Predicting Visual Exemplars of Unseen Classes for Zero-Shot Learning

Soravit Changpinyo
U. of Southern California
Los Angeles, CA
schangpi@usc.edu

Wei-Lun Chao
U. of Southern California
Los Angeles, CA
weilun@usc.edu

Fei Sha
U. of Southern California
Los Angeles, CA
feisha@usc.edu

Abstract

Leveraging class semantic descriptions and examples of known objects, zero-shot learning makes it possible to train a recognition model for an object class whose examples are not available. In this paper, we propose a novel zero-shot learning model that takes advantage of clustering structures in the semantic embedding space. The key idea is to impose the structural constraint that semantic representations must be predictive of the locations of their corresponding visual exemplars. *To this end, this reduces to training multiple kernel-based regressors from semantic representation-exemplar pairs from labeled data of the seen object categories. Despite its simplicity, our approach significantly outperforms existing zero-shot learning methods on standard benchmark datasets, including the ImageNet dataset with more than 20,000 unseen categories.*

1. Introduction

A series of major progresses in visual object recognition can largely be attributed to learning large-scale and complex models with a huge number of labeled training images. There are many application scenarios, however, where collecting and labeling training instances can be laboriously difficult and costly. For example, when the objects of interest are rare (e.g., only about a hundred of northern hairy-nosed wombats alive in the wild) or newly defined (e.g., images of futuristic products such as Tesla's Model S), not only the amount of the labeled training images but also the statistical variation among them is limited. These restrictions do not lead to robust systems for recognizing such objects. More importantly, the number of such objects could be significantly greater than the number of common objects. In other words, the frequencies of observing objects follow a long-tailed distribution [37, 51].

Zero-shot learning (ZSL) has since emerged as a promising paradigm to remedy the above difficulties. Unlike supervised learning, ZSL distinguishes between two types of classes: *seen* and *unseen*, where labeled examples are available for the seen classes only. Crucially, zero-shot learners have access to a shared semantic space that embeds all categories. This semantic space enables transferring and adapting classifiers trained on the seen classes to the unseen ones. Multiple types of semantic information have been exploited in the literature: visual attributes [11, 17], word vector representations of class names [12, 39, 27], textual descriptions [10, 19, 32], hierarchical ontology of classes (such as WordNet [26]) [2, 21, 45], and human gazes [15].

Many ZSL methods take a two-stage approach: (i) predicting the embedding of the image in the semantic space; (ii) inferring the class labels by comparing the embedding to the unseen classes' semantic representations [11, 17, 28, 39, 47, 13, 27, 21]. Recent ZSL methods take a unified approach by jointly learning the functions to predict the semantic embeddings as well as to measure similarity in the embedding space [1, 2, 12, 35, 49, 50, 3]. We refer the readers to the descriptions and evaluation on these representative methods in [44].

Despite these attempts, zero-shot learning is proved to be extremely difficult. For example, the best reported accuracy on the full ImageNet with 21K categories is only 1.5% [3], where the state-of-the-art performance with supervised learning reaches 29.8% [6][1].

There are at least two critical reasons for this. First, class semantic representations are vital for knowledge transfer from the seen classes to unseen ones, but these representations are hard to get right. Visual attributes are human-understandable so they correspond well with our object class definition. However, they are not always discriminative [29, 47], not necessarily machine detectable [9, 13], often correlated among themselves ("brown" and "wooden"') [14], and possibly not category-independent ("fluffy" animals and "fluffy" towels) [5]. Word vectors of class names have shown to be inferior to attributes [2, 3]. Derived from texts, they have little knowledge about or are barely aligned with visual information.

[1] Comparison between the two numbers is not entirely fair due to different training/test splits. Nevertheless, it gives us a rough idea on how huge the gap is. This observation has also been shown on small datasets [4].

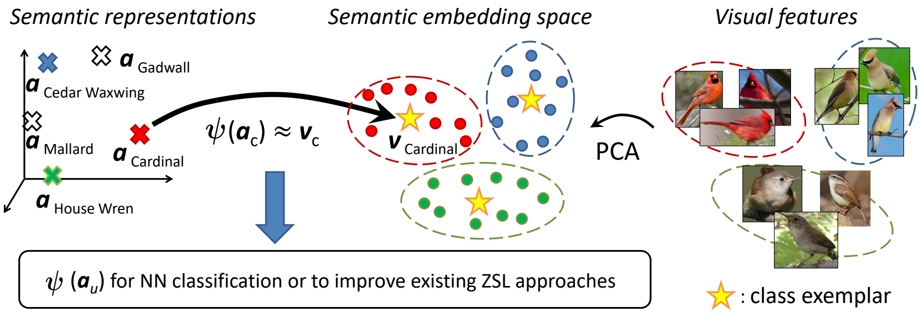
Semantic representations
Semantic embedding space
Visual features
$\boldsymbol{a}$ Gadwall
$\boldsymbol{a}$ Cedar Waxwing
$\boldsymbol{a}$ Mallard
$\boldsymbol{a}$ Cardinal
$\boldsymbol{a}$ House Wren
$\psi(\boldsymbol{a}_c) \approx \boldsymbol{v}_c$
$\boldsymbol{v}$ Cardinal
PCA
ψ ($\boldsymbol{a}_u$) for NN classification or to improve existing ZSL approaches
★ : class exemplar

Figure 1. Given the semantic information and visual features of the seen classes, our method learns a **kernel-based regressor** $\psi(\cdot)$ such that the semantic representation $\boldsymbol{a}_c$ of class c can **predict well** its class exemplar (center) $\boldsymbol{v}_c$ that characterizes the clustering structure. The learned $\psi(\cdot)$ can be used to predict the visual feature vectors of the unseen classes for nearest-neighbor (NN) classification, or to improve the semantic representations for existing ZSL approaches.

The other reason is that the lack of data for the unseen classes presents a unique challenge for model selection. The crux of ZSL involves learning a compatibility function between the visual feature of an image and the semantic representation of each class. But, how are we going to parameterize this function? Complex functions are flexible but at risk of overfitting to the seen classes and transferring poorly to the unseen ones. Simple ones, on the other hand, will result in poorly performing classifiers on the seen classes and will unlikely perform well either on the unseen ones. For these reasons, the success of ZSL methods hinges critically on the insight of the underlying mechanism for transfer and how well that insight is in accordance with data.

One particular fruitful (and often implicitly stated) insight is the existence of clustering structures in the semantic embedding space. That is, images of the same class, after embedded into the semantic space, will cluster around the semantic embedding of that class. For example, ConSE [27] aligns a convex composition of the classifier probabilistic outputs to the semantic representations. A recent method of synthesized classifiers (SynC) [3] models two aligned manifolds of clusters, one corresponding to the semantic embeddings of all objects and the other corresponding to the "centers"[2] in the visual feature space, where the pairwise distances between entities in each space are used to constrain the shapes of both manifolds. These lines of insights have since yielded excellent performance on ZSL.

In this paper, we propose a simple yet very effective ZSL algorithm that assumes and leverages more *structural relations* on the clusters. The main idea is to exploit the intuition that the semantic representation can **predict well** the location of the cluster characterizing all visual feature vectors from the corresponding class (c.f. Sect. 3.2).

More specifically, the main computation step of our approach is reduced to learning (from the seen classes) a predictive function from semantic representations to their corresponding centers (i.e., *exemplars*) of visual feature vectors. This function is used to predict the locations of visual exemplars of the unseen classes that are then used to construct nearest-neighbor style classifiers, or to improve the semantic information demanded by existing ZSL approaches. Fig. 1 shows the conceptual diagram of our approach.

Our proposed method tackles the two challenges for ZSL simultaneously. First, unlike most of the existing ZSL methods, we acknowledge that semantic representations may not necessarily contain visually discriminating properties of objects classes. As a result, we demand that the *predictive* constraint be imposed explicitly. In our case, we assume that the cluster centers of visual feature vectors are our *target* semantic representations. Second, we leverage structural relations on the clusters to further regularize the model, strengthening the usefulness of the clustering structure assumption for model selection.

We validate the effectiveness of our proposed approach on four benchmark datasets for ZSL, including the full ImageNet dataset with more than 20,000 unseen classes. Despite its simplicity, our approach outperforms other existing ZSL approaches in most cases, demonstrating the potential of exploiting the structural relatedness between visual features and semantic information. Additionally, we complement our empirical studies with extensions from zero-shot to few-shot learning, as well as analysis of our approach.

The rest of the paper is organized as follows. We describe our proposed approach in Sect. 2. We demonstrate the superior performance of our method in Sect. 3. We discuss relevant work in Sect. 4 and finally conclude in Sect. 5.

2. Approach

We describe our methods for addressing zero-shot learning, where the task is to classify images from the unseen classes into the label space of the unseen classes. Our approach is based on the structural constraint that takes advantage of the clustering structure assumption in the semantic embedding space. The constraint forces the semantic representations to be predictive of their visual exemplars (i.e.,

[2] The centers are defined as the normals of the hyperplanes separating different classes.

cluster centers). In this section, we describe how we achieve this goal. First, we describe how we learn a function to predict the visual exemplars from the semantic representations. Second, given a novel semantic representation, we describe how we apply this function to perform zero-shot learning.

Notations We follow the notation system introduced in [3] to facilitate comparison. We denote by $\mathcal{D} = \{(\boldsymbol{x}_n \in \mathbb{R}^{\mathsf{D}}, y_n)\}_{n=1}^{\mathsf{N}}$ the training data with the labels from the label space of *seen* classes $\mathcal{S} = \{1, 2, \cdots, \mathsf{S}\}$. we denote by $\mathcal{U} = \{\mathsf{S}+1, \cdots, \mathsf{S}+\mathsf{U}\}$ the label space of *unseen* classes. For each class $c \in \mathcal{S} \cup \mathcal{U}$, let $\boldsymbol{a}_c$ be its semantic representation.

2.1. Learning to predict the visual exemplars from the semantic representations

For each class c, we would like to find a transformation function $\boldsymbol{\psi}(\cdot)$ such that $\boldsymbol{\psi}(\boldsymbol{a}_c) \approx \boldsymbol{v}_c$, where $\boldsymbol{v}_c \in \mathbb{R}^{\mathsf{d}}$ is the visual exemplar for the class. In this paper, we create the visual exemplar of a class by averaging the PCA projections of data belonging to that class. That is, we consider $\boldsymbol{v}_c = \frac{1}{|I_c|}\sum_{n \in I_c} \boldsymbol{M}\boldsymbol{x}_n$, where $I_c = \{i : y_i = c\}$ and $\boldsymbol{M} \in \mathbb{R}^{\mathsf{d} \times \mathsf{D}}$ is the PCA projection matrix computed over training data of the seen classes. We note that $\boldsymbol{M}$ is fixed for all data points (i.e., not class-specific) and is used in Eq. (1).

Given training visual exemplars and semantic representations, we learn d support vector regressors (SVR) with the RBF kernel — each of them predicts each dimension of visual exemplars from their corresponding semantic representations. Specifically, for each dimension $d = 1, \ldots, \mathsf{d}$, we use the ν-SVR formulation [38]. Details are in the supplementary material.

Note that the PCA step is introduced for both the computational and statistical benefits. In addition to reducing dimensionality for faster computation, PCA decorrelates the dimensions of visual features such that we can predict these dimensions independently rather than jointly.

See Sect. 3.3.4 for analysis on applying SVR and PCA.

2.2. Zero-shot learning based on the predicted visual exemplars

Now that we learn the transformation function $\boldsymbol{\psi}(\cdot)$, how do we use it to perform zero-shot classification? We first apply $\boldsymbol{\psi}(\cdot)$ to all semantic representations $\boldsymbol{a}_u$ of the unseen classes. We consider two main approaches that depend on how we interpret these predicted exemplars $\boldsymbol{\psi}(\boldsymbol{a}_u)$.

2.2.1 Predicted exemplars as training data

An obvious approach is to use $\boldsymbol{\psi}(\boldsymbol{a}_u)$ as data directly. Since there is only one data point per class, a natural choice is to use a nearest neighbor classifier. Then, the classifier outputs the label of the closest exemplar for each novel data point $\boldsymbol{x}$ that we would like to classify:

$$\hat{y} = \arg\min_u \quad dis_{NN}(\boldsymbol{M}\boldsymbol{x}, \boldsymbol{\psi}(\boldsymbol{a}_u)), \tag{1}$$

where we adopt the (standardized) Euclidean distance as dis_{NN} in the experiments.

2.2.2 Predicted exemplars as the ideal semantic representations

The other approach is to use $\boldsymbol{\psi}(\boldsymbol{a}_u)$ as the *ideal* semantic representations ("ideal" in the sense that they have knowledge about visual features) and plug them into any existing zero-shot learning framework. We provide two examples.

In the method of convex combination of semantic embeddings (ConSE) [27], their original semantic embeddings are replaced with the corresponding predicted exemplars, while the combining coefficients remain the same. In the method of synthesized classifiers (SynC) [3], the predicted exemplars are used to define the similarity values between the unseen classes and the bases, which in turn are used to compute the combination weights for constructing classifiers. In particular, their similarity measure is of the form $\frac{\exp\{-dis(\boldsymbol{a}_c, \boldsymbol{b}_r)\}}{\sum_{r=1}^{\mathsf{R}} \exp\{-dis(\boldsymbol{a}_c, \boldsymbol{b}_r)\}}$, where dis is the (scaled) Euclidean distance and $\boldsymbol{b}_r$'s are the semantic representations of the base classes. In this case, we simply need to change this similarity measure to $\frac{\exp\{-dis(\boldsymbol{\psi}(\boldsymbol{a}_c), \boldsymbol{\psi}(\boldsymbol{b}_r))\}}{\sum_{r=1}^{\mathsf{R}} \exp\{-dis(\boldsymbol{\psi}(\boldsymbol{a}_c), \boldsymbol{\psi}(\boldsymbol{b}_r))\}}$.

We note that, recently, Chao et al. [4] empirically show that existing semantic representations for ZSL are far from the optimal. Our approach can thus be considered as a way to improve semantic representations for zero-shot learning.

2.3. Comparison to related approaches

One appealing property of our approach is its scalability: we learn and predict at the exemplar (class) level so the runtime and memory footprint of our approach depend only on the number of seen classes rather the number of training data points. This is much more efficient than other ZSL algorithms that learn at the level of each individual training instance [11, 17, 28, 1, 47, 12, 39, 27, 13, 23, 2, 35, 49, 50, 21, 3].

Several methods propose to learn visual exemplars[3] by preserving structures obtained in the semantic space [3, 43, 20]. However, our approach *predicts* them with a regressor such that they may or may not strictly follow the structure in the semantic space, and thus they are more flexible and could even better reflect similarities between classes in the visual feature space.

Similar in spirit to our work, [24] proposes using nearest class mean classifiers for ZSL. The Mahalanobis metric learning in this work could be thought of as learning a linear

[3] *Exemplars* are used loosely here and do not necessarily mean class-specific feature averages.

Table 1. Key characteristics of the datasets

Dataset	# of seen classes	# of unseen classes	# of images
AwA[†]	40	10	30,475
CUB[‡]	150	50	11,788
SUN[‡]	645/646	72/71	14,340
ImageNet[§]	1,000	20,842	14,197,122

[†]: on the prescribed split in [18].
[‡]: on 4 (or 10, respectively) random splits [3], reporting average.
[§]: Seen and unseen classes from ImageNet ILSVRC 2012 1K [36] and Fall 2011 release [8, 12, 27].

transformation of semantic representations (their "zero-shot prior" means, which are in the visual feature space). Our approach learns a highly non-linear transformation. Moreover, our EXEM (1NNS) (cf. Sect. 3.1) learns a (simpler, i.e., diagonal) metric over the learned exemplars. Finally, the main focus of [24] is on *incremental*, not zero-shot, learning settings (see also [34, 31]).

[48] proposes to use a deep feature space as the semantic embedding space for ZSL. Though similar to ours, they do not compute average of visual features (exemplars) but train neural networks to predict *all* visual features from their semantic representations. Their model learning takes significantly longer time than ours. Neural networks are more prone to overfitting and give inferior results (cf. Sect. 3.3.4). Additionally, we provide empirical studies on much larger-scale datasets for both zero-shot and few-shot learning, and analyze the effect of PCA.

3. Experiments

We evaluate our methods and compare to existing state-of-the-art models on four benchmark datasets with diverse domains and scales. Despite variations in datasets, evaluation protocols, and implementation details, we aim to provide a comprehensive and fair comparison to existing methods by following the evaluation protocols in [3]. Note that [3] reports results of many other existing ZSL methods based on their settings. Details on these settings are described below and in the supplementary material.

3.1. Setup

Datasets We use four benchmark datasets for zero-shot learning in our experiments: **Animals with Attributes (AwA)** [18], **CUB-200-2011 Birds (CUB)** [42], **SUN Attribute (SUN)** [30], and **ImageNet** (with full 21,841 classes) [36]. Table 1 summarizes their key characteristics. The supplementary material provides more details.

Semantic representations We use the publicly available 85, 312, and 102 dimensional continuous-valued attributes for **AwA**, **CUB**, and **SUN**, respectively. For **ImageNet**, there are two types of semantic representations of the class names. First, we use the 500 dimensional word vectors [3] obtained from training a skip-gram model [25] on Wikipedia. We remove the class names without word vectors, making the number of unseen classes to be 20,345 (out of 20,842). Second, we derive 21,632 dimensional semantic vectors of the class names using multidimensional scaling (MDS) on the WordNet hierarchy, as in [21]. We normalize the class semantic representations to have unit ℓ_2 norms.

Visual features We use GoogLeNet features (1,024 dimensions) [40] provided by [3] due to their superior performance [2, 3] and prevalence in existing literature on ZSL.

Evaluation protocols For **AwA**, **CUB**, and **SUN**, we use the multi-way classification accuracy (averaged over classes) as the evalution metric. On **ImageNet**, we describe below additional metrics and protocols introduced in [12] and followed by [3, 21, 27].

First, two evaluation metrics are employed: Flat hit@K (F@K) and Hierarchical precision@K (HP@K). F@K is defined as the percentage of test images for which the model returns the true label in its top K predictions. Note that, F@1 is the multi-way classification accuracy (averaged over samples). HP@K is defined as the percentage of overlapping (i.e., precision) between the model's top K predictions and the ground-truth list. For each class, the ground-truth list of its K closest categories is generated based on the ImageNet hierarchy [8]. See the Appendix of [12, 3] for details. Essentially, this metric allows for some errors as long as the predicted labels are semantically similar to the true one.

Second, we evaluate ZSL methods on three subsets of the test data of increasing difficulty: *2-hop*, *3-hop*, and *All*. *2-hop* contains 1,509 (out of 1,549) unseen classes that are within 2 tree hops of the 1K seen classes according to the ImageNet hierarchy. *3-hop* contains 7,678 (out of 7,860) unseen classes that are within 3 tree hops of seen classes. Finally, *All* contains all 20,345 (out of 20,842) unseen classes in the ImageNet 2011 21K dataset that are not in the ILSVRC 2012 1K dataset.

Note that word vector embeddings are missing for certain class names with rare words. For the MDS-WordNet features, we provide results for *All* only for comparison to [21]. In this case, the number of unseen classes is 20,842.

Baselines We compare our approach with several state-of-the-art and recent competitive ZSL methods summarized in Table 3. Our main focus will be on SYNC [3], which has recently been shown to have superior performance against competitors under the same setting, especially on large-scale datasets [44]. Note that SYNC has two versions: one-versus-other loss formulation SYNC$^{\text{o-v-o}}$ and the Crammer-Singer formulation [7] SYNC$^{\text{struct}}$. On small datasets, we also report results from recent competitive baselines LATEM [45] and BIDILEL [43]. For additional details regarding other (weaker) baselines, see the supplementary material. Finally, we compare our approach to all ZSL methods that provide results on **ImageNet**. When using word vectors of the class names as semantic representations, we com-

Table 2. We compute the Euclidean distance matrix between the *unseen* classes based on semantic representations ($\boldsymbol{D}_{\boldsymbol{a}_u}$), predicted exemplars ($\boldsymbol{D}_{\boldsymbol{\psi}(\boldsymbol{a}_u)}$), and real exemplars ($\boldsymbol{D}_{\boldsymbol{v}_u}$). Our method leads to $\boldsymbol{D}_{\boldsymbol{\psi}(\boldsymbol{a}_u)}$ that is better correlated with $\boldsymbol{D}_{\boldsymbol{v}_u}$ than $\boldsymbol{D}_{\boldsymbol{a}_u}$ is. See text for more details.

Dataset name	Correlation to $\boldsymbol{D}_{\boldsymbol{v}_u}$	
	Semantic distances $\boldsymbol{D}_{\boldsymbol{a}_u}$	Predicted exemplar distances $\boldsymbol{D}_{\boldsymbol{\psi}(\boldsymbol{a}_u)}$
AwA	0.862	**0.897**
CUB	0.777 ± 0.021	**0.904** ± 0.026
SUN	0.784 ± 0.022	**0.893** ± 0.019

pare our method to ConSE [27] and SynC [3]. When using MDS-WordNet features as semantic representations, we compare our method to SynC [3] and CCA [21].

Variants of our ZSL models given predicted exemplars The main step of our method is to predict visual exemplars that are well-informed about visual features. How we proceed to perform zero-shot classification (i.e., classifying test data into the label space of unseen classes) based on such exemplars is entirely up to us. In this paper, we consider the following zero-shot classification procedures that take advantage of the predicted exemplars:

- EXEM (*ZSL method*): ZSL method with predicted exemplars as semantic representations, where *ZSL method* = ConSE [27], LatEm [45], and SynC [3].
- EXEM (1NN): 1-nearest neighbor classifier with the Euclidean distance to the exemplars.
- EXEM (1NNs): 1-nearest neighbor classifier with the *standardized* Euclidean distance to the exemplars, where the standard deviation is obtained by averaging the intra-class standard deviations of all seen classes.

EXEM (*ZSL method*) regards the predicted exemplars as the ideal semantic representations (Sect. 2.2.2). On the other hand, EXEM (1NN) treats predicted exemplars as data prototypes (Sect. 2.2.1). The standardized Euclidean distance in EXEM (1NNs) is introduced as a way to scale the variance of different dimensions of visual features. In other words, it helps reduce the effect of *collapsing* data that is caused by our usage of the average of each class' data as cluster centers.

Hyper-parameter tuning We simulate zero-shot scenarios to perform 5-fold cross-validation during training. Details are in the supplementary material.

3.2. Predicted visual exemplars

We first show that predicted visual exemplars better reflect visual similarities between classes than semantic representations. Let $\boldsymbol{D}_{\boldsymbol{a}_u}$ be the pairwise Euclidean distance matrix between *unseen* classes computed from semantic representations (i.e., U by U), $\boldsymbol{D}_{\boldsymbol{\psi}(\boldsymbol{a}_u)}$ the distance matrix computed from predicted exemplars, and $\boldsymbol{D}_{\boldsymbol{v}_u}$ the distance matrix computed from real exemplars (which we do not have access to). Table 2 shows that the correlation between $\boldsymbol{D}_{\boldsymbol{\psi}(\boldsymbol{a}_u)}$ and $\boldsymbol{D}_{\boldsymbol{v}_u}$ is much higher than that between $\boldsymbol{D}_{\boldsymbol{a}_u}$ and $\boldsymbol{D}_{\boldsymbol{v}_u}$. Importantly, we improve this correlation without access to any data of the unseen classes. See also similar results using another metric in the supplementary material.

We then show some t-SNE [41] visualization of predicted visual exemplars of the *unseen* classes. Ideally, we would like them to be as close to their corresponding real images as possible. In Fig. 2, we demonstrate that this is indeed the case for many of the unseen classes; for those unseen classes (each of which denoted by a color), their real images (crosses) and our predicted visual exemplars (circles) are well-aligned.

The quality of predicted exemplars (in this case based on the distance to the real images) depends on two main factors: the predictive capability of semantic representations and the number of semantic representation-visual exemplar pairs available for training, which in this case is equal to the number of seen classes S. On **AwA** where we have only 40 training pairs, the predicted exemplars are surprisingly accurate, mostly either placed in their corresponding clusters or at least closer to their clusters than predicted exemplars of the other unseen classes. Thus, we expect them to be useful for discriminating among the unseen classes. On **ImageNet**, the predicted exemplars are not as accurate as we would have hoped, but this is expected since the word vectors are purely learned from text.

We also observe relatively well-separated clusters in the semantic embedding space (in our case, also the visual feature space since we only apply PCA projections to the visual features), confirming our assumption about the existence of clustering structures. On **CUB**, we observe that these clusters are more mixed than on other datasets. This is not surprising given that it is a fine-grained classification dataset of bird species.

3.3. Zero-shot learning results

3.3.1 Main results

Table 3 summarizes our results in the form of multi-way classification accuracies on all datasets. We significantly outperform recent state-of-the-art baselines when using GoogLeNet features. In the supplementary material, we provide additional quantitative and qualitative results, including those on generalized zero-shot learning task [4].

We note that, on **AwA**, several recent methods obtain higher accuracies due to using a more optimistic evaluation metric (per-sample accuracy) and new types of deep features [48, 49]. This has been shown to be unsuccessfully replicated (cf. Table 2 in [44]). See the supplementary material for results of these and other less competitive baselines.

Our alternative approach of treating predicted visual exemplars as the ideal semantic representations significantly outperforms taking semantic representations as

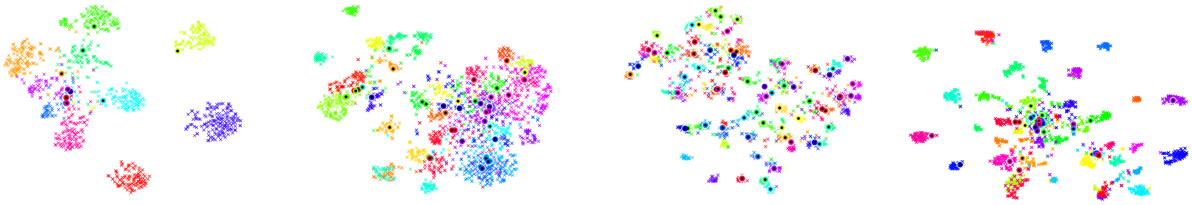

Figure 2. t-SNE [41] visualization of randomly selected real images (crosses) and predicted visual exemplars (circles) for the *unseen* classes on (from left to right) **AwA**, **CUB**, **SUN**, and **ImageNet**. Different colors of symbols denote different unseen classes. Perfect predictions of visual features would result in well-aligned crosses and circles of the same color. Plots for **CUB** and **SUN** are based on their first splits. Plots for **ImageNet** are based on randomly selected 48 unseen classes from *2-hop* and word vectors as semantic representations. Best viewed in color. See the supplementary material for larger figures.

Table 3. Comparison between existing ZSL approaches in multi-way classification accuracies (in %) on four benchmark datasets. For each dataset, we mark the best in red and the second best in blue. *Italic* numbers denote *per-sample* accuracy instead of *per-class* accuracy. On **ImageNet**, we report results for both types of semantic representations: Word vectors (wv) and MDS embeddings derived from WordNet (hie). All the results are based on GoogLeNet features [40].

Approach	**AwA**	**CUB**	**SUN**	**ImageNet**	
				wv	hie
ConSE† [27]	63.3	36.2	51.9	*1.3*	-
BiDiLEL [43]	72.4	49.7§	-	-	-
LatEm‡ [45]	72.1	48.0	64.5	-	-
CCA [21]	-	-	-	-	*1.8*
SynC$^{\text{o-vs-o}}$ [3]	69.7	53.4	62.8	*1.4*	***2.0***
SynC$^{\text{struct}}$ [3]	72.9	54.5	62.7	*1.5*	-
EXEM (ConSE)	70.5	46.2	60.0	-	-
EXEM (LatEm)‡	72.9	56.2	**67.4**	-	-
EXEM (SynC$^{\text{o-vs-o}}$)	73.8	56.2	66.5	*1.6*	***2.0***
EXEM (SynC$^{\text{struct}}$)	**77.2**	**59.8**	66.1	-	-
EXEM (1NN)	76.2	56.3	**69.6**	***1.7***	***2.0***
EXEM (1NNs)	**76.5**	**58.5**	**67.3**	***1.8***	***2.0***

§: on a particular split of seen/unseen classes. †: reported in [3].
‡: based on the code of [45], averaged over 5 different initializations.

given. EXEM (SynC), EXEM (ConSE), EXEM (LatEm) outperform their corresponding *base* ZSL methods relatively by 5.9-6.8%, 11.4-27.6%, and 1.1-17.1%, respectively. This again suggests improved quality of semantic representations (on the predicted exemplar space).

Furthermore, we find that there is no clear winner between using predicted exemplars as ideal semantic representations or as data prototypes. The former seems to perform better on datasets with fewer seen classes. Nonetheless, we remind that using 1-nearest-neighbor classifiers clearly scales much better than zero-shot learning methods; EXEM (1NN) and EXEM (1NNs) are more efficient than EXEM (SynC), EXEM (ConSE), and EXEM (LatEm).

Finally, we find that in general using the standardized Euclidean distance instead of the Euclidean distance for nearest neighbor classifiers helps improve the accuracy, especially on **CUB**, suggesting there is a certain effect of collapsing actual data during training. The only exception is on **SUN**. We suspect that the standard deviation values computed on the seen classes on this dataset may not be robust enough as each class has only 20 images.

3.3.2 Large-scale zero-shot classification results

We then provide expanded results for **ImageNet**, following evaluation protocols in the literature. In Table 4 and 5, we provide results based on the exemplars predicted by word vectors and MDS features derived from WordNet, respectively. We consider SynC$^{\text{o-v-o}}$, rather than SynC$^{\text{struct}}$, as the former shows better performance on **ImageNet** [3]. Regardless of the types of metrics used, our approach outperforms the baselines significantly when using word vectors as semantic representations. For example, on *2-hop*, we are able to improve the F@1 accuracy by 2% over the state-of-the-art. However, we note that this improvement is not as significant when using MDS-WordNet features as semantic representations.

We observe that the 1-nearest-neighbor classifiers perform better than using predicted exemplars as more powerful semantic representations. We suspect that, when the number of classes is very high, zero-shot learning methods (ConSE or SynC) do not fully take advantage of the *meaning* provided by each dimension of the exemplars.

3.3.3 From zero-shot to few-shot learning

In this section, we investigate what will happen when we allow ZSL algorithms to *peek* into some labeled data from part of the unseen classes. Our focus will be on *All* categories of **ImageNet**, two ZSL methods (SynC$^{\text{o-vs-o}}$ and EXEM (1NN)), and two evaluation metrics (F@1 and F@20). For brevity, we will denote SynC$^{\text{o-vs-o}}$ and EXEM (1NN) by SynC and EXEM, respectively.

Setup We divide images from each unseen class into two sets. The first 20% are reserved as training examples that may or may not be revealed. This corresponds to on average 127 images per class. If revealed, those *peeked* unseen classes will be marked as seen, and their labeled data can be used for training. The other 80% are for testing. The test

Table 4. Comparison between existing ZSL approaches on **ImageNet** using **word vectors** of the class names as semantic representations. For both metrics (in %), the higher the better. The best is in red. The numbers of unseen classes are listed in parentheses. †: reported in [3].

Test data	Approach	Flat Hit@K					Hierarchical precision@K			
	K=	1	2	5	10	20	2	5	10	20
2-hop (1,509)	ConSE† [27]	8.3	12.9	21.8	30.9	41.7	21.5	23.8	27.5	31.3
	Sync$^{\text{o-vs-o}}$ [3]	10.5	16.7	28.6	40.1	52.0	25.1	27.7	30.3	32.1
	EXEM (Sync$^{\text{o-vs-o}}$)	11.8	18.9	31.8	43.2	54.8	25.6	28.1	30.2	31.6
	EXEM (1NN)	11.7	18.3	30.9	42.7	54.8	25.9	28.5	**31.2**	**33.3**
	EXEM (1NNs)	**12.5**	**19.5**	**32.3**	**43.7**	**55.2**	**26.9**	**29.1**	31.1	32.0
3-hop (7,678)	ConSE† [27]	2.6	4.1	7.3	11.1	16.4	6.7	21.4	23.8	26.3
	Sync$^{\text{o-vs-o}}$ [3]	2.9	4.9	9.2	14.2	20.9	7.4	23.7	26.4	28.6
	EXEM (Sync$^{\text{o-vs-o}}$)	3.4	5.6	10.3	15.7	22.8	7.5	24.7	27.3	29.5
	EXEM (1NN)	3.4	5.7	10.3	15.6	22.7	8.1	**25.3**	**27.8**	**30.1**
	EXEM (1NNs)	**3.6**	**5.9**	**10.7**	**16.1**	**23.1**	**8.2**	25.2	27.7	29.9
All (20,345)	ConSE† [27]	1.3	2.1	3.8	5.8	8.7	3.2	9.2	10.7	12.0
	Sync$^{\text{o-vs-o}}$ [3]	1.4	2.4	4.5	7.1	10.9	3.1	9.0	10.9	12.5
	EXEM (Sync$^{\text{o-vs-o}}$)	1.6	2.7	5.0	7.8	11.8	3.2	9.3	11.0	12.5
	EXEM (1NN)	1.7	2.8	5.2	8.1	12.1	**3.7**	**10.4**	**12.1**	**13.5**
	EXEM (1NNs)	**1.8**	**2.9**	**5.3**	**8.2**	**12.2**	3.6	10.2	11.8	13.2

Table 5. Comparison between existing ZSL approaches on **ImageNet** (with 20,842 unseen classes) using **MDS embeddings derived from WordNet [21]** as semantic representations. The higher, the better (in %). The best is in red.

Test data	Approach	Flat Hit@K				
	K=	1	2	5	10	20
All (20,842)	CCA [21]	1.8	3.0	5.2	7.3	9.7
	Sync$^{\text{o-vs-o}}$ [3]	**2.0**	3.4	6.0	8.8	12.5
	EXEM (Sync$^{\text{o-vs-o}}$)	**2.0**	3.3	6.1	9.0	12.9
	EXEM (1NN)	**2.0**	**3.4**	**6.3**	**9.2**	13.1
	EXEM (1NNs)	**2.0**	**3.4**	6.2	**9.2**	**13.2**

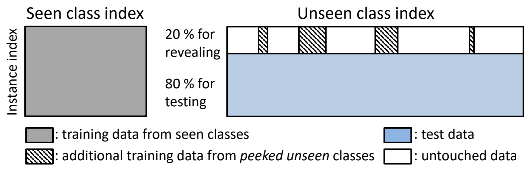

Figure 3. Data split for zero-to-few-shot learning on **ImageNet**

set is always fixed such that we have to do few-shot learning for peeked unseen classes and zero-shot learning on the rest of the unseen classes. Fig. 3 summarizes this protocol.

We then vary the number of peeked unseen classes B. Also, for each of these numbers, we explore the following subset selection strategies (more details are in the supplementary material): (i) **Uniform random**: Randomly selected B unseen classes from the uniform distribution; (ii) **Heavy-toward-seen random** Randomly selected B classes that are semantically similar to seen classes according to the WordNet hierarchy; (iii) **Light-toward-seen random** Randomly selected B classes that are semantically far away from seen classes; (iv) **K-means clustering for coverage** Classes whose semantic representations are nearest to each cluster's center, where semantic embeddings of the unseen classes are grouped by k-means clustering with $k = B$; (v) **DPP for diversity** Sequentially selected classes by a greedy algorithm for fixed-sized determinantal point processes (k-DPPs) [16] with the RBF kernel computed on semantic representations.

Results For each of the ZSL methods (EXEM and Sync), we first compare different subset selection methods when the number of peeked unseen classes is small (up to 2,000) in Fig. 4. We see that the performances of different subset selection methods are consistent across ZSL methods. Moreover, *heavy-toward-seen classes* are preferred for *strict* metrics (Flat Hit@1) but *clustering* is preferred for *flexible* metrics (Flat Hit@20). This suggests that, for a strict metric, it is better to *pick the classes that are semantically similar to what we have seen*. On the other hand, if the metric is flexible, we should focus on providing *coverage* for all the classes so each of them has knowledge they can transfer from.

Next, using the best performing heavy-toward-seen selection, we focus on comparing EXEM and Sync with larger numbers of peeked unseen classes in Fig. 5. When the number of peeked unseen classes is small, EXEM outperforms Sync. (In fact, EXEM outperforms Sync for *each* subset selection method in Fig. 4.) However, we observe that Sync will finally catch up and surpass EXEM. This is not surprising; as we observe more labeled data (due to the increase in peeked unseen set size), the setting will become more similar to supervised learning (few-shot learning), where linear classifiers used in Sync should outperform nearest center classifiers used by EXEM. Nonetheless, we note that EXEM is more computationally advantageous than Sync. In particular, when training on 1K classes of **ImageNet** with over 1M images, EXEM takes 3 mins while Sync 1 hour. We provide additional results under this scenario in the supplementary material.

3.3.4 Analysis

PCA or not? Table 6 investigates the effect of PCA. In general, EXEM (1NN) performs comparably with and without PCA. Moreover, decreasing PCA projected dimension d from 1024 to 500 does not hurt the performance. Clearly, a

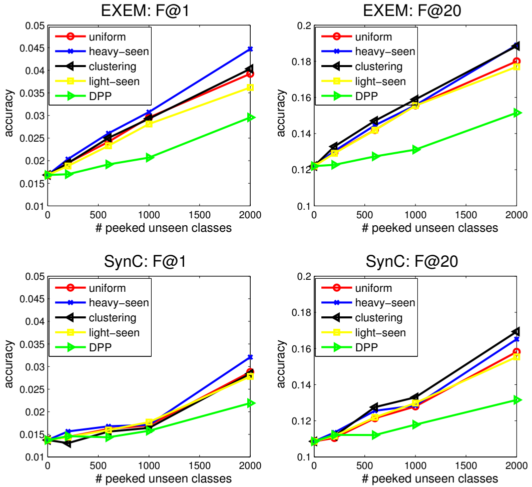

Figure 4. Accuracy vs. the number of peeked unseen classes for EXEM (top) and SYNC (bottom) **across different subset selection methods**. Evaluation metrics are F@1 (left) and F@20 (right).

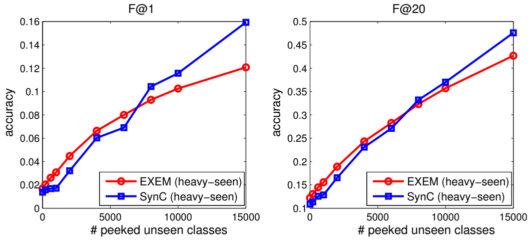

Figure 5. Accuracy vs. the number of peeked unseen classes for EXEM and SYNC for heavy-toward-seen class selection strategy. Evaluation metrics are F@1 (left) and F@20 (right).

Table 6. Accuracy of EXEM (1NN) on **AwA**, **CUB**, and **SUN** when predicted exemplars are from original visual features (No PCA) and PCA-projected features (PCA with d = 1024 and d = 500).

Dataset name	No PCA d = 1024	PCA d = 1024	PCA d = 500
AwA	**77.8**	76.2	76.2
CUB	55.1	**56.3**	**56.3**
SUN	69.2	**69.6**	**69.6**

Table 7. Comparison between EXEM (1NN) with support vector regressors (SVR) and with 2-layer multi-layer perceptron (MLP) for predicting visual exemplars. Results on **CUB** are for the first split. Each number for MLP is an average over 3 random initialization.

Dataset name	How to predict exemplars	No PCA d = 1024	PCA d = 1024	PCA d = 500
AwA	SVR	**77.8**	76.2	**76.2**
	MLP	76.1 ± 0.5	**76.4** ± 0.1	75.5 ± 1.7
CUB	SVR	**57.1**	**59.4**	**59.4**
	MLP	53.8 ± 0.3	54.2 ± 0.3	53.8 ± 0.5

smaller PCA dimension leads to faster computation due to fewer regressors to be trained. See additional results with other values for d in the supplementary material.

Kernel regression vs. Multi-layer perceptron We compare two approaches for predicting visual exemplars: kernel-based support vector regressors (SVR) and 2-layer multi-layer perceptron (MLP) with ReLU nonlinearity. MLP weights are ℓ_2 regularized, and we cross-validate the regularization constant. Additional details are in the supplementary material.

Table 7 shows that SVR performs more robustly than MLP. One explanation is that MLP is prone to overfitting due to the small training set size (the number of seen classes) as well as the model selection challenge imposed by ZSL scenarios. SVR also comes with other benefits; it is more efficient and less susceptible to initialization.

4. Related Work

ZSL has been a popular research topic in both computer vision and machine learning. A general theme is to make use of semantic representations such as attributes or word vectors to relate visual features of the seen and unseen classes, as summarized in [1].

Our approach for predicting visual exemplars is inspired by [12, 27]. They predict an image's semantic embedding from its visual features and compare to unseen classes' semantic embeddings. As mentioned in Sect. 2.3, we perform "inverse prediction": given an unseen class's semantic representation, we predict where the exemplar visual feature vector for that class is in the semantic embedding space.

There has been a recent surge of interest in applying deep learning models to generate images [22, 33, 46]. Most of these methods are based on probabilistic models (in order to incorporate the statistics of natural images). Unlike them, our prediction is to purely deterministically predict visual exemplars (features). Note that, generating features directly is likely easier and more effective than generating realistic images first and then extracting visual features from them.

5. Discussion

We have proposed a novel ZSL model that is simple but very effective. Unlike previous approaches, our method directly solves ZSL by predicting visual exemplars — cluster centers that characterize visual features of the unseen classes of interest. This is made possible partly due to the well separate cluster structure in the deep visual feature space. We apply predicted exemplars to the task of zero-shot classification based on two views of these exemplars: ideal semantic representations and prototypical data points. Our approach achieves state-of-the-art performance on multiple standard benchmark datasets. Finally, we also analyze our approach and compliment our empirical studies with an extension of zero-shot to few-shot learning.

Acknowledgements This work is partially supported by USC Graduate Fellowship, NSF IIS-1065243, 1451412, 1513966/1632803, 1208500, CCF-1139148, a Google Research Award, an Alfred. P. Sloan Research Fellowship and ARO# W911NF-12-1-0241 and W911NF-15-1-0484.

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013. 1, 3, 8

[2] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015. 1, 3, 4

[3] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016. 1, 2, 3, 4, 5, 6, 7

[4] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, 2016. 1, 3, 5

[5] C.-Y. Chen and K. Grauman. Inferring analogous attributes. In *CVPR*, 2014. 1

[6] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman. Project Adam: Building an efficient and scalable deep learning training system. In *OSDI*, 2014. 1

[7] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2002. 4

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[9] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, 2012. 1

[10] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *ICCV*, 2013. 1

[11] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1, 3

[12] 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. 1, 3, 4, 8

[13] D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In *NIPS*, 2014. 1, 3

[14] D. Jayaraman, F. Sha, and K. Grauman. Decorrelating semantic visual attributes by resisting the urge to share. In *CVPR*, 2014. 1

[15] N. Karessli, Z. Akata, A. Bulling, and B. Schiele. Gaze embeddings for zero-shot image classification. In *CVPR*, 2017. 1

[16] A. Kulesza and B. Taskar. k-dpps: Fixed-size determinantal point processes. In *ICML*, 2011. 7

[17] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 1, 3

[18] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 36(3):453–465, 2014. 4

[19] J. Lei Ba, K. Swersky, S. Fidler, and R. Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *ICCV*, 2015. 1

[20] Y. Long, L. Liu, L. Shao, F. Shen, G. Ding, and J. Han. From zero-shot learning to conventional supervised classification: Unseen visual data synthesis. In *CVPR*, 2017. 3

[21] Y. Lu. Unsupervised learning of neural network outputs: with application in zero-shot learning. In *IJCAI*, 2016. 1, 3, 4, 5, 6, 7

[22] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov. Generating images from captions with attention. In *ICLR*, 2016. 8

[23] T. Mensink, E. Gavves, and C. G. Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*, 2014. 3

[24] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *TPAMI*, 35(11):2624–2637, 2013. 3, 4

[25] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR Workshops*, 2013. 4

[26] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 1

[27] 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. In *ICLR*, 2014. 1, 2, 3, 4, 5, 6, 7, 8

[28] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *NIPS*, 2009. 1, 3

[29] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR*, 2011. 1

[30] G. Patterson, C. Xu, H. Su, and J. Hays. The sun attribute database: Beyond categories for deeper scene understanding. *IJCV*, 108(1-2):59–81, 2014. 4

[31] S.-A. Rebuffi, A. Kolesnikov, and C. H. Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, 2017. 4

[32] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 1

[33] S. Reed, Z. Akata, X. Yan, L. Logeswaran, H. Lee, and B. Schiele. Generative adversarial text to image synthesis. In *ICML*, 2016. 8

[34] M. Ristin, M. Guillaumin, J. Gall, and L. Van Gool. Incremental learning of random forests for large-scale image classification. *TPAMI*, 38(3):490–503, 2016. 4

[35] B. Romera-Paredes and P. H. S. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 1, 3

[36] 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. *IJCV*, 2015. 4

[37] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011. 1

[38] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000. 3

[39] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013. 1, 3

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 4, 6

[41] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 9(2579-2605):85, 2008. 5, 6

[42] 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. 4

[43] Q. Wang and K. Chen. Zero-shot visual recognition via bidirectional latent embedding. *arXiv preprint arXiv:1607.02104*, 2016. 3, 4, 6

[44] Y. Xian, Z. Akata, and B. Schiele. Zero-shot learning – the Good, the Bad and the Ugly. In *CVPR*, 2017. 1, 4, 5

[45] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016. 1, 4, 5, 6

[46] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *ECCV*, 2016. 8

[47] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang. Designing category-level attributes for discriminative visual recognition. In *CVPR*, 2013. 1, 3

[48] L. Zhang, T. Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2017. 4, 5

[49] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *ICCV*, 2015. 1, 3, 5

[50] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In *CVPR*, 2016. 1, 3

[51] X. Zhu, D. Anguelov, and D. Ramanan. Capturing long-tail distributions of object subcategories. In *CVPR*, 2014. 1