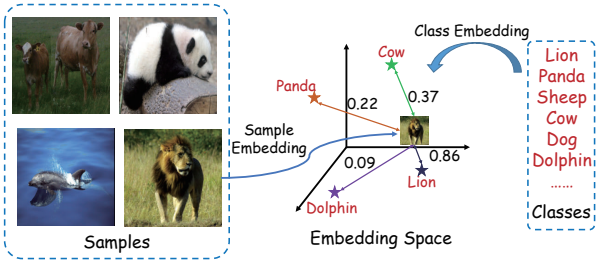
Zero-shot Learning with Transferred Samples

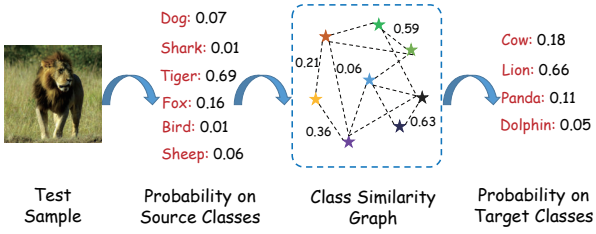
Yuchen Guo, Guiguang Ding, Jungong Han, and Yue Gao

***Abstract*—By transferring knowledge from the abundant labeled samples of known source classes, zero-shot learning (ZSL) makes it possible to train recognition models for novel target classes that have no labeled samples. Conventional ZSL approaches usually adopt a two-step recognition strategy, in which the test sample is projected into an intermediary space in the first step, and then the recognition is carried out by considering the similarity between the sample and target classes in the intermediary space. Due to this redundant intermediate transformation, information loss is unavoidable, thus degrading the performance of overall system. Rather than adopting this two-step strategy, in this paper, we propose a novel *one-step* recognition framework that is able to perform recognition in the original feature space by using directly trained classifiers. To address the lack of labeled samples for training supervised classifiers for the target classes, we propose to transfer samples from source classes with pseudo labels assigned, in which the transferred samples are selected based on their transferability and diversity. Moreover, to account for the unreliability of pseudo labels of transferred samples, we modify the standard SVM formulation such that the unreliable positive samples can be recognized and suppressed in the training phase. The entire framework is fairly general with the possibility of further extensions to several common ZSL settings. Extensive experiments on four benchmark datasets demonstrate the superiority of the proposed framework, compared to the state-of-the-art approaches, in various settings.**

***Index Terms*—Zero-shot Learning, Transfer Learning, Robust SVM, Inductive Learning, Transductive Learning, Experiment**



(a) Embedding-space based ZSL



(b) Class-similarity based ZSL

Fig. 1: Two existing ZSL frameworks. Both of them adopt a two-step strategy. Firstly, the test image is transformed into a intermediary (semantic/distribution) space. Secondly, the final prediction for target classes is generated by considering the relationship between the sample and target classes in the space.

I. Introduction

WHEN training a supervised classifier, sufficient labeled samples for target classes are required in the conventional supervised learning framework [1]. However, collecting and labeling a large quantity of samples is quite expensive in many cases. For instance, many objects "in the wild" follow a long-tailed distribution such that they do not occur frequently enough to collect and label a large set of representative exemplars to build the corresponding recognizers [2]. To address this problem, zero-shot learning (ZSL), which transfers knowledge from abundantly labeled source classes to help build classifiers for target classes that have no labeled samples available, has recently attracted considerable attention from computer vision and machine learning communities [3].

The task of ZSL is generally described as follows. There are no labeled samples for target classes but abundant labeled samples for source classes in the training phase, where source classes and target classes are different but related via some auxiliary information. ZSL constructs a classifier to predict the presence or absence of target classes for a test sample. The key to ZSL is how to effectively transfer knowledge between source classes and target classes. Existing ZSL approaches generally follow two kinds of **two-step** frameworks based on what relationship between the classes is given. The first is embedding-space based approaches [4], [5], [6], [7], [8], [9], [10], [11], [12] in which each class/label is represented by an attribute vector that represents the shared attributes between classes [4] or a word semantic embedding [5] learned from a large text corpus . In this way, both source and target classes are embedded into a shared vector space. Then, by utilizing the fully labeled samples from source classes, a sample embedding function can be learned to project a sample from its feature space into the shared embedding space. Because the classes are related in the shared space, the function learned from the source classes can also work for the target classes. In the test stage, the learned function, in turn, projects a given test sample into the embedding space, on which the similarity/distance between target classes can be measured. Differently, class-similarity based approaches subtly explore the availability of the similarity among all source and target classes [3], [13], [14]. Specifically, a n-way classifier for the source classes is learned from the labeled data at first. Then given a test

This research was supported by the National Natural Science Foundation of China Grant No. 61571269 and 61671267, and the Royal Society Newton Mobility Grant IE150997.

Yuchen Guo, Guiguang Ding, and Yue Gao are with the School of Software, Tsinghua University, Beijing China. Email: yuchen.w.guo@gmail.com, {dinggg, gaoyue}@tsinghua.edu.cn. Corresponding author: Guiguang Ding.

Jungong Han is with the Northumbria University, Newcastle, UK. Email: jungong.han@northumbria.ac.uk.

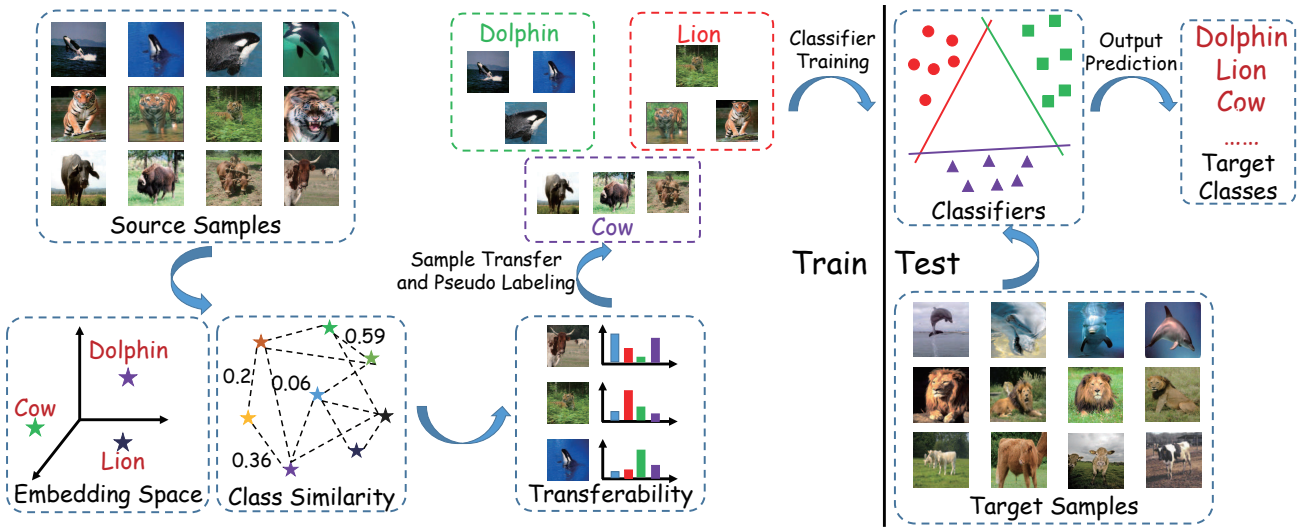


Fig. 2: Overview of the proposed sample-transfer based framework. In the training stage, we compute the transferability of each source sample for each target class via the embedding space or the class similarity. Based on the transferability and diversity, some source samples are selected for each target class and assigned by the corresponding pseudo labels. Then with the transferred samples and pseudo labels, we can train a supervised classifier to perform sample-to-class classification in one step. In the test stage, the classifier takes a target sample as the input and directly outputs the prediction on the target classes.

sample, its probability distribution on the source classes can be produced by the learned classifier. Based on the class similarity, the probability distribution on the target classes is computed. The basic ideas of these two frameworks are illustrated in Fig. 1(a) and Fig. 1(b) respectively, on which it is clear that they both adopt a two-step strategy and there is a need for an intermediate transformation in the test stage.

A. Motivation and Contribution

Despite the fact that the intermediate transformation efficiently bridges these two steps, this additive procedure inevitably causes the information loss, thus degrading the performance of overall system. In this paper, we propose a novel **one-step** ZSL framework which directly predicts the class of a sample without using the intermediary space in the test stage. Just like in the conventional supervised learning, the sample-to-class prediction is performed directly in the original feature space by normal classifiers. Without using the intermediary space in our framework, the unnecessary information leak can be avoided, thus achieving better performance. However, it has to confront the problem that no labeled sample is available for the target classes. To address this problem, as opposed to the **class-based** transfer in the existing approaches, we propose a novel **sample-based** transfer method for training, in which some samples belonging to the source classes are selected and assigned with pseudo labels for each target class. Having the transferred samples and their corresponding pseudo labels in place enables any supervised learning algorithms, e.g., SVM, to train one-step classifiers for the target classes. Here, it is noted that our approach also adopts an embedding space-like or class similarity-like intermediate space for training in order to be compatible to the most existing settings. However, such a space is not required in the online test phase at all, since our classifier can directly predict the labels for the target samples, which is completely different from existing ZSL approaches. We briefly illustrate the whole proposed framework in Fig. 2.

Specifically, we have to tackle two problems arisen in our proposed framework. Firstly, we need a metric that helps select proper samples from the source classes for pseudo labeling. In our design, both transferability and diversity are taken into account, in which the former considers the probabilities of a source sample being assigned to different classes while the latter one takes care of the distribution diversity of the selected samples. Here, transferability reflects whether we should transfer a source sample into a target class. In fact, a sample labeled as one class may have high probability to be another class too. For instance, an image showing a man riding a horse and labeled as "human", can also be labeled as "horse" and even "grass". In addition, a proper "tiger" image can also contribute to building a classifier that intends to distinguish "lion" from some other kinds of animals. It is believed that such a scheme is very logical because human being often uses one category to deduce another one as long as they share the same characteristics. The transferability is measured by the probability that it belongs to a target class via the auxiliary information, i.e., embedding space or class similarity. We also consider the diversity with the objective of selecting samples that have less redundancy. The reason can be explained in the following example. Suppose that there is a high transferability image, it is very likely that another image similar to it will also have high transferability. If they are both selected, the transferred samples will be redundant and thus cannot comprehensively capture the characteristics of the target class. Therefore, we require the selected samples to be diverse to some extent. The second problem is the label noise due to the fact that the pseudo labels are not the true

labels. To alleviate the influence of the label noise on classifier training, we propose to modify the objective function of the SVM classifier which results in a more robust SVM classifier. In summary, we make the following contributions in this paper:

1. Unlike existing two-step frameworks, we propose a novel one-step framework with transferred samples. The classifiers are directly trained in the original feature space using pseudo labels as in conventional supervised learning. In the test stage, such classifiers allow to directly predict the labels from test samples without using an intermediate space, which helps to avoid the unnecessary information loss in the whole procedure.

2. To select good samples for transferring and pseudo labeling, we consider the transferability and the diversity of the selected samples. The selection is formulated as a quadratic optimization problem and an efficient solution based on the augmented Lagrange multiplier framework [15] is proposed.

3. To cope with the label noise existing in the pseudo labels, we modify the standard SVM and adopt a more robust formulation that is able to suppress the unreliability of the transferred positive samples, thereby yielding better results.

4. The framework is generic in the sense that it can be easily extended to various settings including both inductive ones and transductive ones. To this end, our system selectively takes either class embedding or class similarity depending on the settings for training . To the best of our knowledge, this is the first ZSL framework that can be applied to all these settings.

5. We conduct extensive experiments on several benchmark datasets. The results demonstrate that the proposed one-step framework outperforms state-of-the-art two-step approaches.

II. Related Work

A. Problem and Notation

The problem of ZSL is defined as follows. We have a set of source classes $\mathcal{C}^s = \{c_1^s, ..., c_{k_s}^s\}$ together with n_s labeled source samples $\mathcal{D}^s = \{(\mathbf{x}_1^s, \mathbf{y}_1^s), ..., (\mathbf{x}_{n_s}^s, \mathbf{y}_{n_s}^s)\}$, where $\mathbf{x}_i^s \in \mathbb{R}^d$ is the feature vector and $\mathbf{y}_i^s \in \{0,1\}^{k_s}$ is the corresponding label vector which has $y_{ij}^s = 1$ if the sample i belongs to class c_j^s or 0 otherwise. We are given some target samples $\mathcal{D}^t = \{\mathbf{x}_1^t, ..., \mathbf{x}_{n_t}^t\}$ from k_t target classes $\mathcal{C}^t = \{c_1^t, ..., c_{k_t}^t\}$ satisfying $\mathcal{C}^s \cap \mathcal{C}^t = \emptyset$, in which no label information about the samples is given. The goal of ZSL is to build classification models which can predict the label $c(\mathbf{x}_i^t)$ given $\mathbf{x}_i^t$ with no labeled training data for target classes. In the inductive setting, the unlabeled samples from the target classes are not available in the training stage, whereas they do appear in the transductive setting [8]. Because the source classes and target classes are different, some auxiliary information is required for knowledge transfer. As introduced in Section I, there are two kinds of relationships to connect them. The first is class embedding where each class $c_i \in \mathcal{C}^s \cup \mathcal{C}^t$ has a corresponding embedding vector $\mathbf{a}_i \in \mathbb{R}^q$ from a shared attribute space [4] or word vector space [5]. The second is class similarity that can be formulated as a similarity graph $\mathbf{G}$ where g_{ij} denotes the similarity between class c_i and c_j from $\mathcal{C}^s \cup \mathcal{C}^t$. Note that most of the existing approaches focus on only one of the aforementioned settings, while our framework can be applied to all the settings mentioned above.

B. Related Work

Transfer Learning has been used and achieved state-of-the-art performance in many applications, like data retrieval [16], [17], object detection [18], and multispectral imagery change detection [19]. In this paper we focus on the zero-shot setting, and thus we introduce the related works for this problem.

With the problem and notations, the embedding-space based ZSL approaches can be formulated as the following function:

$$c(\mathbf{x}^t) = \text{argmin}_{c \in \mathcal{C}^t}\ dis(\varphi(\mathbf{x}^t), \psi(\mathbf{a}_c)) \tag{1}$$

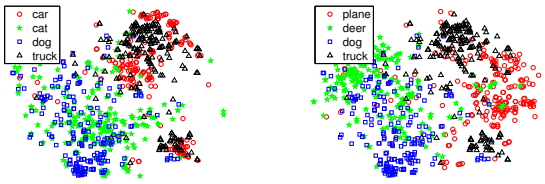
where $dis(\cdot,\cdot)$ is a distance or similarity measure, $\varphi(\mathbf{x}^t)$ is a sample projection function to the embedding space and $\psi(\mathbf{a}_c)$ is a class projection function which transforms the embedding vector in some ways. Because $\psi(\mathbf{a}_c)$ is fixed for a test sample, classifying $\mathbf{x}^t$ consists of two steps. Firstly, the sample is projected into the intermediate space by φ. Secondly, the distance to each target class is measured in the intermediate space. Existing works distinguish from each other by the specific choices of φ, ψ, and $dis(\cdot,\cdot)$. In Direct Attribute Prediction [6], they adopted linear classifiers, identity function, and Euclidean distance respectively. In Cross-modal Transfer [5], nonlinear projection, identity function, and isometric Gaussian probability were adopted. In Shared Model Space Learning [7], [9], linear projection, identity function, and inner product similarity were used, and the similar idea was also adopted by [20] where the deep CNN model was used as the image projection function. In Semantic Similarity Embedding [21], the class-dependent nonlinear projections, sparse reconstruction based projection, and inner project similarity were used. In Synthesized Classifiers [2], its complicated formulation could be also simplified as the combination of a linear projection by virtual classifiers, an exponential transformation with phantom class embedding vector for $\mathbf{a}_c$, and inner product similarity. These approaches may have different specific formulations in the literatures, but they can be summarized into this same general objective.

On the other hand, the class-similarity based ZSL approaches can be summarized by the following general formulation:

$$c(\mathbf{x}^t) = \text{argmax}_{c \in \mathcal{C}^t} prob(c | f^s(\mathbf{x}^t), \mathbf{G}) \tag{2}$$

where $prob()$ is the conditional probability on a target class, f^s is a multi-class classifier for the source classes which outputs the probability distribution of a test sample on the source classes, and $\mathbf{G}$ is the class similarity among all source and target classes. Generally, the class-similarity based approaches also adopt a two-step strategy in the test stage. Firstly, the probability distribution on source classes is produced by f^s. Secondly, the probability is transferred/propagated to the target classes based on the class similarity $\mathbf{G}$. By choosing different ways to build the multi-class classifier and compute the conditional probability, several approaches have been developed. In Indirect Attribute Prediction [3], a multi-class probability classifier and a linear transformation were adopted. In Convex Combination [14], a convolutional neural network classifier and a linear transformation were utilized. In Semantic Manifold Distance [13], the multi-class logistic regression classifier was used for f^s and absorbing Markov chain process was exploited to propagate the probability to the target classes.

Fig. 3: Selected highly transferable images for car and dog. We use them with pseudo labels to train a car-dog classifier.



(a) (b)

Fig. 4: t-SNE visualization of some samples.

In addition, some extra information from target samples is considered by some transductive approaches [8], [9], [10], [22]. In Propagated Semantic Transfer [22], the local manifold structure of target samples was utilized to constrain the predictions on target samples. In Transductive Multi-view Embedding [8], the projection domain shift problem was discussed and the unlabeled target samples were incorporated to learn domain consistent projection. In Unsupervised Domain Adaptation [10], a regularized sparse coding framework was proposed to overcome the projection domain shift problem. Although more information is available, these approaches still follow the two-step strategy summarized in Eq. (1) or Eq. (2).

III. The Proposed Framework

A. Observation

Transferring knowledge across classes is the key issue to ZSL. Different from the existing approaches that transfer knowledge in the embedding space or the class similarity graph, this paper considers the sample transfer for ZSL. Specifically, we take advantage of the source samples to capture the characteristics of target classes in the original feature space. In fact, it is believed that such a sample-transfer scheme is very logical because human being often uses one category to deduce another one as long as they share the same characteristics. Here we take CIFAR10 [23] dataset which contains 10 classes to explain this observation. Suppose we aim to construct a dog-car classifier, but we have only the labeled samples from the other 8 classes with no labeled samples for these two classes available. By the proposed sample transfer method introduced later, we select 500 samples with high transferability from 8 source classes for both target classes and 20 of them are shown in Fig. 3 (most are from "truck" and "cat"). We can observe that the selected samples can well describe the characteristics of target classes. In particular, we assign pseudo labels "car" and "dog" to the $1,000$ selected samples respectively, and on top of it, we train a linear SVM dog-car classifier. Empirically, we found out that the classifier is able to achieve 92.80% recognition accuracy in the dog-car test set. Another example is shown in Fig. 4, where we use t-SNE [24] to visualize the relationship of some samples. Now suppose we aim to construct a dog-truck classifier. In Fig. 4(a), we can see the samples from "cat" and "car" are helpful to capture the characteristics of "dog" and "truck", just like in Fig. 3. In Fig. 4(b), we can see that some (not all) samples from "deer" and "plane" can also contribute to describe the target classes. Empirically, we select some samples with high transferability from "deer" and "plane" and assign pseudo labels from them to train a SVM classifier. The obtained classifier yields 91.1% accuracy for dog-truck classification. From the observations, we can see the sample-based transfer is indeed an effective way for ZSL.

B. Sample Transfer

Now we are going to introduce how to select samples for transfer for each target class. Because there is no labeled samples for target classes, we still need the embedding space or class similarity to help transfer knowledge. However, they are only used for offline training and classifying a test sample does not involve any intermediate transformations, which is a clear difference with the existing works where the intermediate transformation is compulsory for both training and test stages.

To select suitable samples, it is necessary to define how well a sample can capture the characteristics of a target class, i.e., its transferability. The probability of a sample belonging to a class can be a good measure of the transferability [3], [5]. For example, a cat image is more likely than a dolphin image to be classified as a dog, and thus it is more reasonable to transfer a cat image to the dog class. Based on this idea, we utilize the auxiliary information for helping compute the probability.

Embedding space. Given the embedding space, each class (both source and target) is represented as an embedding vector in the space where we measure the probability. Therefore, the embedding functions are learned from the training samples by:

$$\begin{aligned}(\varphi,\psi) = \operatorname{argmin}_{(\varphi,\psi)} & \sum_{i=1}^{n_s} \mathcal{L}(\varphi(\mathbf{x}_i^s), \psi(\mathbf{a}_{c(\mathbf{x}_i^s)})) \\ & + \sum_{j=1}^{n_t} \alpha_j \mathcal{L}(\varphi(\mathbf{x}_j^t), \psi(\mathbf{a}_{\tilde{c}(\mathbf{x}_j^t)}))\end{aligned} \tag{3}$$

where $\mathcal{L}(\cdot,\cdot)$ is a loss between two vectors, like Euclidean distance, $\tilde{c}(\mathbf{x}_j^t)$ is the estimated label for a target sample and α_j is the weight for the target sample. In the inductive setting, the second term is ignored because there is no target samples for training. We will discuss the details about the second term in the transductive setting later. As introduced in Section II, it is possible to choose several settings for φ, ψ and $\mathcal{L}$. Because they are only for helping select samples instead of classifying and this is not the focus of this paper, we simply use a linear projection for φ, an identity function for ψ and the squared Euclidean distance

for $\mathcal{L}$. Then denote $\mathbf{X} = [\mathbf{x}_1^s; ...; \mathbf{x}_{n_s}^s; \alpha_1 \mathbf{x}_1^t; ...; \alpha_{n_t} \mathbf{x}_{n_t}^t]$, $\mathbf{A} = [\mathbf{a}_{c(\mathbf{x}_1^s)}; ...; \mathbf{a}_{c(\mathbf{x}_{n_s}^s)}; \alpha_1 \mathbf{a}_{\bar{c}(\mathbf{x}_1^t)}; ...; \alpha_{n_t} \mathbf{a}_{\bar{c}(\mathbf{x}_{n_t}^t)}]$, and $\varphi(\mathbf{x}) = \mathbf{xP}$, where $\mathbf{P} \in \mathbb{R}^{d \times q}$ is a linear projection matrix for φ. The solution to the learning problem in Eq. (3) is given as follows

$$\mathbf{P} = (\mathbf{X}'\mathbf{X} + \epsilon \mathbf{I}_d)^{-1} \mathbf{X}'\mathbf{A} \tag{4}$$

where ϵ is a small positive value (say, 10^{-4}) to avoid numeric problems. Then based on the linear sample projection matrix $\mathbf{P}$, we adopt the isometric Gaussian probability [5] to compute the transferability of a training sample $\mathbf{x}_i$ to a target class c_j^t:

$$p_i^j = \mathcal{N}(\mathbf{x}_i \mathbf{P} | \mathbf{a}_{c_j^t}, \sigma^2 \mathbf{I}) \tag{5}$$

Class similarity. When the class similarity is given, we first learn a one-vs-all base classifier for each class, denoted as f^c where $c \in \mathcal{C}^s$ in inductive setting and $c \in \mathcal{C}^s \cup \mathcal{C}^t$ for transductive setting. In this paper, we choose the linear SVM for source classes which is trained with labeled samples and robust SVM introduced later for target classes which is trained with pseudo labeled samples. Then given a training sample $\mathbf{x}_i$, each classifier produces an output o_i^c for it, and the probability distribution on all classes can be computed by soft-max [1]:

$$\tilde{p}_i^c = e^{o_i^c} / \sum\nolimits_{c'} e^{o_i^{c'}} \tag{6}$$

Then with the class similarity graph, we can easily transform the initial distribution into the final distribution on target classes. In this paper, we adopt a simple linear transformation:

$$p_i^j = \frac{1}{Z} \sum\nolimits_c \tilde{p}_i^c g_{jc} \tag{7}$$

where Z is a normalization factor to ensure that $\sum_j p_i^j = 1$, and g_{jc} is the similarity between class c and target class c_j^t.

Sample selection and transfer. Based on Eq. (5) or Eq. (7), the probability that a training sample belongs to a target class, i.e., the transferability, can be computed. A larger transferability indicates that training sample $\mathbf{x}_i$ can better capture the characteristics of target class c and we should transfer this sample to c. Therefore, it is reasonable that we employ the transferability as the measurement to select the samples for each target class. To be efficient, we perform here class-wise selection and transfer, in which we select samples for each target class one by one. In principle, it is desirable that the selected and transferred samples have high transferability, which can be translated into the following objective function:

$$\begin{aligned} (r_1, ..., r_n) &= \text{argmax}_{r_i} \sum_{i=1}^{n} r_i \times p_i^c + \mathcal{R}(\mathbf{r}) \\ s.t. \ &\sum_{i=1}^{n} r_i = \rho > 0, r_i \geq 0 \end{aligned} \tag{8}$$

where r_i is the ranking score for training sample $\mathbf{x}_i$. The samples with higher ranking scores are transferred to the target class. The scale constraint parameter ρ is imposed to avoid arbitrary scaling on r_i. $\mathcal{R}(\cdot)$ denotes the regularization term on the ranking scores which reflects the diversity in this paper.

Here we formulate the objective function to be general by not specifying the meaning of n such that it can be applied to different settings. Specifically, we can set $n = n_s$ in the inductive setting. In the transductive setting, we have two choices. The first is to set $n = n_s + n_t$, i.e., we put labeled source samples and unlabeled target samples together for sample selection. The second one is more sophisticated, which adopts a disjoint selection procedure consisting of three steps. In the first step, we set $n = n_s$ intending to select samples only from the labeled source samples. Next, we enforce $n = n_t$ in order to choose samples from the unlabeled target samples. Eventually, we mix them together to form the final training set. Empirically, we find that the second strategy works better.

Without a proper regularization, solving Eq. (8) will simply assign large ranking scores to samples with high transferability. However, this will give rise to the redundant samples in the target class, which is not expected. For example, if two samples are similar to each other, it is likely that they have similar transferability. Hence, if one's transferability is high, the other's will be high as well. Although they can both capture the characteristics of the target class, they cannot provide enough diversity such that they fail to comprehensively describe the target class which may lead to an ineffective classifier [1]. To address this issue, we propose to incorporate **diversity** as the regularization term. Specifically, we first define a heat kernel matrix [25] to measure the similarity between samples as $K_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma^2)$ where σ is set to the mean Euclidean distance between feature vectors in the training set. Note that our classifier training and test are performed directly in the original feature space, we expect the selected samples to be diverse in the original feature space. To that end, the kernel matrix is defined in the original feature space, but not in the intermediate space. With the similarity kernel matrix, it is straightforward to define the diversity regularization term:

$$\mathcal{R}(\mathbf{r}) = -\frac{1}{2} \sum_{i,j=1}^{n} K_{ij} r_i r_j \tag{9}$$

Obviously, if two training samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar, i.e., K_{ij} is large, assigning large ranking scores to r_i and r_j simultaneously will result in large loss. By targeting to minimize the loss, the redundance can be controlled and the selected samples can provide sufficient diversity. After incorporating the diversity regularization, the objective function becomes:

$$\mathbf{r} = \text{argmin}_{\mathbf{r}} \ \frac{\beta}{2} \mathbf{r}\mathbf{K}\mathbf{r}' - \mathbf{r}\mathbf{p}', \ s.t. \ \mathbf{r}\mathbf{1}_n' = \rho, r_i \geq 0, \tag{10}$$

where $\mathbf{p} = [p_1^c, ..., p_n^c]$ is the vector for target class c, and β is the balance parameter between the diversity (the first term) and the transferability (the second term). By optimizing Eq. (10), we obtain the ranking scores of all training samples for target class c and the top m samples are selected and transferred.

Optimization algorithm. The objective function in Eq. (10) is a standard quadratic programming (QP) problem. There are some ready-made packages to solve this problem, such as the quadprog function in MATLAB. However, we notice that the time complexity of a typical QP solver is usually high which reaches $\mathcal{O}(n^3)$. To make the optimization faster, in this paper, we adopt a more efficient algorithm to solve Eq. (10) based on the augmented Lagrange multiplier (ALM) framework [15], [26]. Specifically, we first rewrite the constrained optimization

Algorithm 1 Optimization algorithm for Eq. (10)

Input: The transferability vector of training samples $\mathbf{p}$; Sample-sample similarity matrix $\mathbf{K}$;
Output: Ranking score r_i for each sample;
1: Initialize: $\tau > 1$, $\mu > 0$, $r_i = p_i / \sum_{i=1}^{n} p_i$, $\mathbf{u} = \mathbf{r}$, $\eta_1 = \mathbf{0}_n$, and $\eta_2 = 0$;
2: **repeat**
3: Update $\mathbf{A} = \beta\mathbf{K} + \mu\mathbf{I}_n + \mu\mathbf{1}'\mathbf{1}$;
4: Update $\mathbf{b} = \mathbf{p} + \mu\rho\mathbf{1}_n + \mu\mathbf{u} - \eta_1 - \eta_2\mathbf{1}_n$;
5: Update $\mathbf{r}$ by solving linear system $\mathbf{rA} = \mathbf{b}$;
6: Update $\mathbf{u}$ by Eq. (14);
7: Update η_1, η_2 and μ by Eq. (15);
8: **until** Convergence;
9: Return r_i;

problem in Eq. (10) into the ALM framework as an unconstrained problem by incorporating the penalty terms and the Lagrange multiplier terms for the constraints as below:

$$\begin{aligned}\mathcal{L}(\mathbf{r},\mathbf{u},\mu,\eta_1,\eta_2) = \frac{\beta}{2}\mathbf{r}\mathbf{K}\mathbf{r}' - \mathbf{r}\mathbf{p}' + \frac{\mu}{2}\|\mathbf{r}\mathbf{1}_n' - \rho\|^2 \\ + \frac{\mu}{2}\|\mathbf{r}-\mathbf{u}\|^2 + (\mathbf{r}-\mathbf{u})\eta_1' + (\mathbf{r}\mathbf{1}_n' - \rho)\eta_2, s.t. u_i \geq 0\end{aligned} \tag{11}$$

where μ is a scalar, η_1 and η_2 are the Lagrange coefficients for the corresponding constraints, and $\mathbf{u}$ is an auxiliary vector. Based on the theory of ALM framework, to find the solution to Eq. (10), we just need to update the variables in $\mathcal{L}$ iteratively according to some rules until the convergence is achieved. The final $\mathbf{r}$ is the global optimum to Eq. (10). Please refer to [15] for the proof. In particular, the updating rules are as below.

Update $\mathbf{r}$. When the other variables are fixed, it is straightforward to show the following equivalence with respect to $\mathbf{r}$:

$$\min_{\mathbf{r}} \ \mathcal{L} \Leftrightarrow \min \frac{1}{2}\mathbf{r}\mathbf{A}\mathbf{r}' - \mathbf{r}\mathbf{b}' \tag{12}$$

where $\mathbf{A} = \beta\mathbf{K} + \mu\mathbf{I}_n + \mu\mathbf{1}'\mathbf{1}$ and $\mathbf{b} = \mathbf{p} + \mu\rho\mathbf{1}_n + \mu\mathbf{u} - \eta_1 - \eta_2\mathbf{1}_n$. By setting the derivative of the function with respect to $\mathbf{r}$ to 0, the solution to the unconstrained problem is given by solving a linear system $\mathbf{rA} = \mathbf{b}$. Apparently, $\mathbf{A}$ is a positive defined matrix and thus the linear system has a unique solution. To efficiently solve it, we adopt the algorithm proposed by [27] which gives a nearly linear time complexity.

Update $\mathbf{u}$. By fixing the other variables, the Lagrange function $\mathcal{L}$ with respect to the auxiliary vector $\mathbf{u}$ is reduced to

$$\min_{u_i \geq 0} \mathcal{L} \Leftrightarrow \min_{u_i \geq 0} \frac{\mu}{2}\|\mathbf{r}-\mathbf{u}\|^2 + (\mathbf{r}-\mathbf{u})\eta_1' \tag{13}$$

and the solution to the nonnegativity-constrained problem is

$$u_i = \max(0, r_i + \eta_{1i}/\mu) \tag{14}$$

Update η_1, η_2 *and* μ. Following the pipeline of ALM framework, η_1, η_2 and μ are updated respectively as follows:

$$\eta_1 \leftarrow \eta_1 + \mu(\mathbf{r}-\mathbf{u}), \ \eta_2 \leftarrow \eta_2 + \mu(\mathbf{r}\mathbf{1}_n' - \rho), \ \mu \leftarrow \tau\mu \tag{15}$$

where $\tau > 1$ is a parameter. The optimization algorithm is summarized in Algorithm 1. After applying the above steps, we can obtain the ranking scores for training samples and then we select some training samples with top ranking scores and assign target class c_j^t for them as the pseudo label. We can perform the sample selection and transfer for each target class. Finally, we obtain a set of samples assigned by pseudo labels for target classes, which can be used for training classifiers.

C. Robust SVM

Until now, we have successfully transferred some training samples attached with the pseudo labels for each target class. Next, we train the classifiers directly in the original feature space, given the transferred samples and pseudo labels. It can be seen that the classification in the test is achieved in one single step, omitting the intermediate space. Theoretically, we can train any supervised classifiers by using the pseudo labels, such as Logistic Regression and kNN classifier. In this paper, we adopt the SVM classifier as the classification model considering its good generalization ability [28]. However, we need to deal with the label noise caused by the fact that the pseudo labels are not the true labels. To achieve better performance, we modify the standard SVM formulation in our scenario so as to enhance its robustness against the label noise.

Formally, suppose we have totally m transferred samples $\{\mathbf{x}_1, ..., \mathbf{x}_m\}$ from training set and each sample has a pseudo label from $\mathcal{C}^t$. To handle the multi-class classification, we train k_t one-vs-all SVM classifiers [29] where each classifier f^c treats class c as positive and the other target classes as negative. With the k_t classifiers, the final decision is given by

$$c(\mathbf{x}^t) = \text{argmax}_{c \in \mathcal{C}^t} f^c(\mathbf{x}^t) \tag{16}$$

To train the classifier $f^c (c \in \mathcal{C}^t)$, we first construct the one-vs-all pseudo label vector $\mathbf{l}^c \in \{-1, 1\}^m$ for target class c where $l_i^c = 1$ if the sample $\mathbf{x}_i$ is assigned by the pseudo label c , or $l_i^c = -1$ otherwise. Based on these data, we consider the following dual formulation of the binary SVM learning:

$$\begin{aligned}\min_{\alpha^c} \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i^c \alpha_j^c l_i^c l_j^c K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{m} \alpha_i^c \\ s.t. \ 0 \leq \alpha_i^c \leq C, \sum_{i=1}^{m} \alpha_i^c l_i^c = 0,\end{aligned} \tag{17}$$

where $K(\cdot,\cdot)$ is the kernel function for SVM and the classifier is represented as $f^c(\mathbf{x}) = \sum_i \alpha_i^c l_i^c K(\mathbf{x}_i, \mathbf{x})$. Because of the label noise, we need to consider the situation that a sample with $l_i^c = 1$ should in fact be labeled as -1, i.e., the label flip [30]. Specifically, we consider that the pseudo label has a probability to be the flipped version of the true label $\tilde{l}_i^c = l_i^c(1 - 2\epsilon_i)$ where ϵ_i is a binary variable with $p(\epsilon_i = 1) = \theta_i$ (flipped) and $p(\epsilon_i = 0) = 1 - \theta_i$ (not flipped). Furthermore, given a positive sample with large transferability computed from Eq. (5) or Eq. (7), it is reasonable to assume that its pseudo label is reliable, i.e., θ_i is small. On the other hand, for a sample with $l_i^c = -1$ (negative sample for c), its θ_i should be small too because the highly transferable samples only makes up a small proportion of the large training set such that its true label is unlikely to be 1. Based on these rules, we define the value of θ_i for each transferred samples as follows:

$$\theta_i = \begin{cases} (1 + e^{\delta p_i^c})^{-1}, & \text{if } l_i^c = 1 \\ 0, & \text{if } l_i^c = -1 \end{cases} \tag{18}$$

where δ is a scale parameter and we set it to 5 in this paper.

Then, to take the noise (label flip probability) into account, we replace l_i^c in Eq. (17) with $l_i^c(1-2\epsilon_i)$. Now we denote $M_{ij}=(1-2\epsilon_i)(1-2\epsilon_j)$, and the expected value of M_{ij} is:

$$\mathbb{E}_\epsilon[M_{ij}]=\begin{cases}1-2\theta_i-2\theta_j+4\theta_i\theta_j, & \text{if } i\neq j\\ 1, & \text{if } i=j\end{cases} \tag{19}$$

By replacing $l_i^c l_j^c$ with $\mathbb{E}_\epsilon[l_i^c l_j^c(1-2\epsilon_i)(1-2\epsilon_j)]=l_i^c l_j^c \mathbb{E}_\epsilon[M_{ij}]$ in Eq. (17), we obtain the objective function of robust SVM:

$$\begin{aligned}&\min_{\alpha^c}\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i^c\alpha_j^c l_i^c l_j^c \tilde{K}_\epsilon(\mathbf{x}_i,\mathbf{x}_j)-\sum_{i=1}^{m}\alpha_i^c\\ &s.t.\ 0\leq\alpha_i^c\leq C,\sum_{i=1}^{m}\alpha_i^c l_i^c=0,\end{aligned} \tag{20}$$

where $\tilde{K}_\epsilon(\mathbf{x}_i,\mathbf{x}_j)=\mathbb{E}_\epsilon[M_{ij}]K(\mathbf{x}_i,\mathbf{x}_j)$. This formulation is intrinsically identical to the standard SVM dual formulation with a kernel matrix $\tilde{K}_\epsilon$, which can be efficiently solved by the existing tools, like LIBSVM [31]. In fact, we can observe that the label noise only influences the **similarity** between training samples in the dual formulation, i.e., $i\neq j$. Hence, our formulation actually aims to decrease the similarity between samples to alleviate the influence of the label noise such that an unreliable positive sample gains less weight during training. For example, if a sample has low transferability such that its pseudo label is not confidential at all. A large θ_i is assigned to it such that its kernel similarity $\tilde{K}_\epsilon(\mathbf{x}_i,\mathbf{x}_j)$ is almost 0. In this case, it has little influence on the objective function.

D. Extensions and Summary

In the above sections, we introduced how to transfer training samples to each target class and train robust SVM classifiers. Next we will discuss how to apply it to different settings.

Embedding space and class similarity. Under different situations, the kinds of relationship among classes can be different. For our framework, the auxiliary relationship is only used in training stage to help compute the transferability of each training sample to each target class and we do not need it at all in the test stage. In Eq. (5) and Eq. (7), how to compute the transferability based on the embedding space and the class similarity is given respectively and the sample selection and transfer step in Eq. (10) only requires the transferability of training samples. Therefore, it is straightforward to apply the proposed framework to embedding space or class similarity.

Inductive learning and transductive learning. In the inductive setting where the unlabeled target samples are unavailable, we can only utilize the source samples for training. For this setting, we just need to sequentially perform transferability computing by Eq. (5) or Eq. (7), sample selection and transfer by Eq. (10), and robust SVM training by Eq. (20). Finally, we obtain the one-step classification model for the target classes.

On the other hand, we can make use of the unlabeled target samples in the transductive setting. Although the target samples are all unlabeled, they can also provide important information about the target classes, for example, the influence of domain shift problem [8], [9], [10] can be significantly alleviated by using the information of unlabeled samples. In this paper, we propose an iterative refinement procedure to improve the learning performance. Specifically, the estimated labels of the target samples are produced by the current model, and then we use the estimated labels to help train the next model. For example, in Eq. (3), we can use the estimated labels of target samples to learn a better projection and we use the transferability of $\mathbf{x}_j^t$ as the weight α_j. With a better projection, the whole procedure is re-executed and we normally expect to generate more effective classifiers for target classes which refine the estimated labels for target samples. For the class similarity case, we can also use the current robust SVM classifiers as the base classifier f^c. Here is a "cold-start" problem for the iterative procedure because the estimated labels are not available at first. To address this issue, we can first ignore the target samples and construct an initial model under the inductive setting and then use the initial model to generate the initial estimated samples. In the coming experiment, we will demonstrate that the iterative refinement can always lead to better results. In addition, considering the source samples and target samples have different distributions, we adopt the disjoint selection strategy in the transductive setting. Specifically, for each target class, we first use only source samples to solve Eq. (10), i.e., $n=n_s$, to transfer m_s source samples, and then use only target samples to solve Eq. (10), i.e., $n=n_t$, to transfer m_t target samples. Therefore, there are m_s+m_t samples transferred to each target class.

Summary. We summarize the whole procedure of the proposed framework in Algorithm 2. We can notice that the framework is so flexible that it can be applied to different settings. In line 2, we utilize the auxiliary relationship between source and target classes for knowledge transfer. In line 4 to 7, we transfer training samples for each target class. In line 9, the target classes' classifier is trained using the transferred samples and pseudo labels. In the transductive setting, an iterative refinement (line 1 to 10) is adopted. Based on the

Algorithm 2 Zero-shot learning with transferred samples

Input: Labeled source samples $\mathbf{x}_i^s$;
Unlabeled target samples $\mathbf{x}_j^t$; /*transductive*/
Class embedding $\mathbf{a}_c$ or class similarity $\mathbf{G}$;
Output: Classifiers for target classes;

1: **while** not convergent /*in the transductive setting, we use the iterative procedure to refine the model*/ **do**
2: Learning the projection or base classifiers; /*estimated labels are used in transductive setting*/
3: **for** $c\in\mathcal{C}^t$ **do**
4: Compute the transferability for each sample to each target class by Eq. (5) or Eq. (7);
5: Solve Eq. (10) with source samples;
6: Solve Eq. (10) with target samples; /*transductive*/
7: Assign pseudo label c to selected samples;
8: **end for**
9: Train Robust SVM by transferred samples and pseudo labels by Eq. (20);
10: **end while**
11: Return robust SVM for each target class;

learned classifier, the classification can be performed directly in the original feature space and the auxiliary relationship is no longer required in the test stage. Thus, with less information loss in the test stage, we can expect better ZSL performance.

E. Complexity Analysis

For Algorithm 1, we adopt the ALM to solve the quadratic programming problem in Eq. 10. Specifically, suppose there are n training samples to solve Eq. (12), we adopt the algorithm from [27] which has a nearly linear complexity. In updating operations in Eq. (14) and Eq. (15) is linear to n. Therefore, the complexity of Algorithm 1 is approximately $\mathcal{O}(Tn)$ where T is the number of iterations to convergence. Typically, the algorithm can converge very fast. In Algorithm 2, for each target class, we need to perform Algorithm 1 to select pseudo labeled samples, which leads to $\mathcal{O}(nk_t)$ complexity. The complexity of training robust SVM is approximate to training standard SVM by using LIBSVM. In summary, the complexity of the approach is linear to the number of training sample n, the number of target class k_s, and the number of iterations in the transductive setting where is usually no more than 10.

IV. Experiment

A. Experiment Settings

Data preparation. In our experiment, we adopt four widely used benchmark datasets for ZSL. The first dataset is CIFAR10 [23] which contains 10 categories like "dog" and "truck" with $6,000$ images for each category. Following the setting in [5], [21], we use 8 categories with $48,000$ images as the source classes and the other 2 categories with $12,000$ images as the target classes. Therefore, there are $C_{10}^2 = 45$ different source-target splits. The second dataset is Animal-with-Attributes (AwA) [3]. This dataset has 50 animal categories and $30,475$ images. It has a standard source-target split suggested by [3] where 40 categories with $24,295$ images are source classes and the other 10 categories with $6,180$ images are target classes. The third dataset is aPascal-aYahoo (aPY) [4] with two subsets. aPascal has 20 objects designed for PASCAL VOC2008 challenge [32], such as "people" and "dog". It contains in total $12,695$ images. aYahoo dataset was collected from Yahoo image search. It has 12 classes which are similar but different from the ones in aPascal, such as "centaur" and "wolf". It contains $2,644$ images. Following the common setting, we regard aPascal as the source classes and aYahoo as the target classes. The last dataset is Caltech-UCSD Birds-200-2011 dataset (CUB) [33]. This dataset contains 200 kinds of birds and $11,788$ images. Following [2], we randomly split the classes into 4 parts where each part has 50 classes. We utilize one part as the target classes and the other three as the source classes. We report the average result over 4 parts.

The recent years have witnessed the success of deep convolutional neural network (CNN) features on computer vision and related fields, such as zero-shot learning [2], [8], [10], [21], hashing [34], [35], detection [36], image annotation [37]. Therefore, we also adopt CNN features in our experiment. Specifically, we use the Caffe [38] tool with the pre-trained VGG-19 model [39] and we employ the $4,096$-dimensional output of the last fc layer as the feature vector for each image.

To construct the class embedding for CIFAR10 dataset, following [5] we adopt the 50-dimensional word representation learned by [40]. For AwA dataset, we utilize the provided binary attribute representation for each class [3] which is a 85-dimensional vector. For aPY and CUB, the attributes are annotated to each image. Following previous works [8], [21], we take the means of attribute vectors from the same class to generate the class embedding. To construct the class similarity graph among classes, analogous to [13], we adopt the squared cosine similarity between the class embeddings[1] mentioned above for g_{ij} and we further normalize $\mathbf{g}_{i*}$ to control the scale.

Baselines and evaluation metric. Because our framework can be applied to several settings, we select many related ZSL approaches as baselines. They cover both transductive and inductive settings, and utilize either class embedding or class similarity as auxiliary information. To evaluate the performance, following the standard setting, we adopt the multi-way classification accuracy on target classes as metric.

Implementation detail. We use the following settings to implement our framework under different scenarios. When training (robust) SVM classifiers used as the base classifiers for class similarity and as the target class classifiers, we need to choose the parameter C. We adopt 5-fold sample-wise cross validation using the (pseudo) labeled samples, and C is chosen from $\{0.01, 0.05, 0.1, ..., 5, 10\}$. For the robust SVM, we adopt linear kernel, i.e., $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i\mathbf{x}_j'$, because it is the most easy to control. There is an important parameter β in Eq. (10) which balances the weight between transferability and diversity. To determine this parameter, we adopt the **class-wise** cross validation [2], [9], [21]. Analogous to the sample-wise one whose aim is to guarantee the model learned with labeled samples can generalize well for new samples, class-wise cross validation tries to construct a model using labeled source classes that can generalize well for new target classes. Specifically, we split the source **classes** into k folds in which one fold is regarded as the validation set and the others as the training set. Then we can simulate the ZSL setting by treating the samples in validation set as unlabeled. Based on the class-wise cross validation, the optimal value for parameters can be determined and then we train the final model using all training data. In our experiment, we use 4-fold class-wise cross validation and the parameter β is chosen from $\{10^{-3}, ..., 10^3\}$. When transferring samples from source samples, we set $m_s = 1000, 500, 200, 50$ for CIFAR10, AwA, aPY, and CUB respectively, i.e., for each target class, m_s samples are transferred for it and assigned by the corresponding pseudo labels. On the other hand, when transferring from unlabeled target samples (by the disjoint selection in the transductive setting), we set $m_t = 500, 200, 50, 10$ respectively. The effect of m_s and m_t will be discussed later. We denote the inductive version as STZSL-I, and the transductive version as STZSL-T.

[1]Because the similarity graph is not available, we can only use the class embedding to help construct it. When the class similarity is adopted as the auxiliary relationship, all approaches cannot get access to the class embedding.

TABLE I: The zero-shot classification accuracy(%) on four benchmark datasets. The approaches listed in this table take the **class embedding** as the auxiliary information. The symbol ‡ indicates that the result is produced under the transductive setting.

	CIFAR10	Animal with Attributes	aPascal-aYahoo	Caltech-UCSD-Birds
Farhadi *et al.* 2009 [4]			32.5	
Socher *et al.* 2013 [5]	72.8			
Fu *et al.* 2014 [8]		$77.8^{\ddagger}$		$45.2^{\ddagger}$
Jayaraman *et al.* 2014 [41]		43.01 ± 0.07	26.02 ± 0.05	
Akata *et al.* 2015 [42]		66.7		50.1
Kodirov *et al.* 2015 [10]		$75.6^{\ddagger}$	$26.5^{\ddagger}$	$40.6^{\ddagger}$
Li *et al.* 2015 [43]		$56.88 \pm 1.74^{\ddagger}$	$27.02 \pm 1.25^{\ddagger}$	
Li *et al.* 2015 [44]		$52.06 \pm 1.52^{\ddagger}$	$25.98 \pm 1.19^{\ddagger}$	
RomeraParedes *et al.* 2015 [7]	81.22 ± 1.04	75.32 ± 2.28	24.22 ± 2.89	39.04 ± 0.57
Zhang *et al.* 2015 [21]	88.30	76.33 ± 0.83	46.23 ± 0.53	30.41 ± 0.20
Al-Halah *et al.* 2016 [45]		67.5	37.0	
Changpinyo *et al.* 2016 [2]		72.9		**54.5**
Guo *et al.* 2016 [9]	$86.30 \pm 0.77^{\ddagger}$	$78.47 \pm 1.06^{\ddagger}$	$39.03 \pm 0.77^{\ddagger}$	$43.10 \pm 0.32^{\ddagger}$
Xian *et al.* 2016 [46]		76.1		47.4
Zhang *et al.* 2016 [47]		79.12 ± 0.53	$\mathbf{50.23 \pm 2.97}$	42.11 ± 0.55
STZSL-I	$\mathbf{89.72 \pm 0.56}$	$\mathbf{79.73 \pm 0.68}$	$\mathbf{51.67 \pm 0.32}$	$\mathbf{55.36 \pm 0.30}$
STZSL-T	$\mathbf{90.99 \pm 0.27}^{\ddagger}$	$\mathbf{83.71 \pm 0.82}^{\ddagger}$	$\mathbf{54.37 \pm 0.44}^{\ddagger}$	$\mathbf{58.70 \pm 0.29}^{\ddagger}$

TABLE II: The zero-shot classification accuracy (%) on four benchmark datasets. The approaches listed in this table take the **class similarity** as the auxiliary information. The symbol ‡ indicates that the result is produced under the transductive setting.

	CIFAR10	Animal with Attributes	aPascal-aYahoo	Caltech-UCSD-Birds
Fu *et al.* 2013 [48]		55.3 ± 0.41		
Norouzi *et al.* 2013 [14]		45.2 ± 0.73		
Deng *et al.* 2014 [49]		44.2		
Lampert *et al.* 2014 [3]	70.08 ± 1.02	53.2	$22.10 \pm 0,64$	32.50 ± 0.87
Fu *et al.* 2015 [13]	73.89 ± 1.42	56.0	23.02 ± 0.42	34.87 ± 0.44
STZSL-I	$\mathbf{80.07 \pm 1.09}$	$\mathbf{62.21 \pm 0.72}$	$\mathbf{31.02 \pm 0.41}$	$\mathbf{38.78 \pm 0.10}$
STZSL-T	$\mathbf{84.39 \pm 0.51}^{\ddagger}$	$\mathbf{67.10 \pm 0.31}^{\ddagger}$	$\mathbf{35.57 \pm 0.45}^{\ddagger}$	$\mathbf{41.19 \pm 0.29}^{\ddagger}$

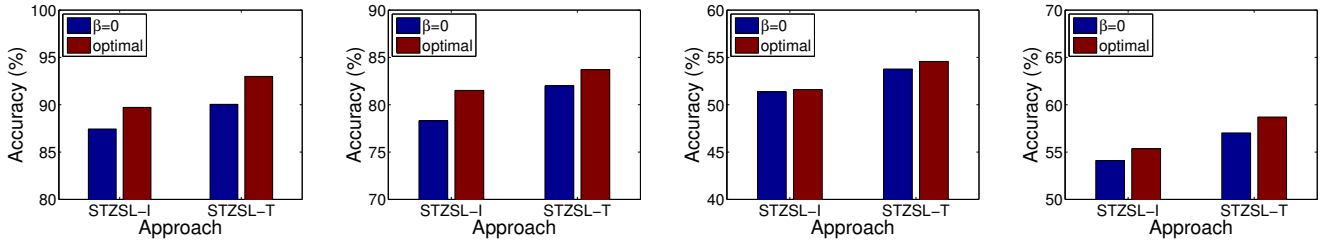
B. Experiment Results

Benchmark comparison. We firstly compare the proposed framework to the state-of-the-arts on four benchmark datasets. In Table I, we present the performance of approaches which take the class embedding as the auxiliary information. In Table II, we show the results of approaches which utilize the class similarity for knowledge transfer. Furthermore, we compare the proposed framework to both inductive approaches and transductive approaches. We can observe that the proposed framework performs better than, at least comparable to, the state-of-the-art baselines in the corresponding settings, which demonstrates the effectiveness of the proposed framework. In addition, we have the following observations from the results.

Firstly, there are two important baselines [3], [5] we need to mention since the way our framework computes the transferability is highly analogous to their classification methods. However, our framework does not simply perform classification in the intermediate space. Instead, it only utilizes the intermediate space for sample transfer and trains classifiers directly in the original feature space. The superiority of the proposed framework to [3], [5] demonstrates that classifying target samples in the intermediate space may suffer from poor performance because of the information loss and that training models in the original feature space can avoid this problem.

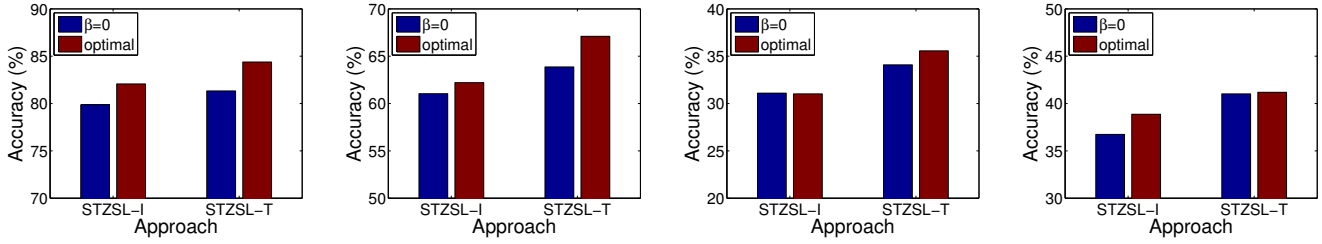
Secondly, we noticed that there are some approaches that construct classification model in the original feature space, like [7], [9], [43]. But in fact, their classifier is formulated as $f^c(\mathbf{x}) = \mathbf{x}\mathbf{W}\mathbf{a}_c'$ where $\mathbf{a}_c$ is the class embedding for target class $c \in \mathcal{C}^t$ and $\mathbf{W}$ is a parameter matrix learned from training data. Obviously, this formulation is equivalent to the two-step strategy where a test sample is firstly projected into the class embedding space by $\mathbf{W}$ and then the classification is done by measuring the inner product similarity between the projected feature and target classes' embeddings. Therefore, they also follow the general framework in Eq. (1). Although they achieve state-of-the-art performance among all baselines, they still suffer from the problem of the two-step strategy such that they still perform worse than the proposed framework.

Thirdly, among all baselines, [2] and [47] adopt the most complicated projection functions which simultaneously project the sample and class embedding into an intermediate space. Because of their complicated projections, they show supe-



(a) CIFAR10 (b) AwA (c) aPY (d) CUB

Fig. 5: Effect of diversity regularization with class embedding as the auxiliary information.



(a) CIFAR10 (b) AwA (c) aPY (d) CUB

Fig. 6: Effect of diversity regularization with class similarity as the auxiliary information.

rior performance and are even comparable to the proposed framework in some specific datasets, for example, [2] in CUB and [47] in aPY. However, their overall performance is still worse than the proposed framework. In addition, in TABLE II, [13] adopts complicated semantic manifold distance for similarity measure in the intermediate space, which achieves best performance in the baselines. But the proposed framework performs much better than it. Both results demonstrate again the superiority of the proposed one-step framework to the two-step strategy because it trains classifiers directly in the original feature space such that it suffers from less information loss.

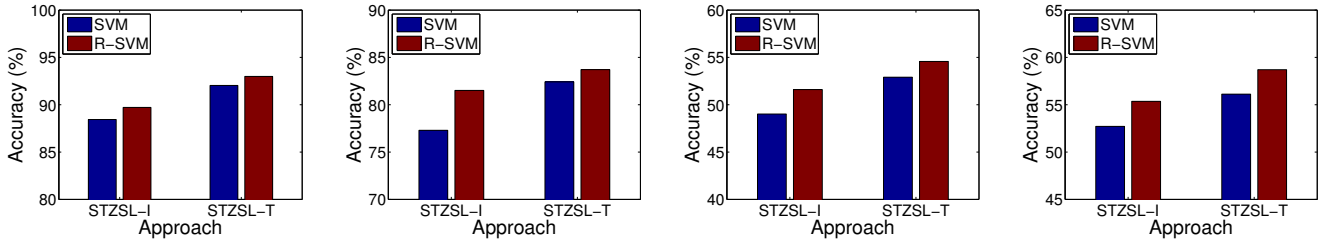
Fourthly, the average results in TABLE II is worse than in TABLE I because the class similarity provides less information than the class embedding although it is relatively easier to obtain. This phenomenon is more significant for the baselines since their classification highly relies on the class similarity and intermediate space. On the contrary, the proposed framework utilizes the class similarity only for sample transfer and the classifiers are trained directly in the original space with the samples. Therefore, the proposed framework performs much better and is even comparable to the state-of-the-arts using class embedding. It is noticed that STZSL in TABLE II is worse than in TABLE I because the less information indeed has influence on the sample transfer since some "bad" samples are transferred. But the superior performance of the proposed framework in both settings still validates its generalizability.

C. More Investigation

The effect of diversity regularization. The key of the proposed framework is to transfer from source domain the similar samples to target domain classes. To remove the redundancy of transferred samples, we propose to incorporate a diversity regularization term into Eq. (10). Now we investigate the effect of the diversity regularization term. Specifically, we run our approaches by setting $\beta = 0$ in Eq. (10) and compare the results to the optimal value for β chosen by the class-wise cross validation. The other parameters are set to the optimal values. The comparisons on four datasets under different settings are summarized in Fig. 5 and 6. In particular, we have the following three observations based on the results.

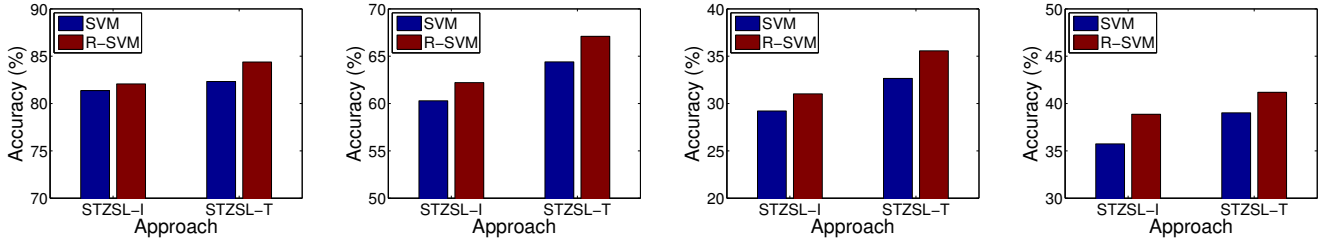
Firstly, the performance when $\beta = 0$ drops with significance in most cases, indicating that removing the diversity regularization leads to worse transferred samples. In fact, based on Eq. (5) or Eq. (7), it is straightforward to observe that similar samples have similar transferability. Therefore, if one sample is transferred because of its high transferability, its neighbors will be transferred too because they are also highly transferable. If so, the transferred samples can cover the target class only from a few aspects, but not comprehensively from most aspects, i.e., they are redundant, which may lead to ineffective classifiers. Actually, this phenomenon is common for human being. For example, if one is trained with images from the back view of "tiger", he/she may fail to recognize a "tiger" from a front view image. By incorporating the diversity regularization into Eq. (10), this problem is well addressed.

Secondly, the diversity regularization has different effects on different datasets. Specifically, it has more influence on CIFAR10 and AwA but less on aPY. This is caused by the properties of datasets. In CIFAR10, there are a large number of candidate samples and there only 2 target classes and the variety of samples is relatively small. Therefore, without diversity regularization, the transferred samples can be very redundant. On the other hand, in aPY, the number of candidate samples is small and the samples in this dataset are more



(a) CIFAR10 (b) AwA (c) aPY (d) CUB

Fig. 7: Effect of robust SVM with class embedding as the auxiliary information.



(a) CIFAR10 (b) AwA (c) aPY (d) CUB

Fig. 8: Effect of robust SVM with class similarity as the auxiliary information.

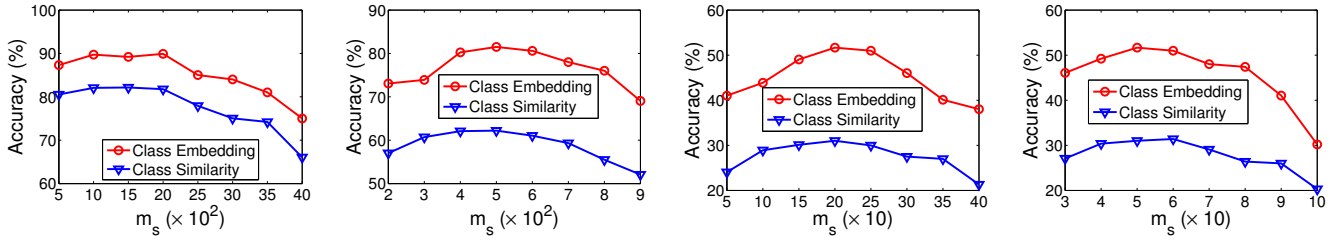
diverse compared to CIFAR10. Consequently, even without diversity regularization, the transferred samples themselves can provide enough diversity such that the regularization term has less influence on it. However, in real-world applications, especially the Web-based ones, the candidate set is quite large, and thus the regularization is important for our framework.

Thirdly, even without the diversity regularization, the proposed framework still yields state-of-the-art performance in most cases. This phenomenon is another evidence to demonstrate that the one-step recognition framework with transferred samples is indeed better than the existing two-step approaches.

The effect of robust SVM. Because we assign the pseudo labels to the transferred samples, it is necessary to consider the noise in the pseudo labels considering the fact that they are not true labels. In this paper we propose to adopt robust SVM to take the noise into account. Now we investigate the effect of the robust SVM on the proposed framework. Specifically, we utilize the conventional linear SVM as the classifier, i.e., we set $\theta_i = 0$ for all samples in Eq. (18), and compare it against the one with robust SVM (R-SVM). We summarize the comparisons under different settings in Fig. (7) and (8).

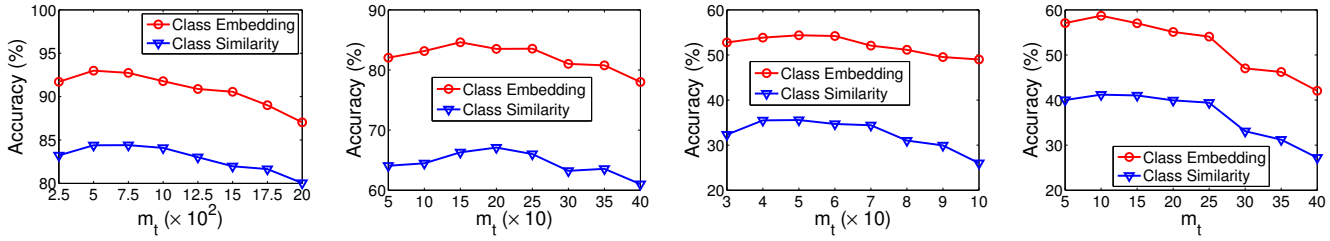
From the results, it can be observed that the robust SVM indeed has significant contribution to the proposed framework. In particular, adopting robust SVM increases the accuracy by 2.61% in average on 4 datasets under 2 settings and 2 auxiliary information sources. Generally, the improvements caused by robust SVM are more significant for inductive setting than transductive setting. This is a reasonable phenomenon because of the following reason. The robust SVM is to address the noise in the pseudo labels caused by the fact that they are not the true labels. In the target domain, the selected samples come from the target classes, and they can better capture the distribution of target classes. Therefore, their pseudo labels are more reliable such that conventional SVM and robust SVM have similar performance. On the other hand, the transferred source samples do not exactly belong to the target classes. Although the transferred ones may capture the characteristics of target classes, they are more likely to be the noisy data compared to the target samples, especially when the diversity regularization is incorporated such that the algorithm is forced to transfer some dissimilar samples. In fact, if we set a large value for β in Eq. (10) (say, 10^5), the performance gap between robust SVM and conventional SVM becomes much larger because the robust SVM will assign a large value (close to 0.5) to θ_i for unreliable samples based on Eq. (18) such that the unreliable samples contribute little to the similarity between samples because $\mathbb{E}_\epsilon[M_{ij}]$ is nearly 0, while the conventional SVM still utilizes the highly noisy pseudo labels for training.

In addition, the performance gap between robust SVM and conventional SVM is relatively smaller in CIFAR10 dataset than the other datasets. The reason is analogous to our discussion mentioned before. Because the CIFAR10 dataset has only 2 target classes and a lot of source samples as candidates, the pseudo labels of transferred samples are more reliable since it is more likely to select the source samples that can well capture the characteristics of target classes. Therefore, the label flip probability θ_i is small for most samples given their high transferability p_i^c based on the definition. In this case, the robust SVM works in a similar way as the conventional SVM because we have $\mathbb{E}_\epsilon[M_{ij}] \approx 1$ given small θ_i and θ_j for robust SVM while $\mathbb{E}_\epsilon[M_{ij}] = 1$ for conventional SVM. On the other hand, the other three datasets have fewer candidate source samples for transfer and thus there may exist some unreliable transferred samples. If so, the effect of robust SVM becomes more significant. However, even



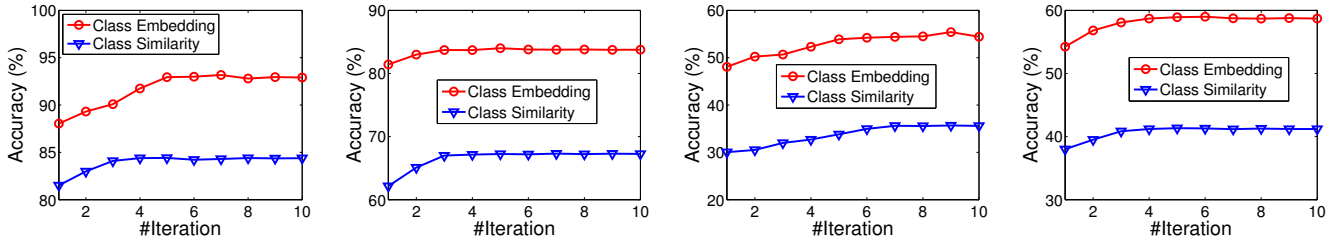
(a) CIFAR10 (b) AwA (c) aPY (d) CUB

Fig. 9: Effect of m_s on STZSL-I.



(a) CIFAR10 (b) AwA (c) aPY (d) CUB

Fig. 10: Effect of m_t on STZSL-T.



(a) CIFAR10 (b) AwA (c) aPY (d) CUB

Fig. 11: Effect of iterative refinement on STZSL-T.

though the effect of robust SVM is small in CIFAR10, the improvement over conventional SVM is still significant. In particular, the average improvement is 1.36% in 4 different settings. Overall speaking, the consistent improvements on 4 datasets and several settings given by robust SVM demonstrate that it is indeed an important part for the proposed framework.

The effect of transferred samples. The key of the proposed framework is to transfer samples for each target class. It is necessary to study the effect of the number of the transferred samples. Firstly, we investigate the effect of m_s, i.e., the number of samples transferred from source domain to each target class. We adopt the inductive approach STZSL-I in the following experiments. We plot the accuracy curves w.r.t. m_s on four datasets in Fig. 9. Obviously, we obtain bell-shaped curves in all settings. Specifically, we have two observations from both sides of the curves. On one hand, when m_s is small, increasing it can help improve the classification accuracy on the target classes. This is quite reasonable because more transferred samples can provide more knowledge for target classes which is beneficial for training effective classifiers [50]. On the other hand, when m_s is large, the performance drops significantly when we keep increasing it. The fundamental of the proposed framework is to transfer samples that can well capture the characteristics of target classes. However, the number of these samples is limited in a dataset so that increasing it too much will force the algorithm to transfer samples that have negative effect. Moreover, the diversity regularization further makes the drop earlier because the good samples are missed if its neighbors are transferred. Without the diversity regularization, the top accuracy of the curves is lower, as was discussed before. But the curves can remain stably high accuracy with larger m_s compared to the current setting because similar/redundant samples can be transferred simultaneously. However, it should be noticed that the curves will also drop significantly at last like in Fig. 9 when m_s becomes too large since too many bad samples are transferred.

In the transductive setting, we also select and transfer samples from unlabeled target domain, and the number of pseudo labeled samples for each target class is denoted as m_t. By fixing m_s to the optimal values mentioned before, we plot

the accuracy w.r.t. m_t on four datasets in Fig. 10. Analogous to the results for m_s, the curves are bell-shaped. One difference is that the curves for m_t have higher start points compared to m_s. This is because of the abundant knowledge transferred from the source samples such that even a few transferred target samples can generate good results. Surely, given reasonably more target samples, the distribution of target classes can be better captured, and thus better performance is achieved. In addition, because we use the pseudo labels, increasing m_t dramatically can only bring in too many unreliable target samples with noisy pseudo labels, which decreases the result.

The effect of iterative refinement. In the transductive setting, we propose an iterative refinement method to progressively fine-tune the model. Here we investigate the effect of the iterative refinement. Specifically, we plot the accuracy on the target classes w.r.t. the number of iterations (line 1 to 10 in Algorithm 1). The performance curves are presented in Fig. 11. The results clearly demonstrate the increasing accuracy with more iterations, which validates that the iterative procedure indeed refines the model. In each iteration, better estimated labels can lead to better projection or base classifiers, which in return refine the estimated labels. As we introduced above, the estimated labels for unlabeled target samples are initialized in the inductive way in the first iteration. One may notice that the results in the first iteration from Fig. 11 are slightly different from the ones we reported in TABLE I and II in some cases. This is because we use class-wise cross validation to choose model parameters and consequently the inductive and transductive extensions may have different optimal parameters.

V. Conclusion

In this paper we address the zero-shot learning problem which transfers knowledge from labeled source classes to unlabeled target classes. Conventional ZSL approaches adopt a two-step recognition strategy in the test stage including a projection step and a similarity measure step in an intermediate space. Information loss is unavoidable because of the intermediate transformation, which degrades the performance. In this paper, we propose a novel one-step ZSL framework which performs classification in the original feature space with directly trained classifiers. To construct label information for classifier training where target classes have no labeled samples available, we propose a novel sample transfer strategy that transfers samples based on their transferability and diversity from source classes to target classes and assign pseudo labels for them to train classifiers. To address the noise of pseudo labels, we adopt robust SVM classifier. We extend the framework into inductive and transductive settings, and with class embedding or class similarity as auxiliary information, which is the first framework that can work in all these settings. We carried out extensive experiments on four benchmarks for ZSL, the results demonstrate the superiority of the proposed framework to the state-of-the-art approaches in many settings.

References

[1] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. Springer, New York, 2006, vol. 1.

[2] S. Changpinyo, W. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5327–5336.

[3] 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, 2014.

[4] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth, "Describing objects by their attributes," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1778–1785.

[5] R. Socher, M. Ganjoo, C. D. Manning, and A. Y. Ng, "Zero-shot learning through cross-modal transfer," in *27th Annual Conference on Neural Information Processing Systems 2013*, 2013, pp. 935–943.

[6] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009, pp. 951–958.

[7] B. Romera-Paredes and P. H. S. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 2152–2161.

[8] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong, "Transductive multi-view embedding for zero-shot recognition and annotation," in *Computer Vision - ECCV 2014 - 13th European Conference*, 2014, pp. 584–599.

[9] Y. Guo, G. Ding, X. Jin, and J. Wang, "Transductive zero-shot recognition via shared model space learning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 3434–3500.

[10] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *IEEE International Conference on Computer Vision*, 2015.

[11] Z. Ji, Y. Xie, Y. Pang, L. Chen, and Z. Zhang, "Zero-shot learning with multi-battery factor analysis," *CoRR*, vol. abs/1606.09349, 2016.

[12] X. Li, S. Liao, W. Lan, X. Du, and G. Yang, "Zero-shot image tagging by hierarchical semantic embedding," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 879–882.

[13] Z. Fu, T. A. Xiang, E. Kodirov, and S. Gong, "Zero-shot object recognition by semantic manifold distance," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2635–2644.

[14] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean, "Zero-shot learning by convex combination of semantic embeddings," *CoRR*, vol. abs/1312.5650, 2013.

[15] D. Bertsekas, *Nonlinear programming*. Belmont,MA: Athena Scientific, 1999.

[16] G. Ding, Y. Guo, J. Zhou, and Y. Gao, "Large-scale cross-modality search via collective matrix factorization hashing," *IEEE Trans. Image Processing*, vol. 25, no. 11, pp. 5427–5440, 2016.

[17] Z. Lin, G. Ding, J. Han, and J. Wang, "Cross-view retrieval via probability-based semantics-preserving hashing," *IEEE Transactions on Cybernetics*, no. 99, doi: 10.1109/TCYB.2016.2608906, 2016.

[18] D. Zhang, J. Han, J. Han, and L. Shao, "Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 27, no. 6, pp. 1163–1176, 2016.

[19] X. Lu, Y. Yuan, and X. Zheng, "Joint dictionary learning for multispectral change detection," *IEEE Trans. Cybernetics*, vol. 47, no. 4, pp. 884–897, 2017.

[20] L. J. Ba, K. Swersky, S. Fidler, and R. Salakhutdinov, "Predicting deep zero-shot convolutional neural networks using textual descriptions," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015, pp. 4247–4255.

[21] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *IEEE International Conference on Computer Vision*, 2015, pp. 4166–4174.

[22] M. Rohrbach, S. Ebert, and B. Schiele, "Transfer learning in a transductive setting," in *27th Annual Conference on Neural Information Processing Systems*, 2013, pp. 46–54.

[23] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Tech Report. Univ. of Toronto*, 2009.

[24] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

[25] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems 14*, 2001, pp. 585–591.

[26] F. Delbos and J. C. Gilbert, "Global linear convergence of an augmented lagrangian algorithm for solving convex quadratic optimization problems," Ph.D. dissertation, INRIA, 2003.

[27] D. A. Spielman and S. Teng, "Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems," in *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, 2004, pp. 81–90.

[28] V. Vapnik, *Statistical learning theory*. Wiley, 1998.

[29] C. Hsu and C. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[30] H. Xiao, H. Xiao, and C. Eckert, "Adversarial label flips attack on support vector machines," in *ECAI 2012 - 20th European Conference on Artificial Intelligence*, 2012, pp. 870–875.

[31] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM TIST*, vol. 2, no. 3, p. 27, 2011.

[32] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results," http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[33] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep., 2011.

[34] Y. Guo, G. Ding, L. Liu, J. Han, and L. Shao, "Learning to hash with optimized anchor embedding for scalable retrieval," *IEEE Trans. Image Processing*, vol. 26, no. 3, pp. 1344–1354, 2017.

[35] X. Lu, X. Zheng, and X. Li, "Latent semantic minimal hashing for image retrieval," *IEEE Trans. Image Processing*, vol. 26, no. 1, pp. 355–368, 2017.

[36] D. Zhang, J. Han, C. Li, J. Wang, and X. Li, "Detection of co-salient objects by looking deep and wide," *International Journal of Computer Vision*, vol. 120, no. 2, pp. 215–232, 2016.

[37] Y. Gu, X. Qian, Q. Li, M. Wang, R. Hong, and Q. Tian, "Image annotation by latent community detection and multikernel learning," *IEEE Trans. Image Processing*, vol. 24, no. 11, pp. 3450–3463, 2015.

[38] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31th International Conference on Machine Learning*, 2014, pp. 647–655.

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[40] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *The 50th Annual Meeting of the Association for Computational Linguistics*, 2012, pp. 873–882.

[41] D. Jayaraman and K. Grauman, "Zero-shot recognition with unreliable attributes," in *Annual Conference on Neural Information Processing Systems 2014*, 2014, pp. 3464–3472.

[42] Z. Akata, S. E. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2927–2936.

[43] X. Li, Y. Guo, and D. Schuurmans, "Semi-supervised zero-shot classification with label representation learning," in *IEEE International Conference on Computer Vision*, 2015, pp. 4211–4219.

[44] X. Li and Y. Guo, "Max-margin zero-shot learning for multi-class classification," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015.

[45] Z. Al-Halah, M. Tapaswi, and R. Stiefelhagen, "Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5975–5984.

[46] Y. Xian, Z. Akata, G. Sharma, Q. N. Nguyen, M. Hein, and B. Schiele, "Latent embeddings for zero-shot classification," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 69–77.

[47] Z. Zhang and V. Saligrama, "Zero-shot learning via joint latent similarity embedding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 6034–6042.

[48] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S. Chang, "Designing category-level attributes for discriminative visual recognition," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 771–778.

[49] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam, "Large-scale object classification using label relation graphs," in *Computer Vision - ECCV 2014 - 13th European Conference*, 2014, pp. 48–64.

[50] X. Qian, H. Wang, Y. Zhao, X. Hou, R. Hong, M. Wang, and Y. Y. Tang, "Image location inference by multisaliency enhancement," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 813–821, 2017.

Yuchen Guo received his B. Sc. degree from School of Software, and B. Ec. from School of Economics and Management, Tsinghua University, Beijing, China in 2013, and currently is a Ph. D. candidate in School of Software in the same campus. His research interests include multimedia information retrieval, computer vision, and machine learning.

Guiguang Ding received his Ph.D degree in electronic engineering from Xidian University, China, in 2014. He is currently an associate professor of School of Software, Tsinghua University. Before joining school of software in 2006, he has been a postdoctoral research fellow in the Department of Automation, Tsinghua University. He has published 80 papers in major journals and conferences, including the IEEE TIP, TMM, TKDE, SIG IR, AAAI, ICML, IJCAI, CVPR, and ICCV. His current research centers on the area of multimedia information retrieval, computer vision and machine learning.

Jungong Han is a senior lecturer with the Department of Computer Science at Northumbria University, UK. Previously, he was a senior scientist (2012-2015) with Civolution Technology (a combining synergy of Philips CI and Thomson STS), a research staff (2010-2012) with the Centre for Mathematics and Computer Science, and a researcher (2005-2010) with the Technical University of Eindhoven in Netherlands.

Yue Gao (SM'14) received his B.S. degree from Harbin Institute of Technology, Harbin, China and M.E. and Ph.D. degrees from Tsinghua University, Beijing, China, respectively.