Zero-Shot Learning via Category-Specific Visual-Semantic Mapping and Label Refinement

Li Niu, Jianfei Cai, *Senior Member, IEEE*, Ashok Veeraraghavan, and Liqing Zhang

***Abstract*— Zero-shot learning (ZSL) aims to classify a test instance from an unseen category based on the training instances from seen categories in which the gap between seen categories and unseen categories is generally bridged via visual-semantic mapping between the low-level visual feature space and the intermediate semantic space. However, the visual-semantic mapping (i.e., projection) learnt based on seen categories may not generalize well to unseen categories, which is known as the projection domain shift in ZSL. To address this projection domain shift issue, we propose a method named adaptive embedding ZSL (AEZSL) to learn an adaptive visual-semantic mapping for each unseen category, followed by progressive label refinement. Moreover, to avoid learning visual-semantic mapping for each unseen category in the large-scale classification task, we additionally propose a deep adaptive embedding model named deep AEZSL sharing the similar idea (i.e., visual-semantic mapping should be category specific and related to the semantic space) with AEZSL, which only needs to be trained once, but can be applied to arbitrary number of unseen categories. Extensive experiments demonstrate that our proposed methods achieve the state-of-the-art results for image classification on three small-scale benchmark datasets and one large-scale benchmark dataset.**

***Index Terms*— Zero-shot learning (ZSL), domain adaptation.**

I. Introduction

CONVENTIONAL classification tasks require the training and test categories to be identical, but collecting annotated data for all categories is time-consuming and expensive in large-scale or fine-grained classification tasks. Therefore, Zero-Shot Learning (ZSL) [1]–[3], which aims to recognize the test instances from the categories previously unseen in the training stage, has become increasingly popular in the field of computer vision. In ZSL, the gap between seen (*i.e.*, training) categories and unseen (*i.e.*, testing) categories is generally bridged based on the intermediate semantic representations. Semantic representation of each category means the representation of this category in the semantic embedding space. One popular semantic representation is manually designed attribute vector [1], [4], which is a high-level description for each category, such as the shape (*e.g.*, cylindrical), material (*e.g.*, cloth), and color (*e.g.*, white). Besides, there also exist automatically generated semantic representations [5]–[8] including the textual features extracted from online corpus corresponding to this category or the word vector of this category name.

Recently, based on the intermediate semantic representation (*i.e.*, semantic embedding), many ZSL approaches have been proposed, which can be roughly categorized into Semantic Relatedness (SR) and Semantic Embedding (SE) methods according to [9] and [10]. In particular, the first category SR methods [11]–[13] tend to learn the visual classifiers for unseen categories using the similarities between unseen categories and seen categories while the second category SE methods attract more research interest [1], [5], [6], [10], [14]–[27], which aim to build the semantic link between visual features and semantic representations using mapping functions. More details of SR and SE methods can be found in Section II. For Semantic Embedding (SE) methods, a mapping function needs to be learnt from the seen categories in the training stage, and then applied to the unseen categories in the testing stage. However, the visual-semantic mappings between the seen categories and unseen categories might be considerably different. In this sense, the learnt mapping (*i.e.*, projection) may not generalize well to the test set and thus results in poor performance, leading to the recently highlighted projection domain shift, which was first mentioned in [20] and then theoretically proved in [24].

To cope with the projection domain shift, some semi-supervised or transductive (the two terminologies semi-supervised and transductive are often interchangeably used in the field of ZSL) ZSL approaches have been proposed by utilizing unlabeled test instances from unseen categories in the training stage. In general, these methods either learn a common mapping function for both seen and unseen categories, or adapt the mapping function learnt based on the seen categories to the unseen categories. Nevertheless, this may be insufficient to tackle the projection domain shift because the mapping function of each individual category varies significantly.

Manuscript received December 13, 2017; revised July 4, 2018; accepted September 13, 2018. Date of publication September 28, 2018; date of current version October 23, 2018. This work was supported in part by the Key Basic Research Program of Shanghai under Grant 15JC1400103 and in part by the National Basic Research Program of China under Grant 2015CB856004. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aydin Alatan. *(Corresponding author: Li Niu.)*

L. Niu was with the Electric and Computer Engineering Department, Rice University, Houston, TX 77005 USA. He is now with the Computer Science and Engineering Department, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: ustcnewly@sjtu.edu.cn).

J. Cai is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: asjfcai@ntu.edu.sg).

A. Veeraraghavan is with the Electric and Computer Engineering Department, Rice University, Houston, TX 77005 USA (e-mail: vashok@rice.edu).

L. Zhang is with the Computer Science and Engineering Department, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zhang-lq@cs.sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2872916

For example, as shown in Fig. 3, the categories "badminton court" and "bedchamber" share the same "cloth" attribute, but in fact the visual appearances of cloth from these two categories vary greatly. Another example is that the categories "recycling plant" and "landfill" both have the attribute "cluttered space", but their manifestation are also considerably different (see Section V for more details).

To this end, we focus on the projection domain shift and tend to address this problem by learning a category-specific mapping function for each unseen category, which has never been studied before. Since we do not have labeled instances for the unseen categories, the mapping functions of unseen categories can only be learnt by transferring from those of seen categories. However, it is hard to tell which seen categories are semantically closer to a given unseen category *w.r.t.* certain entry in the semantic representation. Thus, we make a simple yet effective assumption that when the semantic representations of two categories are similar, the common non-zero entries of these two semantic representations should be semantically similar, and thus the mapping functions of these two categories should also be similar. Specifically, for each unseen category, we calculate the similarities between this unseen category and all the seen categories based on their semantic representations, and assign higher weights to the classification tasks corresponding to the more similar seen categories. This idea can be unified with many existing ZSL works with various forms of classification losses. As an instantiation, we build our method upon ESZSL [24] considering its simplicity and effectiveness, which is named as Adaptive Embedding ZSL (AEZSL). Note that for AEZSL, we only utilize the semantic representations of unseen categories to learn the mapping functions. In order to further utilize the unlabeled instances from unseen categories like previous semi-supervised or transductive ZSL works [10], [15], [18]–[20], we propose a progressive label refinement strategy following AEZSL.

One problem of AEZSL is its inefficiency for large-scale classification task with a large number of unseen categories because one visual-semantic mapping needs to be learnt for each unseen category. Thus, we aim to design a model which only needs to be trained once on seen categories but can be applied to any unseen category without retraining the model. With this aim, we develop a Deep Adaptive Embedding ZSL (DAEZSL) model, which only utilizes seen categories in the training stage but has the generalization ability to an arbitrary number of unseen categories. Specifically, instead of learning one mapping for each unseen category based on its semantic representation as for AEZSL, we target at learning a projection function from semantic representation to visual-semantic mapping. In this sense, given a new unseen category, its visual-semantic mapping can be easily generated based on its semantic representation. In the testing stage, given a set of unseen categories associated with semantic representations, our model is able to generate category-specific feature weights for each unseen category, which can better fit the classification tasks for unseen categories.

Our contributions are threefold: 1) this is the first work to address the projection domain shift by learning category-specific visual-semantic mappings with the idea that higher weights should be assigned to the classification tasks of more relevant seen categories for each unseen category. As an instantiation, we build our AEZSL method upon ESZSL [24], followed by a progressive label refinement strategy; 2) we additionally propose a deep adaptive embedding model DAEZSL for large-scale ZSL, which needs to be trained only once on seen categories but can be applied to arbitrary unseen category; 3) comprehensive experiments are conducted on three small-scale datasets and one large-scale dataset to demonstrate the effectiveness of our approaches.

II. Related Work

As discussed in Section I, the existing Zero-Shot Learning (ZSL) methods can be categorized into Semantic Relatedness (SR) approaches and Semantic Embedding (SE) approaches. From another perspective, ZSL methods can be categorized into standard ZSL and transductive/semi-supervised ZSL based on whether to use the unlabeled test instances from unseen categories in the training stage. Moreover, several deep learning models have been proposed for ZSL. Additionally, since the terminology domain shift is commonly used in the field of domain adaptation, we briefly describe the difference of this terminology used in ZSL and domain adaptation.

A. Semantic Relatedness (SR) and Semantic Embedding (SE) ZSL

For SR approaches, the methods in [11]–[13] construct the visual classifiers for unseen categories from those for seen categories based on the semantic similarities between seen categories and unseen categories. Then, an extension was made in [28], which assumes that the visual classifiers for both seen and unseen categories can be represented by a set of base classifiers. The above SR approaches do not exhibit obvious projection domain shift problem, but they cannot take full advantage of the semantic representations. Moreover, the above works did not discuss how to utilize the unlabeled test instances in the training stage. In contrast, our methods can fully exploit the semantic representations and leverage the unlabeled test instances during the training procedure.

For SE approaches, they can be further divided into three groups based on different strategies to build the semantic link between the visual feature space and the semantic representation space. The first group of methods [1], [14]–[16] are proposed for projecting the visual features to semantic representations based on the learnt attribute classifiers. The second group of methods [18], [19], [29], [30] target at projecting the semantic representations to visual features based on the learnt dictionary. It is worth mentioning that pseudo training samples are generated for unseen categories in [30] based on the learnt mapping. The third group of methods [5], [6], [10], [20]–[27] tend to map the visual feature space and the semantic representation space into a common space, or learn a mapping function which measures the compatibility between visual features and semantic representations. Among the above SE approaches, the approaches in [1], [5], [14], [22], [23],

and [26] do not address the projection domain shift while the remaining transductive or semi-supervised approaches can somehow account for the projection domain shift problem, which will be detailed next.

B. Transductive or Semi-Supervised ZSL

Some transductive or semi-supervised Semantic Embedding (SE) methods [10], [15], [18]–[21], [25], [31] can alleviate the projection domain shift by utilizing the unlabeled test instances from unseen categories in the training phase. Specifically, the projection domain shift is rectified in [20] by projecting visual features and multiple semantic embeddings of test instances into a common subspace via Canonical Correlation Analysis (CCA). Label propagation for zero-shot learning is used in both [20] and [31]. The approach in [18] learns dictionaries for the seen categories and the unseen categories separately while enforcing these two dictionaries to be close. The methods in [10], [19], [21], and [25] learn the mapping function and simultaneously infer the test labels. In [15] and [21], a Laplacian regularizer is employed based on the semantic representations or the label representations of test instances. The above transductive or semi-supervised ZSL approaches are able to account for the projection domain shift problem. However, the approach in [20] requires multi-view semantic embeddings, which are often unavailable. Besides, the methods in [10], [15], [18], [19], [21], and [25] either learn a common mapping function for all the categories or adapt the mapping function learnt from the seen categories to the unseen categories, which is likely to be improper for the real-world applications since the mapping function of each category may vary significantly. In contrast, our methods learn a category-specific mapping for each unseen category, which can capture diverse semantic meanings of the same attribute for different categories.

C. Deep ZSL

Compared with traditional ZSL methods based on extracted deep learning features, there are fewer end-to-end deep ZSL models [6], [7], [32]–[34]. These works learn the mapping from visual features to semantic representations [7], map visual feature space and semantic space to a common space [6], [32], [34], or directly learn classifiers for different categories based on their semantic representations [33]. However, none of the above deep ZSL methods mentions the projection domain shift issue. In contrast, we focus on tackling the projection domain shift in this paper and propose a deep adaptive embedding model DAEZSL which can adapt the visual-semantic mapping to different categories implicitly by learning category-specific feature weight.

D. Domain Adaptation

Domain adaptation methods aim to address the domain shift issues, *i.e.*, to reduce the domain distribution mismatch between the source domain (*i.e.*, training set) and the target domain (*i.e.*, test set). Regarding the domain shift problem, domain adaptation methods focus on the difference of marginal probability or class conditional probability between the source domain and the target domain while ZSL methods focus on the difference of visual-semantic mappings (*i.e.*, projections) between seen categories and unseen categories, which is referred to as projection domain shift. Thus, the terminology of domain shift has quite different meanings in these two fields. In this paper, we focus on the projection domain shift problem in ZSL.

III. Background

In this paper, for ease of presentation, a vector/matrix is denoted by a lowercase/uppercase letter in boldface. The transpose of a vector/matrix is denoted using the superscript $'$. We use $\mathbf{A}\circ\mathbf{B}$ to denote the dot product of two matrices. Moreover, we use $\mathbf{I}$ to denote the identity matrix and $\mathbf{A}^{-1}$ to denote the inverse matrix of $\mathbf{A}$. We use upperscript s (*resp.*, t) to indicate seen (*resp.*, unseen) categories while omitting the upperscript for not being specific with seen or unseen categories.

A. Problem Definition

Assume we have C^s seen categories and C^t unseen categories. Let us denote the training (*resp.*, test) data from seen (*resp.*, unseen) categories as $\mathbf{X}^s \in \mathcal{R}^{d\times n^s}$ (*resp.*, $\mathbf{X}^t \in \mathcal{R}^{d\times n^t}$), where d is the dimensionality of visual features and n^s (*resp.*, n^t) is the number of training (*resp.*, test) instances from seen (*resp.*, unseen) categories. We assume each category is associated with an a-dim semantic representation and thus the semantic representations of seen (*resp.*, unseen) categories can be stacked as $\mathbf{A}^s \in \mathcal{R}^{a\times C^s}$ (*resp.*, $\mathbf{A}^t \in \mathcal{R}^{a\times C^t}$). In order to bridge the gap between visual features and semantic representations and simultaneously exploit the discriminative capacity of semantic representations, inspired by Romera-Paredes and Torr [24], Guo *et al.* [25], and Qiao *et al.* [35], we use the mapping matrix $\mathbf{W} \in \mathcal{R}^{d\times a}$ to match visual features with semantic representations, *i.e.*, $\mathbf{X}^{s\prime}\mathbf{W}\mathbf{A}^s$, which measures the compatibility between instances and categories. Ideally, the most compatible semantic representation of each training instance should be from its ground-truth category. In the testing stage, a test instance $\mathbf{x}^t$ is assigned to the category corresponding to the maximum value in the C^t-dim vector $\mathbf{x}^{t\prime}\mathbf{W}\mathbf{A}^t$, in which $\mathbf{W}\mathbf{A}^t$ is essentially the stacked visual classifiers for unseen categories.

B. Embarrassingly Simple Zero-Shot Learning (ESZSL)

Before introducing our method, we briefly introduce Embarrassingly Simple Zero-Shot Learning (ESZSL) [24], based on which we build our own method considering its simplicity and effectiveness. ESZSL is formulated as

$$\min_{\mathbf{W}} \|\mathbf{X}^{s\prime}\mathbf{W}\mathbf{A}^s - \mathbf{Y}^s\|_F^2 + \gamma\|\mathbf{W}\mathbf{A}^s\|_F^2 + \lambda\|\mathbf{X}^{s\prime}\mathbf{W}\|_F^2 + \beta\|\mathbf{W}\|_F^2, \quad (1)$$

in which γ, λ, and β are trade-off parameters, and $\mathbf{Y}^s \in \mathcal{R}^{n^s\times C^s}$ is a binary label matrix with the i-th row being the label vector of the i-th training instance. Note that in (1), $\|\mathbf{X}^{s\prime}\mathbf{W}\mathbf{A}^s - \mathbf{Y}^s\|_F^2$ can fully exploit the discriminability of semantic representations by assigning each instance to the category with the most compatible semantic representation.

Specifically, the multi-class classification error is minimized by both learning how to predict attributes via $\mathbf{X}^{s\prime}\mathbf{W}$ and also considering the importance of each attribute contributing to the final classification. $\|\mathbf{W}\mathbf{A}^s\|_F^2$ is used to control the complexity of the projection of semantic embeddings onto the visual feature space, which allows fair comparisons of semantic embeddings from different categories. $\|\mathbf{X}^{s\prime}\mathbf{W}\|_F^2$ is used to bound the variance of the projection of visual features onto the semantic embedding space, which makes the approach invariant to diverse feature distribution. $\|\mathbf{W}\|_F^2$ is a standard weight penalty to control the complexity of $\mathbf{W}$. By setting the derivative of (1) *w.r.t.* $\mathbf{W}$ to zero, we obtain

$$(\mathbf{X}^s\mathbf{X}^{s\prime}+\gamma\,\mathbf{I})\mathbf{W}\mathbf{A}^s\mathbf{A}^{s\prime}+\lambda(\mathbf{X}^s\mathbf{X}^{s\prime}+\frac{\beta}{\lambda}\mathbf{I})\mathbf{W}=\mathbf{X}^s\mathbf{Y}^s\mathbf{A}^{s\prime}. \quad (2)$$

when setting $\beta=\gamma\,\lambda$, the problem in (2) has a close-form solution:

$$\mathbf{W}=(\mathbf{X}^s\mathbf{X}^{s\prime}+\gamma\,\mathbf{I})^{-1}\mathbf{X}^s\mathbf{Y}^s\mathbf{A}^{s\prime}(\mathbf{A}^s\mathbf{A}^{s\prime}+\lambda\mathbf{I})^{-1}. \quad (3)$$

IV. Our Methods

In this section, we first build our AEZSL method upon ESZSL to learn category-specific mapping matrix in Section IV-A1 followed by a label refinement strategy in Section IV-A2. Moreover, we propose a deep adaptive embedding model named DAEZSL for large-scale ZSL in Section IV-B, which shares the similar idea with AEZSL.

A. Adaptive Embedding Zero-Shot Learning

In this section, we first introduce how to learn category-specific visual-semantic mapping, followed by describing our label refinement strategy.

1) Category-Specific Visual-Semantic Mapping: Since the visual-semantic mappings of different categories could be largely different, we aim to learn a category-specific mapping matrix for each unseen category (*i.e.*, $\mathbf{W}^c$ for the c-th unseen category) to address the projection domain shift. Because no labeled instances are provided for the unseen categories, we can only transfer from the mapping matrices of seen categories. However, it is a challenging task to determine which seen categories are more semantically similar to a given unseen category *w.r.t.* certain entry in the semantic representation. To facilitate the transfer, we assume that when the semantic representations of two categories are similar, the common non-zero entries shared by these two semantic representations should be semantically similar, and thus the mapping matrices of these two categories should also be similar.

Specifically, recall that in (1), each column in $\mathbf{X}^{s\prime}\mathbf{W}\mathbf{A}^s-\mathbf{Y}^s$ corresponds to the classification task for each seen category, and the tasks for different seen categories are dependent on one another with a common $\mathbf{W}$. To avoid ambiguity, in the remainder of this paper, we use c to denote the index of unseen category and $\tilde{c}$ to denote the index of seen category. Given the c-th unseen category, in order to ensure that $\mathbf{W}^c$ is close to the mapping matrices of those similar seen categories, we assign higher weights to the classification tasks for those more similar seen categories. In particular, we formulate this idea as $(\mathbf{X}^{s\prime}\mathbf{W}^c\mathbf{A}^s-\mathbf{Y}^s)\mathbf{S}^c$, where $\mathbf{S}^c\in\mathcal{R}^{C^s\times C^s}$ is a diagonal matrix with the $\tilde{c}$-th diagonal element being the cosine similarity

$$s_{\tilde{c}}^c=\frac{\mathbf{a}_c^{t\prime}\mathbf{a}_{\tilde{c}}^s}{\|\mathbf{a}_c^t\|\|\mathbf{a}_{\tilde{c}}^s\|},$$

which measures the similarity between the semantic representation $\mathbf{a}_c^t$ of the c-th unseen category and the semantic representation $\mathbf{a}_{\tilde{c}}^s$ of the $\tilde{c}$-th seen category. Note that $(\mathbf{X}^{s\prime}\mathbf{W}^c\mathbf{A}^s-\mathbf{Y}^s)\mathbf{S}^c$ is equivalent to multiplying the $\tilde{c}$-th column of $\mathbf{X}^{s\prime}\mathbf{W}^c\mathbf{A}^s-\mathbf{Y}^s$ by $s_{\tilde{c}}^c$. For better explanation, assuming that the c-th unseen category is similar to the $\tilde{c}$-th seen category and their semantic representations share the common non-zero entry indices $\{j_1, j_2, \ldots, j_l\}$, then a higher weight $s_{\tilde{c}}^c$ should be assigned to $\mathbf{X}^{s\prime}\mathbf{W}^c\mathbf{a}_{\tilde{c}}^s-\mathbf{y}_{\tilde{c}}^s$ with $\mathbf{y}_{\tilde{c}}^s$ being the $\tilde{c}$-th column of $\mathbf{Y}^s$. In this way, the learnt $\mathbf{W}^c$ should be closer to the mapping matrix of the $\tilde{c}$-th seen category *w.r.t.* the $\{j_1, j_2, \ldots, j_l\}$-th columns. To this end, we tend to solve $\mathbf{W}^c$'s for all unseen categories simultaneously using the following formulation:

$$\min_{\mathbf{W}^c}\ \frac{1}{2}\sum_{c=1}^{C^t}\|(\mathbf{X}^{s\prime}\mathbf{W}^c\mathbf{A}^s-\mathbf{Y}^s)\mathbf{S}^c\|_F^2+\frac{\lambda_1}{2}\sum_{c=1}^{C^t}\|\mathbf{X}^{s\prime}\mathbf{W}^c\|_F^2 + \frac{\lambda_2}{2}\sum_{c=1}^{C^t}\|\mathbf{W}^c\|_F^2+\frac{\lambda_3}{2}\sum_{c<\bar{c}}\|\mathbf{W}^c-\mathbf{W}^{\bar{c}}\|_F^2, \quad (4)$$

where λ_1, λ_2, and λ_3 are trade-off parameters, which can be obtained using cross-validation, and $\sum_{c<\bar{c}}\|\mathbf{W}^c-\mathbf{W}^{\bar{c}}\|_F^2$ is a co-regularizer which encourages $\mathbf{W}^c$'s for different unseen categories to share some common parts. Note that we omit the regularizer $\|\mathbf{W}^c\mathbf{A}^s\|_F^2$ in (1) for ease of optimization. When λ_3 approaches Infinity, the mappings of all categories are enforced to be the same $\mathbf{W}$. In this case, the problem in (4) reduces to

$$\min_{\mathbf{W}}\ \frac{1}{2}\|(\mathbf{X}^{s\prime}\mathbf{W}\mathbf{A}^s-\mathbf{Y}^s)\tilde{\mathbf{S}}\|_F^2+\frac{\lambda_1}{2}\|\mathbf{X}^{s\prime}\mathbf{W}\|_F^2+\frac{\lambda_2}{2}\|\mathbf{W}\|_F^2, \quad (5)$$

in which the $\tilde{\mathbf{S}}$ is a diagonal matrix with the $\tilde{c}$-th diagonal element being

$$\sqrt{\frac{\sum_{c=1}^{C^t}(s_{\tilde{c}}^c)^2}{C^t}}.$$

Compared with (1), we assign higher weights on the classification tasks corresponding to the seen categories which are closer to the overall unseen categories based on $\tilde{\mathbf{S}}$.

The problem in (4) can be solved in an alternating fashion by updating one $\mathbf{W}^c$ while fixing all the other $\mathbf{W}^{\bar{c}}$'s. We leave the detailed solution to Appendix A. After learning $\mathbf{W}^c$'s, we can obtain the visual classifier for the c-th unseen category as $\mathbf{p}^c=\mathbf{W}^c\mathbf{a}_c^t$. In the testing stage, given a test instance $\mathbf{x}^t$ from unseen categories, we can obtain its decision value for the c-th unseen category as $\mathbf{p}^{c\prime}\mathbf{x}^t$.

Discussion: The projection domain shift problem has been theoretically proved in [24], in which the seen (*resp.*, useen) categories are referred to as the source (*resp.*, target) domain. The source (*resp.*, target) domain sample can be represented as $\tilde{\mathbf{x}}_{i,c}^s=vec(\mathbf{x}_i^s\mathbf{a}_c^{s\prime})$ (*resp.*, $\tilde{\mathbf{x}}_{i,c}^t=vec(\mathbf{x}_i^t\mathbf{a}_c^{t\prime})$). By assuming that the difference of visual feature distribution between two domains ($\mathbf{x}_i^s$'s and $\mathbf{x}_i^t$'s) is negligible, the domain shift mainly depends on the difference of semantic representations between two domains ($\mathbf{a}_c^s$'s and $\mathbf{a}_c^t$'s). Then, two extreme

cases are presented in [24] when the semantic representations of two domains are identical or orthogonal. Specifically, when the semantic representations of two domains are identical (*i.e.*, $\mathbf{a}_c^t = \mathbf{a}_{\hat{c}}^s$), the error bound of the learnt classifier is approximate to that of a standard classifier without projection domain shift. In this case, we have $s_{\hat{c}}^c = 1$, which allows the maximum transfer. When their semantic representations are orthogonal (*i.e.*, ${\mathbf{a}_c^t}'\mathbf{a}_{\hat{c}}^s = 0$), the error bound is vacuous and no transfer can be done. In this case, we have $s_{\hat{c}}^c = 0$, which means no transfer. So the analysis for our problem accords with that in [24], which verifies that it is reasonable to control how much to transfer from seen categories based on the cosine similarities.

In this paper, we build our AEZSL method upon ESZSL due to its simplicity and effectiveness. However, it is worth mentioning that the idea of AEZSL, *i.e.*, assigning higher weights on the classification tasks of more similar seen categories for each unseen category, can be incorporated into many existing ZSL frameworks with slight modification based on their learning paradigms and used losses.

2) Progressive Label Refinement: Note that in Section IV-A1, we only utilize the semantic representations of unseen categories to learn the adaptive mapping matrices. In order to further adapt the mapping matrices to the unseen categories by utilizing the unlabeled test instances, we propose a progressive approach to update visual classifiers and refine predicted test labels alternatively, similar to some progressive semi-supervised learning approaches like [36]. Unlike traditional semi-supervised learning which requires the labeled and unlabeled instances to be from the same set of categories, zero-shot learning does not have any labeled instances from the unseen categories. So we divide the test set into a confident set $\mathcal{L}$ and an unconfident set $\mathcal{U}$, in which the labels in $\mathcal{L}$ (*i.e.*, $\mathbf{Y}^l$) are expected to be relatively more accurate than those in $\mathcal{U}$ (*i.e.*, $\mathbf{Y}^u$). Note that $\mathbf{Y}^l$ and $\mathbf{Y}^u$ are both binary label matrices, similar to $\mathbf{Y}^s$ in (4). Initially, $\mathcal{L}$ is an empty set and $\mathcal{U}$ is the entire test set with initial labels $\mathbf{Y}^u$ predicted by the initial classifiers $\mathbf{p}^c$'s, which are obtained based on $\mathbf{W}^c$'s from Section IV-A1. Then, we move k most confident instances from $\mathcal{U}$ into $\mathcal{L}$, and update the visual classifiers based on the new confident and unconfident sets. We repeat the above process iteratively until $\mathcal{U}$ becomes empty and output $\mathbf{Y}^l$ as final predicted test labels. The details of each iteration will be described as follows, in which we stack $\mathbf{p}^c$'s for all unseen categories as $\mathbf{P}$, and split $\mathbf{X}^t$ into $\mathbf{X}^l$ and $\mathbf{X}^u$ corresponding to $\mathcal{L}$ and $\mathcal{U}$.

In each iteration, we first use the visual classifiers $\mathbf{P}$ from the previous iteration to select k most confident instances from $\mathcal{U}$ based on the confidence score, which is defined as a softmax function

$$\text{conf}(\mathbf{x}^t) = \frac{\exp({\mathbf{p}^{c(\mathbf{x}^t)}}'\mathbf{x}^t)}{\sum_{\hat{c}} \exp({\mathbf{p}^{\hat{c}}}'\mathbf{x}^t)}$$

with $c(\mathbf{x}^t)$ being the assigned label of $\mathbf{x}^t$ corresponding to $\mathbf{p}^{c(\mathbf{x}^t)}$ which achieves the highest prediction score on $\mathbf{x}^t$. Note that the labels of the selected k instances are updated as $c(\mathbf{x}^t)$ while the labels of the remaining instances in $\mathcal{U}$ stay unchanged.

After moving the k most confident instances from $\mathcal{U}$ into $\mathcal{L}$, we update $\mathbf{P}$ by changing some predicted labels in $\mathcal{U}$ (*i.e.*, $\mathbf{X}^{u\prime}\mathbf{P}$) while using $\mathcal{L}$ as weak supervision. In particular, we update $\mathbf{P}$ with the aim to flip the labels of set $\mathcal{U}$ predicted by $\mathbf{P}$ (*i.e.*, $\mathbf{X}^{u'}\mathbf{P}$) based on a group-lasso regularizer $\|\mathbf{X}^{u\prime}\mathbf{P} - \mathbf{Y}^u\|_{2,1}$, which allows some rows in $\mathbf{X}^{u'}\mathbf{P}$ to be inconsistent with those in $\mathbf{Y}^u$. This means that the group-lasso regularizer encourages the predicted labels of some unconfident instances (the rows in $\mathbf{X}^{u'}\mathbf{P} - \mathbf{Y}^u$ with non-zero entries) to be flipped. Moreover, we rely on the similarities among unconfident instances and the similarities among unseen categories to regulate the label flipping, which will be explained as follows. 1) For the similarities among unconfident instances, we employ a standard Laplacian regularizer based on the smoothness assumption that the predicted label vectors of two unconfident instances should be close to each other when their visual features are similar. 2) For the similarities among unseen categories, we construct a transition matrix $\hat{\mathbf{S}}$ to characterize the probabilities that one category label is flipped to another, in which $\hat{S}_{i,j}$ is the cosine similarity

$$\frac{{\mathbf{a}_i^t}'\mathbf{a}_j^t}{\|\mathbf{a}_i^t\|\|\mathbf{a}_j^t\|},$$

similar to that in Section IV-A1. With the transition matrix $\hat{\mathbf{S}}$, we employ a coherent regularizer $\text{tr}(\mathbf{Y}^u\hat{\mathbf{S}}\mathbf{P}'\mathbf{X}^u)$, which enforces the predicted labels $\mathbf{P}'\mathbf{X}^u$ to be coherent with the expected transited labels $\mathbf{Y}^u\hat{\mathbf{S}}$ based on the transition probabilities. To be more specific, let us denote $\mathbf{Y}_1 = \mathbf{X}^{u'}\mathbf{P}$ and $\mathbf{Y}_2 = \mathbf{Y}^u\hat{\mathbf{S}}$, then maximizing $\text{tr}(\mathbf{Y}_2\mathbf{Y}_1')$ will enforce each row of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ (*i.e.*, label vector of each sample) to be coherent. Note that we set the diagonal elements of $\hat{\mathbf{S}}$ as zeros to encourage the labels to be flipped. To this end, the formulation to update $\mathbf{P}$ in each iteration can be written as

$$\min_{\mathbf{P}} \ \frac{1}{2}\|\mathbf{X}^{l\prime}\mathbf{P} - \mathbf{Y}^l\|_F^2 + \frac{\gamma_1}{2}\|\mathbf{X}^{u\prime}\mathbf{P} - \mathbf{Y}^u\|_{2,1} - \gamma_2\text{tr}(\mathbf{Y}^u\hat{\mathbf{S}}\mathbf{P}'\mathbf{X}^u) + \frac{\gamma_3}{2}\text{tr}(\mathbf{P}'\mathbf{X}^u\mathbf{H}^u\mathbf{X}^{u\prime}\mathbf{P}), \quad (6)$$

where γ_1, γ_2, and γ_3 are trade-off parameters, which can be obtained using cross-validation, and $\mathbf{H}^u$ is the Laplacian matrix constructed based on $\mathbf{X}^u$ following [21]. Specifically, we construct the similarity matrix $\tilde{\mathbf{H}}^u$ with each entry $\tilde{H}_{i,j}^u$ being the inverse Euclidean distance between $\mathbf{x}_i^u$ and $\mathbf{x}_j^u$, and then produce the Laplacian matrix $\mathbf{H}^u = diag(\tilde{\mathbf{H}}^u\mathbf{1}) - \tilde{\mathbf{H}}^u$. The problem in (18) is not easy to solve due to the regularizer $\|\mathbf{X}^{u\prime}\mathbf{P} - \mathbf{Y}^u\|_{2,1}$. We leave the detailed solution to (6) in Appendix B. We refer to our AEZSL with Label Refinement as AEZSL_LR.

Discussion: Since $\mathbf{p}^c = \mathbf{W}^c\mathbf{a}_c^t$ (see Section IV-A1) and $\mathbf{a}_c^t$'s are fixed, updating $\mathbf{p}^c$'s implies updating $\mathbf{W}^c$. In other words, by updating $\mathbf{p}^c$ based on the similarities among test instances and among unseen categories, we actually adapt the mapping matrices $\mathbf{W}^c$'s to the test set implicitly.

B. Deep Adaptive Embedding Zero-Shot Learning

One of the important applications of zero-shot learning is large-scale classification since it is difficult to obtain sufficient well-labeled training data exhaustively for all the categories.

However, our AEZSL method is not very efficient in this scenario, because we need to learn one visual-semantic mapping for each unseen category (*e.g.*, over 20,000 test categories in the ImageNet 2011 21K dataset). This concern motivates us to design a model which is more suitable for large-scale ZSL. In order to mitigate the burden induced by a large set of unseen categories, we design a deep adaptive embedding model named Deep Adaptive Embedding ZSL (DAEZSL), which only needs to be trained once but can generalize to arbitrary number of unseen categories.

Similar as in Section IV-A1, we assume the visual-semantic mappings should be category-specific and related to the semantic representation of each category. To avoid learning one mapping for each unseen category as in Section IV-A1, one possible approach is learning a projection $f(\cdot)$ from semantic representation $\mathbf{a}_c$ to visual-semantic mapping $\mathbf{W}_c$, *i.e.*, $f(\mathbf{a}_c) = \mathbf{W}_c$, so that given a new unseen category with semantic representation $\mathbf{a}_{\tilde{c}}$, we can easily obtain its visual-semantic mapping $\mathbf{W}_{\tilde{c}}$ by using $\mathbf{W}_{\tilde{c}} = f(\mathbf{a}_{\tilde{c}})$.

However, the size of $\mathbf{W}_c$ is usually very large (*i.e.*, $d \times a$), so we adopt an alternative approach for the sake of computational efficiency, that is, learning the projection $g(\cdot)$ from semantic representation $\mathbf{a}_c$ to feature weight $\mathbf{m}_c$, *i.e.*, $g(\mathbf{a}_c) = \mathbf{m}_c$. The feature weight $\mathbf{m}_c$ is applied on the visual feature via elementwise product, and thus has the same dimension d as visual feature. Since the size of $\mathbf{m}_c$ (*i.e.*, d) is much smaller than that of $\mathbf{W}_c$ (*i.e.*, $d \times a$), it is much more efficient to learn the projection $g(\cdot)$ instead of $f(\cdot)$. In fact, applying the feature weights of different categories can be treated as implicitly adapting the visual-semantic mapping to different categories, which will be explained as follows. Assume we have a common visual-semantic mapping $\mathbf{W}$, given the i-th training instance with visual feature $\mathbf{x}_i^s$, its decision values after applying the feature weight $\mathbf{m}_c^s$ is

$$(\mathbf{x}_i^{s'} \circ \mathbf{m}_c^{s\prime})\mathbf{W}\mathbf{A}^s = \mathbf{x}_i^{s'}(\bar{\mathbf{M}}_c^s \circ \mathbf{W})\mathbf{A}^s, \tag{7}$$

in which $\bar{\mathbf{M}}_c^s$ is horizontally stacked a copies of $\mathbf{m}_c^s$. Then, we define the implicit visual-semantic mapping $\bar{\mathbf{W}}_c^s$ as

$$\bar{\mathbf{W}}_c^s = \bar{\mathbf{M}}_c^s \circ \mathbf{W}, \tag{8}$$

from which we can see that learning feature weight $\mathbf{m}_c^s$ is equivalent to adapting $\mathbf{W}$ to $\bar{\mathbf{W}}_c^s$ by using the weight $\bar{\mathbf{M}}_c^s$. Note that we simplify the task of adapting $\mathbf{W}$ by using the weight $\bar{\mathbf{M}}_c^s$ with all columns being the same $\mathbf{m}_c^s$, so that the number of variables (*i.e.*, size of $\mathbf{m}_c^s$) generated by $g(\cdot)$ is greatly reduced compared with the size of visual-semantic mapping.

In practice, we only need to learn $\mathbf{W}$ and $g(\cdot)$ without explicitly producing $\bar{\mathbf{W}}_c^s$. In the remainder of this section, we use implicit $\bar{\mathbf{W}}_c^s$ merely for the purpose of better explanation. Specifically, we expect implicit $\bar{\mathbf{W}}_c^s$ to satisfy two properties: generalizability and specificity, analogous to AEZSL in (4) (category-specific $\mathbf{W}^c$'s for specificity and co-regularizer for generalizability).

1) Generalizability: On one hand, we expect any $\bar{\mathbf{W}}_c^s$ can correctly classify the training instances from any category, which can be achieved by minimizing the square loss $\sum_{c=1}^{C^s} \|\mathbf{X}^{s\prime}\bar{\mathbf{W}}_c^s\mathbf{A}^s - \mathbf{Y}^s\|_F^2$ with $\mathbf{X}^s$ and $\mathbf{Y}^s$ being the same as defined in (4). After defining the one-hot label vector of $\mathbf{x}_i^s$ as $\mathbf{y}_i^s$ with the $c(i)$-th entry being 1 and $\mathbf{M}^s \in \mathcal{R}^{d\times C^s}$ as the aggregated feature weights over all C^s categories with the c-th column being $\mathbf{m}_c^s$, we can have

$$\begin{aligned}
&\sum_{c=1}^{C^s} \|\mathbf{X}^{s\prime}\bar{\mathbf{W}}_c^s\mathbf{A}^s - \mathbf{Y}^s\|_F^2, \\
&= \sum_{c=1}^{C^s}\sum_{i=1}^{n^s} \|\mathbf{x}_i^{s\prime}\bar{\mathbf{W}}_c^s\mathbf{A}^s - \mathbf{y}_i^s\|^2 \\
&= \sum_{i=1}^{n^s}\sum_{c=1}^{C^s} \|\mathbf{x}_i^{s\prime}(\bar{\mathbf{M}}_c^s \circ \mathbf{W})\mathbf{A}^s - \mathbf{y}_i^s\|^2 \\
&= \sum_{i=1}^{n^s}\sum_{c=1}^{C^s} \|(\mathbf{x}_i^{s'} \circ \mathbf{m}_c^{s'})\mathbf{W}\mathbf{A}^s - \mathbf{y}_i^s\|^2 \\
&= \sum_{i=1}^{n^s} \|(\bar{\mathbf{X}}_i^{s'} \circ \mathbf{M}^{s'})\mathbf{W}\mathbf{A}^s - \bar{\mathbf{Y}}_i^s\|_F^2,
\end{aligned} \tag{9}$$

in which $\bar{\mathbf{X}}_i^s \in \mathcal{R}^{d\times C^s}$ is horizontally stacked C^s copies of $\mathbf{x}_i^s$ and $\bar{\mathbf{Y}}_i^s \in \mathcal{R}^{C^s\times C^s}$ is vertically stacked C^s copies of $\mathbf{y}_i^s$. Hence, in our implementation, given the i-th training instance, we duplicate $\mathbf{x}_i^s$ and $\mathbf{y}_i^s$ to C^s copies, and apply the aggregated feature weights $\mathbf{M}^s$ over all C^s categories on the duplicated visual features $\bar{\mathbf{X}}_i^s$. Then, the decision value matrix of the i-th training instance $(\bar{\mathbf{X}}_i^{s'} \circ \mathbf{M}^{s'})\mathbf{W}\mathbf{A}^s$ can be easily obtained.

2) Specificity: On the other hand, we expect that $\bar{\mathbf{W}}_c^s$ can better fit the classification task corresponding to the c-th category. For the i-th training instance, we use $\mathbf{J}^i = (\bar{\mathbf{X}}_i^{s'} \circ \mathbf{M}^{s'})\mathbf{W}\mathbf{A}^s$ to denote its decision value matrix, in which $J^i_{c_1,c_2}$ is the decision value of c_2-th category obtained by using $\bar{\mathbf{W}}_{c_1}^s$. In the following, we focus on the decision values of its ground-truth category $c(i)$, *i.e.*, the $c(i)$-th column of $\mathbf{J}^i$. In this case, similar as in (7), the decision value of $c(i)$-th seen category obtained by using $\bar{\mathbf{W}}_{\tilde{c}}^s$ is

$$J^i_{\tilde{c},c(i)} = (\mathbf{x}_i^{s'} \circ \mathbf{m}_{\tilde{c}}^{s'})\mathbf{W}\mathbf{a}^s_{c(i)} = \mathbf{x}_i^{s'}\bar{\mathbf{W}}_{\tilde{c}}^s\mathbf{a}^s_{c(i)}. \tag{10}$$

Then, we expect $J^i_{c(i),c(i)}$ obtained by $\bar{\mathbf{W}}^s_{c(i)}$ should be larger than $J^i_{\tilde{c},c(i)}$ obtained by $\bar{\mathbf{W}}_{\tilde{c}}^s$ for $\tilde{c} \neq c(i)$. With this aim, we employ the hinge loss $R^i = \sum_{\tilde{c}\neq c(i)} \max(0, J^i_{\tilde{c},c(i)} - J^i_{c(i),c(i)} + \rho)$ to push $J^i_{c(i),c(i)}$ to be larger than $J^i_{\tilde{c},c(i)}$ by margin ρ for $\tilde{c} \neq c(i)$. In our experiments, we empirically set ρ as 0.5 considering that $\mathbf{J}^i$ is regressed to binary label matrix.

By taking both generalizability and specificity into consideration, and incorporating CNN used to extract visual features into our model, the loss function of our end-to-end DAEZSL model is designed as

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n^s} (\|\mathbf{J}^i - \bar{\mathbf{Y}}_i^s\|_F^2 + R^i), \tag{11}$$

in which $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\text{CNN}}, \boldsymbol{\theta}_{g(\cdot)}, \mathbf{W}\}$ with $\boldsymbol{\theta}_{\text{CNN}}$ (*resp.*, $\boldsymbol{\theta}_{g(\cdot)}$) being the parameters of CNN (*resp.*, $g(\cdot)$). Based on (11), we aim to have implicit $\bar{\mathbf{W}}_c^s$'s which can generalize to other categories by minimizing $\|\mathbf{J}^i - \bar{\mathbf{Y}}_i^s\|_F^2$ and simultaneously better fit the classification task corresponding to its own category by minimizing R^i. Note that semantic representations of unseen

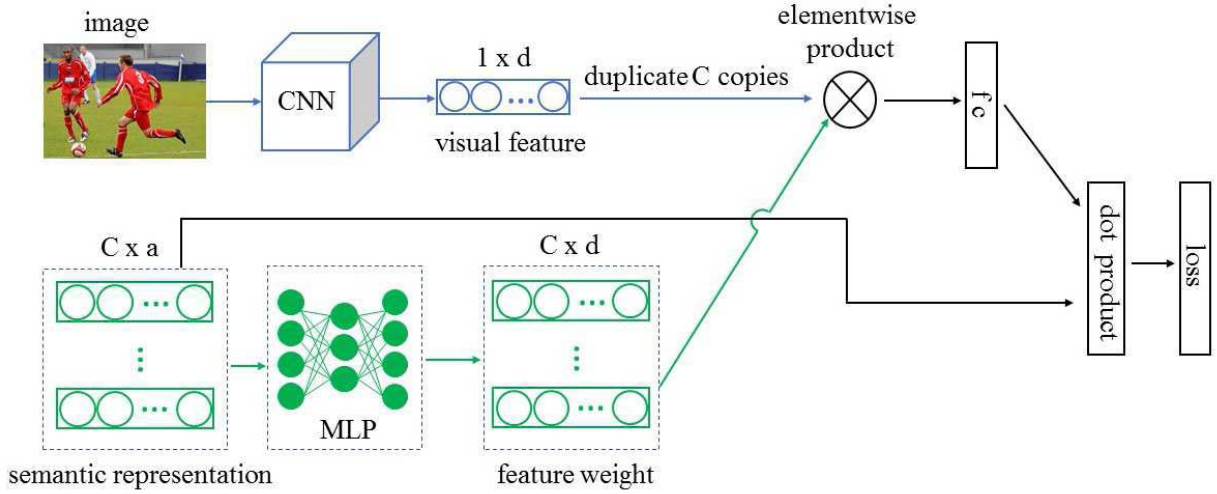

Fig. 1. Illustration of our Deep Adaptive Embedding Zero-Shot Learning (DAEZSL) model. In the top flow, the feature of each input image is duplicated to C copies. In the bottom flow, the semantic representations of all C categories pass through multi-layer perceptrons (MLP) and generate the feature weights for C categories, which are applied on the duplicated features via elementwise product. Then, the weighted features are fed into a fully connected (fc) layer (*i.e.*, visual-semantic mapping), followed by dot product with semantic representations, and finally output the decision value matrix, based on which we minimize the training loss in the training stage and predict test instances in the testing stage.

categories and unlabeled test instances are not utilized in the training stage.

Our DAEZSL model is illustrated in Fig. 1, from which we can see that $g(\cdot)$ is modeled by multi-layer perceptrons (MLP) and $\mathbf{W}$ is modeled by fully connected (fc) layer. Specifically, during the training process, we input the training images and the semantic representations of all C^s seen categories, *i.e.*, $\mathbf{A}^{s'} \in \mathcal{R}^{C^s \times a}$. In the top flow, d-dim feature of the i-th training image is duplicated to C^s copies, *i.e.*, $\bar{\mathbf{X}}_i^{s'} \in \mathcal{R}^{C^s \times d}$. In the bottom flow, $\mathbf{A}^{s'}$ passes through MLP and generates the feature weights $\mathbf{M}^{s'} \in \mathcal{R}^{C^s \times d}$ for C^s categories. After applying the generated feature weights on the duplicated visual features via elementwise product, the weighted features $\bar{\mathbf{X}}_i^{s'} \circ \mathbf{M}^{s'}$ are fed into fc layer (*i.e.*, the common visual-semantic mapping matrix $\mathbf{W}$), and output $(\bar{\mathbf{X}}_i^{s'} \circ \mathbf{M}^{s'})\mathbf{W}$. Finally, we perform dot product on $(\bar{\mathbf{X}}_i^{s'} \circ \mathbf{M}^{s'})\mathbf{W}$ and $\mathbf{A}^s$, leading to the decision value matrix $\mathbf{J}^i$, based on which we employ the loss function in (11). Note that we train an end-to-end system to minimize the loss function in (11), during which the parameters of CNN, MLP, and fc layer in Fig. 1 are jointly optimized. More details about the network architecture and training process can be found in Section V-B.

In the prediction stage, given an unseen category with semantic representation $\mathbf{a}_{\tilde{c}}^t$, we can easily generate its feature weight $\mathbf{m}_{\tilde{c}}^t$ via the projection function $g(\cdot)$, which implies an adaptive visual-semantic mapping $\bar{\mathbf{W}}_{\tilde{c}}^t$. Intuitively, if $\mathbf{a}_{\tilde{c}}^t$ is similar to $\mathbf{a}_c^s$ of the c-th seen category, their generated feature weights $\mathbf{m}_{\tilde{c}}^t$ and $\mathbf{m}_c^s$ should be similar. Furthermore, their visual-semantic mapping matrices $\bar{\mathbf{W}}_{\tilde{c}}^t$ and $\bar{\mathbf{W}}_c^s$ should be close to each other. Therefore, the visual-semantic mapping of a given unseen category is expected to be in correlation with those of similar seen categories, analogous to learning $\mathbf{W}^c$ based on the similarity matrix $\mathbf{S}^c$ in (4).

We further elaborate on the prediction procedure based on Fig. 1. In particular, given the i-th test image $\mathbf{x}_i^t$ and the set of unseen categories with size C^t, we use this test image and the semantic representations of all C^t unseen categories $\mathbf{A}^{t'} \in \mathcal{R}^{C^t \times a}$ as input. The test image passes through the top flow and generates C^t duplicated copies of visual features, *i.e.*, $\bar{\mathbf{X}}_i^{t'} \in \mathcal{R}^{C^t \times d}$, while $\mathbf{A}^{t'}$ passes through the bottom flow and generates the feature weights $\mathbf{M}^{t'} \in \mathcal{R}^{C^t \times d}$ for all unseen categories. After performing $(\bar{\mathbf{X}}_i^{t'} \circ \mathbf{M}^{t'})\mathbf{W}\mathbf{A}^t$, we can obtain a decision value matrix $\mathbf{J}^i \in \mathcal{R}^{C^t \times C^t}$ for the i-th test image. Finally, we get the diagonal of $\mathbf{J}^i$ as decision value vector and classify this test image as the category corresponding to the highest decision value. Note that only the diagonal of $\mathbf{J}^i$ is used for prediction because we assume that for the c-th unseen category, the decision value $J_{c,c}^i = \mathbf{x}_i^{t'} \bar{\mathbf{W}}_c^t \mathbf{a}_c^t$ (see (10)) obtained based on $\bar{\mathbf{W}}_c^t$ can best fit the classification task for this category.

3) Relation to ESZSL: When fixing the feature weights for all categories, *i.e.*, $\mathbf{M}^s$, as all-one matrix without learning MLP in Fig. 1, our DAEZSL model approximately reduces to ESZSL, in which the visual-semantic mappings of all categories are the same.

4) Relation to AEZSL: Our DAEZSL model shares the similar idea with our AEZSL method, that is, visual-semantic mapping should be category-specific and related to the semantic representation of each category. Besides, we design the training loss in (11) considering the generalizability and specificity of visual-semantic mappings, in analogy to learning category-specific $\mathbf{W}^c$'s with co-regularizer in (4). Moreover, in our DAEZSL model, the visual-semantic mapping of a given unseen category is expected to be close to those of seen categories with similar semantic representations, resembling the first regularizer based on the similarity matrix $\mathbf{S}^c$ in (4).

V. Experiments

In this section, we conduct experiments for image classification on three small-scale datasets (*e.g.*, CUB, SUN, and Dogs) and one large-scale dataset (*e.g.*, ImageNet). On the small-scale datasets, we compare our AEZSL and AEZSL_LR

methods with their special cases as well as standard/semi-supervised ZSL baseline methods. Note that our DAEZSL method tends to learn a mapping from semantic representation to feature weight, which is in demand of adequate seen categories. With scarce seen categories in the training stage, the learnt DAEZSL model cannot generalize well to unseen categories. Thus, we omit the results of DAEZSL on the small-scale datasets. On the large-scale dataset, due to the inefficiency of AEZSL for a large number of unseen categories, we only evaluate our DAEZSL method, which is specifically designed for large-scale ZSL, and compare with recently reported state-of-the-art results.

A. Zero-Shot Learning on Small-Scale Datasets

1) Experimental Settings: We conduct experiments on the following three popular benchmark datasets which are commonly used for zero-shot learning tasks:

- CUB [37]: Caltech-UCSD Bird (CUB) has in total $11,788$ images distributed in 200 bird categories. Following the ZSL setting in [38], we use the standard train-test split with 150 (*resp.*, 50) categories as seen (*resp.*, unseen) categories. The CUB dataset contains a 312-dim binary human specified attribute vector for each image, so we average the attribute vectors of the images within each category and use it as the semantic representation of that category.
- SUN [39]: Scene UNderstanding (SUN) dataset has 20 images in each scene category. Following the ZSL setting in [40], we use the provided list of 10 categories as unseen categories and the rest of 707 categories as seen categories. Similar to CUB, the averaged 102-dim attribute vector is used as the semantic representation for each category.
- Dogs [41]: ZSL was firstly performed on the Stanford Dogs dataset in [5], which uses $19,501$ images from 113 breeds of dogs. We follow the provided train-test split in [5], *i.e.*, 85 (*resp.*, 28) categories as seen (*resp.*, unseen) categories. Since there is no manually annotated attribute for the Dogs dataset, we combine two types of output embeddings learned from online corpus (*i.e.*, $3,850$-dim Bag-of-Words embedding and 163-dim WordNet-derived similarity embedding) as the semantic representation for each category, which has demonstrated superior performance in [5].

For CUB and SUN datasets, we extract the $4,096$-dim output of the 6-th layer of the pretrained VGG [45] as the visual feature for each image, following myriad of previous ZSL works such as [13], [14], [23], and [44]. For the Dogs dataset, we use the $1,024$-dim output of the top layer of the pretrained GoogleNet [46] following [5] and [22].

2) Baselines: We compare our AEZSL and AEZSL_LR methods with two sets of ZSL baselines: standard ZSL methods and semi-supervised/transductive ZSL methods. For the first set, we include the following methods [1], [5], [9], [11]–[14], [22], [23], [28], [42], [43], which do not utilize unlabeled test instances in the training stage. For the second set, we compare with the following transductive or semi-supervised ZSL methods [10], [15], [18], [19], [21], [25], [44], which utilize the unlabeled test instances and semantic representations of unseen categories in the training phase. Note that our AEZSL belongs to standard ZSL methods while AEZSL_LR belongs to semi-supervised ZSL methods.

Besides two sets of baselines mentioned above, we consider two special cases of our AEZSL method: AEZSL_sim without using the co-regularizer $\sum_{c<\bar{c}} \|\mathbf{W}^c - \mathbf{W}^{\bar{c}}\|_F^2$ by setting $\lambda_3 = 0$ and AEZSL_inf with λ_3 being very large ($\lambda_3 = 10^{10}$). We also report the results of one special cases of our AEZSL_LR method, namely, AEZSL_LR (one-step), which is a non-progressive approach by treating the entire test set as the unconfident set $\mathcal{U}$ and using the method in (6) without the regularizer $\|\mathbf{X}^{l'}\mathbf{P} - \mathbf{Y}^l\|_F^2$.

We use multi-class accuracy for performance evaluation. For the baselines, if the experimental settings (*i.e.*, train-test split, visual feature, semantic representation, evaluation metric, *etc*) mentioned in their papers are exactly the same as ours, we directly copy their reported results. Otherwise, we run their methods using our experimental setting for fair comparison.

3) Parameters: Our AEZSL method has three trade-off parameters λ_1, λ_2, and λ_3 in (4). Besides, our label refinement strategy has three additional trade-off parameters γ_1, γ_2, and γ_3 (see (6)). We use cross-validation strategy to determine the above trade-off parameters. Specifically, following [19], we choose the first C^c categories based on the default category indices from C^s seen categories as validation categories, in which C^c satisfies $\frac{C^c}{C^s} = \frac{C^t}{C^s+C^t}$. In our experiments, we first learn models based on the seen categories excluding the validation categories with λ_1, λ_2, and λ_3 set within the range $[10^{-3}, 10^{-2}, \ldots, 10^3]$, and then test the learnt models on the validation categories. After determining the optimal trade-off parameters through the grid search, we learn the final model based on all seen categories. Note that we also have a hyper parameter k, which is the number of selected confident test instances in each iteration. We empirically fix k as 100 for CUB and Dogs datasets, and 10 for the SUN dataset given that there are only 200 test instances in the SUN dataset.

Based on our experimental observation, our methods are relatively robust when the parameters are set within certain range. By taking λ_3 and k as examples, we vary λ_3 (*resp.*, k) in the range of $[10^{-1}, \ldots, 10^3]$ (*resp.*, $[80, 90, \ldots, 120]$) and evaluate our AEZSL_LR method on the Dogs dataset. The performance variance is illustrated in the middle and right subfigure in Fig. 5, from which we can see that our method is insensitive to λ_3 and k within certain range. We have similar observations for the other parameters on the other datasets.

4) Experimental Results: The experimental results for our AEZSL and AEZSL_LR methods as well as all the baselines are reported in Table I. It can be seen that ESZSL achieves competitive results compared with other standard ZSL baselines, which demonstrates the effectiveness of ESZSL despite its simplicity. By comparing AEZSL_sim with AEZSL, we observe that AEZSL achieves better results after employing the co-regularizer, so it is beneficial to encourage the mappings from different categories to share some common parts. We also observe that our AEZSL method not only outperforms ESZSL and AEZSL_sim, but also outperforms the standard ZSL

TABLE I

ACCURACIES (%) OF DIFFERENT BASELINE METHODS AND OUR METHODS ON THREE BENCHMARK DATASETS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Dataset	CUB	SUN	Dogs	Avg
ESZSL [24]	49.74	82.50	40.27	57.50
LatEm [22]	45.50	83.50	37.41	55.47
Zhang and Saligrama [23]	46.11	83.83	43.42	57.79
Bucher *et al.* [14]	43.29	84.41	36.18	54.63
AMP [9]	43.12	82.50	38.02	54.55
COSTA [11]	44.19	76.00	33.98	51.39
SSE [13]	40.64	82.50	38.63	53.92
SJE [5]	51.70	84.00	37.11	57.60
DAP/IAP [1]	41.40	72.00	37.11	50.17
SYNC [28]	54.50	83.50	42.01	60.50
ConSE [12]	37.33	73.00	23.49	44.61
RKT [42]	53.73	81.00	42.11	58.95
EXEM [43]	58.54	86.00	42.90	62.48
Li *et al.* [21]	43.94	87.50	43.50	58.31
Kodirov *et al.* [18]	57.42	86.00	48.59	64.00
Zhang and Saligrama [44]	56.90	89.50	48.53	64.98
Shojaee and Baghshah [19]	58.80	86.16	47.49	64.15
Li and Guo [10]	57.14	88.50	47.84	64.49
SMS [25]	57.14	83.50	47.95	62.86
Xu *et al.* [15]	55.74	85.50	43.09	61.44
AEZSL_inf	49.98	83.00	40.91	57.96
AEZSL_sim	57.55	87.50	43.98	63.01
AEZSL	59.73	88.00	44.62	64.12
AEZSL_LR (one-step)	60.08	89.00	48.90	65.99
AEZSL_LR	**64.44**	**92.50**	**50.62**	**69.18**

TABLE II

ACCURACIES (%) OF DIFFERENT METHODS UNDER THE NEW SETTING PROPOSED IN [47] ON CUB AND SUN DATASETS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Dataset	CUB2	SUN2
ESZSL [24]	53.9	54.5
ALE [38]	54.9	58.1
DeVISE [6]	52.0	56.5
SJE [5]	53.9	53.7
SYNC [28]	55.6	56.3
Zhang *et al.* [48]	57.1	61.7
SE-ZSL [49]	59.6	63.4
AEZSL	60.9	65.8
AEZSL_LR	**63.2**	**69.5**

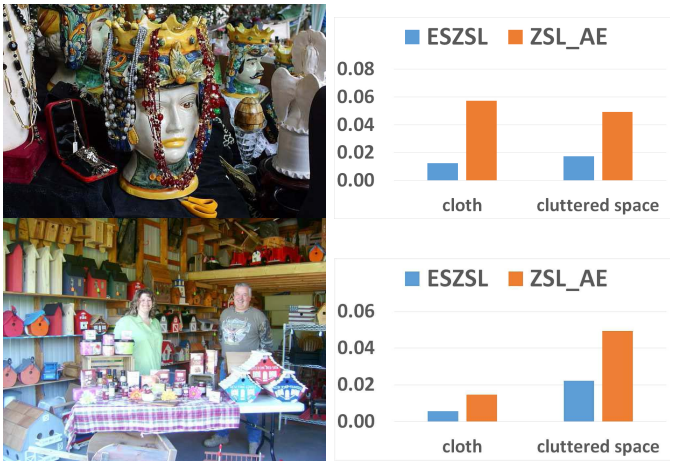

Fig. 2. Illustration of two instance images from the category "flea market" and their corresponding mapped values for the attributes "cloth" and "cluttered space" obtained by using ESZSL and our AEZSL method.

baselines [1], [5], [9], [11]–[14], [22], [23], [28], [42], [43], which indicates the advantage of learning a visual-semantic mapping for each unseen category.

From Table I, we also observe that the transductive or semi-supervised baselines [10], [15], [18], [19], [21], [25], [44] generally achieve better results than those standard ZSL baselines, indicating that it is helpful to utilize the unlabeled test instances in the training stage. Another observation is that our AEZSL_LR method outperforms AEZSL, which demonstrates the effectiveness of the progressive label refinement. Moreover, AEZSL_LR also performs better than AEZSL_LR (one-step). This is because the selected confident set with more accurate predicted labels can provide weak supervision. Finally, our AEZSL_LR method outperforms all the baselines and achieves the state-of-the-art results on three datasets, which again shows the advantage of adapting the visual-semantic mapping to each unseen category.

5) New Experimental Setting on CUB and SUN Datasets: In [47], new experimental settings on benchmark datasets were proposed for zero-shot learning. For fair comparison with the results reported in [47] and recent papers following the setting [47], we additionally conduct experiments using the training-testing split as well as ResNet visual features provided in [47]. The results of our methods on the CUB (*resp.*, SUN) dataset are reported in Table II, which are referred to as CUB2 (*resp.*, SUN2). We observe that our methods outperform the other baselines, which demonstrates the consistent superiority of our methods under different experimental settings.

6) Qualitative Analysis of the Attributes Learnt by AEZSL: In order to show the advantage of learning category-specific visual-semantic mappings intuitively, we take the SUN dataset as an example to investigate why our AEZSL can correctly classify the test instances which are misclassified by ESZSL. As illustrated in Figure 2, the two images are correctly classified as "flea market" by our AEZSL method but are misclassified as "shoe shop" by ESZSL. Based on the semantic representations of "flea market" and "shoe shop", we find that among the five attributes with maximum values in the semantic representation of "flea market", only two attributes (*i.e.*, "cloth" and "cluttered space") have larger values than those of "shoe shop". We simply treat these two attributes as the representative attributes to distinguish "flea market" from "shoe shop". Figure 2 shows the mapped values of the above two confusing images corresponding to two representative attributes obtained by using ESZSL and our AEZSL method, which are calculated by using $\mathbf{X}^{s\prime}\mathbf{W}$ and $\mathbf{X}^{s\prime}\mathbf{W}^c$ ($\mathbf{W}^c$ is the category-specific mapping matrix for "flea market") respectively. It can be seen that our AEZSL method obtains larger mapped values for the two representative attributes than ESZSL, which contributes to the correct classification of two confusing images as "flea market".

Based on the fact that AEZSL obtains larger mapped values corresponding to the two representative attributes, we conjecture that the learnt mapping using our method can bet-

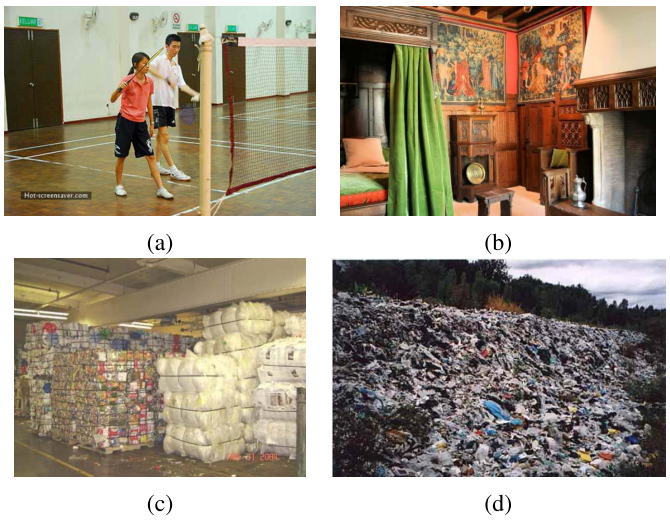

Fig. 3. The first (*resp.*, second) row contains the instance images from different categories with the attribute "cloth" (*resp.*, "cluttered space"). (a) Badminton court (indoor). (b) bedchamber. (c) recycling plant. (d) landfill.

ter capture the semantic meanings of "cloth" and "cluttered space". To verify this point, we first show some instance images from different categories with the attribute "cloth" (*resp.*, "cluttered space") in the first (*resp.*, second) row in Figure 3. Together with the instance images in Figure 2, we can observe that for different categories, the visual appearances of "cloth" and "cluttered space" are considerably different as discussed in Section I. Thus, a general mapping matrix learnt by ESZSL cannot tell the subtle discrepancy in the semantic meanings of the same attribute between different categories. In contrast, our AEZSL method learns the category-specific mapping matrix for "flea market" based on the similarities between this unseen category and each seen category, in which the major transfer comes from more similar seen categories. In Figure 4, we show the instance images from four nearest neighboring seen categories of "flea market" with the attributes "cloth" and "cluttered space", from which it can be seen that in terms of the visual appearances of "cloth" and "cluttered space", these neighboring categories resemble "flea market" much better than other categories such as those in Figure 3. Therefore, AEZSL can learn a better fitting visual-semantic mapping for "flea market". We have similar observations for the other unseen categories and on the other datasets.

7) Performance Variation w.r.t. *Iteration in AEZSL_LR:* Since our proposed label refinement method is a progressive approach, we are interested in the variation of the accuracy of the predicted test labels (*i.e.*, the union of $\mathbf{Y}^l$ and $\mathbf{Y}^u$) *w.r.t.* the number of iterations. By taking the Dogs dataset as an example, we plot the label accuracy *w.r.t.* the number of iterations in the left subfigure in Figure 5, from which we can observe that the accuracy of predicted test labels increases from 44.62% to 50.62% steadily within 50 iterations. Recall that in each iteration, we move the top k confident instances from the unconfident set with their refined predicted labels into the confident set. Thus, we can infer that in most iterations, the accuracy of the predicted labels of the selected k confident instances is improved using the updated visual classifiers, which verifies the effectiveness of progressive label refinement. We have similar observations on the other datasets.

8) Comparison Between Different Semantic Representations: In Table I, we use continuous attribute vector as the semantic representation on the CUB and SUN datasets. Instead, we can also use binary attribute vector or word vector. Specifically, we binarize the category-level continuous attribute vector based on the threshold 0 as the binary attribute vector and use 400-dim word2vec [50] provided in [22] corresponding to each category name. By taking CUB dataset as an example, we report the results in Table IV, in which the results obtained by using continuous attribute vector are significantly better. However, in the other two cases, our AEZSL method is still better than ESZSL and further improved by AEZSL_LR.

B. Zero-Shot Learning on Large-Scale Dataset

In this section, we apply our deep adaptive embedding model DAEZSL on the ImageNet dataset and compare with the state-of-the-art reported results.

1) Experimental Settings: We strictly follow the experimental settings in [6] and [28]. Specifically, we use the 1000 categories in ImageNet ILSVRC 2012 1K [51] as seen categories, and perform evaluation on three test scenarios, which are chosen from ImageNet 2011 21K dataset and built based on ImageNet label hierarchy with increasing difficulty. The three test sets are listed as follows.

- 2-hop test set: 1,509 unseen categories within two tree hops of the 1000 seen categories according to the ImageNet label hierarchy,[1] which are semantically and visually similar with 1000 seen categories.
- 3-hop test set: 7,678 unseen categories within three tree hops of 1000 seen categories, which is constructed in a similar way to 2-hop test set.
- "all" test set: all the 20,345 unseen categories in the ImageNet 2011 21K dataset which do not belong to the ILSVRC 2012 1K dataset.

Note that 2-hop (*resp.*, 3-hop) test set is a subset of 3-hop (*resp.*, "all") test set, and all the test sets have no overlap with the training set (*e.g.*, ImageNet ILSVRC 2012 1K).

2) Semantic Representations: For both seen categories and unseen categories, we use the 500-dim word vector for each category provided in [28], which is obtained based on a skip-gram language model [50] trained on the latest Wikipedia corpus. Note that one ImageNet category may have more than one word according to its synset, we simply average the word vectors of all the words appearing in its synset as the word vector for that category.

3) Evaluation Metrics: Evaluating ZSL methods on the large-scale ImageNet dataset is a nontrivial task considering the large number of unseen categories and the semantic overlap of different categories in the ImageNet label hierarchy. So we use different evaluation metrics compared with multi-class accuracy used on three small-scale datasets (*i.e.*, CUB, SUN, and Dog) in Section V-A. Following [6], we use two metrics

[1] http://www.image-net.org/api/xml/structure_released.xml

(a) (b) (c) (d)

Fig. 4. The instance images from four nearest neighboring seen categories of the unseen category "flea market" with the attributes "cloth" and "cluttered space". (a) Bazzar (indoor). (b) Thrift shop. (c) Market (indoor). (d) General store (indoor).

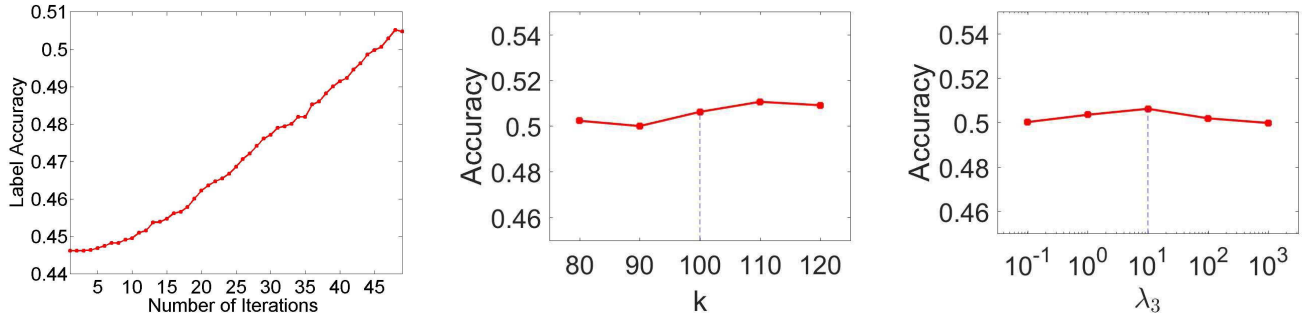

Fig. 5. The left subfigure shows the accuracy variation of predicted test labels *w.r.t.* the number of iterations in label refinement on the Dogs dataset. The middle (*resp.*, right) subfigure shows the performance variance of our AEZSL_LR method *w.r.t.* parameter k (*resp.*, λ_3) on the Dogs dataset, in which the dash line indicates the parameter we use in Table I.

Flat hit@K and Hierarchical precision@K for performance evaluation. To be exact, Flat hit@K is defined as the percentage of test instances for which the ground-truth category is in its top K predictions. When $K = 1$, Flat hit@K is identical with the multi-class accuracy. Compared with Flat hit@K, Hierarchical precision@K takes the hierarchical structure of categories into consideration. Given each test image, we generate a feasible set of K nearest categories of its ground-truth category in the ImageNet label hierarchy and calculate the overlap ratio between the feasible set and the top K predictions of this test image, that is, the precision of top K predictions. When generating a feasible set of K nearest categories for a ground-truth category, we enlarge the searching radius around the ground-truth category in the ImageNet label hierarchy iteratively and add the searched categories belonging to the test set to the feasible set. This procedure is repeated until the size of feasible set exceeds K. For more details, please see the Appendix of [6] or the Supplementary of [28]. Note that when $K = 1$, Flat hit@K is equal to Hierarchical precision@K, so we omit Hierarchical precision@1 in Table III to avoid redundancy.

4) Network Structure: In terms of CNN and MLP in Fig. 1, we use GoogleNet as the CNN structure in our experiments, which is initialized with released model [46] pretrained on ImageNet dataset. The dimension of output from CNN is $1,024$. The MLP we use has one hidden layer with its size empirically set as 750, which is approximately the average of feature dimension ($d = 1024$) and attribute dimension ($a = 500$), *i.e.*, $\lfloor \frac{d+a}{2} \rfloor$. Besides, we add a dropout layer following the hidden layer in MLP with 50% ratio of dropped output. Additionally, we add a sigmoid layer following MLP to normalize the feature weight in the range of $(0, 1)$. The network is implemented using TensorFlow and we use AdaGrad optimizer for training, with batchsize as 128 and learning rate as 0.001.

5) Experimental Results: To the best of our knowledge, there are few ZSL papers [6], [12], [28], [43] reporting their performances on the ImageNet dataset in 2-hop, 3-hop, and "all" test settings. We compare our DAESL method with the reported results of DeVISE [6], ConSE [12], EXEM [43], and SYNC [28] in Table III. We also include ESZSL as a baseline, in which we train the DAEZSL network using all-one feature weights without learning MLP. From Table III, we can observe that DAEZSL achieves far better results than ESZSL, which demonstrates the advantage of category-specific feature weight. Our DAEZSL also outperforms all the baseline methods in most cases (25 out of 27), indicating that it is beneficial to learn deep adaptive embedding model which can implicitly adapt visual-semantic mapping to different categories by using category-specific feature weight.

Additionally, in Table V, we report the results of our DAEZSL method without fine-tuning CNN model parameters. We also report the results without using ReLU activation in the hidden layer of MLP, as well as those with various numbers of hidden layers (*i.e.*, l_h) in MLP or various numbers of nodes in each hidden layer (*i.e.*, n_h). From Table V, we observe that DAEZSL ($l_h = 1$ or 2, $n_h = 750$) with ReLU in MLP and CNN fine-tuning generally performs more favorably. Recall that we employ MLP to learn a mapping from semantic representation to feature weight, which requires adequate seen categories in the training stage. With scarce seen categories in the training stage, the learnt MLP may not generalize

TABLE III

ACCURACIES (%) OF DIFFERENT BASELINE METHODS AND OUR DAEZSL METHOD ON THE IMAGENET DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Setting	Method	Flat Hit@K					Hierarchical precision@K			
		1	2	5	10	20	2	5	10	20
2-hop test	DeVISE [6]	6.0	10.0	18.1	26.4	36.4	15.2	19.2	21.7	23.3
	ConSE [12]	9.4	15.1	24.7	32.7	41.8	21.4	24.7	26.9	28.4
	SYNC [28]	10.5	16.7	28.6	40.1	52.0	25.1	27.7	30.3	32.1
	EXEM [43]	12.5	19.5	32.3	43.7	55.2	26.9	29.1	31.2	33.3
	ESZSL	8.6	13.8	24.1	35.0	47.4	21.8	24.7	27.4	30.2
	DAEZSL	**13.9**	**21.3**	**33.8**	**45.2**	**57.1**	**28.4**	**29.8**	**32.5**	**34.8**
3-hop test	DeVISE [6]	1.7	2.9	5.3	8.2	12.5	3.7	19.1	21.4	23.6
	ConSE [12]	2.7	4.4	7.8	11.5	16.1	5.3	20.2	22.4	24.7
	SYNC[28]	2.9	4.9	9.2	14.2	20.9	8.0	23.7	26.4	28.6
	EXEM [43]	3.6	5.9	10.7	16.1	23.1	**8.2**	25.3	27.8	30.1
	ESZSL	2.4	4.1	7.6	11.8	17.5	5.8	20.9	23.1	25.2
	DAEZSL	**4.3**	**6.6**	**11.9**	**17.6**	**24.5**	8.0	**26.7**	**28.9**	**31.8**
"all" test	DeVISE [6]	0.8	1.4	2.5	3.9	6.0	1.7	7.2	8.5	9.6
	ConSE [12]	1.4	2.2	3.9	5.8	8.3	2.5	7.8	9.2	10.4
	SYNC [28]	1.5	2.4	4.5	7.1	10.9	3.6	9.6	11.0	12.5
	EXEM [43]	1.8	2.9	5.3	8.2	12.2	**3.7**	10.4	12.1	13.5
	ESZSL	1.2	2.0	3.8	5.9	9.1	2.8	9.3	10.7	11.9
	DAEZSL	**2.0**	**3.2**	**5.9**	**8.7**	**13.6**	3.5	**11.8**	**12.9**	**14.7**

TABLE IV

ACCURACIES (%) OF DIFFERENT METHODS USING WORD VECTOR OR ATTRIBUTE VECTOR (BINARY OR CONTINUOUS) ON THE CUB DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Methods	ESZSL	AEZSL	AEZSL_LR
attribute (binary)	30.96	42.55	46.91
attribute (continuous)	**49.74**	**59.73**	**64.44**
word vector	34.84	45.28	48.04

TABLE V

ACCURACIES (%) OUR DAEZSL METHOD WITH DIFFERENT CONFIGURATIONS UNDER THE 2-HOP TEST SETTING ON THE IMAGENET DATASET. WE USE l_h TO DENOTE THE NUMBER OF HIDDEN LAYERS IN MLP AND n_h TO DENOTE THE NUMBER OF NODES IN EACH HIDDEN LAYER. NOTE THAT THE DEFAULT CONFIGURATION OF DAEZSL IS $l_h = 1$, $n_h = 750$. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Method	Flat Hit@K				
	1	2	5	10	20
DAEZSL ($l_h = 0$)	10.4	17.1	27.5	36.7	47.7
DAEZSL ($l_h = 1$, $n_h = 250$)	12.6	18.8	30.0	39.4	49.5
DAEZSL ($l_h = 1$, $n_h = 500$)	13.8	21.3	33.5	45.0	56.5
DAEZSL ($l_h = 1$, $n_h = 1000$)	13.8	21.2	33.7	44.8	55.9
DAEZSL ($l_h = 2$, $n_h = 750$)	13.9	20.9	33.6	**45.7**	**57.5**
DAEZSL (w/o ReLU in MLP)	13.6	21.1	33.1	44.2	55.8
DAEZSL (with fixed CNN)	**14.0**	21.1	33.5	45.1	56.7
DAEZSL	13.9	**21.3**	**33.8**	45.2	57.1

well to unseen categories in the testing stage. To verify this point, we vary the number of seen categories and plot the performance curve of AEZSL and DAEZSL in Fig. 6. Based on Fig. 6, we observe that DAEZSL can achieve better results than AEZSL with adequate seen categories while performing worse than AEZSL when the number of seen categories is very small.

C. Generalized Zero-Shot Learning

Most of existing ZSL methods assume that in the testing stage, the test instances only come from the unseen categories,

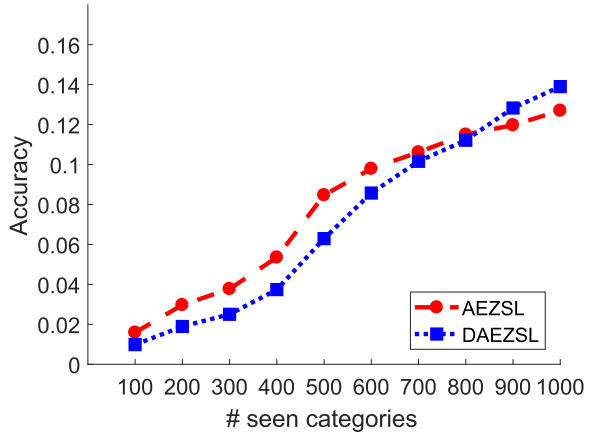

Fig. 6. Accuracy (*i.e.*, Flat Hit@1) of our AEZSL and DAEZSL methods under the 2-hop test setting on the ImageNet dataset with different numbers of seen categories in the training stage.

which is actually an unrealistic setting because the instances from seen categories may also be encountered in the testing stage. So it is more useful to predict a test instance from either seen categories or unseen categories instead of assuming that the test instances are only from unseen categories. However, when using the mixture of test instances from both seen and unseen categories for testing, the performance will be significantly degraded due to the bias of prediction scores between seen category label space and unseen category label space, as demonstrated in [7] and [52]. This more challenging test setting is referred to as generalized zero-shot learning (GZSL) in [52]. To validate the effectiveness of our AEZSL method under the more realistic GZSL setting, we additionally conduct experiments by mixing the test instances from both seen and unseen categories as the test set. In particular, we follow the setting in [52], that is, we move 20% of the instances from each seen category to the test set, and thus the test set includes both seen categories and unseen categories.

TABLE VI
ACCURACIES (%) OF DIFFERENT METHODS UNDER THE GENERALIZED ZSL SETTING ON FOUR BENCHMARK DATASETS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Dataset	CUB	SUN	Dogs	ImageNet
ESZSL	26.35	22.06	37.73	7.94
AEZSL	30.76	26.78	42.75	12.92
AEZSL(CS)	**42.08**	**36.53**	**43.66**	**13.29**
Chao *et al.* [52]	35.60	30.28	40.90	9.37

When applying our AEZSL method to generalized ZSL setting, we adopt the same strategy as in (4) and the only difference is that we learn $(C^s + C^t)$ instead of C^t category-specific visual-semantic mappings. Particularly, we learn category-specific mappings for both unseen and seen categories by assigning different weights on different classification tasks of different seen categories based on the similarity between each category and all the seen categories. For fair comparison with state-of-the-art results under the generalized ZSL setting, we further employ the existing calibrated stacking strategy proposed in [52] on the results obtained by our AEZSL method. The idea of calibrated stacking strategy is simple yet very effective, that is, to reduce the prediction scores in the seen category label space by a threshold, which is learnt based on a performance metric called Area Under SeenUnseen accuracy Curve (AUSUC) on the validation set, according to the observation that the prediction scores in the seen category label space are often higher than those in the unseen category label space. Thus, after employing the calibrated stacking strategy, we expect to obtain unbiased prediction scores.

In Table VI, we report the results obtained by ESZSL and our AEZSL method as well as our AEZSL after employing Calibrated Stacking (CS) strategy, which is referred to as AEZSL(CS). We also compare with [52], which is specifically designed for generalized ZSL. From Table VI, we observe that our AEZSL method outperforms ESZSL on all datasets, which shows that our AEZSL method is also effective under the generalized ZSL setting. Moreover, after employing the calibrated stacking strategy, the accuracy obtained by our AEZSL method is greatly improved, which demonstrates that our AEZSL method can be perfectly integrated with the existing calibrated stacking strategy. Finally, we observe that AEZSL(CS) achieves significantly better result than the other baselines on all datasets, which again indicates the effectiveness of our method under the generalized ZSL setting.

VI. CONCLUSION

In this paper, we have proposed our AEZSL method, which aims to learn a category-specific visual-semantic mapping for each unseen category based on the similarities between unseen categories and seen categories, followed by progressive label refinement. Moreover, we additionally propose a deep adaptive embedding model named DAEZSL for large-scale ZSL, which only needs to be trained once on the seen categories but can be applied to arbitrary number of unseen categories. Comprehensive experimental results demonstrate the effectiveness of our proposed methods.

APPENDIX

A. Solution to (4)

We rewrite the problem in (4) as follows,

$$\min_{\mathbf{W}^c} \frac{1}{2}\sum_{c=1}^{C^t}\|(\mathbf{X}^{s\prime}\mathbf{W}^c\mathbf{A}^s - \mathbf{Y}^s)\mathbf{S}^c\|_F^2 + \frac{\lambda_1}{2}\sum_{c=1}^{C^t}\|\mathbf{X}^{s\prime}\mathbf{W}^c\|_F^2 + \frac{\lambda_2}{2}\sum_{c=1}^{C^t}\|\mathbf{W}^c\|_F^2 + \frac{\lambda_3}{2}\sum_{c<\tilde{c}}\|\mathbf{W}^c - \mathbf{W}^{\tilde{c}}\|_F^2, \quad (12)$$

To solve the problem in (12), we update each $\mathbf{W}^c$ by fixing all the other $\mathbf{W}^{\tilde{c}}$'s for $\tilde{c} \neq c$ in an alternating fashion. Specifically, by setting the derivative of (12) *w.r.t.* $\mathbf{W}^c$ to zeros, we obtain the following equation:

$$(\mathbf{X}^s\mathbf{X}^{s\prime})\mathbf{W}^c(\mathbf{A}^s\mathbf{S}^c\mathbf{S}^{c\prime}\mathbf{A}^{s\prime} + \lambda_1\mathbf{I}) + ((C^t-1)\lambda_3 + \lambda_2)\mathbf{W}^c = \mathbf{X}^s\mathbf{Y}^s\mathbf{S}^c\mathbf{S}^{c\prime}\mathbf{A}^{s\prime} + \lambda_3\sum_{\tilde{c}\neq c}\mathbf{W}^{\tilde{c}}. \quad (13)$$

By denoting $\mathbf{L} = \mathbf{X}^s\mathbf{X}^{s\prime}$, $\mathbf{T} = \mathbf{A}^s\mathbf{S}^c\mathbf{S}^{c\prime}\mathbf{A}^{s\prime} + \lambda_1\mathbf{I}$, $\mathbf{N} = \mathbf{X}^s\mathbf{Y}^s\mathbf{S}^c\mathbf{S}^{c\prime}\mathbf{A}^{s\prime} + \lambda_3\sum_{\tilde{c}\neq c}\mathbf{W}^{\tilde{c}}$, and $\mu = (C^t-1)\lambda_3 + \lambda_2$, the problem in (13) becomes a special case of Sylvester equation *w.r.t.* $\mathbf{W}^c$ as follows,

$$\mathbf{L}\mathbf{W}^c\mathbf{T} + \mu\mathbf{W}^c = \mathbf{N}. \quad (14)$$

Since $\mathbf{L}$ and $\mathbf{T}$ are symmetric real matrices, the problem in (14) has an efficient solution. Inspired by Simoncini [53], we perform Singular Value Decomposition (SVD) on $\mathbf{L}$ and $\mathbf{T}$, *i.e.*, $\mathbf{L} = \hat{\mathbf{U}}\mathbf{\Sigma}^L\hat{\mathbf{U}}'$ and $\mathbf{T} = \hat{\mathbf{V}}\mathbf{\Sigma}^T\hat{\mathbf{V}}'$ with $\mathbf{\Sigma}^L$ (*resp.*, $\mathbf{\Sigma}^T$) being a diagonal matrix, in which the i-th diagonal element is σ_i^L (*resp.*, σ_i^T). Then, we can rewrite (14) as

$$\hat{\mathbf{U}}\mathbf{\Sigma}^L\hat{\mathbf{U}}'\mathbf{W}^c\hat{\mathbf{V}}\mathbf{\Sigma}^T\hat{\mathbf{V}}' + \mu\mathbf{W}^c = \mathbf{N}. \quad (15)$$

After multiplying (15) by $\hat{\mathbf{U}}'$ (*resp.*, $\hat{\mathbf{V}}$) on the left (*resp.*, right) and considering the orthonormality of $\hat{\mathbf{U}}$ (*resp.*, $\hat{\mathbf{V}}$), we have

$$\mathbf{\Sigma}^L\hat{\mathbf{U}}'\mathbf{W}^c\hat{\mathbf{V}}\mathbf{\Sigma}^T + \mu\hat{\mathbf{U}}'\mathbf{W}^c\hat{\mathbf{V}} = \hat{\mathbf{U}}'\mathbf{N}\hat{\mathbf{V}}. \quad (16)$$

By denoting $\hat{\mathbf{W}}^c = \hat{\mathbf{U}}'\mathbf{W}^c\hat{\mathbf{V}}$ and $\hat{\mathbf{N}} = \hat{\mathbf{U}}'\mathbf{N}\hat{\mathbf{V}}$, we can arrive at

$$\mathbf{\Sigma}^L\hat{\mathbf{W}}^c\mathbf{\Sigma}^T + \mu\hat{\mathbf{W}}^c = \hat{\mathbf{N}}. \quad (17)$$

Based on (17), we can easily solve each entry in $\hat{\mathbf{W}}^c$ as

$$\hat{W}_{ij}^c = \frac{\hat{N}_{ij}}{\sigma_i^L\sigma_j^T + \mu}.$$

Note that $\mathbf{L}$ is positive semi-definite, $\mathbf{T}$ is positive definite, and $\mu > 0$, so $\sigma_i^L\sigma_j^T + \mu \neq 0,\ \forall i, j$, and the problem in (17) has unique solution. Then, we can recover $\mathbf{W}^c$ by using $\mathbf{W}^c = \hat{\mathbf{U}}\hat{\mathbf{W}}^c\hat{\mathbf{V}}'$. Because SVD on $\mathbf{L}$ and $\mathbf{T}$ can be precomputed, our algorithm is quite efficient. We update each $\mathbf{W}^c$ alternatively until the objective of (12) converges. We name this method as Adaptive Embedding ZSL (AEZSL) and the algorithm to solve (12) is listed in Algorithm 1.

Algorithm 1 The Algorithm to Solve AEZSL (12)

Input: $\mathbf{X}^s, \mathbf{Y}^s, \mathbf{A}^s, \mathbf{S}^c, \lambda_1, \lambda_2, \lambda_3$
Initialize all $\mathbf{W}^c$'s equally using ESZSL [24].
repeat
  for $c = 1 : C^t$ **do**
    Update $\mathbf{W}^c$ by solving (14).
  end for
until The objective of (12) converges.
Output: $\mathbf{W}^c$'s.

Algorithm 2 The Algorithm of Progressive Label Refinement

Input: $\mathbf{X}^u, \mathbf{Y}^u, \hat{\mathbf{S}}, k, \gamma_1, \gamma_2, \gamma_3$
Initialize $\mathcal{L}$ as $\emptyset$ and $\mathcal{U}$ as the entire test set. Initialize $\mathbf{P}$ based on the learnt $\mathbf{W}^c$'s from Algorithm 1.
repeat
  Move k most confident instances selected by $\mathbf{P}$ from $\mathcal{U}$ to $\mathcal{L}$. Update $\mathbf{X}^l$, $\mathbf{X}^u$, $\mathbf{Y}^l$, $\mathbf{Y}^u$, $\mathbf{H}^u$ accordingly.
  repeat
    Update $\mathbf{D}$ based on its definition below (19).
    Update $\mathbf{P}$ by using (20).
  until The objective of (18) converges.
until The unconfident set $\mathcal{U}$ is empty.
Output: $\mathbf{Y}^l$, $\mathbf{P}$.

B. Solution to (6)

We rewrite the problem in (6) as follows,

$$\min_{\mathbf{P}} \frac{1}{2}\|\mathbf{X}^{l\prime}\mathbf{P} - \mathbf{Y}^l\|_F^2 + \frac{\gamma_1}{2}\|\mathbf{X}^{u\prime}\mathbf{P} - \mathbf{Y}^u\|_{2,1} - \gamma_2 \mathrm{tr}(\mathbf{Y}^u\hat{\mathbf{S}}\mathbf{P}'\mathbf{X}^u) + \frac{\gamma_3}{2}\mathrm{tr}(\mathbf{P}'\mathbf{X}^u\mathbf{H}^u\mathbf{X}^{u\prime}\mathbf{P}). \quad (18)$$

According to [54], solving (18) is equivalent to solving the following problem iteratively:

$$\min_{\mathbf{P}} \frac{1}{2}\|\mathbf{X}^{l\prime}\mathbf{P} - \mathbf{Y}^l\|_F^2 + \frac{\gamma_1}{2}\mathrm{tr}((\mathbf{P}'\mathbf{X}^u - \mathbf{Y}^{u\prime})\mathbf{D}(\mathbf{X}^{u\prime}\mathbf{P} - \mathbf{Y}^u)) - \gamma_2 \mathrm{tr}(\mathbf{Y}^u\hat{\mathbf{S}}\mathbf{P}'\mathbf{X}^u) + \frac{\gamma_3}{2}\mathrm{tr}(\mathbf{P}'\mathbf{X}^u\mathbf{H}^u\mathbf{X}^{u\prime}\mathbf{P}), \quad (19)$$

where $\mathbf{D}$ is a diagonal matrix with the i-th diagonal element being $\frac{1}{2\|\mathbf{q}_i\|_2}$, in which $\mathbf{q}_i$ is the i-th row of $\mathbf{Q}$ with $\mathbf{Q} = \mathbf{X}^{u\prime}\mathbf{P} - \mathbf{Y}^u$. In each iteration of solving $\mathbf{P}$, we calculate $\mathbf{D}$ based on previous $\mathbf{P}$ and then update $\mathbf{P}$ by solving (19). As claimed in [54], the updated $\mathbf{P}$ in each iteration will decrease the objective of (18).

By setting the derivative of (19) *w.r.t.* $\mathbf{P}$ to zeros, we can obtain the close-form solution to $\mathbf{P}$ as

$$\mathbf{P} = (\mathbf{X}^l\mathbf{X}^{l\prime} + \gamma_1\mathbf{X}^u\mathbf{D}\mathbf{X}^{u\prime} + \gamma_3\mathbf{X}^u\mathbf{H}^u\mathbf{X}^{u\prime} + \nu\mathbf{I})^{-1} \times (\mathbf{X}^l\mathbf{Y}^l + \gamma_1\mathbf{X}^u\mathbf{D}\mathbf{Y}^u + \gamma_2\mathbf{X}^u\mathbf{Y}^u\hat{\mathbf{S}}), \quad (20)$$

in which ν is a very small number (*i.e.*, $\nu \to 0$). We update $\mathbf{P}$ by solving (19) iteratively until the objective of (18) converges. According to [54], the output $\mathbf{P}$ is the global optimum solution to (18) because $\frac{1}{2}\|\mathbf{X}^{l\prime}\mathbf{P} - \mathbf{Y}^l\|_F^2 - \gamma_2\mathrm{tr}(\mathbf{Y}^u\hat{\mathbf{S}}\mathbf{P}'\mathbf{X}^u) + \frac{\gamma_3}{2}\mathrm{tr}(\mathbf{P}'\mathbf{X}^u\mathbf{H}^u\mathbf{X}^{u\prime}\mathbf{P})$ is convex *w.r.t.* $\mathbf{P}$. The whole algorithm of the proposed progressive label refinement is summarized in Algorithm 2.

REFERENCES

[1] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 453–465, Mar. 2014.
[2] Y. Guo, G. Ding, J. Han, and Y. Gao, "Zero-shot learning with transferred samples," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3277–3290, Jul. 2017.
[3] C. Luo, Z. Li, K. Huang, J. Feng, and M. Wang, "Zero-shot learning via attribute regression and class prototype rectification," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 637–648, Feb. 2018.
[4] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Conf. CVPR*, Miami, FL, USA, Jun. 2009, pp. 1778–1785.
[5] Z. Akata, S. Reed, and D. Walter, "Evaluation of output embeddings for fine-grained image classification," in *Proc. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 2927–2936.
[6] A. Frome, G. S. Corrado, and J. Shlens, "Devise: A deep visual-semantic embedding model," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2013, pp. 2121–2129.
[7] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Proc. 27th Annu. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2013, pp. 935–943.
[8] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S. F. Chang, "Designing category-level attributes for discriminative visual recognition," in *Proc. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 771–778.
[9] Z. Fu, T. Xiang, and E. Kodirov, "Zero-shot object recognition by semantic manifold distance," in *Proc. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 2635–2644.
[10] X. Li and Y. Guo, "Max-margin zero-shot learning for multi-class classification," in *Proc. Int. Conf. Art. Intell. Stat.*, San Diego, CA, USA, May 2015, pp. 626–634.
[11] T. Mensink, E. Gavves, and C. G. M. Snoek, "COSTA: Co-occurrence statistics for zero-shot classification," in *Proc. 27th IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 2441–2448.
[12] M. Norouzi *et al.*, "Zero-shot learning by convex combination of semantic embeddings," in *Proc. Int. Conf. Learn. Represent.*, Banff, AB, Canada, Apr. 2014.
[13] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4166–4174.
[14] M. Bucher, S. Herbin, and F. Jurie, "Improving semantic embedding consistency by metric learning for zero-shot classiffication," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 730–746.
[15] X. Xu, T. Hospedales, and S. Gong, "Transductive zero-shot action recognition by word-vector embedding," *Int. J. Comput. Vis.*, vol. 123, no. 3, pp. 309–333, 2017.
[16] D. Jayaraman, F. Sha, and K. Grauman, "Decorrelating semantic visual attributes by resisting the urge to share," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 1629–1636.
[17] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 4447–4456.
[18] E. Kodirov, T. Xiang, and Z. Fu, "Unsupervised domain adaptation for zero-shot learning," in *Proc. Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 2452–2460.
[19] S. M. Shojaee and M. S. Baghshah. (2016). "Semi-supervised zero-shot learning by a clustering-based approach." [Online]. Available: https://arxiv.org/abs/1605.09016
[20] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong, "Transductive multi-view embedding for zero-shot recognition and annotation," in *Proc. 13th Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Sep. 2014, pp. 584–599.
[21] X. Li, Y. Guo, and D. Schuurmans, "Semi-supervised zero-shot classification with label representation learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4211–4219.
[22] Y. Q. Xian, Z. Akata, and G. Sharma, "Latent embeddings for zero-shot classification," in *Proc. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 69–77.
[23] Z. Zhang and V. Saligrama, "Zero-shot learning via joint latent similarity embedding," in *Proc. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 6034–6042.
[24] B. Romera-Paredes and P. H. S. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, Jul. 2015, pp. 2152–2161.

[25] Y. Guo, G. Ding, X. Jin, and J. Wang, "Transductive zero-shot recognition via shared model space learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, Feb. 2016, pp. 3434–3500.
[26] Y. Long, L. Liu, and L. Shao, "Attribute embedding with visual-semantic ambiguity removal for zero-shot learning," in *Proc. Brit. Mach. Vis. Conf.*, York, U.K., Sep. 2016, pp. 40.1–40.11.
[27] Z. Al-Halah, M. Tapaswi, and R. Stiefelhagen, "Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 5975–5984.
[28] S. Changpinyo, W. L. Chao, and B. Gong, "Synthesized classifiers for zero-shot learning," in *Proc. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 5327–5336.
[29] M. Ye and Y. Guo, "Zero-shot classification with discriminative semantic representation learning," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 7140–7148.
[30] S. Changpinyo, W.-L. Chao, and F. Sha, "Predicting visual exemplars of unseen classes for zero-shot learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 3496–3505.
[31] M. Rohrbach, S. Ebert, and B. Schiele, "Transfer learning in a transductive setting," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2013, pp. 46–54.
[32] Y. Yang and T. M. Hospedales, "A unified perspective on multi-domain and multi-task learning," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015.
[33] L. Ba, K. Swersky, and S. Fidler, "Predicting deep zero-shot convolutional neural networks using textual descriptions," in *Proc. Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 4247–4255.
[34] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Apr. 2017, pp. 2021–2030.
[35] R. Qiao, L. Liu, and C. Shen, "Less is more: Zero-shot learning from online textual documents with noise suppression," in *Proc. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 2249–2257.
[36] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, New York, NY, USA, Jul. 1998, pp. 92–100.
[37] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-UCSD birds-200-2011 dataset," California Institute of Technology, Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.
[38] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for attribute-based classification," in *Proc. 26th IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 819–826.
[39] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 3485–3492.
[40] D. Jayaraman and K. Grauman, "Zero-shot recognition with unreliable attributes," in *Proc. 28th Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 3464–3472.
[41] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit. Workshop Fine-Grained Vis. Categorization (FGVC)*, Colorado Springs, CO, USA, Jun. 2011.
[42] D. Wang, Y. Li, Y. Lin, and Y. Zhuang, "Relational knowledge transfer for zero-shot learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, Feb. 2016, pp. 2145–2151.
[43] S. Changpinyo, W.-L. Chao, and F. Sha, "Predicting visual exemplars of unseen classes for zero-shot learning," in *Proc. 16th Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 3476–3485.
[44] Z. Zhang and V. Saligrama, "Zero-shot recognition via structured prediction," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 533–548.
[45] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[46] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. CVPR*, Boston, MA, USA, Jun. 2015, pp. 1–9.
[47] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning—The good, the bad and the ugly," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3077–3086.
[48] H. Zhang and P. Koniusz, "Zero-shot kernel learning," in *Proc. 31st IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, Jun. 2018, pp. 7670–7679.
[49] V. K. Verma, G. Arora, A. Mishra, and P. Rai, "Generalized zero-shot learning via synthesized examples," in *Proc. 31th IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, Jun. 2018, pp. 4281–4289.
[50] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2013, pp. 3111–3119.
[51] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[52] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 52–68.
[53] V. Simoncini, "Computational methods for linear matrix equations," *SIAM Rev.*, vol. 58, no. 3, pp. 377–441, 2016.
[54] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint ℓ2,1-norms minimization," in *Proc. 24th Annu. Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2010, pp. 1813–1821.

Li Niu received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2011, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2017. He was a Post-Doctoral Associate with Rice University, Houston, TX, USA. He is currently an Associate Professor with the Computer Science and Engineering Department, Shanghai Jiao Tong University, Shanghai, China. His current research interests include machine learning, deep learning, and computer vision.

Jianfei Cai (S'98–M'02–SM'07) received the Ph.D. degree from the University of Missouri–Columbia. He is currently a Professor and the Head of the Visual and Interactive Computing Division, School of Computer Science and Engineering, Nanyang Technological University, Singapore, where he is also the Head of the Computer Communication Division. He has authored over 200 technical papers in international journals and conferences. His major research interests include computer vision, multimedia, and machine learning. He is currently an AE of the IEEE TMM. He served as an AE for the IEEE TIP and TCSVT.

Ashok Veeraraghavan received the bachelor's degree in electrical engineering from IIT Madras in 2002 and the M.S. and Ph.D. degrees from the Department of Electrical and Computer Engineering, University of Maryland, College Park, in 2004 and 2008, respectively. He spent three wonderful and fun-filled years as a Research Scientist with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. He is currently an Associate Professor of electrical and computer engineering with Rice University, TX, USA, where he directs the Computational Imaging and Vision Lab. His research interests are broadly in the areas of computational imaging, computer vision, and robotics. His thesis received the Doctoral Dissertation Award from the Department of Electrical and Computer Engineering, University of Maryland. His work received numerous awards, including the NSF CAREER Award in 2017 and the Hershel M. Rich Invention Award in 2016 and 2017.

Liqing Zhang received the Ph.D. degree from Zhongshan University, Guangzhou, China, in 1988. He was with the South China University of Technology, where he was promoted to a Full Professor in 1995. He was a Research Scientist with the RIKEN Brain Science Institute, Japan, from 1997 to 2002. He is currently a Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. He has authored over 250 papers in journals and international conferences. His current research interests cover computational theory for cortical networks, brain–computer interface, visual perception and image search, and statistical learning and inference.