over these superclass prototypes for label inference.

3.4.2 Full ZSL Algorithm

The results of the proposed projection learning method can be used to infer the labels of test unseen-class samples as follows. First, we start at the top of the class hierarchy and have $l = n_r, \dots, 1$ (where l denotes the index-number of each superclass layer). Given the l -th superclass layer, we predict the top-3 superclass labels of each test sample I_j using the optimal projection matrix learned by Algorithm 1. Second, derived from the obtained top-3 superclass labels of I_j , we generated the set of the most likely superclass labels of I_j for the $l-1$ -th superclass layer by collecting all the superclasses contained by the top-3 superclass labels at the l -th superclass layer. After that, we use the projection learning method in Algorithm 1 to select the top-3 superclass at the

$l-1$ -th superclass layer. In the same way, the set of the most likely unseen class labels $\mathcal{N}(I_j)$ can be acquired for the single class/leaf layer (i.e. $l = 0$). Finally, we learn the projection function using seen class samples to infer the labels of test samples by solving:

$$\begin{aligned} \min_{W, \tilde{Z}_s} L(W, \tilde{Z}_s) \\ = \|F_s W - \tilde{Z}_s\|_F^2 + \mu^c \|F_s - \tilde{Z}_s W^T\|_F^2 \\ + \epsilon^c \text{tr}(\tilde{Z}_s^T \mathcal{L} \tilde{Z}_s) + \nu^c \|\tilde{Z}_s - Z_s\|_F^2 + \eta^c \|W\|_F^2 \end{aligned} \quad (11)$$

where $W \in \mathbb{R}^{d_f \times d_z}$ is the projection matrix, and $\tilde{Z}_s \in \mathbb{R}^{N_s \times d_z}$ is the updated training seen-class semantic matrix during model learning. To solve this optimisation problem, we can develop a solver similar to Algorithm 1 (also need to tune parameters α^c, β^c and ϵ^c as Algorithm 1) and the only difference is that the training superclass-semantic matrix is replaced by the class-level semantic matrix.

When the best projection matrix is learned, we can project the class-level semantic prototypes from the test set into the feature space. The nearest neighbor search is then performed (over the set of the most likely unseen class labels $\mathcal{N}(I_j)$) in the feature space to predict the label of a test image. In summary, the projection learning over superclasses (in Sect. 3.4.1) is exploited for generating the candidate set for zero-shot recognition over test samples. Note that the ablative results in Table 5 show that the performance with only class-level semantics is improved with 2.6% by adding superclass-level semantics (see ‘PL w. Class Semantic’ vs. ‘Our PL Model’).

Different from existing projection learning methods that mainly rely on the class-level semantics for label inference, we exploit both class-level and superclass-level semantics to recognise the unseen class samples in this paper. Since the superclasses in our hierarchy are shared across the seen and unseen-class domains, our projection learning method can alleviate the projection domain shift and thus benefit the unseen-class image recognition (see ‘PL w. Class Semantic’

vs. ‘Our PL Model’ in Table 5). Finally, by combining class hierarchy construction, deep feature learning, and projection function learning together for inductive ZSL, our full algorithm is outlined in Algorithm 2.

3.5 Extension to Few-Shot Learning

Although the proposed model is originally designed for ZSL, it can be easily extended to FSL (Fei-Fei et al. 2006; Lake et al. 2013; Vinyals et al. 2016; Snell et al. 2017; Finn et al. 2017; Sung et al. 2018). Under a standard FSL setting, the dataset is split into three parts: a training set of many labelled base/seen class samples, a support set of few labelled novel/unseen class samples, and a test set of the rest novel class samples. In this work, we first construct a tree-structured class hierarchy using all base and novel class prototypes as in Sect. 3.2. With the obtained class hierarchy, we further train our full model (including deep feature learning and projection function learning) over the whole training set as before, and predict the labels of test samples as in Sect. 3.4.2. Specifically, we first train our CNN-RNN model with all training seen class samples from scratch. That is, the class hierarchy containing all base and novel classes is used to learn the CNN-RNN model, even for the 5-way 1-shot/5-shot FSL setting. Further, we adopt our projection learning method formulated in Eq. 4 to identify a set of candidates in FSL. After that, the projection learning method formulated in Eq. 11 is used to predict the labels of test novel class samples from the candidate set. To obtain better FSL results, we exploit both average visual features of few shot samples per novel class and the projected novel class prototypes for nearest neighbor search in the feature space. The experiments on mini-ImageNet show that our model can achieve promising results in the FSL task.

4 Experiments

4.1 Zero-Shot Learning

4.1.1 Datasets and Settings

Datasets Four widely-used benchmark datasets are selected. Three of them are of medium-size: Animals with Attributes (AwA) (Lampert et al. 2014), Caltech UCSD Birds (CUB) (Wah et al. 2011), and SUN Attribute (SUN) (Patterson et al. 2014). One large-scale dataset is ILSVRC2012/2010 (ImNet) (Russakovsky et al. 2015), where the 1000 classes of ILSVRC2012 are used as seen classes and 360 classes of ILSVRC2010 (not included in ILSVRC2012) are used as unseen classes, as in Fu and Sigal (2016). The details of these benchmark datasets are given in Table 1.

Table 1 Details of four benchmark datasets

Dataset	# instances	SS	SS-D	# seen/unseen
AwA	30,475	A	85	40/10
CUB	11,788	A	312	150/50
SUN	14,340	A	102	645/72
ImNet	218,000	W	1000	1000/360

‘SS’—semantic space, ‘SS-D’—the dimension of semantic space, ‘A’—attribute, and ‘W’—word vector

Semantic Space We use two types of semantic spaces. For the three medium-scale datasets, attributes are used as the semantic representations. For the large-scale ImNet dataset, the semantic word vectors are employed to form the semantic space. In this paper, we train a skip-gram text model on a corpus of 4.6M Wikipedia documents to obtain the word2vec (Norouzi et al. 2014) word vectors.

ZSL Settings (1) Pure ZSL: A new ‘pure’ ZSL setting (Xian et al. 2017) was proposed to overcome the weakness in the standard ZSL setting. Specifically, most recent ZSL models extract the visual features using ImageNet ILSVRC2012 1K class pretrained CNN models, but the unseen classes of the three medium-scale datasets in the standard splits may overlap with the 1K ImageNet classes. The zero-shot rule is thus violated. Under the new ZSL setting, new benchmark splits are provided to ensure that the unseen classes have no overlap with the ImageNet ILSVRC2012 1K classes. (2) Generalised ZSL: Another recently appearing setting is the generalised ZSL setting (Xian et al. 2017), under which the test set contains data samples from both seen and unseen classes. This setting is more suitable for real-world applications.

Evaluation Metrics (1) Pure ZSL: For the three medium-scale datasets, we compute average per-class top-1 accuracy as in Xian et al. (2017). For the large-scale ImNet dataset, the flat h@5 classification accuracy is computed as in Fu and Sigal (2016); Kodirov et al. (2017), where h@5 means that a test image is classified to a ‘correct label’ if it is among the top five labels. (2) Generalised ZSL: Three evaluation metrics are defined as follows: (1) acc_s —the accuracy of classifying the data samples from the seen classes to all the classes (both seen and unseen); (2) acc_u —the accuracy of classifying the data samples from the unseen classes to all the classes; (3) HM—the harmonic mean of acc_s and acc_u . The overall performance is evaluated with HM.

Compared Methods A wide range of existing ZSL models are selected for comparison. Under each ZSL setting, we focus on the recent and representative ZSL models that have achieved the state-of-the-art results.

4.1.2 Implementation Details

Class Hierarchy Construction In our feature learning model, the number of superclass layers and the number of super-

Table 2 Comparative accuracies (%) of pure ZSL

Model	Visual features	Inductive?	AwA	CUB	SUN	ImNet
CMT (Socher et al. 2013)	RES	Yes	39.5	34.6	39.9	–
DeViSE (Frome et al. 2013)	RES	Yes	54.2	52.0	56.5	12.8
DAP (Lampert et al. 2014)	RES	Yes	44.1	40.0	39.9	–
ConSE (Norouzi et al. 2014)	RES	Yes	45.6	34.3	38.8	15.5
SSE (Zhang and Saligrama 2015)	RES	Yes	60.1	43.9	51.5	–
SJE (Akata et al. 2015)	RES	Yes	65.6	53.9	53.7	–
ALE (Zhang and Saligrama 2016a)	RES	Yes	59.9	54.9	58.1	–
SynC (Changpinyo et al. 2016)	RES	Yes	54.0	55.6	56.3	–
SP-AEN (Chen et al. 2018)	RES	Yes	58.5	55.4	59.2	–
CVAE (Mishra et al. 2017)	VGG	Yes	71.4	52.1	61.7	24.7
SAE (Kodirov et al. 2017)	GOO	Yes	61.3	48.2	59.2	27.2
DEM (Zhang et al. 2017)	GOO	Yes	68.4	51.7	61.9	25.7
CLN + KRR (Long et al. 2018)	Feat-Learn	Yes	68.2	58.1	60.0	–
WGAN + ALE (Xian et al. 2018)	Feat-Learn	Yes	68.2	61.5	62.0	–
Ours	Feat-Learn	Yes	72.2	65.8	62.9	28.6

For ImNet, the hit@5 accuracy is used. Notations: ‘GOO’—GoogLeNet (Szegedy et al. 2015); ‘VGG’—VGG Net (Simonyan and Zisserman 2014); ‘RES’—ResNet (He et al. 2016); ‘Feat-Learn’—deep feature learning Best/highest result along each column are given in bold

Table 3 Comparative results (%) of generalised ZSL

Model	AwA			CUB			SUN		
	acc_s	acc_u	HM	acc_s	acc_u	HM	acc_s	acc_u	HM
CMT (Socher et al. 2013)	86.9	8.4	15.3	60.1	4.7	8.7	28.0	8.7	13.3
DeViSE (Frome et al. 2013)	68.7	13.4	22.4	53.0	23.8	32.8	27.4	16.9	20.9
SSE (Zhang and Saligrama 2015)	80.5	7.0	12.9	46.9	8.5	14.4	36.4	2.1	4.0
SJE (Akata et al. 2015)	74.6	11.3	19.6	59.2	23.5	33.6	30.5	14.7	19.8
LATEM (Xian et al. 2016)	71.7	7.3	13.3	57.3	15.2	24.0	28.8	14.7	19.5
ALE (Akata et al. 2016)	76.1	16.8	27.5	62.8	23.7	34.4	33.1	21.8	26.3
ESZSL (Romera-Paredes and Torr 2015)	75.6	6.6	12.1	63.8	12.6	21.0	27.9	11.0	15.8
SynC (Changpinyo et al. 2016)	87.3	8.9	16.2	70.9	11.5	19.8	43.3	7.9	13.4
SAE (Kodirov et al. 2017)	71.3	31.5	43.5	36.1	28.0	31.5	25.0	15.8	19.4
DEM (Zhang et al. 2017)	84.7	32.8	47.3	57.9	19.6	29.2	34.3	20.5	25.6
SP-AEN (Chen et al. 2018)	90.9	23.3	37.1	70.6	34.7	46.6	38.6	24.9	30.3
WGAN + ALE (Xian et al. 2018)	57.2	47.6	52.0	59.3	40.2	47.9	31.1	41.3	35.5
Ours	68.2	44.5	53.9	46.9	51.1	48.9	30.2	31.4	30.7

The *overall* performance is evaluated by the HM metric Best/highest result along each column are given in bold

classes at each layer are important hyperparameters for class hierarchy construction. As in Kodirov et al. (2017), our model selects these hyperparameter values by cross-validation on the seen class data: Firstly, the set of seen classes is split into training and validation data/classes; Secondly, the hyperparameters are tuned on the validation data; Finally, we fix the hyperparameters and train our feature learning model using all seen classes.

Deep Network Training In our CNN-RNN model, the first five groups of convolutional layers in VGG-16 Net

(Simonyan and Zisserman 2014) are used as the CNN subnet. The convolutional layers of this CNN subnet are pre-trained on ILSVRC 2012 (Russakovsky et al. 2015), while the other layers (including the LSTMs) are trained from scratch. Stochastic gradient descent (SGD) (LeCun et al. 1989) is used for model training with a base learning rate of 0.001. For those layers trained from scratch, their learning rates are 10 times of the base learning rate. The deep learning is implemented with Caffe (Jia et al. 2014).

Projection Function Learning Our projection function learning model has three groups of parameters to tune: $\{\alpha^l : l = 1, \dots, n_r\}$ [in Eq. (8)], $\{\beta^l : l = 1, \dots, n_r\}$ [in Eq. (10)], and $\{\epsilon^l : l = 1, \dots, n_r\}$ [in Eq. (10)]. In this paper, we select $\alpha^1, \beta^1, \epsilon^1$ by cross-validation on the training data as in Kodirov et al. (2017). Then, we directly use them in other superclass-level projection learning and class-level projection learning [in Eq. (11)].

4.1.3 Comparative Evaluation

Pure ZSL By following the same ‘pure’ ZSL setting in Xian et al. (2017), the overlapped ImageNet ILSVRC2012 1K classes are removed from the test set of unseen classes for the three medium-scale datasets, while the split of the ImNet dataset naturally satisfies the ‘pure’ ZSL setting. Table 2 presents the comparative results under the pure ZSL setting. It can be seen that: (1) our model yields the best results on all four datasets and achieves about 1–4% performance improvement over the best competitors. This validates the effectiveness of our model for this stricter ZSL setting. (2) On the large-scale ImNet dataset, our model achieves about 1–4% improvements over the state-of-the-art ZSL models (Mishra et al. 2017; Kodirov et al. 2017; Zhang et al. 2017), showing the scalability of our model for large-scale ZSL problems. (3) Our model clearly outperforms the state-of-the-art feature learning model (Xian et al. 2018), due to our transferrable feature and projection learning with the superclasses for zero-shot recognition.

Generalised ZSL We follow the same generalised ZSL setting of Xian et al. (2017). Table 3 provides the comparative results of generalised ZSL on the three medium-scale datasets. We can observe that: (1) different ZSL models take a different trade-off between acc_u and acc_s , and the overall performance is thus measured by the HM metric. (2) Our model achieves similar results on seen and unseen class domains while existing models favor one over the other. That is, our model has the strongest generalisation ability under this more challenging setting. (3) Our model clearly yields the best overall performance on AwA and CUB datasets, while is outperformed by the state-of-the-art model (Xian et al. 2018) on the SUN dataset. This is still very impressive, given that Xian et al. (2018) exploits a much superior CNN model (i.e. ResNet-101 He et al. 2016) for feature extraction.

4.1.4 Further Evaluations

Ablation Study on Key Components We compare our full ZSL model with a number of stripped-down versions to evaluate the effectiveness of the key components of our model. Three of such feature learning models are compared, each of which uses the same projection learning model (i.e. the baseline model defined by Eq. (5), where only class-level semantics

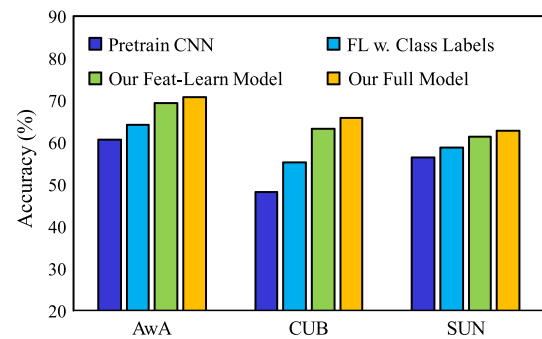


Fig. 5 Ablation study results on the three medium-scale datasets under the pure ZSL setting

are exploited) and differs only in how the visual features are obtained: ‘Pretrain CNN’—the same CNN model trained on ImageNet without any finetuning on seen class data; ‘FL w. Class Labels’—feature learning by finetuning with only class labels of seen class images; ‘Our Feat-Learn Model’: feature learning by the proposed model in Sect. 3.3. The ablation study results in Fig. 5 show that: (1) the transferrable feature learning with the class hierarchy leads to significant improvements (see Our Feat-Learn Model vs. FL w. Class Labels), ranging from 3 to 9%. This provides strong supports for our main contribution on deep feature learning for ZSL. (2) Our full model with extra transferrable projection learning achieves about 1–2% improvements (see Our Full Model vs. Our Feat-Learn Model), showing the effectiveness of our transferrable projection learning for inductive ZSL. (3) As expected, finetuning the CNN model can learn better deep features for ZSL, yielding 2–7% gains. However, the finetuned CNN model can not extract transferrable features and thus has very limited practicality in ZSL.

Qualitative Results and Analysis We give qualitative results to show why adding more components into our feature learning model benefits ZSL in the above ablation study. Figure 6 shows the tSNE visualisation of the visual features of test unseen-class samples. The visual features of test unseen-class samples are obtained by the feature learning methods used in the above ablation study, while the labels of test unseen-class samples are directly set as the ground-truth labels. It can be clearly seen that the visual features of the test samples become more separable when more components are added into our feature learning model, enabling us to obtain better recognition results.

In addition, some qualitative results are given to show how the proposed projection learning method benefits unseen class recognition. Figure 7 presents examples of candidate classes obtained by projection learning with superclasses. In this figure, the seen class candidates are in blue, unseen class candidates are in red, and the ground truth labels are in bold. According to Algorithm 1, our projection learning method finds the best predicted label for each test image by nearest

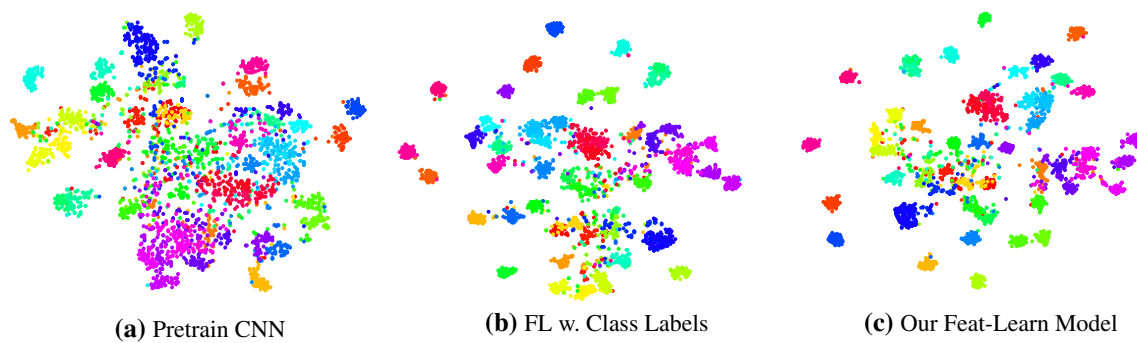


Fig. 6 The tSNE visualisation of the visual features of the test unseen-class samples from the CUB dataset. The visual features of test samples are obtained by the feature learning methods used in the ablation study,

while the labels (marked with different colors) of test samples are directly set as the ground-truth labels (Color figure online)


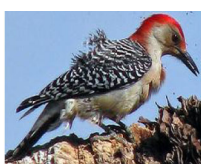






Unseen class images	Candidate Classes	Unseen class images	Candidate Classes
	Common tern ; California gull; Glaucous winged gull; Herring gull; Ivory gull; Ring billed gull ; Western gull; Red legged kittiwake; Artic tern ; Caspian tern; Laysan albatross; Elegant tern ; Forster's tern; Least tern ;		Red bellied woodpecker ; Least auklet; Pied kingfisher; Black and white warbler; Red cockaded woodpecker ; Crested auklet; Red headed woodpecker; Parakeet auklet; Downy woodpecker ; Horned puffin; Rose breasted grosbeak; Pigeon guillemot ; American three toed woodpecker
	Green kingfisher ; Sooty albatross; Black footed albatross; Eared grebe; Horned grebe; Western grebe ; Long tailed jaeger; Pacific loon; Pomarine jaeger ; Belted kingfisher; Ringed kingfisher ; Brown pelican; Hooded merganser; Black tern;		Cape May warbler ; Evening grosbeak; Yellow throated vireo; Hooded warbler ; Blue winged warbler ; Kentucky warbler; Magnolia warbler; Mourning warbler; Nashville warbler; Pine warbler ; Prairie warbler; Common yellowthroat
	Least flycatcher ; Mangrove cuckoo; Acadian flycatcher; Mockingbird; Great crested flycatcher; Palm warbler ; Olive sided flycatcher; Warbling vireo; Yellow bellied flycatcher ; Sayornis; Western wood pewee; Seaside sparrow ; Bank swallow; Red eyed vireo;		Tree sparrow ; Gray crowned rosy finch; American Pipit; Whip poor will; Baird sparrow ; Brewer sparrow; Chipping sparrow; Clay colored sparrow; House sparrow; Fieldsparrow; Grasshopper Sparrow; Harris Sparrow ; Henslow sparrow ; Vesper sparrow;
	Baltimore oriole ; Brewer blackbird; Eastern towhee; American redstart; Shiny cowbird ; Boat tailed grackle; Orchard oriole; White necked raven; Yellow headed blackbird; Bobolink; Hooded oriole; Scott oriole		Winter wren ; Chuck will widow; Fox sparrow; Lincoln sparrow; Savannah sparrow; Song sparrow ; Brown thrasher; Northern waterthrush; Carolina wren; House wren ; Cactus wren; Bewick wren ; Marsh wren; Wilson warbler

Fig. 7 Examples of candidate classes obtained by our ZSL model. The seen class candidates are in blue, unseen class candidates are in red, and the ground truth labels are in red bold. The number of candidate classes

for each unseen class sample is forced to become smaller by projection function learning with superclasses.

neighbor searching among unseen classes in several superclasses, rather than search among all unseen classes. This explains the superior performance of our projection learning method using superclasses.

Alternative Feature Learning Models To validate the effectiveness of the proposed feature learning model, we compare our feature learning model with several feature learning alternatives. Each of them is used to extract visual features

and the label inference is then achieved by the projection learning method proposed in Sect. 3.4. These feature learning alternatives respectively differ from our feature learning model in that: (1) ‘FL w. Seen’—all seen classes are used as leaves (i.e., class nodes) for hierarchy construction; (2) ‘FL w. MLP’—‘LSTM’ modules are replaced with three-layer Multi-Layer Perceptions for hierarchical structure modelling; (3) ‘FL w. Res-MLP’—‘LSTM’ modules

Table 4 Comparative results obtained by feature learning alternatives on the CUB dataset under the pure ZSL setting

No.	Feature learning alternatives	Accuracy (%)
1	‘FL w. Seen’	65.7
2	‘FL w. MLP’	63.8
3	‘FL w. Res-MLP’	64.5
4	‘FL w/o HSM’	62.6
5	‘FL w. Top-Down’	64.7
6	‘FL w. 1 time-step LSTMs’	63.9
7	‘FL w. 3 time-step LSTMs’	65.6
8	‘Our FL Model’	65.8

Best/highest result along each column are given in bold

are replaced with three-layer Multi-Layer Perceptions with residual connections for hierarchical structure modelling; (4) ‘FL w/o HSM’—hierarchical structure modelling is removed in feature learning (removing LSTM for Fig. 2); (5) ‘FL w. Top-Down’—for hierarchical structure modelling, the superclass-level layers and class-level layer are combined in a top-down manner: each LSTM predicts the superclass/class labels by combining the outputs of the FC layers corresponding to the current class/superclass layer and its upper superclass layer; (6) ‘FL w. 1 time-step LSTMs’—our two-time-step LSTM modules are replaced with one-time-step LSTM modules for hierarchical structure modelling; (7) ‘FL w. 3 time-step LSTMs’—our two-time-step LSTM modules are replaced with three-time-step LSTM modules for hierarchical structure modelling.

Table 4 presents the comparative results obtained by different feature learning models on the CUB dataset under the pure ZSL setting. It can be observed that: (1) our feature learning model (denoted as ‘Our FL Model’) achieves similar performance on the hierarchy constructed only with seen classes and the hierarchy constructed with all seen/unseen classes (see ‘FL w. Seen’ vs. ‘Our Full Model’). This indicates that a new unknown class label (prototype) can be directly coped with at test time, without any additional post-processing steps. (2) The proposed LSTM modules in the hierarchical structure modelling are shown to be effective for transferrable feature learning (see ‘FL w. MLP’ vs. ‘Our FL Model’, ‘FL w. Res-MLP’ vs. ‘Our FL Model’ and ‘FL w/o HSM’ vs. ‘Our FL Model’). (3) Our bottom-up layer combination strategy for hierarchical structure modelling is more suitable for transferrable feature learning than the top-down strategy (see ‘FL w. Top-Down’ vs. ‘Our FL Model’). (4) The proposed two-time-step LSTM module in the hierarchical structure modelling outperform the one-time-step LSTM module and achieve similar results w.r.t. the three-time-step LSTM module. This indicates that the two-time-step LSTM module is the most suitable architecture for our CNN-RNN model.

Table 5 Comparative results obtained by projection learning alternatives on the CUB dataset under the pure ZSL setting

No.	Projection learning alternatives	Accuracy(%)
1	‘PL: F to S’	58.2
2	‘PL: S to F’	60.4
3	‘PL: FS to I’	61.7
4	‘PL w. Class Semantic’	63.2
5	‘PL w/o. Laplacian’	64.2
6	‘PL w/o. Image-Level Semantic’	64.4
7	‘Our PL Model’	65.8

Best/highest result along each column are given in bold

Table 6 Comparative h@5 classification accuracies obtained by using the human-annotated/our hierarchy on the ImNet dataset

Model	Human-annotated	Ours
Our Feat-Learn Model	27.8	28.0
Our Full Model	28.3	28.6

Table 7 Comparative classification accuracies obtained by using the human-annotated/our hierarchy on the CUB dataset

Model	Human-annotated	Ours
Our Feat-Learn Model	64.6	64.7
Our Full Model	65.6	65.8

5) Overall, the comparative results indicate that our feature learning model is effective for ZSL.

Alternative Projection Learning Models To validate the effectiveness of the proposed projection learning model, we compare our projection learning model with six projection learning alternatives. Each of them is used for label inference given the visual features extracted by our transferrable feature learning model proposed in Sect. 3.3. The six projection learning alternatives respectively differ from our projection learning models in that: (1) ‘PL: F to S’—projection learning from the visual feature space to the semantic space; (2) ‘PL: S to F’—projection learning from the semantic space to the visual feature space; (3) ‘PL: FS to I’—learning to project the feature space and the semantic space into an intermediate space; (4) ‘PL w. Class Semantic’—projection learning with only class-level semantic representation according to Eq. (11), i.e. we simply perform the nearest neighbour search over all classes in the feature space; (5) ‘PL w/o. Laplacian’—projection learning without using Laplacian regularization term in Eqs. (4) and (11); (6) ‘PL w/o. Image-Level Semantic’—projection learning without updating the image-level semantic representation in Eqs. (4) and (11).

Table 5 provides the comparative results obtained by different projection learning models on the CUB dataset under

Table 8 Comparative results obtained by our feature learning model using the hierarchies with different numbers of superclass layers on the CUB dataset under the pure ZSL setting

No.	Hierarchy structure	Accuracy (%)
1	$n_r = 1, r_1 = 25$	63.2
2	$n_r = 2, r_1 = 33, r_2 = 5$	64.1
3	$n_r = 3, r_1 = 50, r_2 = 12, r_3 = 3$	65.8
4	$n_r = 4, r_1 = 100, r_2 = 50, r_3 = 25, r_4 = 12$	65.1

n_r —the total number of superclass layers; r_i —the number of superclasses in the i -th superclass layer
Best/highest result along each column are given in bold

Table 9 Comparative results obtained by our feature learning model using the hierarchies with different numbers of superclasses at each layer on the CUB dataset under the pure ZSL setting

No.	Hierarchy structure	Accuracy (%)
1	$n_r = 3, r_1 = 25, r_2 = 6, r_3 = 3$	65.1
2	$n_r = 3, r_1 = 50, r_2 = 12, r_3 = 3$	65.8
3	$n_r = 3, r_1 = 100, r_2 = 25, r_3 = 3$	65.3

The notations are exactly the same as in Table 8

Best/highest result along each column are given in bold

the pure ZSL setting. It can be observed that: (1) our bidirectional projection learning (denoted as ‘Our PL Model’) is shown to be more effective than the single directional projection learning and learning to project into an intermediate space (see ‘PL: F to S’ vs. ‘Our PL Model’, ‘PL: S to F’ vs. ‘Our PL Model’, and ‘PL: FS to I’ vs. ‘Our PL Model’). (2) Our superclass-level projection learning is shown to benefit the transferrable projection learning (see ‘PL w. Class Semantic’ vs. ‘Our PL Model’). (3) Both Laplacian regularization and image-level semantic representation updating are effective for our transferrable projection learning (see ‘PL w/o. Laplacian’ vs. ‘Our PL Model’, and ‘PL w/o. Image-Level Semantic’ vs. ‘Our PL Model’). Overall, the comparative results demonstrate the effectiveness of our projection learning model.

Human-Annotated Class Hierarchy The proposed ZSL model can be easily generalised to the human-annotated tree-structured class hierarchy (e.g. the biological taxonomy tree for animal classes, and the hierarchy tree of object classes provided by ImageNet). Concretely, we directly use seen/unseen classes as the leaf class layer and the higher-level classes are thus assigned as superclass layers (e.g. for animal classes, genus-level, family-level, and order-level layers can be used as superclass layers). Similarly, we can learn transferrable deep features by using the proposed feature learning model in Sect. 3.3. With the learned features, we can infer the labels of test unseen-class images by using the proposed projection learning method in Sect. 3.4.

Tables 6 and 7 give the comparative classification results with the human-annotated class hierarchy and our class hierarchy on the ImNet and CUB datasets under the pure ZSL setting, respectively. The human-annotated class hier-

archy used in the CUB dataset is the biological taxonomy tree collected from Wikipedia, while the human-annotated class hierarchy used in the ImNet dataset is collected by WordNet³ (Miller 1995). As shown in these two tables, our class hierarchy achieves better performance than the human-annotated class hierarchy. Importantly, the two groups of results obtained by repeating 10 trials have a p value < 0.05 (i.e. their difference is significant). This is reasonable because the human-annotated class hierarchy is prone to the superclass imbalance (i.e. the number of classes belonging to different superclasses varies greatly), while our class hierarchy often provides more balanced clustering results.

Hyperparameter Selection for Hierarchy Construction As we have mentioned in Sect. 4.1.2, our model selects the hyperparameters for hierarchy construction by cross-validation on the seen class data, as in Kodirov et al. (2017). For example, this strategy determines a 3-superclass-layer class hierarchy for the CUB dataset, where the three superclass layers respectively have 50, 12, and 3 superclasses. In the following, we provide experimental results to validate the effectiveness of the selected hierarchy structure.

First, to validate the effectiveness of the selected number of superclass layers, we construct four class hierarchies with different structures for comparison, and then train our feature learning model with these class hierarchies, as in Sec 3.3. In the same hierarchy, each cluster has similar numbers of nodes on average. Table 8 provides the comparative results obtained by our model using the four class hierarchies on the CUB dataset. It can be observed that the class hierarchy with 3 superclass layers yields the best results. Second, given the best number of superclass layers, we then conduct experiments to validate the selected number of superclasses at each superclass layer. Concretely, we first construct two additional class hierarchies with 3 superclass layers for comparison and then train our feature learning model with these two hierarchies. The comparative results obtained by our model with these hierarchies on the CUB dataset are given in Table 9. It can be seen that the selected number of superclass layers yields the best results. This indicates that the hyperparameter values obtained by cross-validation are indeed optimal.

³ The class hierarchy is available at <http://www.image-net.org>.

Table 10 Comparative 5-way accuracies (%) with 95% confidence intervals for FSL on the mini-ImageNet dataset

Model	CNN	1 shot	5 shot
Nearest Neighbor	Simple	41.8 ± 0.70	51.04 ± 0.65
Matching Net (Vinyals et al. 2016)	Simple	43.56 ± 0.84	55.31 ± 0.73
Meta-Learn LSTM (Ravi and Larochelle 2016)	Simple	43.33 ± 0.77	60.60 ± 0.71
MAML (Finn et al. 2017)	Simple	48.70 ± 1.84	63.11 ± 0.92
Prototypical Net (Snell et al. 2017)	Simple	49.42 ± 0.78	68.20 ± 0.66
mAP-SSVM (Triantafillou et al. 2017)	Simple	50.32 ± 0.80	63.94 ± 0.72
Relation Net (Sung et al. 2018)	Simple	50.44 ± 0.82	65.32 ± 0.70
SNAIL (Mishra et al. 2016)	ResNet20	55.71 ± 0.99	68.88 ± 0.92
PPA (Qiao et al. 2018)	Simple	54.53 ± 0.40	67.87 ± 0.20
PPA (Qiao et al. 2018)	WRN	59.60 ± 0.41	73.74 ± 0.19
Ours	Simple	53.92 ± 0.63	65.13 ± 0.61
Ours	WRN	57.14 ± 0.78	74.45 ± 0.73

Best/highest result along each column are given in bold

4.2 Few-Shot Learning

4.2.1 Dataset and Settings

To directly compare our method to the standard few-shot learning methods (Snell et al. 2017; Qiao et al. 2018; Finn et al. 2017; Sung et al. 2018; Triantafillou et al. 2017; Mishra et al. 2017; Vinyals et al. 2016), we further apply it to FSL on mini-ImageNet as in Snell et al. (2017). This dataset consists of 100 ImageNet classes, which is significantly smaller than the large-scale ImNet. The semantic space is formed as in Sect. 4.1.1, while the visual features are extracted with two CNN models trained from scratch with the training set of mini-ImageNet: (1) simple—four conventional blocks as in Snell et al. (2017); (2) WRN—wide residual networks (Zagoruyko and Komodakis 2016) as in Qiao et al. (2018). The 5-way accuracy is computed by randomly selecting 5 classes from the 20 novel classes for each test trial, and the average accuracy over 600 test trials is used as the evaluation metric.

4.2.2 Comparative Evaluation

The comparative 5-way accuracies on mini-ImageNet are given in Table 10. We can make the following observations: (1) our model achieves state-of-the-art performance under the 5-way 5-shot setting and competitive results under 5-way 1-shot. This suggests that our model is effective not only for ZSL but also for FSL. (2) Stronger visual features extracted for FSL yield significantly better results, when PPA and our model are concerned.

5 Conclusion

In this paper, we tackle the domain gap challenge in ZSL by leveraging superclasses as the bridge between seen and unseen classes. We first build a tree-structured class hierarchy that consists several superclass layers and one single class layer. Then, we exploit the superclasses to overcome the domain gap challenge in two aspects: deep feature learning and projection function learning. In the deep feature learning phase, we introduce a recurrent neural network (RNN) defined with the superclasses from the class hierarchy into a convolutional neural network (CNN). A novel CNN-RNN model is thus proposed for transferrable deep feature learning. In the projection function learning phrase, we align the seen and unseen class domains using the superclasses from the class hierarchy. Extensive experiments on four benchmark datasets show that the proposed ZSL model yields significantly better results than the state-of-the-art alternatives. Furthermore, the proposed model can be extended to few-shot learning and also achieves promising results. In our ongoing research, we will implement our current projection learning method using a deep autoencoder framework and then connect it to the CNN-RNN model so that the entire network can be trained in an end-to-end manner. Moreover, we also expect that our CNN-RNN model can be generalised to other ZSL-related vision problems such as multi-label ZSL and large-scale FSL.

Acknowledgements This work was supported in part by National Key R&D Program of China (2018YFB1402600), National Natural Science Foundation of China (61976220, 61573026, 61832017), and Beijing Natural Science Foundation (L172037).

References

- Akata, Z., Reed, S., Walter, D., Lee, H., & Schiele, B. (2015). Evaluation of output embeddings for fine-grained image classification. In *Proceedings of CVPR*, pp. 2927–2936.
- Akata, Z., Perronnin, F., Harchaoui, Z., & Schmid, C. (2016). Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7), 1425–1438.
- Bartels, R. H., & Stewart, G. W. (1972). Solution of the matrix equation $ax + xb = c$ [f4]. *Communications of the ACM*, 15, 820–826.
- Bo, L., Ren, X., & Fox, D. (2011). Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *NIPS*, pp. 2115–2123.
- Bucher, M., Herbin, S., & Jurie, F. (2017). Generating visual representations for zero-shot classification. In *ICCV workshops: Transferring and adapting source knowledge in computer vision*, pp. 2666–2673.
- Changpinyo, S., Chao, W. L., Gong, B., & Sha, F. (2016). Synthesized classifiers for zero-shot learning. In *Proceedings of CVPR*, pp. 5327–5336.
- Chao, W. L., Changpinyo, S., Gong, B., & Sha, F. (2016). An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *Proceedings of ECCV*, pp. 52–68.
- Chen, L., Zhang, H., Xiao, J., Liu, W., & Chang, S. F. (2018). Zero-shot visual recognition using semantics-preserving adversarial embedding network. In *Proceedings of CVPR*, pp. 1043–1052.
- Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., et al. (2014). Large-scale object classification using label relation graphs. In *Proceedings of ECCV*, pp. 48–64.
- Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., et al. (2014). DeCAF: A deep convolutional activation feature for generic visual recognition. In *Proceedings of ICML*, pp. 647–655.
- Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4), 594–611.
- Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of ICML*, pp. 1126–1135.
- Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A., et al. (2013). DeViSE: A deep visual-semantic embedding model. In *NIPS*, pp. 2121–2129.
- Fu, Y., & Sigal, L. (2016). Semi-supervised vocabulary-informed learning. In *Proceedings of CVPR*, pp. 5337–5346.
- Fu, Y., Hospedales, T. M., Xiang, T., & Gong, S. (2015a). Transductive multi-view zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11), 2332–2345.
- Fu, Z., Xiang, T., Kodirov, E., & Gong, S. (2015b). Zero-shot object recognition by semantic manifold distance. In *Proceedings of CVPR*, pp. 2635–2644.
- Fu, Z., Xiang, T., Kodirov, E., & Gong, S. (2018). Zero-shot learning on semantic class prototype graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(8), 2009–2022.
- Graves, A., Mohamed, A., & Hinton, G. E. (2013). Speech recognition with deep recurrent neural networks. In *Proceedings of ICASSP*, pp. 6645–6649.
- Guo, Y., Ding, G., Jin, X., & Wang, J. (2016). Transductive zero-shot recognition via shared model space learning. In *Proceedings of AAAI*, pp. 3494–3500.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of CVPR*, pp. 770–778.
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of CVPR*, pp. 2261–2269.
- Hwang, S. J., & Sigal, L. (2014). A unified semantic embedding: Relating taxonomies and attributes. In *NIPS*, pp. 271–279.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of ACMM*, pp. 675–678.
- Kankuekul, P., Kawewong, A., Tangruamsub, S., & Hasegawa, O. (2012). Online incremental attribute-based zero-shot learning. In *Proceedings of CVPR*, pp. 3657–3664.
- Kodirov, E., Xiang, T., Fu, Z., & Gong, S. (2015). Unsupervised domain adaptation for zero-shot learning. In *Proceedings of ICCV*, pp. 2452–2460.
- Kodirov, E., Xiang, T., & Gong, S. (2017). Semantic autoencoder for zero-shot learning. In *Proceedings of CVPR*, pp. 3174–3183.
- Lake, B. M., Salakhutdinov, R. R., & Tenenbaum, J. (2013). One-shot learning by inverting a compositional causal process. In *NIPS*, pp. 2526–2534.
- 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(3), 453–465.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 541–551.
- Lei Ba, J., Swersky, K., Fidler, S., & Salakhutdinov, R. (2015). Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of ICCV*, pp. 4247–4255.
- Li, A., Lu, Z., Wang, L., Xiang, T., & Wen, J. R. (2017). Zero-shot scene classification for high spatial resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7), 4157–4167.
- Li, X., Guo, Y., & Schuurmans, D. (2015). Semi-supervised zero-shot classification with label representation learning. In *Proceedings of ICCV*, pp. 4211–4219.
- Liu, F., Xiang, T., Hospedales, T. M., Yang, W., & Sun, C. (2017). Semantic regularisation for recurrent image annotation. In *Proceedings of CVPR*, pp. 4160–4168.
- Long, T., Xu, X., Shen, F., Liu, L., Xie, N., & Yang, Y. (2018). Zero-shot learning via discriminative representation extraction. *Pattern Recognition Letters*, 109, 27–34.
- Lu, Y. (2015). Unsupervised learning on neural network outputs: With application in zero-shot learning. [arXiv:1506.00990](https://arxiv.org/abs/1506.00990).
- Miller, G. A. (1995). Wordnet: An online lexical database. *Communications of the ACM*, 38(11), 39–44.
- Mishra, A., Reddy, M. S. K., Mittal, A., & Murthy, H. A. (2017). A generative model for zero shot learning using conditional variational autoencoders. [arXiv:1709.00663](https://arxiv.org/abs/1709.00663).
- Mishra, N., Rohaninejad, M., Chen, X., & Abbeel, P. (2016). A simple neural attentive meta-learner. In *Proceedings of ICLR*.
- Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., et al. (2014). Zero-shot learning by convex combination of semantic embeddings. In *Proceedings of ICLR*.
- Patterson, G., Xu, C., Su, H., & Hays, J. (2014). The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1), 59–81.
- Qiao, S., Liu, C., Shen, W., & Yuille, A. L. (2018). Few-shot image recognition by predicting parameters from activations. In *Proceedings of CVPR*, pp. 7229–7238.
- Radovanović, M., Nanopoulos, A., & Ivanović, M. (2010). Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(9), 2487–2531.
- Ravi, S., & Larochelle, H. (2016). Optimization as a model for few-shot learning. In *Proceedings of ICLR*.
- Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *CVPR*, pp. 7263–7271.
- Reed, S., Akata, Z., Lee, H., & Schiele, B. (2016). Learning deep representations of fine-grained visual descriptions. In *Proceedings of CVPR*, pp. 49–58.

- Rohrbach, M., Ebert, S., & Schiele, B. (2013). Transfer learning in a transductive setting. In *NIPS*, pp. 46–54.
- Romera-Paredes, B., & Torr, P. H. S. (2015). An embarrassingly simple approach to zero-shot learning. In *Proceedings of 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(3), 211–252.
- Shigeto, Y., Suzuki, I., Hara, K., Shimbo, M., & Matsumoto, Y. (2015). Ridge regression, hubness, and zero-shot learning. In *Proceedings of European conference on machine learning and principles and practice of knowledge discovery in databases*, pp. 135–151.
- Shojaee, S. M., & Baghshah, M. S. (2016). Semi-supervised zero-shot learning by a clustering-based approach. [arXiv:1605.09016](https://arxiv.org/abs/1605.09016).
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In *ICLR*.
- Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In *NIPS*, pp. 4080–4090.
- Socher, R., Ganjoo, M., Manning, C. D., & Ng, A. (2013). Zero-shot learning through cross-modal transfer. In *NIPS*, pp. 935–943.
- Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., & Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In *Proceedings of CVPR*, pp. 1199–1208.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Proceedings of CVPR*, pp. 1–9.
- Triantafillou, E., Zemel, R., & Urtasun, R. (2017). Few-shot learning through an information retrieval lens. In *NIPS*, pp. 2255–2265.
- Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. In *NIPS*, pp. 3630–3638.
- Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). *The caltech-ucsd birds-200-2011 dataset*. Technical Report of CNS-TR-2011-001. California Institute of Technology.
- Wang, F., & Zhang, C. (2008). Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineer*, 20(1), 55–67.
- Wang, Q., & Chen, K. (2017). Zero-shot visual recognition via bidirectional latent embedding. *International Journal of Computer Vision*, 124(3), 356–383.
- Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., & Schiele, B. (2016). Latent embeddings for zero-shot classification. In *Proceedings of CVPR*, pp. 69–77.
- Xian, Y., Schiele, B., & Akata, Z. (2017). Zero-shot learning—The good, the bad and the ugly. In *Proceedings of CVPR*, pp. 4582–4591.
- Xian, Y., Lorenz, T., Schiele, B., & Akata, Z. (2018). Feature generating networks for zero-shot learning. In *Proceedings of CVPR*, pp. 5542–5551.
- Xu, X., Hospedales, T., & Gong, S. (2015). Semantic embedding space for zero-shot action recognition. In *Proceedings of IEEE conference on image processing*, pp. 63–67.
- Xu, X., Hospedales, T., & Gong, S. (2017). Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision*, 123(3), 309–333.
- Ye, M., & Guo, Y. (2017). Zero-shot classification with discriminative semantic representation learning. In *Proceedings of CVPR*, pp. 7140–7148.
- Yu, Y., Ji, Z., Li, X., Guo, J., Zhang, Z., Ling, H., et al. (2017). Transductive zero-shot learning with a self-training dictionary approach. [arXiv:1703.08893](https://arxiv.org/abs/1703.08893)
- Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. In *Proceedings of British machine vision conference*, pp. 87.1–87.12.
- Zhang, L., Xiang, T., & Gong, S. (2017). Learning a deep embedding model for zero-shot learning. In *Proceedings of CVPR*, pp. 2021–2030.
- Zhang, Z., & Saligrama, V. (2015). Zero-shot learning via semantic similarity embedding. In *Proceedings of ICCV*, pp. 4166–4174.
- Zhang, Z., & Saligrama, V. (2016a). Zero-shot learning via joint latent similarity embedding. In *Proceedings of CVPR*, pp. 6034–6042.
- Zhang, Z., & Saligrama, V. (2016b). Zero-shot recognition via structured prediction. In *Proceedings of ECCV*, pp. 533–548.
- Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., et al. (2015). Conditional random fields as recurrent neural networks. *Proceedings of CVPR*, pp. 1529–1537.
- Zhu, Y., Elhoseiny, M., Liu, B., & Elgammal, A. M. (2018). A generative adversarial approach for zero-shot learning from noisy texts. In *Proceedings of CVPR*, pp. 1004–1013.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.