A Unified approach for Conventional Zero-shot, Generalized Zero-shot and Few-shot Learning

Shafin Rahman, Salman H. Khan and Fatih Porikli

***Abstract*—Prevalent techniques in zero-shot learning do not generalize well to other related problem scenarios. Here, we present a unified approach for conventional zero-shot, generalized zero-shot and few-shot learning problems. Our approach is based on a novel Class Adapting Principal Directions (CAPD) concept that allows multiple embeddings of image features into a semantic space. Given an image, our method produces one principal direction for each seen class. Then, it learns how to combine these directions to obtain the principal direction for each unseen class such that the CAPD of the test image is aligned with the semantic embedding of the true class, and opposite to the other classes. This allows efficient and class-adaptive information transfer from seen to unseen classes. In addition, we propose an automatic process for selection of the most useful seen classes for each unseen class to achieve robustness in zero-shot learning. Our method can update the unseen CAPD taking the advantages of few unseen images to work in a few-shot learning scenario. Furthermore, our method can generalize the seen CAPDs by estimating seen-unseen diversity that significantly improves the performance of generalized zero-shot learning. Our extensive evaluations demonstrate that the proposed approach consistently achieves superior performance in zero-shot, generalized zero-shot and few/one-shot learning problems.**

***Index Terms*—Zero-Shot learning, Few-shot learning, Generalized Zero-Shot learning, Class Adaptive Principal Direction**

I. Introduction

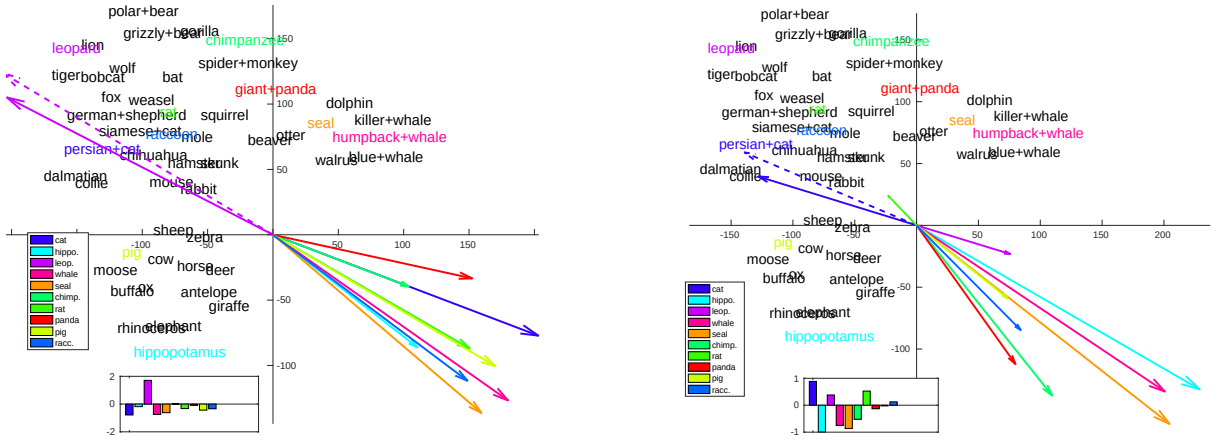
Being one of the most fundamental tasks in visual understanding, object classification has long been the focus of attention in computer vision. Recently, significant advances have been reported, in particular for supervised learning using deep learning based techniques that are driven by the emergence of large-scale annotated datasets, fast computational platforms, and efficient optimization methods [42], [44].

Towards an ultimate visual object classification, this paper addresses three inherent handicaps of supervised learning approaches. The **first** one is the dependence on the availability of labeled training data. When object categories grow in number, sufficient annotations cannot be guaranteed for all objects beyond simpler and frequent single-noun classes. For composite and exotic concepts (such as American crow and auto racing paddock) not only the available images do not suffice as the number of combinations would be unbounded, but often the annotations can be made only by experts [24], [47]. The **second** challenge is the appearance of new classes after the learning stage. In real world situations, we often need to deal with an ever-growing set of classes without representative images. Conventional approaches, in general, cannot tackle such recognition tasks in the wild. The **last** shortcoming is that supervised learning, in its customarily contrived forms, disregards the notion of wisdom. This can be exposed in the fact that we can identify a new object by just having a description of it, possibly leveraging its similarities with the previously learned concepts, without requiring an image of the new object [26].

In the absence of object annotations, zero-shot learning (ZSL) aims at recognizing object classes not seen at the training stage. In other words, ZSL intends to bridge the gap between the seen and unseen classes using semantic (and syntactic) information, which is often derived from textual descriptions such as word embeddings and attributes. Emerging work in ZSL attempt to predict and incorporate semantic embeddings to recognize unseen classes [34], [49], [26], [55], [29]. As noted in [22], semantic embedding itself might be noisy. Instead of a direct embedding, some methods [4], [50], [37], [60] utilize global compatibility functions, e.g. a single projection in [60], that project image features to the corresponding semantic representations. Intuitively, different seen classes contribute differently to describe each unseen class. Enforcing all seen and unseen classes into a single global projection undermines the subtle yet important differences among the seen classes. It eventually limits ZSL approaches by over-fitting to a specific dataset, visual and semantic features (supervised or unsupervised). Besides, incremental learning with newly added unseen classes using a global projection is also problematic due to its less flexibility.

Traditionally, ZSL approaches (e.g., [7], [59], [39]) assume that only the unseen classes are present in the test set. This is not a realistic setting for recognition in the wild where both unseen, as well as seen classes, can appear during the test phase. Recently [51], [9] tested several ZSL methods in generalized zero-shot learning (GZSL) settings and reported their poor performance in this real world scenario. The main reason of such failure is the strong bias of existing approaches towards seen classes where almost all test unseen instances are categorized as one of the seen classes. Another obvious extension of ZSL is few/one-shot learning (F/OSL) where few labeled instances of each unseen class are revealed during training. The existing ZSL approaches, however, do not scale well to the GZSL and FSL settings [1], [7], [50], [60], [28].

To provide a comprehensive and flexible solution to ZSL, GZSL and FSL problem settings, we introduce the concept of principal directions that adapt to classes. In simple terms, CAPD is an embedding of the input image into the semantic space such that, when projected onto CAPDs, the semantic space embedding of the true class gives the highest response. A visualization of the CAPD concept is presented in Fig. 1. As illustrated, the CAPDs of a Leopard (Fig. 1a) and a Persian cat image (Fig. 1b) point to their true semantic label embedding shown in violet and blue respectively, which gives the highest projection response in each case.



(a) Input image from unseen class Leopard

(b) Input image from unseen class Persian cat

Fig. 1: Visualization of class adapting principal directions (CAPD) on a 2D tSNE [46] for illustration. Text labels on the plot represent the seen (black) and unseen (colored) semantic space embeddings of AwA classes in 2D space. For a given test input (a) Leopared (b) Persian cat, CAPDs of the unseen classes are drawn with the same color of the unseen class label text. The bars indicate the responses of a semantic space embeddings projected onto their corresponding CAPDs. Our approach classifies an input to the class that has the maximum response. We also introduce an improved approach to use a reduced set of CAPDs (shown as dashed line) while obtaining better alignment with the correct unseen class embedding (see Sec. III-B).

Our proposed approach utilizes three main sources of knowledge to generalize learning from seen to unseen classes. **First**, we model the relationships between the visual features and semantics for seen classes using the proposed 'Class Adapting Principal Directions' (CAPDs). CAPDs are computed using class-specific discriminative models which are learned for each seen category in the '*visual domain*' (Sec. III-A1). **Second**, our approach effectively models the relationships between the seen and unseen classes in the '*semantic space*' defined by CAPDs. To this end, we introduce a mixing transformation, which learns the optimal combination of seen semantics which are sufficient to reconstruct the semantic embedding of an unseen class (Sec. III-A2). **Third**, we learn a distance metric for the seen CAPDs such that samples belonging to the same class are clustered together, while different classes are mapped further apart (Sec. III-A2). This learned metric transfers cross domain knowledge from visual domain to semantic embedding space. Such a mapping is necessary because the class semantics, especially those collected from unsupervised sources (e.g., word2vec), can be noisy and highly confusing. The distance metric is then used to robustly estimate the seen-unseen semantic relationships.

While most of the approaches in the literature focus on specific sub-problems and do not generalize well to other related settings, we present a unified solution which can easily adapt to ZSL, GZSL and F/OSL settings. We attribute this strength to two key features in our approach: **a)** a highly 'modular learning' scheme and **b)** the two-way inter-domain 'knowledge sharing'. Specifically for the GZSL, we present a novel method to generalize seen CAPDs that avoids the inherent bias of prediction towards seen classes (Sec. III-C). The generalized *seen* CAPD balances the seen-unseen diversity in the semantic space, without any direct supervision from the visual data. In contrast to ZSL and GZSL, the F/OSL setting allows few or a single training instance of the unseen classes. This information is used to update *unseen* CAPDs based on the learned relationships between visual and semantic domains for unseen classes (Sec. III-D). The overall pipeline of our learning and prediction process is illustrated in Fig. 2.

We hypothesize that not all seen classes are instrumental in describing a novel unseen category. To validate this claim, we introduce a new constraint during the reconstruction of semantic embedding of the unseen classes. We show that automatically reducing the number of seen classes in the mixing process to obtain CAPD of each unseen class results in a significant performance boost (Sec. III-B). We perform extensive experimental evaluations on four benchmark datasets and compare with several state-of-the-art methods. Our results demonstrate that the proposed CAPD based approach provides superior performance in supervised and unsupervised settings of ZSL, GZSL and F/OSL.

To summarize, our main contributions are:

- We present a unified solution to ZSL, GZSL and F/OSL by introducing the notion of class adapting principal directions that enable efficient and discriminative embeddings of unseen class images in the semantic space.
- We propose a semantic transformation to link the embeddings for seen and unseen classes based on a learned distance measure.
- We provide an automatic solution to select a reduced set of relevant seen classes resulting in a better performance.

II. Related Work

Class Label Description: It is a common practice to employ class label descriptions to transfer knowledge from seen to unseen class in ZSL. Such descriptions may come

from either supervised or unsupervised learning settings. For the supervised case, class attributes can be one source as well [12], [25], [35], [47]. These attributes are often generated manually, which is a laborious task. As a workaround, word semantic space embeddings derived from a large corpus of unannotated text (e.g. from Wikipedia) can be used. Among such unsupervised word semantic embeddings, word2vec [31], [30] and GloVe [36] vectors are frequently employed in ZSL [58], [50]. These ZSL methods are sometimes (arguably confusingly) referred as unsupervised zero-shot learning [5], [1]. Supervised features tend to provide better performance than the unsupervised ones. Nevertheless, unsupervised features provide more scalability and flexibility since they do not require expert annotation. Recent approaches attempt to advance unsupervised ZSL by mapping textual representations (e.g. word2vec or GloVe) as attribute vectors using heuristic measures [23], [5]. In our work, we use both types of features and evaluate on both supervised and unsupervised ZSL to demonstrate the strength of our approach.

Embedding Space: ZSL strategies aim to map between two different sources of information and two spaces: image and label embeddings. Based on the mapping scheme, ZSL approaches can be grouped into two categories. The **first** category is attribute/word vector prediction. Given an image, they attempt to approximate label embedding and then classify an unseen class image based on the similarity of predicted vector with unseen attribute/word vector. For example, in an early seminal work, [34] introduced a semantic output code classifier by using a knowledge base of attributes to predict unseen classes. [49], [26] proposed direct and indirect attribute prediction methods via a probabilistic realization. [55] formulated a discriminative model of category level attributes. [29] proposed an approach of transferring semantic knowledge from seen to unseen classes by a linear combination of classifiers. The main problem with such direct attribute prediction is the poor performance when noisy or biased attribute annotations are available. Jayaraman and Grauman [22] addressed this issue and proposed a discriminative model for ZSL.

Instead of predicting word vectors, the **second** category of approaches learn a compatibility function between image and label embeddings, which returns a compatibility score. An unseen instance is then assigned to the class that gives the maximum score. For example, [2] proposed a label embedding function that ranks correct class of unseen image higher than incorrect classes. In [39], authors use the same principle but an improved loss function and regularizer. Qiao *et al.* [37] further improved the former approach by incorporating a component for noise suppression. In a similar work, Xian *et al.* [50] added latent variables in the compatibility function which can learn a collection of label embeddings and select the correct embedding for prediction. Our method also has similar compatibility function based on inner product of CAPD and corresponding semantic vector. The use of CAPDs provide an effective avenue to recognition.

Similarity Matching: This type of approaches build linear or nonlinear classifiers for each seen class, and then relate those classifiers with unseen classes based on class-wise similarity measures [7], [11], [18], [29], [38]. Our method finds similar relation but instead of classifiers, we relate CAPDs of seen and unseen classes. Moreover, we compute this relation on a learned metric of semantic embedding space which let us consider subtle discriminative details.

Few/One-shot Learning: FSL has a long history of investigation where few instances of some classes are used as labeled during training [41], [13]. Although ZSL problem can easily be extended to FSL, established ZSL methods are not evaluated in FSL settings. A recent work [45] reports FSL performance of only two ZSL methods e.g. [43], [15]. In another work, [8], [19] presented FSL results on ImageNet. In this paper, we extend our approach to FSL settings and compare our method with the reported performance in [45].

Generalized Zero-shot Learning: GZSL setting significantly increases the complexity of the problem by allowing both seen and unseen classes during testing phase [51], [9], [8]. This idea is related to open set recognition problem where methods consider to reject unseen objects in conjunction with recognizing known objects [6], [21]. In open set case, methods consider all unseen objects as one outlier class. In contrast, GZSL represents unseen classes as individual separate categories. Very few of the ZSL methods reported results on GZSL setting [8], [27], [52]. [15] proposed a joint visual-semantic embedding model to facilitate the generalization of ZSL. [43] offered a novelty detection mechanism which can detect whether the test image came from seen or unseen category. Chao *et al.* [9] proposed a calibration mechanism to balance seen-unseen prediction score which any ZSL algorithm can adopt at decision making stage and proposed an evaluation method called Area Under Seen-Unseen accuracy Curve (AUSUC). Later, several other works [8], [52] adopted this evaluation strategy. In another recent work, Xian *et al.* [51] reported benchmarking results for both ZSL and GZSL performance of several established methods published in the literature. In this paper, we describe extension of our ZSL approach to efficiently adapt with GZSL settings.

III. Our Approach

Problem Formulation: Suppose, the set of all class labels is $\mathbf{y} = \mathbf{y}^{\mathcal{S}} \cup \mathbf{y}^{\mathcal{U}}$ where $\mathbf{y}^{\mathcal{S}} = \{1, ..., \mathrm{S}\}$ and $\mathbf{y}^{\mathcal{U}} = \{\mathrm{S}+1, ..., \mathrm{S}+\mathrm{U}\}$ are the sets of seen and unseen class labels respectively, with no overlap i.e., $\mathbf{y}^{\mathcal{S}} \cap \mathbf{y}^{\mathcal{U}} = \emptyset$. Here, S and U denote the total number of seen and unseen classes, respectively. For all classes in the seen and unseen class sets, we have associated semantic class embeddings (either attributes or word vectors) denoted by the sets $\mathbf{E}^{\mathcal{S}} = \{\mathbf{e}_s : s \in \mathbf{y}^{\mathcal{S}}\}$ and $\mathbf{E}^{\mathcal{U}} = \{\mathbf{e}_u : u \in \mathbf{y}^{\mathcal{U}}\}$ respectively, where $\mathbf{e}_s, \mathbf{e}_u \in \mathbb{R}^d$. For every seen ($s$) and unseen ($u$) class, we have a number of instances denoted by n_s and n_u respectively. The matrices $\mathbf{X}_s = [\mathbf{x}_s^1, ..., \mathbf{x}_s^{n_s}]$ for $s \in \mathbf{y}^{\mathcal{S}}$, and $\mathbf{X}_u = [\mathbf{x}_u^1, ..., \mathbf{x}_u^{n_u}]$ for $u \in \mathbf{y}^{\mathcal{U}}$ represent the image features for the seen class s and the unseen class u, respectively, such that $\mathbf{x}_s, \mathbf{x}_u \in \mathbb{R}^k$. Below, we define the three problem instances addressed in this paper:

- **Zero Shot Learning (ZSL):** The image features of the unseen classes $\mathbf{X}_u$ are not available during the training stage. The goal is to assign an unseen class label $u \in \mathbf{y}^{\mathcal{U}}$ to a given unseen image using its feature vector $\mathbf{x}_u$.

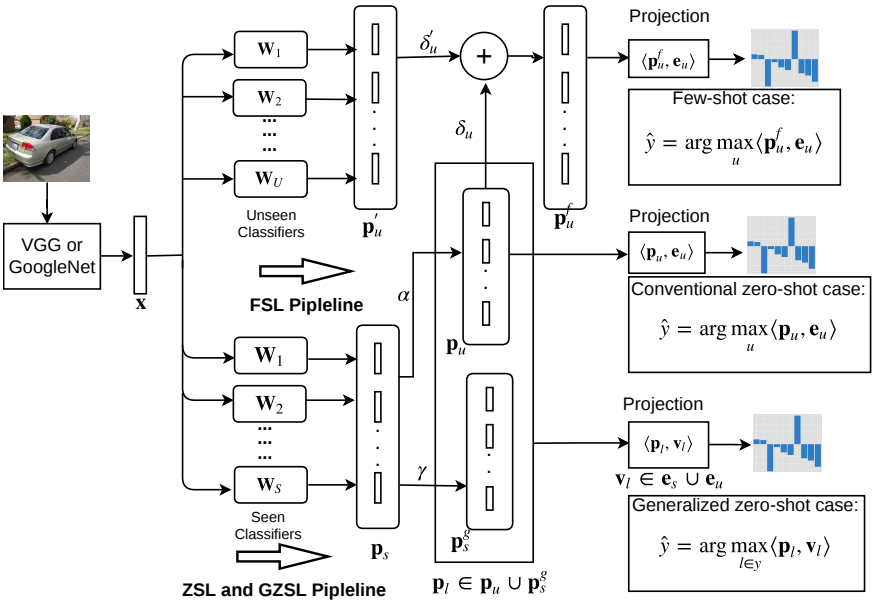


Fig. 2: Overall pipeline of conventional ZSL, FSL and GZSL method. **Conventional ZSL:** An image is passed through a deep network to get an image feature $\mathbf{x}$. Then, $\mathbf{x}$ is fed to seen classifiers $\mathbf{W}_s$ to produce seen CAPDs, $\mathbf{p}_s$. Afterwards, unseen CAPDs, $\mathbf{p}_u$ are computed by linearly combining seen CAPDs using α (or β for reduced case). Finally, prediction is done by computing the maximum projection response of $\mathbf{p}_u$ and unseen semantic embeddings $\mathbf{e}_u$. **FSL:** $\mathbf{x}$ is fed to unseen classifiers $\mathbf{W}_u$ to produce another version of unseen CAPDs $\mathbf{p}'_u$ which are combined with previously computed $\mathbf{p}_u$ through δ'_u and δ_u to find an updated version of unseen CAPDs $\mathbf{p}^f_u$. Final prediction is done by maximum response of $\mathbf{p}^f_u$ and $\mathbf{e}_u$. **GZSL:** Seen CAPDs, $\mathbf{p}_s$ of conventional ZSL setting are generalized using γ to produce generalized seen CAPDs, $\mathbf{p}^g_s$. For prediction, both $\mathbf{p}^g_s$ and $\mathbf{p}_u$ are considered for calculating maximum response of CAPDs and their corresponding semantic embeddings, $\mathbf{e}_s$ and $\mathbf{e}_u$.

- **Generalized Zero Shot Learning (GZSL):** The image features of the unseen classes $\mathbf{X}_u$ are not available during the training stage similar to ZSL. The goal is to assign a class label $l \in \mathbf{y}$ to a given image using its feature vector $\mathbf{x}$. Notice that, the true class of $\mathbf{x}$ may belong to either a seen or an unseen class.
- **Few/One Shot Learning (FSL):** Only a few/one randomly chosen image features from $\mathbf{X}_u$ are available as labeled examples during the training stage. The goal is same as the ZSL setting above.

In Secs. III-A and III-B, we first provide a general framework of our approach mainly focused on ZSL. Afterwards, in Secs. III-D and III-C we extend our approach to FSL and GZSL settings, respectively. Before describing our extensive experimental evaluations in Sec. V, we also provide an in-depth comparison with the existing literature in Sec. IV.

A. Class Adapting Principal Direction

We introduce the concept of 'Class Adapting Principal Direction' (CAPD), which is a projection of image features onto the semantic space. The CAPD is computed for both seen and unseen classes, however the derivation of the CAPD is different for both cases. In the following, we first introduce our approach to learn CAPDs for seen classes and then use the learned principal directions to derive CAPDs for unseen classes.

1) Learning CAPD for Seen Classes: For a given image feature $\mathbf{x}_s$ belonging to the seen class s, we define its CAPD $\mathbf{p}_s$ in terms of a linear mapping parametrized by $\mathbf{W}_s$ as,

$$\mathbf{p}_s = \mathbf{W}_s^T \mathbf{x}_s. \tag{1}$$

Our goal is to learn the class-specific weights $\mathbf{W}_s$ such that the output principal directions are highly discriminative in the semantic space (rather than the image feature space). To this end, we introduce a novel loss function which uses the corresponding semantic space embedding $\mathbf{e}_s$ of seen class s to achieve maximum separability.

Proposed Objective Function: Given the training samples $\mathbf{X}_s$ for the seen class s, $\mathbf{W}_s$ is learned such that the projection of $\mathbf{p}_s$ on the semantic space embedding $\mathbf{e}_s$, defined by the inner product $\langle \mathbf{p}_s, \mathbf{e}_s \rangle$, generates a strong response. Precisely, the following objective function is minimized:

$$\min_{\mathbf{W}_s} \frac{1}{\kappa} \sum_{c=1}^{\mathrm{S}} \sum_{m=1}^{n_c} \log\left(1 + \exp\left\{L(\mathbf{x}_c^m; \mathbf{W}_s)\right\}\right) + \frac{\lambda_s}{2} \parallel \mathbf{W}_s \parallel_2^2 \tag{2}$$

where L is the cost for a specific input $\mathbf{x}_c^m$, λ_s is the regularization weight set using cross validation and $\kappa = \sum_{c=1}^{\mathrm{S}} n_c$. We define the cost L as:

$$L(\mathbf{x}_c^m; \mathbf{W}_s) = \begin{cases} \langle \mathbf{p}_s, \mathbf{e}_c \rangle - \langle \mathbf{p}_s, \mathbf{e}_s \rangle, & c \neq s \\ \langle \mathbf{p}_s, \frac{1}{\mathrm{S}-1} \sum\limits_{t \neq s} \mathbf{e}_t \rangle - \langle \mathbf{p}_s, \mathbf{e}_s \rangle, & c = s \end{cases}$$

In the above loss function, two different scenarios are tackled depending on whether the training samples (image features) are from the same (positive) or different (negative) classes. For the **negative** samples ($c \neq s$), the projection of $\mathbf{p}_s$ on the correct semantic embedding $\mathbf{e}_s$ is maximized while its projection on the incorrect semantic embedding $\mathbf{s}_c$ is minimized. For the **positive** samples ($c = s$), our proposed formulation directs the projection on the correct semantic embedding $\mathbf{e}_s$ to be higher than the average response of projections on the incorrect semantic embeddings. In both cases, $\langle \mathbf{p}_s, \mathbf{e}_s \rangle$ is constrained to produce a high response. Our loss formulation is motivated by [58], with notable differences such as the class-wise optimization, explicit handling of positive samples and the extension of their ranking loss for image tagging to the ZSL problem. Moreover, the loss of [58] considers a single principal direction for all possible possible tags in the multi-label annotation task whereas our CAPD is specialized to assign a single label for zero-shot recognition.

We optimize Eq. 2 by Stochastic Gradient Descent to obtain $\mathbf{W}_s$ for each seen class. Note that, $\mathbf{p}_s = \mathbf{W}_s^T \mathbf{x}_c^m$ in the above

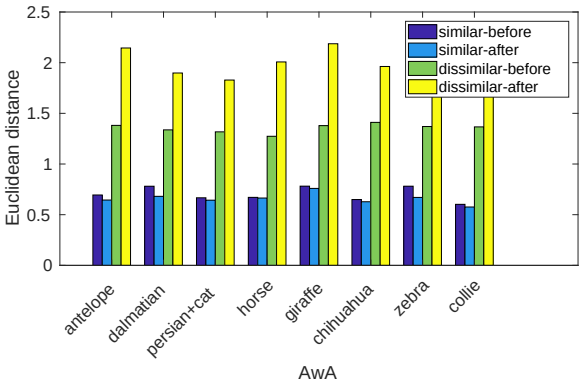


Fig. 3: The effect of metric learning on semantic space with AwA attributes. The average Euclidean distances of top 5 similar and dissimilar classes for example classes from AwA dataset are shown. The distances are illustrated for both with and without the application of metric learning. Metric learning brings similar classes together while pulls dissimilar classes further apart.

cost function, thus for any sample $\mathbf{x}_c^m$, $\mathbf{p}_s$ changes when $\mathbf{W}_s$ is updated at each training iteration. Also, the learning process of $\mathbf{W}_s$ for each seen class is independent of other classes. Therefore, all $\mathbf{W}_s$ can be learned jointly in a parallel fashion. Once the training process is complete, given an input visual feature $\mathbf{x}_c^m$, we generate one CAPD $\mathbf{p}_s$ for each seen class using Eq. 1. As a result, $\mathbf{P}^{\mathcal{S}} = [\mathbf{p}_1 ... \mathbf{p}_{\mathrm{S}}] \in \mathbb{R}^{d \times \mathrm{S}}$ accumulates the CAPDs of all the seen classes. Each CAPD is the mapped version of the image feature on the class specific semantic space. The CAPD vector and its corresponding semantic space embedding vector point to similar direction if the input feature belongs to the same class.

2) Learning CAPD for Unseen Classes: In ZSL settings, the images of the unseen classes are not observed during the training. For this reason, we cannot directly learn a weight matrix to calculate $\mathbf{p}_u$ using the same approach as $\mathbf{p}_s$. Instead, for any unseen sample, we propose to approximate $\mathbf{p}_u$ using the seen CAPD of the same sample. Here, we consider a bilinear map, in particular, a linear combination of the seen class CAPDs to generate the CAPD of the unseen class u:

$$\mathbf{p}_u = \sum_{s=1}^{\mathrm{S}} \theta_{s,u} \mathbf{p}_s = \mathbf{P}^{\mathcal{S}} \theta_u \tag{3}$$

where, $\theta_u = [\theta_{1,u} ... \theta_{\mathrm{S},u}]^T \in \mathbb{R}^{\mathrm{S}}$ is the coefficient vector that, in a way, aggregates the knowledge of seen classes into the unseen one. The computation of θ_u is subject to the relation between CAPDs and semantic embeddings of classes. We detail our approach to approximate θ_u below.

Metric Learning on CAPDs: The CAPDs reside in the semantic embedding space. In this space, we learn a distance metric to better model the similarities and dissimilarities among the CAPDs. To this end, we assemble the sets of similar $\mathbf{A}$ and dissimilar $\bar{\mathbf{A}}$ pairs of CAPDs that correspond to the pairs of training samples belonging to the same and different seen classes, respectively. Our goal is to learn a distance metric $d_{\mathbf{M}}$ such that the similar CAPDs are clustered together and the dissimilar ones are mapped further apart. We minimize the following objective which maximizes the squared distances between the minimally separated dissimilar pairs:

$$\max_{\mathbf{M}} \min_{(i,j) \in \bar{\mathbf{A}}} d^2_{\mathbf{M}}(\mathbf{p}_i, \mathbf{p}_j) \quad s.t. \sum_{(i,j) \in \mathbf{A}} d^2_{\mathbf{M}}(\mathbf{p}_i, \mathbf{p}_j) \leq 1 \tag{4}$$

where $d_{\mathbf{M}} = \sqrt{(\mathbf{p}_i - \mathbf{p}_j)^T \mathbf{M} (\mathbf{p}_i - \mathbf{p}_j)}$ is the Mahalanobis distance metric [54]. After training, the most confusing dissimilar CAPD pairs are pulled apart while the similar CAPDs are clustered together by learning an optimal distance matrix $\mathbf{M}$. Moreover, as metric learning is done on the semantic space it can help to measure the seen-unseen relation.

Our intuition is that, given a learning metric $\mathbf{M}$ in the semantic embedding space, the relation between the semantic label embeddings of the seen $\mathbf{e}_s$ and the unseen classes $\mathbf{e}_u$ is analogous to that of their principal directions. Since the semantic label embedding of unseen classes are available, we can estimate their relation with the seen classes. For simplicity, we consider a linear combination of semantic space embeddings:

$$\hat{\mathbf{e}}_u = \sum_{s=1}^{\mathrm{S}} \alpha_{s,u} \mathbf{e}_s = \mathbf{E}^{\mathcal{S}} \alpha_u \tag{5}$$

where, $\hat{\mathbf{e}}_u$ is the approximated semantic embedding of $\mathbf{e}_u$ corresponding to unseen class u. We compute $\alpha_u = [\alpha_{1,u} ... \alpha_{\mathrm{S},u}]^T \in \mathbb{R}^{\mathrm{S}}$ by solving:

$$\min_{\alpha_u} \quad (\hat{\mathbf{e}}_u - \mathbf{e}_u)^T \mathbf{M} (\hat{\mathbf{e}}_u - \mathbf{e}_u) + \frac{\lambda_u}{2} \parallel \alpha_u \parallel_2^2 \tag{6}$$

where λ_u is a regularization parameter which is set via cross validation.

As we mentioned above, using the learned metric $\mathbf{M}$, the relationship between the seen-unseen semantic embeddings α_u is analogous to the relationship between the seen-unseen CAPDs θ_u, thus $\theta_u \approx \alpha_u$. Here, $\mathbf{M}$ acts as a bridge between visual features and their corresponding class semantics. For example, 'giraffe' is among top 5 close animals of 'deer' in semantic space but after considering the metric $\mathbf{M}$ 'cow' becomes closer than 'giraffe' because of its visual similarity with deer. In Fig. 3, we highlight this behavior by calculating average Euclidean distance (before and after applying $\mathbf{M}$) of top 5 similar and dissimilar classes. Essentially, metric learning brings visually and semantically similar classes together while pulls dissimilar classes further apart. Accordingly, we approximate the unseen CAPDs with seen CAPDs by rewriting Eq. 3 as:

$$\mathbf{p}_u \approx \mathbf{P}^{\mathcal{S}} \alpha_u. \tag{7}$$

We derive a CAPD, $\mathbf{p}_u$ for each unseen class using Eq. 7. In test stage of ZSL setting, we assign a given image feature $\mathbf{x}$ to an unseen class using the maximum projection response:

$$\hat{y} = \arg\max_u \langle \mathbf{p}_u, \mathbf{e}_u \rangle \tag{8}$$

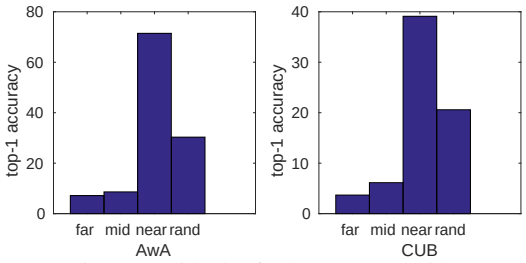


Fig. 4: Experiments with the farthest away, mid-range, nearest, and randomly chosen seen classes, using one third of the total seen classes in each case. Image features are obtained using VGG-verydeep-19 and semantic space vectors are derived from attributes. As shown, the semantic space embeddings of the seen classes that are **near** to the embedding of the unseen class provide more discriminative representations.

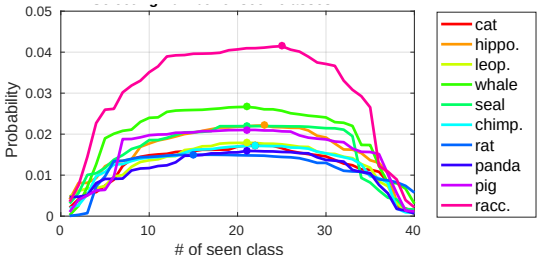


Fig. 5: PDF of distances using the normal distribution with zero mean and a unit standard deviation for each unseen class. (GoogLeNet features and the word2vec semantic embedding for AwA dataset)

B. Reduced Set Description of Unseen Classes

When describing a novel object, we often resort to relating it with the similar known object categories. It is intuitive that a subset of the known objects is sufficient for describing an unknown one.

We incorporate this observation by proposing a modified version of Eq. 5. The term α_u contains the contribution of each seen class to describe the unseen class $u \in \mathbf{y}^{\mathcal{U}}$ by reconstructing $\mathbf{e}_u$ using all seen classes semantic label embeddings. We reconstruct $\mathbf{e}_u$ by only a small number of seen classes ($\mathrm{N} < \mathrm{S}$). These N seen classes can be selected using any similarity measure (Mahalanobis distance in our case). The reconstruction of $\mathbf{e}_u$ becomes:

$$\hat{\mathbf{e}}_u = \sum_{i=1}^{\mathrm{N}} \beta_{i,u} \mathbf{e}_i \tag{9}$$

Here, $\beta_u \in \mathbb{R}^{\mathrm{N}}$ is the coefficients of selected seen classes. We learn β_u by a similar minimization objective as in the Eq. 6. By replacing α_u with β_u in the Eq. 7, it is possible to compute the CAPD of unseen class u using a reduced set of seen classes. Such CAPDs are shown in Fig. 1 in dashed lines.

Appropriate Choice of Seen Classes: In Fig. 4, we show comparisons when different approaches are used to select a subset of seen classes to describe the unseen ones. The results illustrate that the seen classes having the semantic space embeddings close to that of a particular unseen class are more suitable to describe it. Here, we considered N nearest neighbors of the unseen class semantic vector $\mathbf{e}_u$ using the Mahalanobis distance. Using a less number of seen classes is inspired by the work Norouzi *et al.* [33] where they applied convex combination of selected semantic embedding vector based on outputs of the softmax classifier of corresponding seen classes. The main drawback of their approach is that the softmax classifier output does not take the semantic embeddings into consideration, which can ignore important features when describing the unseen class. Instead, our method performs an independent optimization (Eq. 6) that jointly considers image feature, CAPD and semantic embedding relations via the learned metric $\mathbf{M}$. As a result, the proposed strategy is better able to determine the optimal combination of selected seen semantic embeddings (see Sec. V-B).

Automatic N Selection for Each Unseen Class: While [33] proposed a *fixed* number of selected seen classes to describe an unseen class, we suggest a novel technique to automatically select the number of most informative seen classes (N).

First, for an unseen class semantic embedding $\mathbf{e}_u$, we calculate the Mahalanobis distances (using learned metric $\mathbf{M}$) from $\mathbf{e}_u$ to all $\mathbf{e}_s$ and perform mean normalization. Then, we apply kernel density estimation to obtain a Probability Density Function (PDF) for the normalized distances. Fig. 5 shows the PDF for each unseen semantic embedding vector of the AwA dataset. For a specific unseen class, the number of seen classes with the highest probability score is assigned as the value of N. Unlike [33], this scheme allows choosing a *variable* number of the seen classes for different unseen classes. In Sec. V-A of this paper, we have reported an estimation of the average numbers of seen classes selected for the tested unseen classes.

Sparsity: Using a reduced number of the seen classes in Eq. 9 indirectly imposes sparsity in the coefficient vector α_u in the Eq. 5. This is similar to Lasso (ℓ_1) regularization (instead of ℓ_2 regularization) in the loss function in Eq. 6. We observe that the above selection solution is more efficient and accurate than the Lasso-based regularization. This is because the proposed solution is based on the intuition that the semantic embedding of an unseen class can be described by closely embedded seen classes. In contrast, Lasso is a general approach and do not consider any domain specific semantic knowledge.

Having discussed the ZSL setting in Secs. III-A and III-B above, we present the extension of CAPDs to the GZSL problem.

C. Generalized Zero-shot Learning

ZSL setting considers only unseen class images during the test phase. This setting is less realistic, because new images can belong to both seen and unseen classes. To address this scenario, generalized ZSL (GZSL) has recently been introduced as a new line of investigation [51], [9]. Recent works suggest that most of the existing ZSL approaches fail to cope up with the GZSL setting. When both seen and unseen classes come into consideration for prediction, the prediction score function becomes highly biased towards seen classes because only seen classes were used for training. As a result, majority

of the unseen test instances are misclassified as seen examples. In other words, this bias notably decreases the classification accuracy on unseen classes while maintains relatively high accuracy on seen classes. To solve this problem, available techniques attempts to estimate the prior probability of an input belonging to either a seen or an unseen class [43], [9]. However, this scheme heavily depends on the original data distribution used for training.

Considering the above aspects, a competent GZSL method should possess the following properties:

- *Equilibrium:* It should be able to balance seen-unseen diversity so that the performances of both seen and unseen classes achieve a balance.
- *Reduced data dependency:* It should not receive any supervision signal (obtained from either training or validation set images) determining the likelihood of an input belonging to seen or unseen class.
- *Consistency:* It should retain its performance on the conventional ZSL setting as well.

In this work, we propose a novel GZSL algorithm to adequately address these challenges.

Generalized CAPD for Seen Class: In Sec. III-A, we described the CAPD of seen classes for a given input image is $\mathbf{P}^{\mathcal{S}} = [\mathbf{p}_1 ... \mathbf{p}_{\mathrm{S}}]$. Each seen CAPDs is obtained using the class-wise learned classifier matrix $\mathbf{W}_s$. It is obvious that each $\mathbf{W}_s$ is biased to seen class 's'. For the same reason, each $\mathbf{p}_s$ is also biased to class 's'. Since there was no seen instance available during the testing phase in conventional ZSL setting, seen CAPDs were not used for prediction (Eq. 8). Therefore, the inherent bias of seen CAPDs was not affecting ZSL performance. In contrast, for GZSL settings, all seen and unseen CAPDs are considered for prediction. Thus, biased seen CAPDs will dominate as expected and significantly affect the unseen class performances. To solve this problem, we propose to develop a generalized version of each seen CAPD as follows:

$$\mathbf{p}_s^g = \mathbf{P}^{\mathcal{S}} \gamma_s, \tag{10}$$

where, γ_s denotes a parameter vector for seen class 's'.

Proposed Objective Function: Our hypothesis is that the bias towards seen classes that causes high scores during prediction can be resolved using the semantic information of classes. To elaborate, γ_s is computed solely in semantic label embedding domain and later applied to generalize CAPD of seen class instances. We minimize the squared difference of two complementary losses to obtain $\gamma = [\gamma_1 ... \gamma_{\mathrm{S}}] \in \mathbb{R}^{\mathrm{S}\times\mathrm{S}}$, as:

$$\min_{\gamma} \| \overbrace{\frac{1}{\mathrm{S}}\sum_{s=1}^{\mathrm{S}}(\mathbf{E}^{\mathcal{S}}\gamma_s - \mathbf{e}_s)^2}^{\text{mean generalized seen loss}} - \overbrace{\frac{1}{\mathrm{U}}\sum_{u=1}^{\mathrm{U}}(\mathbf{E}^{\mathcal{S}}\alpha_u - \mathbf{e}_u)^2}^{\text{mean unseen reconst. loss}} \|_2^2 + \frac{\lambda_\gamma}{2}\sum_{s=1}^{\mathrm{S}} \| \gamma_s \|_2^2, \tag{11}$$

where λ_γ is the regularization weight set using cross validation.

The objective function in Eq. 11 minimizes the squared difference between the mean of two loss components. The **first** component is the mean generalized seen loss which measures the reconstruction accuracy of seen class embedding $\mathbf{e}_s$ using the generalization parameters γ_s. The **second** component measures the reconstruction accuracy of unseen class embedding $\mathbf{e}_u$ from seen classes. By reducing the squared difference between these two components, we indirectly balance the distribution of seen-unseen diversity which effectively prevents the domination of seen classes in the GZSL setting (the '*equilibrium*' property). The interesting fact is that our proposed generalization mechanism does not directly use CAPDs, yet it is strong enough to stabilize the CAPD of different classes during the prediction stage (the '*less data dependence*' property). Furthermore, the formulation does not affect the computation of unseen CAPDs i.e. $\mathbf{p}_u$ which preserves the conventional ZSL performance (the '*consistency*' property).

Prediction: For a given image feature $\mathbf{x}$, we can derive generalized CAPDs of seen classes $\mathbf{p}_s^g$ and CAPD of unseen classes $\mathbf{p}_u$ using the description in Sec. III-B. In test stage, we consider both the projection responses of seen and unseen classes to predict a class.

$$\hat{y} = \arg\max_{l \in \mathbf{y}} \langle \mathbf{p}_l, \mathbf{v}_l \rangle \tag{12}$$

where, $\mathbf{p}_l \in \mathbf{p}_u \cup \mathbf{p}_s^g$ and $\mathbf{v}_l \in \mathbf{e}_s \cup \mathbf{e}_u$.

D. Few-shot Learning

The few-shot learning (FSL) is a natural extension of ZSL. While ZSL considers no instance of an unseen class during training, FSL relaxes this restriction by allowing a few instances of an unseen class as labeled during the training process. Another variant of FSL is called one-shot learning (OSL), which allows exactly one instance of an unseen class (instead of few) as labeled during training. An ideal ZSL approach should be able to benefit from the labeled data for unseen classes under F/OSL settings. In this section, we explain how our approach is easily adaptable to FSL.

Updated CAPD for Unseen Class. In ZSL setting, for a given input image feature, we can calculate the unseen CAPD, $\mathbf{p}_u$ for every unseen class 'u'. Now, in the FSL setting, we optimally use the newly available labeled unseen data to update $\mathbf{p}_u$. To this end, new classifiers $\mathbf{W}_u$ are learned for each unseen class 'u' similar to the case of seen classes (Sec. III-A). For a given image feature, $\mathbf{x}$, we can calculate unseen CAPDs by $\mathbf{p}'_u = \mathbf{W}_u^T \mathbf{x}$. These CAPDs are fused with $\mathbf{p}_u$, which were derived from the linear combination of seen CAPDs (Eq. 7). The updated CAPD for unseen class 'u' is represented as $\mathbf{p}_u^f$, given by:

$$\mathbf{p}_u^f = \delta_u \mathbf{p}_u + \delta'_u \mathbf{p}'_u, \qquad s.t.\ \ \delta_u + \delta'_u = 1 \tag{13}$$

where, δ_u and δ'_u are the contribution of the respective CAPDs to form an updated CAPD of an unseen class. During prediction, we use $\mathbf{p}_u^f$ instead of $\mathbf{p}_u$ in Eq. 8.

Calculation of δ_u and δ'_u: The weights δ_u and δ'_u are set using training data such that they encode the reliability of $\mathbf{p}_u$ and $\mathbf{p}'_u$ respectively. Recall that our prediction is based on the strength of projection of a CAPD on the semantic embedding vector. Therefore, we need to maximize the correspondence

between a CAPD and the correct semantic embedding vector i.e., a high $\langle \mathbf{p}_u, \mathbf{e}_u \rangle$. The unseen CAPD among $\mathbf{p}_u$ and $\mathbf{p}'_u$ that provides higher projection response with u^{th} unseen class semantic vector gets a strong weight during the combination in Eq. 13.

We derive $\mathbf{p}_u$ and $\mathbf{p}'_u$ for each training image feature, $\mathbf{x} \in \mathbf{X}^{\mathcal{S}} = \{\mathbf{X}_s : s \in \mathbf{y}^{\mathcal{S}}\}$, and the classification matrix of unseen class 'u'. Then, we find the summation of maximum projection response of the CAPD (either $\mathbf{p}_u$ or $\mathbf{p}'_u$) with its respective semantic vector. This maximum projection response finds the response of most similar (or confusing) unseen class of any image. The summation of this response across all training images can estimate the overall quality of CAPDs from the two sources. Finally, we normalize the summations to get δ_u and δ'_u as follows:

$$\delta_u = \frac{\sum_{\mathbf{x}\in\mathbf{X}} s \, \max_u \langle \mathbf{p}_u, \mathbf{e}_u \rangle}{\sum_{\mathbf{x}\in\mathbf{X}} s \, \max_u \langle \mathbf{p}_u, \mathbf{e}_u \rangle + \sum_{\mathbf{x}\in\mathbf{X}} s \, \max_u \langle \mathbf{p}'_u, \mathbf{e}_u \rangle},$$
$$\delta'_u = \frac{\sum_{\mathbf{x}\in\mathbf{X}} s \, \max_u \langle \mathbf{p}'_u, \mathbf{e}_u \rangle}{\sum_{\mathbf{x}\in\mathbf{X}} s \, \max_u \langle \mathbf{p}_u, \mathbf{e}_u \rangle + \sum_{\mathbf{x}\in\mathbf{X}} s \, \max_u \langle \mathbf{p}'_u, \mathbf{e}_u \rangle}. \quad (14)$$

In Fig. 9, we further elaborate the overall effect of δ_u and δ'_u in FSL while using different semantic information.

E. Overall pipeline

The overall learning of this unified approach is summarized in Algorithm 1. The training of all settings (i.e., ZSL, GZSL and F/OSL) consists of four main steps. *First*, the calculation of CAPD is performed for seen classes for the case of ZSL and GZSL ($\mathbf{p}_s$) and also for unseen classes for F/OSL settings ($\mathbf{p}'_u$). *Second*, metric learning is performed to link the semantic and visual domains using $\mathbf{M}$. *Third*, the relationship between seen and unseen semantic embeddings is modeled as α_u for all seen classes and β_u for the reduced set of closely related seen classes. For the specific case of GZSL, γ_s is additionally learned to balance the contribution from seen and unseen CAPDs. The test side optimizations to learn parameters α_u, β_u and γ_s is required to be done only once before starting the test and does not take much computation as the total number of classes are not many (around 1500). *Fourth*, the parameters learned in the previous stage are used to obtain unseen CAPDs for the case of ZSL and F/OSL and to further re-balance the seen CAPDs for the case of GZSL. Finally, the CAPD projections are used to predict the output class.

Algorithm 1: Learning procedure for proposed model

Input: Image features $\{\mathbf{X}_s\}_1^{\mathrm{S}}$, $\{\mathbf{X}_u\}_1^{\mathrm{U}}$, Semantic embeddings $\mathbf{E}^{\mathcal{S}}$, $\mathbf{E}^{\mathcal{U}}$

Training Phase

for $s = 1 : \mathrm{S}$ **do**

1 $\quad \mathbf{W}_s \leftarrow$ learn class specific models using Eq. 2

2 $\quad \mathbf{p}_s \leftarrow$ calculate CAPD of seen class s

if *F/OSL* **then**

$\quad$ **for** $u = 1 : \mathrm{U}$ **do**

3 $\quad\quad \mathbf{W}_u \leftarrow$ learn class specific models using Eq. 2

4 $\quad\quad \mathbf{p}'_u \leftarrow$ calculate CAPD of unseen class u

5 $\mathbf{M} \leftarrow$ learn metric space using Eq. 4

Testing Phase

for $u = 1 : \mathrm{U}$ **do**

6 $\quad \alpha_u/\beta_u \leftarrow$ learn the relationship between seen-unseen embeddings using Eq. 6

7 $\quad \mathbf{p}_u \leftarrow$ calculate CAPD of unseen class u

$\quad$ **if** *F/OSL* **then**

8 $\quad\quad$ Calculate δ_u and δ'_u using Eq. 14

9 $\quad\quad \mathbf{p}^f_u \leftarrow$ fuse $\mathbf{p}'_u$ and $\mathbf{p}_u$ using Eq. 13

if *GZSL* **then**

$\quad$ **for** $s = 1 : \mathrm{S}$ **do**

10 $\quad\quad \gamma_s \leftarrow$ minimize cost function in Eq. 11 to learn generalization parameters γ

11 $\quad\quad \mathbf{p}^g_s \leftarrow$ compute updated seen CAPDs using Eq. 10

12 $\quad$ Predict seen or unseen class using Eq. 12

else

13 $\quad$ Predict unseen class using Eq. 8

Return: Class decision $\hat{y}$

IV. Comparison with Related Work

A. ZSL Settings

Our method has similarities with two streams of previous efforts on ZSL. Here, we discuss the significant differences.

In terms of class-specific learning, a number of recent studies [7], [33] report competitive performances when they rely on handcrafted attributes ('*supervised*' source). However, we observe that these methods fail when they use '*unsupervised*' source of semantics (e.g. word2vec and GloVe). The underlying reason is that they do not leverage on the semantic information during the training of the classifiers. Moreover, the attribute set is less noisy than unsupervised source of semantics. Although our work follows the same spirit, the main novelty lies in using the semantic embedding vectors explicitly during the learning phase for each individual class. This helps the classifiers to easily adapt themselves to a wide variety of semantic information sources, e.g. attributes, word2vec and GloVe.

Another body of work [50], [4] considers semantic information during the training process. In the same vein, [59] designed max-margin based loss formulation considering semantic information, while [60] maximizes the likelihood between latent embeddings of visual and semantic concepts. However, these approaches do not take the benefits of class-specific learning. Using a single classifier, they compute a global projection. Generalizing all classes by one projection is restrictive and it fails to encompass subtle variations among classes. These approaches do not leverage the flexibility of suppressing irrelevant seen classes while describing an unseen class. Besides, the semantic label embeddings are subject to tuning based on the visual image features. As they cannot learn any metric on semantic embedding space, these methods fail to work accurately across different semantic embeddings. Another problem is that these kind of approaches do not extend

well for generalized zero-shot and few shot scenario because the training easily gets biased to seen classes which makes difficult to generalize [51] and cannot utilize newly available test data in few-shot settings. In contrast, by taking the benefits of class-specific learning, our approach computes CAPD for each classifier that can significantly enhance the learned discriminative information. In addition, our approach describes the unseen class with automatically selected informative seen classes and learns a metric on the semantic embedding space to further fine-tune the semantic label information. Moreover, our approach can work simultaneously in GZSL and O/FSL settings.

In terms of relating seen and unseen by a linear combination our method has similarity with some previous efforts [48], [56], [5], [10]. [48], [56] applied the combination to convert ZSL problem to a supervised learning problem by generating virtual or synthesized data. For doing so, these approaches required the names of unseen classes during training time which makes unseen pre-defined. [5], [10] used combination of both attribute vector and word vector together in training to relate seen and unseen in semantic space. However, these approaches do not require attribute vectors during testing. It reduces the costly annotation of unseen classes in testing but still utilize costly manual labeling of attributes of seen classes during training. In contrast, our method utilize the seen-unseen combination in an unique way to solve ZSL, GZSL and O/FSL problem. We do not use the concept of attributes to improve the performance of word vectors. Therefore, the unsupervised version of our work is not depended on strong supervision during training.

Many traditional methods focus ZSL but do not perform well in GZSL (See Table VIII, IX and X). Some other methods need to modify ZSL to trusductive [27] or domain adaptation [52] settings to achieve generalization. Again, many approaches perform FSL but do not have the extendibility to zero-shot settings [41], [13]. Moreover, some approaches seem to overfit on small scale dataset, specific image features, and specific semantic vector i.e. supervised-attributes ([59], [60]) or unsupervised word2vec/Glove ([4], [50]). Our method, on the other hand, consistently provides improved performance across all the different semantic information and problem settings.

B. GZSL settings

We automatically balance the diversity of seen-unseen classes in an unsupervised way, without strongly relying on CAPD or image visual feature. Previous efforts used a supervision mechanism either from training or validation image data to determine if any input image belongs to a seen or an unseen class. Chao et al. [9] proposed a calibration based approach to rescale the seen scores and evaluated using Area Under Seen-Unseen accuracy Curve (AUSUC) [7], [52]. As prediction scores of GZSL are strongly biased to seen classes, they proposed to calibrate seen scores by adding a constant negative bias termed as a calibration factor. This factor is calculated on a validation set and works as a prior likelihood of a data point being from a seen/unseen class. The drawback of such an approach is that it acts as a post-processing mechanism applied at the decision making stage, not dealing with the generalization at the basic algorithmic level.

Another alternative work, CMT method [43] incorporates a novelty detection approach which estimates the outlier probability of an input image. Again, the outlier probability is determined using training images which provides an extra image-based supervision to GZSL model. In contrary, our method considers the seen-unseen biasness in the semantic space at the algorithmic level. The overall prediction scores are then balanced to remove the inherent biasness towards the seen classes. We show that such an approach can be useful for both supervised attributes and unsupervised word2vec/GloVe as semantic embedding information. As our approach does not follow the post-processing strategy like [9], [7], [52], we do not evaluate our work with AUSUC. In line with the recommendation in [50], we use harmonic mean based approach for GZSL evaluation.

V. Experiments

Benchmark Datasets: We use four standard datasets for our experiments; aPascal & aYahoo (aPY) [12], Animals with Attributes (AwA) [25], SUN attributes (SUN) [35], and Caltech-UCSD Birds (CUB) [47]. The statistics of these datasets are given in Table I. We follow the standard protocols (seen/unseen splits of classes) used in the literature. To be specific, we have exactly followed [50] for AwA and CUB datasets, [59], [60] for aPY and SUN-10 and [7] for SUN. To increase the complexity of GZSL task for SUN, we used a different of seen/unseen split introduced in [7]. In line with the standard protocol, the test images correspond to only unseen classes in ZSL settings. In Few/One-shot settings, we randomly choose three/one instances per unseen class to use in training as labeled examples. Again, in GZSL settings, we perform a 80-20% split of each seen class instances; 80% portion is used in training and rest 20% for testing in conjunction with all unseen test data. We report the average results of 10 random trails for Few/One shot or GZSL settings. In a recent work, Xian et al. [51] proposed a different seen/unseen split for the above mentioned four datasets. We perform GZSL experiments on that setting as well. We conduct large scale experiment for ZSL problem on ImageNet (ILSVRC) 2012/2010 dataset with the setting of [16], [57]. The training and testing are done on the images of 1K class of ILSVRC 2012 and non-overlapped 360 classes of ILSVRC 2010 dataset respectively. It makes 1.2M images in training from ILSVRC 2012 and 54K images from ILSVRC 2010.

Image Features: Previous ZSL approaches use both shallow (SIFT, PHOG, Fisher Vector, color histogram) and deep features [4], [7], [37]. As reported repeatedly, deep features outperform shallow features by a significant margin [7]. For this reason, we consider only deep features from the pretrained GoogLeNet [44] and VGG-verydeep-19 [42] models for our comparisons. For feature extraction from GoogLeNet and VGG-verydeep-19, we exactly follow Changpinyo *et al.* [7] and Zhang *et al.* [59], respectively. The dimension of visual features extracted from GoogLeNet is 1024, and VGG-verydeep-19 is 4096. While using the recent Xian et al. [51]

Dataset	seen/unseen	# image	# train	# test
aPY[12]	20/12	15,339	12,695	2,644
AwA[25]	40/10	30,475	24,518	6,180
SUN-10[35]	707/10	14,340	14,140	200
SUN[35]	645/72	14,340	12,900	1,440
CUB[47]	150/50	11,788	8,855	2,933
ImageNet[40]	1000/360	1.25M	1.2M	54K

TABLE I: Statistics of the benchmark datasets.

Using G	aPY	AwA	CUB	SUN	ILSVRC
Total seen	20	40	150	717	1000
Reduced-att	10.17	20.00	74.70	344.40	-
Reduced-w2v	-	21.20	70.96	-	134.89
Reduced-glo	-	19.70	74.14	-	-

TABLE II: Average number of the seen classes for reduced set case. Our method automatically selects an optimal number of the nearest seen classes to describe an unseen class.

seen/unseen split, we use the same 2048-dim features from top-layer pooling units of the 101-layered ResNet [20] for a fair comparison. For large scale experiment on ILSVRC 2012/2010 dataset we use GoogleNet features.

Semantic Space Embeddings: We analyze both supervised and unsupervised settings of ZSL. For the supervised case, we use 64, 85, 102 and 312 dimensional continuous valued semantic attributes for aPY, AwA, SUN, and CUB datasets, respectively. We dismiss the binary version of these attributes since [7] showed that continuous attributes are more useful. For the unsupervised case, we test our approach on AwA and CUB datasets. We consider both word vector embeddings i.e., word2vec (w2v) [31], [30] and GloVe (glo) [36]. We use ℓ_2 normalized 400-dimensional word vectors, similar to [50]. For ILSVRC 2012 and 2010 classes we use 500 dimensional word2vec vectors.

Evaluation Metrics: This line of investigation naturally applies to two different tasks; recognition and retrieval [51], [28], [45]. We measure the recognition performance by the top-1 accuracy, and the retrieval performance by the mean average precision (mAP). The top-1 accuracy is the percentage of the estimated labels (the ones with the highest scores) that match the correct labels. The mean average precision is computed over the precision scores of the test classes. In addition, [51] proposed to use Harmonic Mean (HM) of the accuracies of seen and unseen classes (acc_s and acc_u respectively) to evaluate GZSL performance, as follows:

$$HM = \frac{2 \times acc_s \times acc_u}{acc_s + acc_u}.$$

The main motivation of using HM is its ability to estimate the inherent biasness of any method towards seen classes. If a method is too biased to seen classes then acc_s will be very high compared to acc_u and harmonic mean based GZSL performance drops down significantly [51], [9].

Implementation Details: We initialize each classifier $\mathbf{W}_s$ from a $N(0, \frac{1}{k})$ distribution where k is the dimension of the image feature [50]. We use a constant learning rate over 100 iterations in training of each class: 0.001 for AwA and ImageNet, and 0.005 for aPY, SUN and CUB datasets. For each dataset, we select the value of parameters λ_s, λ_u and λ_γ using a validation set. We use the same value of λ_s, λ_u and λ_γ across all seen and unseen classes in the optimization task (Eq. 2 and 6 respectively). To choose the validation dataset, we randomly divide all seen classes into two groups 80%/20%, and use 20% group as the unseen set (no test data is used during validation). Based on the average performance of five different random validation sets, we choose the optimal parameter values. We notice that λ_s has little effect while learning Eq. 2, thus validation really helps to choose word vectors specific tuning parameters λ_u and λ_γ. We choose a value for those parameters within $10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10$. In Fig. 7, we illustrate a validation experiment to choose a value for λ_u for different semantic vectors of AwA and CUB dataset. Our validation performances are different than our test performance because of the averaging of different validation seen/unseen splits of training data. For the metric learning in Eq. 4, we use the standard implementation of [54].

A. Results for Reduced Set

In the reduced set experiment as describe in Sec. III-B, for each unseen class, we select four subsets of the seen classes having one-third of the total number. First three subsets contain the farthest away, mid-range, and nearest seen classes of each unseen class in the semantic embedding space, and the last one is the random selection. For all subsets, we determine the proximity of the unseen classes by Mahalanobis distance with learned metric $\mathbf{M}$. In our experiments, a different unseen class will get a different set of seen classes to describe it. We report the top-1 accuracy on test data of those four subsets in Fig. 4. We observe that only one-third of seen classes closest to each unseen class perform the best among the four subsets. The farthest away, mid-range and randomly selected subsets fail to describe an unseen class with high accuracy. This experiment suggest that using only some nearest seen classes located in the semantic embedding space can efficiently approximate an unseen class embedding. The nearest case experiment performances are not the best accuracies reported in the paper because we consider an automatic seen class selection process in our final experiments.

From the discussion in Sec. III-B, we also know that for different unseen classes our method automatically chooses different sets of useful seen classes. The numbers of seen classes in those sets can be different. In Table II, we report the average number of seen classes in the sets. One can observe that the average number of the seen classes required is around 50% across different datasets. This means, in general, only half of the total seen classes are useful to describe one unseen class. Such a reduced set description of the unseen class not only maintains the best performance but also reduces the complexity of the sparse representation of each unseen class.

Method	aPY		AwA		SUN		CUB	
	V	G	V	G	V	G	V	G
Ours [all-seen]	45.84	50.64	73.19	64.74	84.5	87.00	39.86	42.31
Ours [reduced-Lasso]	36.54	37.22	74.16	75.76	78.50	84.50	37.47	37.37
Ours [reduced-auto w/o $\mathbf{M}$]	46.90	42.78	76.42	77.51	85.00	89.50	42.34	41.36
Ours [reduced-auto with $\mathbf{M}$]	54.69	55.07	78.53	80.43	85.00	79.00	43.01	45.31

TABLE III: Top-1 accuracy (in %) of various versions of CAPD using attributes. V: VGG-verydeep-19, G: GoogLeNet features.

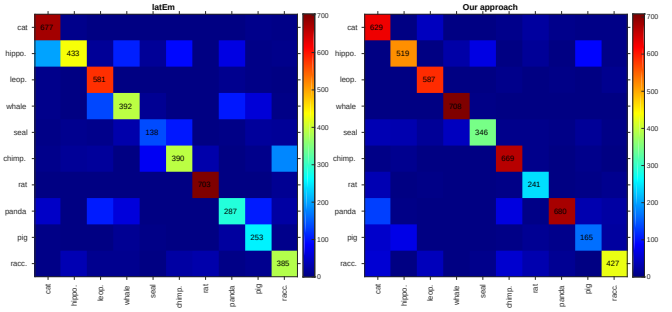


Fig. 6: Confusion matrices on AwA dataset using GoogLeNet as image features and the attributes as semantic space vectors. Left: Xian *et al.* [50]. Right: CAPD. As seen, CAPD provide better overall and class-wise performance.

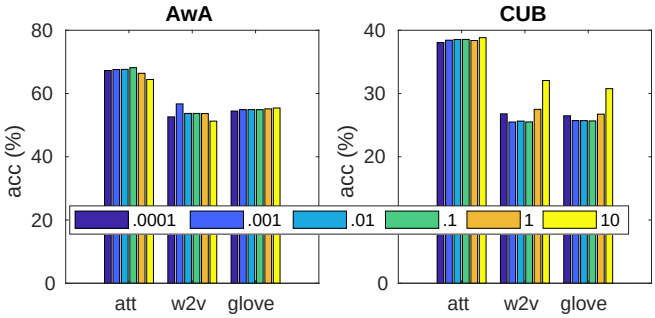


Fig. 7: Validation experiment for choosing λ_u

Using V	aPY	AwA	SUN	CUB
Lampert'14 [26]	38.16	57.23	72.00	31.40
ESZSL'15 [39]	24.22	75.32	82.10	-
SSE-ReLU'15 [59]	46.23	76.33	82.50	30.41
Zhang'16 [60]	50.35	**80.46**	83.30	42.11
Bucher'16 [28]	53.15	77.32	84.41	43.29
DSRL'17[53]	**56.29**	77.38	82.00	**50.26**
MFMR'17[52]	48.20	79.80	84.00	47.70
Ours	54.69	78.53	**85.00**	43.33
Using G	aPY	AwA	SUN	CUB
Lampert'14 [26]	37.10	59.50	-	-
Akata'15 [4]	-	66.70	-	50.10
Changpinyo'16 [7]	-	72.90	-	45.85
Xian'16 [50]	-	72.50	-	45.60
SCoRe'17[32]	-	78.30	-	58.40
MFMR'17[52]	46.40	76.60	81.50	46.20
EXEM'17[8]	-	77.20	-	**59.80**
Ours	**55.07**	**80.83**	**87.00**	45.31

TABLE IV: Supervised ZSL top-1 accuracy (in %) on four standard datasets. V: VGG-verydeep-19 and G: GoogLeNet image features. Results are from the original papers. Only very recent SOTA methods are considered for comparison.

Using V	aPY	AwA	SUN	CUB
SSE-INT'15 [59]	15.43	46.25	58.94	4.69
SSE-ReLU'15 [59]	14.09	42.60	44.55	3.70
Bucher'16 [28]	36.92	68.10	52.68	25.33
Zhang'16 [60]	38.30	67.66	80.10	29.15
MFMR'17 [52]	**45.60**	70.80	77.40	30.60
Ours	43.85	**72.87**	**80.20**	**36.60**

TABLE V: Supervised ZSL retrieval performance (in mAP). V: VGG-verydeep-19 image features.

B. Benchmark Comparisons

We discuss benchmark performances of ZSL recognition and retrieval for both supervised (attributes) and unsupervised semantics (word2vec or GloVe).

1) Results for ZSL with Supervised Attributes[1]*:* We present the top-1 ZSL accuracy results of different versions of our method in Table III. In the all-seen case, we have considered all seen classes to describe an unseen class (Eq. 5). In Lasso, we report the performance using Lasso regularization in place of ℓ_2 in Eq. 6. The results demonstrate that using a reduced number of seen classes with ($\mathbf{M} \neq \mathbb{I}$) or without ($\mathbf{M} = \mathbb{I}$) metric learning to describe an individual unseen class can improve ZSL accuracy. Performance with the metric learning outperforms all other variations of the method in almost every cases except SUN dataset because the implementation of metric learning uses same number images per class for each dataset. However, SUN has a large number of classes but contains less number of images per class. In Table IV, we compare the overall top-1 accuracy of our method (after using validated parameter settings) with many recent ZSL approaches. Our approach outperforms other methods in most of the settings. In Fig. 6, we show confusion matrices of a recent approach [50] and ours. Similar to recognition, ZSL can also perform retrieval task. ZSL retrieval is to search images of unseen classes using their class label embeddings. We test the attributes set as a query to retrieve test images. In Table V, we compare our ZSL retrieval performance with four recent approaches on four datasets. Our approach performs consistently better or comparable to state-of-the-art methods.

2) Results for ZSL with Unsupervised Semantics: ZSL with pretrained word vectors [31], [36] as sematnic embedding is the focus of attention nowadays since it is difficult to generate manually annotated attribute sets in real-world applications.

[1]For fairness, inductive test performances from DSRL [53], MFMR [52] and DMaP [27] are reported in the tables.

Semantic:word2vec	AwA		CUB	
	V	G	V	G
Akata'15 [4]	-	51.20	-	28.40
Xian'16 [50]	-	61.10	-	31.80
Akata'16 [1]	-	-	33.90	-
Changpinyo'16 [7]	-	57.50	-	-
SCoRe'17[32]	-	60.88	-	31.51
DMaP-I'17[27]	-	-	-	26.28
Ours	**66.26**	**66.89**	**34.40**	**32.42**
Semantic: GloVe	AwA		CUB	
	V	G	V	G
Akata'15 [4]	-	58.80	-	24.20
Xian'16 [50]	-	62.90	-	**32.50**
DMaP-I'17[27]	-	-	-	23.69
Ours	**62.01**	**64.73**	**32.08**	29.66

TABLE VI: Unsupervised ZSL performance in top-1 accuracy. V: VGG-verydeep-19, G: GoogLeNet image features. Only very recent SOTA papers are considered for comparison.

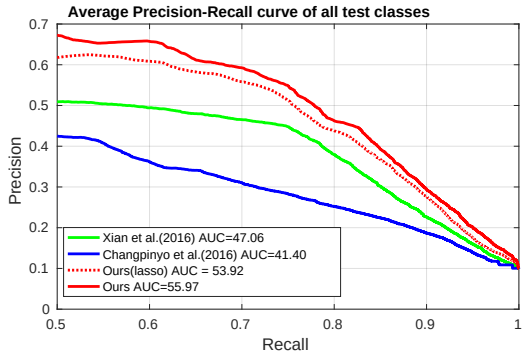


Fig. 8: Average precision recall curve of all test classes of AwA dataset. GoogLeNet and word2vec are used as image feature and semantic label embedding respectively.

Therefore, the ZSL research is pushing forward to eliminate dependency on manually assigned attributes [1], [5], [23], [37], [50]. In line with this view, we adapt our method to unsupervised settings by replacing attribute set with word2vec [31] and GloVe [36] vectors. Our results on two standard datasets, AwA and CUB, are reported in Table VI. We compare with very recent approaches keeping same experimental protocol. One can notice that our approach performs consistently in the unsupervised settings as well in a wide variety of feature and semantic embedding combinations. We provide the average precision-recall curves of ours and two very recent approaches using word2vec embeddings in Fig. 8. As shown, our method is superior to others by a significant margin.

Our observation is that ZSL attains better performance with supervised attributes as semantics than unsupervised ones because the semantic descriptors (word2vec and GloVe) are often noisy and cannot describe a class as good as attributes. To address this performance gap, some works investigate ZSL with transductive learning [52], [27], domain adaptation techniques [11], [23], and class attribute associations [1], [5]. In our study, we consider these improvements as future work.

3) Large-scale ZSL with ILSVRC 2012/2010: To show scalability to large-scale datasets, we evaluate our method on ImageNet for ZSL task. In literature, there are two different standard setups: (1) 800/200 seen/unseen split using only ILSVRC 2010, and (2) 1000/360 split using both ILSVRC 2012/2010. We adopt the second setting because of its larger-scale, difficulty and diversity between train and test sets. Results are reported in Table VII in terms of hit@1 and hit@5 rates. Our approach outperforms several previous techniques, but performs slightly lower compared to the recent [57]. We note that while ours is a stage-wise learning approach, [57] propose an end-to-end method which specifically addresses the hubness problem in large-scale ZSL tasks. Our method achieves performance close to [57] without directly addressing the hubness issue. In future, we will explore the possibility of extending our approach to deal with this problem.

Method	hit@1	hit@5
ConSE[33]	7.8	15.5
DeViSE[15]	5.2	12.8
AMP [17]	6.1	13.1
SS-Voc [16]	9.5	16.8
Zhang'17 [57]	**11.0**	**25.7**
Ours	10.4	23.6

TABLE VII: Comparison on ILSVRC 2012/2010

4) Results for GZSL: GZSL is a more realistic scenario than conventional ZSL because GZSL setting tests a method with not only the unseen class instances but also seen class instances. In this paper, we extend our method to work on GZSL setting as well. Although GZSL is a more interesting problem than ZSL, usually standard ZSL methods do not report any results on GZSL in the original papers. However, recently a few efforts have been published to establish the standard testing protocol for GZSL [51], [9]. In the current work, we test our GZSL method on both testing protocols of [51] and [9].

Xian et al. [51] tested 10 ZSL methods with a new seen-unseen split of datasets ensuring unseen classes are not used during pre-training of deep network (e.g., GoogLeNet, ResNet) which was used to extract image features. They used ResNet as image features and attributes as semantic embedding for SUN, CUB, AwA and aPY dataset. With this exact settings, in Table VIII, we compare our GZSL results with the reported results of [51]. In terms of HM measure, our results consistently outperform other methods by a large margin. Moreover, our method balances the seen-unseen diversity in a robust manner which helps to achieve the best unseen class accuracy (acc_u). In contrast, seen accuracy (acc_u) moves down because of the trade-off while balancing the bias towards seen classes. In the last row, we report the ZSL performance of this experiment where only unseen class test instances are classified to only unseen classes (not considering both seen-unseen classes together). This accuracy is actually an oracle case (upper bound) for acc_u of GZSL case of our method. This is because, if an instance is misclassified in the ZSL case, it must be misclassified in the GZSL case too. Another important point to note is that the parameters of

Top1	SUN			CUB			AWA			aPY		
ResNet	HM	acc_s	acc_u	HM	acc_s	acc_u	HM	acc_s	acc_u	HM	acc_s	acc_u
DAP[26]	7.2	25.1	4.2	3.3	67.9	1.7	0.0	**88.7**	0.0	9.0	78.3	4.8
CONSE[33]	11.6	39.9	6.8	3.1	**72.2**	1.6	0.8	88.6	0.4	0.0	**91.2**	0.0
CMT[43]	13.3	28.0	8.7	8.7	60.1	4.7	15.3	86.9	8.4	19.0	74.2	10.9
SSE[59]	4.0	36.4	2.1	14.4	46.9	8.5	12.9	80.5	7.0	0.4	78.9	0.2
LATEM[50]	19.5	28.8	14.7	24.0	57.3	15.2	13.3	71.7	7.3	0.2	73.0	0.1
ALE[3]	26.3	33.1	21.8	34.4	62.8	23.7	27.5	76.1	16.8	8.7	73.7	4.6
DEVISE[15]	20.9	27.4	16.9	32.8	53.0	23.8	22.4	68.7	13.4	9.2	76.9	4.9
SJE[4]	19.8	30.5	14.7	33.6	59.2	23.5	19.6	74.6	11.3	6.9	55.7	3.7
ESZSL[39]	15.8	27.9	11.0	21.0	63.8	12.6	12.1	75.6	6.6	4.6	70.1	2.4
SYNC[7]	13.4	**43.3**	7.9	19.8	70.9	11.5	16.2	87.3	8.9	13.3	66.6	7.4
Our GZSL	**31.3**	27.8	**35.8**	**43.3**	41.7	**44.9**	**54.5**	68.6	**45.2**	**37.0**	59.5	**26.8**
Our ZSL		49.7			53.8			52.6			39.3	

TABLE VIII: GZSL performance comparison with other established methods in the literature. The experiment setting is exactly same as in [51]. Image features are taken from ResNet and attributes are used as semantic information.

Top1:G	AwA			CUB		
	HM	acc_s	acc_u	HM	acc_s	acc_u
DAP[26]	4.7	77.9	2.4	7.5	55.1	4.0
IAP[26]	3.3	76.8	1.7	2.0	69.4	1.0
ConSE[33]	16.9	75.9	9.5	3.5	69.9	1.8
SynC[7]	0.8	**81.0**	0.4	22.3	**72.0**	13.2
MFMR[52]	29.60	75.6	18.4	-	-	-
Our GZSL	**50.8**	43.2	**61.7**	**29.5**	23.4	**39.9**
Our ZSL	76.2			44.0		

TABLE IX: GZSL performance comparison with the experiment settings of [9]. Image features are taken from GoogLeNet and attributes are used as semantic information.

Top1:Mean	AwA		CUB	
	att	w2v	att	glo
DMaP[27]	17.23	6.44	13.55	2.07
Our GZSL	**52.45**	**43.70**	**31.65**	**18.75**

TABLE X: Comparison with a recent GZSL work DMaP[27]

our method are tuned for GZSL setting in this experiment. Therefore, ZSL performance in the last row may increase if one tunes parameters for the ZSL setting.

Chao et al. [9] experimented GZSL with standard seen-unseen split used in ZSL literature. Keeping this split, they kept random 80% seen class images for training and held out the rest of 20% images for testing stage during GZSL. We perform the same harmonic mean based evaluation like previous setting. In Table IX, we compare our results with the reported results in [9]. Using the same settings, we also compare with two recent methods, MFMR [52] (Table IX) and DMAP [27] (Table X). For the sake of comparison with DMAP [27], we compare mean Top1 accuracy (not standard though) instead of harmonic mean because acc_u and acc_s are not reported separately in the [27]. Again, our method performs consistently well across datasets. More results on GZSL for AwA, CUB, SUN and aPY datasets are reported in Table XII.

5) Results for FSL: As stated earlier, our method can easily take the advantage when new unseen class instances become available as labeled data for training. To test this scenario, in FSL settings, we assume three instances of each unseen class (randomly chosen) are available as labeled during training. In Table XI, we report our results for FSL on AwA and CUB dataset while using attribute, word2vec and GloVe as semantic information. The compared methods, DeViSE [14] and CMT[43], did not report FSL performance in the original paper. But, [45] reimplemented the original work to adapt FSL. The exact three instances of each unseen class used in [45] are not publically available. However, to make our results comparable with others, we report the average performance of 10 random trails. Our method performs consistently better than comparing methods except one case: mAP of CUB-att (58.0 vs 58.5). Another observation from these results is that the performance gap between unsupervised semantics (like word2vec and GloVe) and supervised attribute semantics is significantly reduced compared to ZSL settings where unsupervised semantics always ill-performed than supervised attributes across all methods. The reason is that the FSL setting alleviates the inherent noise of unsupervised semantics to perform better (and as good as) supervised semantic. We also experiment on the OSL task, where all conditions are same as FSL setting except a single randomly picked labeled instance is available for each unseen class during training. More results of OSL and FSL for AwA, CUB, SUN and aPY datasets are reported in Table XII.

For any given image, our FSL method described in Sect. III-D utilizes the contribution of unseen CAPDs coming from two sources: one by combining the CAPDs of seen classes from zero-shot setting and another by using unseen classifier from few-shot setting. In Eq. 13, two constants (δ_u and $\delta_u^{'}$) combine the respective CAPDs to compute the updated CAPD of the unseen class. In this experiment, we visualize the contribution of δ_u and $\delta_u^{'}$ for AwA and CUB dataset in Fig. 9. Few observations from this figure are below:

- In most cases, few-shot contribution from classifier ($\delta_u^{'}$) contributes higher than zero-shot contribution (δ_u). The

Top1: Using G	AwA			CUB		
	att	w2v	glo	att	w2v	glo
DeViSE[14]	80.9	75.3	79.4	54.0	45.7	46.0
CMT[43]	85.1	83.4	84.3	56.7	53.4	52.0
Our	**87.4**	**84.9**	**85.8**	**56.9**	**55.4**	**55.8**
mAP: Using G	AwA			CUB		
	att	w2v	glo	att	w2v	glo
DeViSE[14]	85.0	79.3	84.9	46.4	42.6	42.9
CMT[43]	88.4	88.2	89.2	**58.5**	54.0	52.7
Our	**92.0**	**89.5**	**89.6**	58.0	**56.3**	**56.2**

TABLE XI: FSL performance comparison with the experiment settings of [45]. Image features are taken from GoogLeNet.

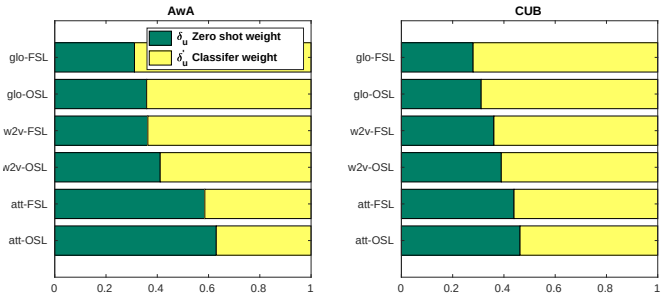


Fig. 9: Contribution of δ_u and $\delta_u^{'}$ to update unseen class CAPD

Dataset	Using G Semantic	OSL	FSL	ZSL	GZSL		
					HM	acc_s	acc_u
aPY	Top1:att	71.2	83.6	40.7	35.7	40.5	32.0
	mAP: att	77.7	88.3	45.1	27.7	24.1	32.7
AwA	Top1:att	82.8	87.4	76.2	50.8	43.2	61.7
	mAP: att	86.9	92.0	71.7	50.0	41.2	63.6
	Top1:w2v	76.9	84.7	56.4	43.6	42.8	44.6
	mAP: w2v	82.0	89.5	50.8	38.5	35.3	42.5
	Top1:glo	78.2	85.8	60.7	44.7	46.4	43.2
	mAP: glo	83.7	89.6	54.3	42.2	37.8	47.9
CUB	Top1:att	46.3	56.9	44.0	29.5	23.4	39.9
	mAP: att	46.9	58.0	40.5	31.8	29.2	34.9
	Top1:w2v	41.7	55.4	33.2	14.9	9.8	31.1
	mAP: w2v	41.9	56.3	29.5	23.2	21.9	24.6
	Top1:glo	41.2	55.8	31.1	11.7	7.2	30.3
	mAP: glo	40.3	56.2	28.3	23.1	22.8	23.4
SUN	SUN (645/72: Seen/Unseen Split)						
	Top1:att	53.7	66.3	59.8	28.3	22.2	39.2
	mAP: att	55.2	68.9	60.5	34.1	27.1	45.9
	SUN-10 (707/10: Seen/Unseen Split)						
	Top1:att	80.8	87.5	77.9	33.6	25.7	48.6
	mAP: att	84.3	90.1	76.8	40.0	32.3	52.7

TABLE XII: All results at a glance on aPY, AwA, CUB and SUN datasets from top to down.

reason is that few instances of unseen class can make better generalization than no instance during training.

- Zero-shot contribution (δ_u) contributes higher on supervised attribute case than word2vec or GloVe across two datasets. The reason is that supervised attributes contain less noise which gives high confidence to zero-shot based CAPD.
- While comparing OSL and FSL, few-shot contribution from classifier ($\delta_u^{'}$) contributes higher in FSL than OSL case. The reason is that in FSL settings, any unseen classifier becomes more confident than OSL settings as FSL observes more than one instances during training.
- While comparing word2vec and GloVe for both OSL and FSL settings, zero-shot contribution (δ_u) contributes higher for word2vec than GloVe semantics. It suggests that word2vec is a better semantic embedding than GloVe for FSL task.
- While comparing AwA and CUB, zero-shot contribution (δ_u) contributes lower than few-shot contribution from classifier ($\delta_u^{'}$) for CUB across all semantics used. The reason is that CUB is a more difficult dataset than AwA in zero-shot setting. One can find that the overall performance on CUB is lower than AwA in all cases (i.e., ZSL, F/OSL and GZSL).

6) All results at a glance.: With experiment setting of [9], we juxtapose all results of OSL, FSL, ZSL and GZSL for AwA, CUB, SUN and aPY datasets in Table XII. Some overall observations from these results are below:

- Performance improves from OSL to FSL settings. This is expected because in FSL setting, more than one (three to be exact) instances of unseen class are used as labeled during training.
- The performance gap between supervised attributes and unsupervised word2vec or GloVe is greatly reduced in OSL and FSL. It suggests that getting few instances as labeled during training helps to greatly compensate the noise of unsupervised semantics.
- O/FSL setting should always outperform ZSL because more information of unseen is revealed in O/FSL settings. However, we got one exception in SUN dataset where OSL perform worse than ZSL. The reason is that the SUN dataset has 717 classes and only one labeled instance of unseen class could not provide discriminative information which eventually confuses our auto unseen CAPD weighting process.
- ZSL results are different from Table IV, V and VI because here our method is tuned for GZSL case not on ZSL. In addition, random selection of 80% training instance of seen classes across 10 different trails affects the result.
- Performance of acc_u of GZSL is always lower than ZSL because ZSL accuracy is the oracle case of acc_u.

7) Complexity Analysis: In Table XIII, we report the training duration for different parameters involved in our approach. These include, class specific weights $\mathbf{W_i}$, learned metric $\mathbf{M}$, seen unseen relationship coefficient α/β, seen generalization coefficient γ and automatic selection of δ_u/δ_u'. As expected, learning $\mathbf{W_i}$ and $\mathbf{M}$ take relatively large time as it involves the main training. Moreover, training time gets increased from AwA, aPY to CUB, SUN and ImageNet as the total number of seen classes as well as number of images are increased. In contrast, learning α/β and γ is very fast because they are based on limited size word vectors only (not involving image features). This means that the overhead our model requires to adapt to GZSL setting is minimal.

Dataset	$\mathbf{W_i}$	$\mathbf{M}$	α/β	γ	δ_u/δ'_u
AwA	92s	2s	2s	2s	10s
CUB	180s	60s	6s	6s	33s
aPY	50s	2s	1s	1s	3s
SUN	250s	110s	20s	50s	100s
ImageNet	48hr	24hr	300s	-	-

TABLE XIII: Training duration for different parameter sets on all datasets.

C. Discussion

Based on our experiments, we draw the following contributions of our work:

Benefits of CAPD: A CAPD points out the most likely class. If a semantic space embedding vector of a class and the CAPD of the image lies close to each other, there is a strong confidence for that class. One important contribution of this paper is the derivation of the CAPD for each unseen class. Conventional ZSL approaches in this vein of thought essentially calculate one principal direction [4], [39], [50], [37], [58]. Generalizing all seen-unseen classes with only one principal direction cannot capture the differences among classes effectively. In our work, each CAPD is obtained with the help of bilinear mapping (matrix multiplication). One can extend this by incorporating latent variables, in line with the work Xian et al. [50] where a collection of bilinear maps along with a selection criterion is used.

Benefits of Nearest Seen Classes: Intuitively, when we describe a novel object, rather than giving a dissimilar object as an example, we use a similar known object. This hints that we can reconstruct the CAPD of an unseen class with the CAPDs of the similar seen classes. This idea helps to improve the prediction performance.

How Many Seen Classes are Required? Results presented in Fig. 4 support the idea that all seen classes are not always necessary. We propose a simple yet effective solution for selecting adaptively the number of similar seen classes for each unseen class (see the discussion in Sec. III-B). This scheme allows different set of useful seen classes required to describe an unseen class.

Extension to GZSL Setting: ZSL methods are biased to assign high prediction scores towards seen classes while performing GZSL task. Due to this reason, conventional ZSL methods fail to achieve good performance in GZSL. Our proposed method solves this problem by adapting seen-unseen class diversity in a novel manner. Unlike [43], [9], our adaptation technique does not take any extra supervision from training/validation image data. We show that class semantic information can be used to adapt seen-unseen diversity.

Extension to Few/One Shot Settings: In some applications, a few images of a new class may become available for training. To adapt with such situations, our method can train a model for the new class without disturbing the previous training. The CAPD from the new model is combined with its previous CAPD (of unseen settings) to obtain an updated CAPD with few-shot refinement. We propose an automatic way of combining CAPDs from two sources by measuring the quality of prediction responses of training images. Our updated CAPD provides better fitness score for unseen class prediction.

VI. Conclusion

We propose a novel unified solution to ZSL, GZSL and F/OSL problems utilizing the concept of class adaptive principal direction (CAPD) that enables efficient and discriminative embeddings of unseen class images in semantic space for recognition and retrieval. We introduce an automatic solution to select a reduced set of relevant seen classes. As demonstrated in our extensive experimental analysis, our method works consistently well in both unsupervised and supervised ZSL settings and achieves the superior performance in particular for the unsupervised case. It provides several benefits including reliable generalization and noise suppression in the semantic space. In addition to ZSL, our method also performs very well in GZSL settings. We propose an easy solution to match the seen-unseen diversity of classes at the algorithmic level. Unlike conventional methods, our GZSL strategy can balance seen-unseen performance to achieve overall better recognition rates. We have extended our CAPD based ZSL approach to adapt with FSL settings. Our approach easily takes the advantage of few examples available in FSL task to fine tune unseen CAPDs to improve classification performance. As a future work, we will extend our approach to transductive settings and domain adaptation.

References

[1] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele. Multi-cue zero-shot learning with strong supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2013.

[3] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-Embedding for Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1425–1438, July 2016.

[4] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June-2015, pages 2927–2936, 2015.

[5] Z. Al-Halah, M. Tapaswi, and R. Stiefelhagen. Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] A. Bendale and T. E. Boult. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1563–1572, 2016.

[7] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-January, pages 5327–5336, 2016.

[8] S. Changpinyo, W.-L. Chao, and F. Sha. Predicting visual exemplars of unseen classes for zero-shot learning. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[9] W.-L. Chao, B. Changpinyo, Soravitand Gong, and F. Sha. *An Empirical Study and Analysis of Generalized Zero-Shot Learning for Object Recognition in the Wild*, pages 52–68. Springer International Publishing, Cham, 2016.

[10] B. Demirel, R. Gokberk Cinbis, and N. Ikizler-Cinbis. Attributes2classname: A discriminative model for attribute-based unsupervised zero-shot learning. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[11] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2584–2591, 2013.

[12] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009.

[13] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, April 2006.

[14] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[15] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2121–2129. Curran Associates, Inc., 2013.

[16] Y. Fu and L. Sigal. Semi-supervised vocabulary-informed learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[17] Z. Fu, T. Xiang, E. Kodirov, and S. Gong. Zero-shot object recognition by semantic manifold distance. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[18] E. Gavves, T. Mensink, T. Tommasi, C. G. M. Snoek, and T. Tuytelaars. Active transfer learning with zero-shot priors: Reusing past datasets for future tasks. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[19] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. volume 2016-January, pages 770–778, 2016. cited By 107.

[21] L. P. Jain, W. J. Scheirer, and T. E. Boult. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision*, pages 393–409. Springer, 2014.

[22] D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3464–3472. Curran Associates, Inc., 2014.

[23] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[24] J. Krause, H. Jin, J. Yang, and L. Fei-Fei. Fine-grained recognition without part annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5546–5555, 2015.

[25] C. 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 Workshops, CVPR Workshops 2009*, pages 951–958, 2009.

[26] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, March 2014.

[27] Y. Li, D. Wang, H. Hu, Y. Lin, and Y. Zhuang. Zero-shot recognition using dual visual-semantic mapping paths. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[28] S. H. Maxime Bucher and F. Jurie. Improving semantic embedding consistency by metric learning for zero-shot classification. In *Proceedings of The 14th European Conference on Computer Vision*, 2016.

[29] T. Mensink, E. Gavves, and C. G. Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2441–2448, 2014.

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

[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[32] P. Morgado and N. Vasconcelos. Semantically consistent regularization for zero-shot recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[33] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

[34] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1410–1418. Curran Associates, Inc., 2009.

[35] G. Patterson, C. Xu, H. Su, and J. Hays. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1-2):59–81, 2014.

[36] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[37] R. Qiao, L. Liu, C. Shen, and A. van den Hengel. Less is more: Zero-shot learning from online textual documents with noise suppression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[38] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1641–1648. IEEE, 2011.

[39] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2152–2161, 2015.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[41] R. Salakhutdinov, J. B. Tenenbaum, and A. Torralba. Learning with hierarchical-deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1958–1971, Aug 2013.

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

[43] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 935–943. Curran Associates, Inc., 2013.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015:1–9, 2015.

[45] Y. H. Tsai, L. Huang, and R. Salakhutdinov. Learning robust visual-semantic embeddings. *CoRR*, abs/1703.05908, 2017.

[46] L. Van Der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of machine learning research*, 15(1):3221–3245, 2014.

[47] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

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

[49] X. Wang and Q. Ji. A unified probabilistic approach modeling relationships between attributes and objects. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2120–2127, 2013.

[50] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[51] Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning - the good, the bad and the ugly. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[52] X. Xu, F. Shen, Y. Yang, D. Zhang, H. T. Shen, and J. Song. Matrix tri-factorization with manifold regularizations for zero-shot learning. In *Proc. of CVPR*, 2017.

[53] M. Ye and Y. Guo. Zero-shot classification with discriminative semantic representation learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[54] Y. Ying and P. Li. Distance metric learning with eigenvalue optimization. *J. Mach. Learn. Res.*, 13(1):1–26, Jan. 2012.

[55] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S. F. Chang. Designing category-level attributes for discriminative visual recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 771–778, June 2013.

[56] J. H. Y. G. Yuchen Guo, Guiguang Ding. Synthesizing samples for zero-shot learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1774–1780, 2017.

[57] L. Zhang, T. Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[58] Y. Zhang, B. Gong, and M. Shah. Fast zero-shot image tagging. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[59] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[60] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.