This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Zero-Shot Learning - The Good, the Bad and the Ugly

Yongqin Xian[1] Bernt Schiele[1] Zeynep Akata[1,2]

[1]Max Planck Institute for Informatics
Saarland Informatics Campus

[2]Amsterdam Machine Learning Lab
University of Amsterdam

Abstract

Due to the importance of zero-shot learning, the number of proposed approaches has increased steadily recently. We argue that it is time to take a step back and to analyze the status quo of the area. The purpose of this paper is threefold. First, given the fact that there is no agreed upon zero-shot learning benchmark, we first define a new benchmark by unifying both the evaluation protocols and data splits. This is an important contribution as published results are often not comparable and sometimes even flawed due to, e.g. pre-training on zero-shot test classes. Second, we compare and analyze a significant number of the state-of-the-art methods in depth, both in the classic zero-shot setting but also in the more realistic generalized zero-shot setting. Finally, we discuss limitations of the current status of the area which can be taken as a basis for advancing it.

1. Introduction

Zero-shot learning aims to recognize objects whose instances may not have been seen during training [17, 22, 23, 30, 40]. The number of new zero-shot learning methods proposed every year has been increasing rapidly, i.e. the good aspects as our title suggests. Although each new method has been shown to make progress over the previous one, it is difficult to quantify this progress without an established evaluation protocol, i.e. the bad aspects. In fact, the quest for improving numbers has lead to even flawed evaluation protocols, i.e. the ugly aspects. Therefore, in this work, we propose to extensively evaluate a significant number of recent zero-shot learning methods in depth on several small to large-scale datasets using the same evaluation protocol both in zero-shot, i.e. training and test classes are disjoint, and the more realistic generalized zero-shot learning settings, i.e. training classes are present at test time.

We benchmark and systematically evaluate zero-shot learning w.r.t. three aspects, i.e. methods, datasets and evaluation protocol. The crux of the matter for all zero-shot learning methods is to associate observed and non observed classes through some form of auxiliary information which encodes visually distinguishing properties of objects. Different flavors of zero-shot learning methods that we evaluate in this work are linear [11, 2, 4, 32] and nonlinear [39, 34] compatibility learning frameworks whereas an orthogonal direction is learning independent attribute [22] classifiers and finally others [42, 7, 26] propose a hybrid model between independent classifier learning and compatibility learning frameworks.

We thoroughly evaluate the second aspect of zero-shot learning, by using multiple splits of several small to large-scale datasets [28, 38, 22, 10, 9]. We emphasize that it is hard to obtain labeled training data for fine-grained classes of rare objects recognizing which requires expert opinion. Therefore, we argue that zero-shot learning methods should be evaluated mainly on least populated or rare classes.

We propose a unified evaluation protocol to address the third aspect of zero-shot learning which is arguably the most important one. We emphasize the necessity of tuning hyperparameters of the methods on a validation class split that is disjoint from training classes as improving zero-shot learning performance via tuning parameters on test classes violates the zero-shot assumption. We argue that per-class averaged top-1 accuracy is an important evaluation metric when the dataset is not well balanced with respect to the number of images per class. We point out that extracting image features via a pre-trained deep neural network (DNN) on a large dataset that contains zero-shot test classes also violates the zero-shot learning idea as image feature extraction is a part of the training procedure. Moreover, we argue that demonstrating zero-shot performance on small-scale and coarse grained datasets, i.e. aPY [10] is not conclusive. We recommend to abstract away from the restricted nature of zero-shot evaluation and make the task more practical by including training classes in the search space, i.e. generalized zero-shot learning setting. Therefore, we argue that our work plays an important role in advancing the zero-shot learning field by analyzing the good and bad aspects of the zero-shot learning task as well as proposing ways to eliminate the ugly ones.

2. Related Work

We review related work on zero-shot and generalized zero-shot learning, we present previous evaluations on the same task and describe the unique aspects of our work.

Zero-Shot Learning. In zero-shot learning setting test and training class sets are disjoint [17, 22, 23, 30, 40] which can be tackled by solving related sub-problems, e.g. learning intermediate attribute classifiers [22, 30, 31] and learning a mixture of seen class proportions [42, 43, 26, 7], or by a direct approach, e.g. compatibility learning frameworks [3, 4, 11, 15, 27, 32, 34, 39, 32, 12, 29, 1, 6, 24, 13, 21]. Among these methods, in our evaluation we choose to use DAP [22] for being one of the most fundamental methods in zero-shot learning research; CONSE [26] for being one of the most widely used representatives of learning a mixture of class proportions; SSE [42] for being a recent method with a public implementation; SJE [4], ALE [3], DEVISE [11] for being recent compatibility learning methods with similar loss functions; ESZSL [32] for adding a regularization term to unregularized compatibility learning methods; [39] and CMT [34] proposing non-linear extensions to bilinear compatibility learning framework and finally SYNC [7] for reporting the state-of-the-art on several benchmark datasets.

Generalized Zero-shot Learning. This setting [33] generalizes the zero-shot learning task to the case with both seen and unseen classes at test time. [19] argues that although ImageNet classification challenge performance has reached beyond human performance, we do not observe similar behavior of the methods that compete at the detection challenge which involves rejecting unknown objects while detecting the position and label of a known object. [11] uses label embeddings to operate on the generalized zero-shot learning setting whereas [41] proposes to learn latent representations for images and classes through coupled linear regression of factorized joint embeddings. On the other hand, [5] introduces a new model layer to the deep net which estimates the probability of an input being from an unknown class and [34] proposes a novelty detection mechanism. We evaluate [34] and [11] for being the most widely used.

Previous Evaluations of Zero-Shot Learning. In the literature some zero-shot vs generalized zero-shot learning evaluation works exist [30, 8]. Among these, [30] proposes a model to learn the similarity between images and semantic embeddings on the ImageNet 1K by using 800 classes for training and 200 for test. [8] provides a comparison between five methods evaluated on three datasets including ImageNet with three standard splits and proposes a metric to evaluate generalized zero-shot learning performance.

Our work. We evaluate ten zero-shot learning methods on five datasets with several splits both for zero-shot and generalized zero-shot learning settings, provide statistical significancy and robustness tests, and present other valuable insights that emerge from our benchmark. In this sense, ours is a more extensive evaluation compared to prior work.

3. Evaluated Methods

We start by formalizing the zero-shot learning task and then we describe the zero-shot learning methods that we evaluate in this work. Given a training set $\mathcal{S} = \{(x_n, y_n), n = 1...N\}$, with $y_n \in \mathcal{Y}^{tr}$ belonging to training classes, the task is to learn $f : \mathcal{X} \rightarrow \mathcal{Y}$ by minimizing the regularized empirical risk:

$$\frac{1}{N}\sum_{n=1}^{N} L(y_n, f(x_n; W)) + \Omega(W) \tag{1}$$

with $L(.)$ being the loss function and $\Omega(.)$ being the regularization term. Here, the mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ from input to output embeddings is defined as:

$$f(x; W) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, F(x, y; W) \tag{2}$$

At test time, in zero-shot learning setting, the aim is to assign a test image to an unseen class label, i.e. $\mathcal{Y}^{ts} \subset \mathcal{Y}$ and in generalized zero-shot learning setting, the test image can be assigned either to seen or unseen classes, i.e. $\mathcal{Y}^{tr+ts} \subset \mathcal{Y}$ with the highest compatibility score.

3.1. Learning Linear Compatibility

Attribute Label Embedding (ALE) [3], Deep Visual Semantic Embedding (DEVISE) [11] and Structured Joint Embedding (SJE) [4] use bi-linear compatibility function to associate visual and auxiliary information:

$$F(x, y; W) = \theta(x)^T W \phi(y) \tag{3}$$

where $\theta(x)$ and $\phi(y)$, i.e. image and class embeddings, both of which are given. $F(.)$ is parameterized by the mapping W, to be learned. Embarassingly Simple Zero Shot Learning (ESZSL) [32] adds a regularization term to this objective. In the following, we provide a unified formulation of these four zero-shot learning methods.

DEVISE [11] uses pairwise ranking objective that is inspired from unregularized ranking SVM [20]:

$$\sum_{y \in \mathcal{Y}^{tr}} [\Delta(y_n, y) + F(x_n, y; W) - F(x_n, y_n; W)]_+ \tag{4}$$

ALE [3] uses weighted approximate ranking objective [37]:

$$\sum_{y \in \mathcal{Y}^{tr}} \frac{l_{r_{\Delta(x_n, y_n)}}}{r_{\Delta(x_n, y_n)}} [\Delta(y_n, y) + F(x_n, y; W) - F(x_n, y_n; W)]_+ \tag{5}$$

where $l_k = \sum_{i=1}^{k} \alpha_i$ and $r_{\Delta(x_n,y_n)}$ is defined as:

$$\sum_{y \in \mathcal{Y}^{tr}} \mathbb{1}(F(x_n, y; W) + \Delta(y_n, y) \geq F(x_n, y_n; W)) \quad (6)$$

Following the heuristic in [18], [3] selects $\alpha_i = 1/i$ which puts high emphasis on the top of the rank list.

SJE [4] gives full weight to the top of the ranked list and is inspired from the structured SVM [36]:

$$[\max_{y \in \mathcal{Y}^{tr}} (\Delta(y_n, y) + F(x_n, y; W)) - F(x_n, y_n; W)]_+ \quad (7)$$

ESZSL [32] adds the following regularization term to the unregularized risk minimization formulation:

$$\gamma \|W\phi(y)\|_{Fro}^2 + \lambda \|\theta(x)^T W\|_{Fro}^2 + \beta \|W\|_{Fro}^2 \quad (8)$$

where γ, λ, β are parameters of this regularizer.

3.2. Learning Nonlinear Compatibility

Latent Embeddings (LATEM) [39] and Cross Modal Transfer (CMT) [34] encode an additional non-linearity in compatibility learning framework.

LATEM [39] constructs a piece-wise linear compatibility:

$$F(x, y; W_i) = \max_{1 \leq i \leq K} \theta(x)^T W_i \phi(y) \quad (9)$$

where every W_i models a different visual characteristic of the data and the selection of which matrix to use to do the mapping is a latent variable. LATEM uses the ranking loss formulated in Equation 4.

CMT [34] first maps images into a semantic space of words, i.e. class names, where a neural network with $\tanh$ nonlinearity learns the mapping:

$$\sum_{y \in \mathcal{Y}^{tr}} \sum_{x \in \mathcal{X}_y} \|\phi(y) - W_1 \tanh(W_2 . \theta(x)\| \quad (10)$$

where (W_1, W_2) are weights of the two layer neural network. This is followed by a novelty detection mechanism that assigns images to unseen or seen classes. The novelty is detected either via thresholds learned using the embedded images of the seen classes or the outlier probabilities are obtained in an unsupervised way.

3.3. Learning Intermediate Attribute Classifiers

Although Direct Attribute Prediction (DAP) [22] has been shown to perform poorly compared to compatibility learning frameworks [3], we include it to our evaluation for being historically one of the most widely used methods in the literature.

DAP [22] learns probabilistic attribute classifiers and makes a class prediction by combining scores of the learned attribute classifiers. A novel image is assigned to one of the unknown classes using:

$$f(x) = \operatorname*{argmax}_c \prod_{m=1}^{M} \frac{p(a_m^c|x)}{p(a_m^c)}. \quad (11)$$

with M being the total number of attributes. We train a one-vs-rest SVM with log loss that gives probability scores of attributes with respect to training classes.

3.4. Hybrid Models

Semantic Similarity Embedding (SSE) [42], Convex Combination of Semantic Embeddings (CONSE) [26] and Synthesized Classifiers (SYNC) [7] express images and semantic class embeddings as a mixture of seen class proportions, hence we group them as hybrid models.

SSE [42] leverages similar class relationships both in image and semantic embedding space. An image is labeled with:

$$\operatorname*{argmax}_{u \in \mathcal{U}} \pi(\theta(x))^T \psi(\phi(y_u)) \quad (12)$$

where π, ψ are mappings of class and image embeddings into a common space. Specifically, ψ is learned by sparse coding and π is by class dependent transformation.

CONSE [26] learns the probability of a training image belonging to a training class:

$$f(x, t) = \operatorname*{argmax}_{y \in \mathcal{Y}^{tr}} p_{tr}(y|x) \quad (13)$$

where y denotes the most likely training label (t=1) for image x. Combination of semantic embeddings (s) is used to assign an unknown image to an unseen class:

$$\frac{1}{Z} \sum_{i=1}^{T} p_{tr}(f(x,t)|x).s(f(x,t)) \quad (14)$$

where $Z = \sum_{i=1}^{T} p_{tr}(f(x,t)|x)$, $f(x,t)$ denotes the t^{th} most likely label for image x and T controls the maximum number of semantic embedding vectors.

SYNC [7] learns a mapping between the semantic class embedding space and a model space. In the model space, training classes and a set of phantom classes form a weighted bipartite graph. The objective is to minimize distortion error:

$$\min_{w_c, v_r} \|w_c - \sum_{r=1}^{R} s_{cr} v_r\|_2^2. \quad (15)$$

Semantic and model spaces are aligned by embedding real (w_c) and phantom classes (v_r) in the weighted graph (s_{cr}).

				Number of Classes			Number of Images								
								At Training Time				At Evaluation Time			
								SS		PS		SS		PS	
Dataset	**Size**	**Detail**	**Att**	$\mathcal{Y}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	**Total**	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$
SUN [28]	medium	fine	102	717	580 + 65	72	14K	12900	0	10320	0	0	1440	2580	1440
CUB [38]	medium	fine	312	200	100 + 50	50	11K	8855	0	7057	0	0	2933	1764	2967
AWA [22]	medium	coarse	85	50	27 + 13	10	30K	24295	0	19832	0	0	6180	4958	5685
aPY [10]	small	coarse	64	32	15 + 5	12	15K	12695	0	5932	0	0	2644	1483	7924

Table 1: Statistics for attribute datasets: SUN [28], CUB [38], AWA [22], aPY [10] in terms of size of the datasets, fine-grained or coarse-grained, number of attributes, number of classes in training + validation ($\mathcal{Y}^{tr}$) and test classes ($\mathcal{Y}^{ts}$), number of images at training and test time for standard split (SS) and our proposed splits (PS).

4. Datasets and Evaluation Protocol

In this section, we provide several components of previously used and our proposed zero-shot and generalized zero-shot learning evaluation protocols, e.g. datasets, image and class encodings and the evaluation protocol.

4.1. Dataset Statistics

Among the most widely used datasets for zero-shot learning, we select two coarse-grained, one small and one medium-scale, and two fine-grained, both medium-scale, datasets with attributes and one large-scale dataset without. Here, we consider between $10K$ and $1M$ images, and, between 100 and $1K$ classes as medium-scale.

Attribute Datasets. Statistics of the attribute datasets are presented in Table 1. Attribute Pascal and Yahoo (aPY) [10] is a small-scale coarse-grained dataset with 64 attributes. Among the total number of 32 classes, 20 Pascal classes are used for training (we randomly select 5 for validation) and 12 Yahoo classes are used for testing. Animals with Attributes (AWA) [22] is a coarse-grained dataset that is medium-scale in terms of the number of images, i.e. $30,475$ and small-scale in terms of number of classes, i.e. 50. [22] introduces a standard zero-shot split with 40 classes for training (we randomly select 13 for validation) and 10 for testing. AWA has 85 attributes. Caltech-UCSD-Birds 200-2011 (CUB) [38] is a fine-grained and medium scale dataset with respect to both number of images and number of classes, i.e. $11,788$ images from 200 different types of birds annotated with 312 attributes. [3] introduces the first zero-shot split of CUB with 150 training (50 validation classes) and 50 test classes. SUN [28] is a fine-grained and medium-scale dataset with respect to both number of images and number of classes, i.e. SUN contains 14340 images coming from 717 types of scenes annotated with 102 attributes. Following [22] we use 645 classes of SUN for training (we randomly select 65 for val) and 72 for testing.

Large-Scale ImageNet. We also evaluate the performance of methods on the large scale ImageNet [9]. Among the total of 21K classes, 1K classes are used for training (we use 200 classes for validation) and the test split is either all the remaining 21K classes or a subset of it, e.g. we determine these subsets based on the hierarchical distance between classes and the population of classes.

4.2. Proposed Evaluation Protocol

We present our proposed unified protocol for image and class embeddings, dataset splits and evaluation criteria.

Image and Class Embedding. We extract image features from the entire image for SUN, CUB, AWA and ImageNet, with no image pre-processing. For aPY, as proposed in [10], we extract image features from bounding boxes. Our image embeddings are 2048-dim top-layer pooling units of the 101-layered ResNet [16] as we found that it performs better than $1,024$-dim top-layer pooling units of GoogleNet [35]. ResNet is pre-trained on ImageNet 1K and not fine-tuned. In addition to ResNet features, we evaluate methods with their published image features. As class embeddings, for aPY, AWA, CUB and SUN, we use per-class attributes. For ImageNet we use Word2Vec [25] provided by [7] as it does not contain attribute annotation for all the classes.

Dataset Splits. Zero-shot learning assumes disjoint training and test classes with the presence of all the images of training classes and the absence of any image from test classes during training. On the other hand, as deep neural network (DNN) training for image feature extraction is actually a part of model training, the dataset used to train DNNs, e.g. ImageNet, should not include any of the test classes. However, we notice from the standard splits (SS) of aPY and AWA datasets that 7 aPY test classes out of 12 (monkey, wolf, zebra, mug, building, bag, carriage), 6 AWA test classes out of 10 (chimpanzee, giant panda, leopard, persian cat, pig, hippopotamus), are among the 1K classes of ImageNet, i.e. are used to pre-train ResNet. On the other hand, the mostly widely used splits, i.e. we term them as standard splits (SS), for SUN from [22] and CUB from [2] shows us that 1 CUB test class out of 50 (Indigo Bunting),

and 6 SUN test classes out of 72 (restaurant, supermarket, planetarium, tent, market, bridge), are also among the 1K classes of ImageNet. We noticed that the accuracy for all methods on those overlapping test classes are higher than others. Therefore, we propose new dataset splits, i.e. proposed splits (PS), insuring that none of the test classes appear in ImageNet 1K, i.e. used to train the ResNet model. We present the differences between the standard splits (SS) and the proposed splits (PS) in Table 1. While in SS and PS no image from test classes is present at training time, at test time SS does not include any images from training classes however our PS does. We designed the PS this way as evaluating accuracy on both training and test classes is crucial to show the generalization of methods.

ImageNet with thousands of classes provides possibilities of constructing several zero-shot evaluation splits. Following [7], our first two standard splits consider all the classes that are 2-hops and 3-hops away from the original 1K classes according to the ImageNet label hierarchy, corresponding to 1509 and 7678 classes. This split measures the generalization ability of the models with respect to the hierarchical and semantic similarity between classes. Our proposed split considers 500, 1K and 5K most populated classes among the remaining 21K classes of ImageNet with ≈ 1756, ≈ 1624 and ≈ 1335 images per class on average. Similarly, we consider 500, 1K and 5K least-populated classes in ImageNet which correspond to most fine-grained subsets of ImageNet with ≈ 1, ≈ 3 and ≈ 51 images per class on average. Our final split considers all the remaining $\approx$ 20K classes of ImageNet with at least 1 image per-class, ≈ 631 images per class on average.

Evaluation Criteria. Single label image classification accuracy has been measured with Top-1 accuracy, i.e. the prediction is accurate when the predicted class is the correct one. If the accuracy is averaged for all images, high performance on densely populated classes is encouraged. However, we are interested in having high performance also on sparsely populated classes. Therefore, we average the correct predictions independently for each class before dividing their cumulative sum w.r.t the number of classes, i.e. we measure average per-class top-1 accuracy.

In generalized zero-shot learning setting, the search space at evaluation time is not restricted to only test classes, but includes also the training classes, hence this setting is more practical. As with our proposed split at test time we have access to some images from training classes, after having computed the average per-class top-1 accuracy on training and test classes, we compute the harmonic mean of training and test accuracies:

$$H = 2 * (acc_{\mathcal{Y}^{tr}} * acc_{\mathcal{Y}^{ts}})/(acc_{\mathcal{Y}^{tr}} + acc_{\mathcal{Y}^{ts}}) \qquad (16)$$

where $acc_{\mathcal{Y}^{tr}}$ and $acc_{\mathcal{Y}^{ts}}$ represent the accuracy of images from seen ($\mathcal{Y}^{tr}$), and images from unseen ($\mathcal{Y}^{ts}$) classes respectively. We choose harmonic mean as our evaluation criteria and not arithmetic mean because in arithmetic mean if the seen class accuracy is much higher, it effects the overall results significantly. Instead, our aim is high accuracy on both seen and unseen classes.

	SUN		AWA	
Model	**R**	**O**	**R**	**O**
DAP [22]	22.1	22.2	41.4	41.4
SSE [42]	83.0	82.5	64.9	76.3
LATEM [39]	–	–	71.2	71.9
SJE [4]	–	–	67.2	66.7
ESZSL [32]	64.3	65.8	48.0	49.3
SYNC [7]	62.8	62.8	69.7	69.7

Table 2: Reproducing zero-shot results: O = Original results published in the paper, R = Reproduced using provided image features and code. We measure top-1 accuracy in %.

5. Experiments

We first provide zero-shot learning results on attribute datasets SUN, CUB, AWA and aPY and then on the large-scale ImageNet dataset. Finally, we present results for the generalized zero-shot learning setting.

5.1. Zero-Shot Learning Results

On attribute datasets, i.e. SUN, CUB, AWA and aPY, we first reproduce the results of each method using their evaluation protocol, then provide a unified evaluation protocol using the same train/val/test class splits, followed by our proposed train/val/test class splits. We also evaluate the robustness of the methods to parameter tuning and visualize the ranking of different methods. Finally, we evaluate the methods on the large-scale ImageNet dataset.

Reproducing Results. For sanity-check, we re-evaluate methods [22, 42, 39, 4, 32, 7] [1] using provided features and code. We chose SUN and AWA as two representative of fine-grained and non-fine-grained datasets having been widely used in the literature. We observe from the results in Table 2 that our reproduced results and the reported results of DAP and SYNC are identical to the reported number in their original publications. For LATEM, we obtain slightly different results which can be explained by the non-convexity and thus the sensibility to initialization. Similarly for SJE random sampling in SGD might lead to slightly different results. ESZSL has some variance because its algorithm randomly picks a validation set during each run, which leads to different hyperparameters. Notable observations on SSE [42] results are as follows. The published code has hard-coded hyperparameters operational on aPY,

[1] [34] has public code available, but is not evaluated on SUN or AWA.

	SUN		CUB		AWA		aPY	
Method	**SS**	**PS**	**SS**	**PS**	**SS**	**PS**	**SS**	**PS**
DAP [22]	38.9	39.9	37.5	40.0	57.1	44.1	35.2	33.8
CONSE [26]	44.2	38.8	36.7	34.3	63.6	45.6	25.9	26.9
CMT [34]	41.9	39.9	37.3	34.6	58.9	39.5	26.9	28.0
SSE [42]	54.5	51.5	43.7	43.9	68.8	60.1	31.1	34.0
LATEM [39]	56.9	55.3	49.4	49.3	74.8	55.1	34.5	35.2
ALE [3]	59.1	**58.1**	53.2	54.9	**78.6**	59.9	30.9	39.7
DEVISE [11]	57.5	56.5	53.2	52.0	72.9	54.2	35.4	**39.8**
SJE [4]	57.1	53.7	**55.3**	53.9	76.7	**65.6**	32.0	32.9
ESZSL [32]	57.3	54.5	55.1	53.9	74.7	58.2	34.4	38.3
SYNC [7]	**59.1**	56.3	54.1	**55.6**	72.2	54.0	**39.7**	23.9

Table 3: Zero-shot on SS = Standard Split, PS = Proposed Split using ResNet features (top-1 accuracy in %).

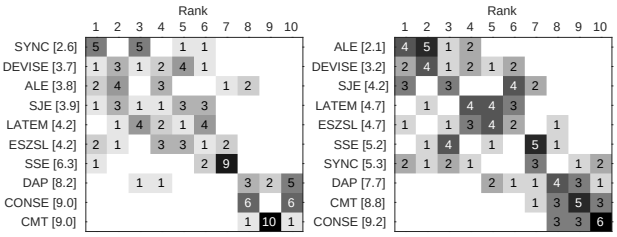

Figure 1: Ranking 10 models by setting parameters on three validation splits on the standard (SS, left) and proposed (PS, right) setting. Element (i, j) indicates number of times model i ranks at jth over all 4×3 observations. Models are ordered by their mean rank (displayed in brackets).

i.e. number of iterations, number of data points to train SVM, and one regularizer parameter γ which lead to inferior results than the ones reported here, therefore we set these parameters on validation sets. On SUN, SSE uses 10 classes (instead of 72) and our results with validated parameters got an improvement of 0.5% that may be due to random sampling of training images. On AWA, our reproduced result being 64.9% is significantly lower than the reported result (76.3%). However, we could not reach the reported result even by tuning parameters on the test set, i.e. we obtain 73.8% in this case.

Reproduced Results vs Standard Split (SS). In addition to [22, 42, 39, 4, 32, 7, 34], we re-implement [26, 11, 3] based on the original publications. We use train, validation, test splits as provided in Table 1 and report results on Table 3 with deep ResNet features. DAP [22] uses hand-crafted image features and thus reproduced results with those features are significantly lower than the results with deep features (22.1% vs 38.9%). When we investigate the results in detail, we noticed two irregularities with reported results on SUN. First, SSE [42] and ESZSL [32] report results on a test split with 10 classes whereas the standard split of SUN contains 72 test classes (74.5% vs 54.5% with SSE [42] and 64.3% vs 57.3% with ESZSL [32]). Second, after careful examination and correspondence with the authors of SYNC [7], we detected that SUN features were extracted with a MIT Places [44] pre-trained model. As MIT Places dataset intersects with both training and test classes of SUN dataset, it is expected to lead to significantly better results than ImageNet pre-trained model (62.8% vs 59.1%).

Results on Standard (SS) and Proposed Splits (PS). We propose new dataset splits (see details in section 4) insuring that test classes do not belong to the ImageNet1K that is used to pre-train ResNet. We compare these results (PS) with previously published standard split (SS) results in Table 3. Our first observation is that the results on PS is significantly lower than SS for AWA. This is expected as most of the test classes in SS is included in ImageNet 1K. On the other hand, for fine-grained datasets CUB and SUN, the results are not significantly effected. Our second observation regarding the method ranking is as follows. On SS, SYNC [7] is the best performing method on SUN (59.1%) and aPY (39.7%) datasets whereas SJE [4] performs the best on CUB (55.3%) and ALE [3] performs the best on AWA (78.6%) dataset. On PS, ALE [3] performs the best on SUN (58.1%), SYNC [7] on CUB (55.6%), SJE [4] on AWA (65.6%) and DEVISE [11] on aPY (39.8%). Note that ALE, SJE and DEVISE all use max-margin bi-linear compatibility learning framework.

Robustness. We evaluate robustness of 10 methods to parameters by setting them on 3 different validation splits while keeping the test split intact. We report results on SS (Figure 2, top) and PS (Figure 2, bottom). On SUN and CUB, the results are stable across methods and across splits. This is expected as these datasets have balanced number of images across classes and due to their fine-grained nature, the validation splits are similar. On the other hand, AWA and aPY being small and coarse-grained datasets have several issues. First, many of the test classes on AWA and aPY are included in ImageNet1K. Second, they are not well balanced, i.e. different validation class splits contain significantly different number of images. Third, the class embeddings are far from each other, i.e. objects are semantically different, therefore different validation splits learn a different mapping between images and classes.

Visualizing Method Ranking. We rank the 10 methods based on their per-class top-1 accuracy using the non-parametric Friedman test [14], which does not assume a distribution on performance but rather uses algorithm ranking. Each entry of the rank matrix on Figure 1 indicates the number of times the method is ranked at the first to tenth rank. We then compute the mean rank of each method and order them based on that. Our general observation is that the highest ranked method on the standard split (SS) is SYNC while on the proposed split (PS) it is ALE. These results

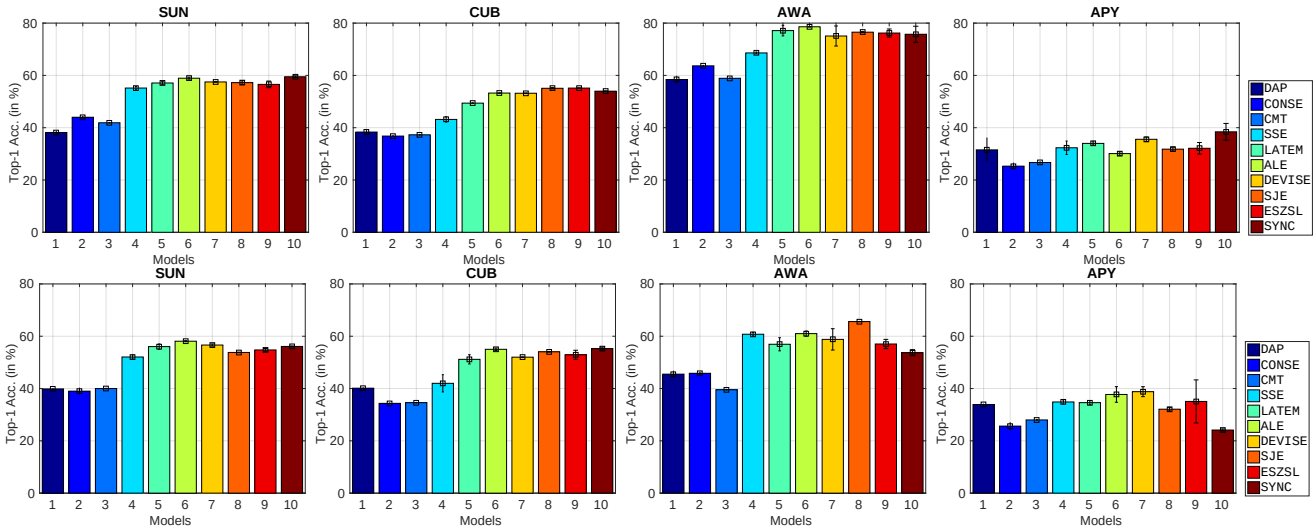

Figure 2: Robustness of 10 methods evaluated on SUN, CUB, AWA, aPY using 3 validation set splits (results are on the same test split). Top: original split, Bottom: proposed split (Image embeddings = ResNet). We measure top-1 accuracy in %.

	Hierarchy		Most Populated			Least Populated			All
Method	**2 H**	**3 H**	**500**	**1K**	**5K**	**500**	**1K**	**5K**	**20K**
CONSE [26]	7.63	2.18	12.33	8.31	3.22	3.53	2.69	1.05	0.95
CMT [34]	2.88	0.67	5.10	3.04	1.04	1.87	1.08	0.33	0.29
LATEM [39]	5.45	1.32	10.81	6.63	1.90	4.53	2.74	0.76	0.50
ALE [3]	5.38	1.32	10.40	6.77	2.00	4.27	2.85	0.79	0.50
DEVISE [11]	5.25	1.29	10.36	6.68	1.94	4.23	2.86	0.78	0.49
SJE [4]	5.31	1.33	9.88	6.53	1.99	4.93	2.93	0.78	0.52
ESZSL [32]	6.35	1.51	11.91	7.69	2.34	4.50	3.23	0.94	0.62
SYNC [7]	**9.26**	**2.29**	**15.83**	**10.75**	**3.42**	**5.83**	**3.52**	**1.26**	**0.96**

Table 4: ImageNet with different splits: 2/3 H = classes with 2/3 hops away from 1K $\mathcal{Y}^{tr}$, 500/1K/5K most populated classes, 500/1K/5K least populated classes, All=20K categories of ImageNet. We measure top-1 accuracy in %.

indicate the importance of choosing zero-shot splits carefully. On the proposed split, the three highest ranked methods are compatibility learning methods, i.e. ALE, DEVISE and SJE whereas the three lowest ranked methods are attribute classifier learning or hybrid methods, i.e. DAP, CMT and CONSE. Therefore, max-margin compatibility learning methods lead to consistently better results in the zero-shot learning task compared to learning independent classifiers.

Results on ImageNet. ImageNet scales the methods to a truly large-scale setting, thus these experiments provide further insights on how to tackle the zero-shot learning problem from the practical point of view. Here, we evaluate 8 methods. We exclude DAP as attributes are not available for all ImageNet classes and SSE due to scalability issues of the public implementation of the method. Table 4 shows that the best performing method is SYNC [7] which may indicate that it performs well in large-scale setting or it can learn under uncertainty due to usage of Word2Vec instead of attributes. Another possibility is Word2Vec may be tuned for SYNC as it is provided by the same authors however making a strong claim requires a full evaluation on class embeddings which is out of the scope of this paper. Our general observation from all the methods is that in the most populated classes, the results are higher than the least populated classes which indicates that fine-grained subsets are more difficult. We consistently observe a large drop in accuracy between 1K and 5K most populated classes which is expected as 5K contains $\approx$ 6.6M images, making the problem much more difficult than 1K ($\approx$ 1624 images). On the other hand, All 20K results are poor for all methods which indicates the difficulty of this problem where there is a large room for improvement.

5.2. Generalized Zero-Shot Learning Results

In real world applications, image classification systems do not have access to whether a novel image belongs to a seen or unseen class in advance. Hence, generalized zero-shot learning is interesting from a practical point of view. Here, we use same models trained on zero-shot learning setting on our proposed splits (PS). We evaluate performance on both $\mathcal{Y}^{tr}$ and $\mathcal{Y}^{ts}$, i.e. using held-out images from $\mathcal{Y}^{ts}$.

As shown in Table 5, generalized zero-shot results are significantly lower than zero-shot results as training classes are included in the search space. Another interesting observation is that compatibility learning frameworks, e.g. ALE, DEVISE, SJE, perform well on test classes. However, methods that learn independent attribute or object classifiers, e.g. DAP and CONSE, perform well on training classes. Due to this discrepancy, we evaluate the harmonic mean which takes a weighted average of training and test

Method	SUN			CUB			AWA			aPY		
	ts	tr	H	ts	tr	H	ts	tr	H	ts	tr	H
DAP [22]	4.2	25.1	7.2	1.7	67.9	3.3	0.0	**88.7**	0.0	4.8	78.3	9.0
CONSE [26]	6.8	**39.9**	11.6	1.6	**72.2**	3.1	0.4	88.6	0.8	0.0	**91.2**	0.0
CMT [34]	8.1	21.8	11.8	7.2	49.8	12.6	0.9	87.6	1.8	1.4	85.2	2.8
CMT* [34]	8.7	28.0	13.3	4.7	60.1	8.7	8.4	86.9	15.3	**10.9**	74.2	**19.0**
SSE [42]	2.1	36.4	4.0	8.5	46.9	14.4	7.0	80.5	12.9	0.2	78.9	0.4
LATEM [39]	14.7	28.8	19.5	15.2	57.3	24.0	7.3	71.7	13.3	0.1	73.0	0.2
ALE [3]	**21.8**	33.1	**26.3**	23.7	62.8	**34.4**	**16.8**	76.1	**27.5**	4.6	73.7	8.7
DEVISE [11]	16.9	27.4	20.9	**23.8**	53.0	32.8	13.4	68.7	22.4	4.9	76.9	9.2
SJE [4]	14.7	30.5	19.8	23.5	59.2	33.6	11.3	74.6	19.6	3.7	55.7	6.9
ESZSL [32]	11.0	27.9	15.8	12.6	63.8	21.0	6.6	75.6	12.1	2.4	70.1	4.6
SYNC [7]	7.9	43.3	13.4	11.5	70.9	19.8	8.9	87.3	16.2	7.4	66.3	13.3

Table 5: Generalized Zero-Shot Learning on Proposed Split (PS) measuring ts = Top-1 accuracy on $\mathcal{Y}^{ts}$, tr=Top-1 accuracy on $\mathcal{Y}^{tr+ts}$), H = harmonic mean (CMT*: CMT with novelty detection). We measure top-1 accuracy in %.

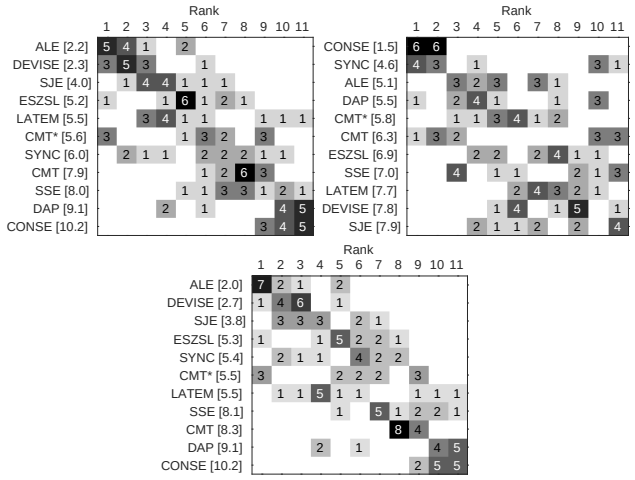

Figure 3: Ranking 11 models on the proposed split (PS) in generalized zero-shot learning setting. Top-Left: on unseen cla sses (ts) accuracy, Top-Right: on seen classes (tr) accuracy, Bottom: on Harmonic mean (H).

class accuracy. H measure ranks ALE as the best performing method on SUN, CUB and AWA datasets whereas on aPY dataset CMT* performs the best. Note that CMT* has an integrated novelty detection phase for which the method receives another supervision signal determining if the image belongs to a train or a test class. As a summary, generalized zero-shot learning setting provides one more level of detail on the performance of zero-shot learning methods. Our take-home message is that the accuracy of training classes is as important as the accuracy of test classes in real world scenarios. Therefore, methods should be designed in a way that they are able to predict labels well in train and test classes.

Visualizing Method Ranking. Similar to the analysis in the previous section, we rank the 11 methods based on per-class top-1 accuracy on train classes, test classes and based on Harmonic mean of the two. Looking at the rank matrix obtained by evaluating on test classes, i.e. Figure 3 top left, highest ranked 5 methods are the same as in Figure 1, i.e. ALE, DEVISE, SJE, LATEM, ESZSL while overall the absolute numbers are lower. Looking at the rank matrix obtained by evaluating the harmonic mean, i.e. Figure 3 bottom, the highest ranked 3 methods are the same as in Figure 1, i.e. ALE, DEVISE, SJE. Looking at the rank matrix obtained by evaluating on train classes, i.e. Figure 3 top right, our observations are different from Figure 1. ALE is ranked the 3rd but other highest ranked methods are at the bottom of this rank list. These results clearly suggest that we should not only optimize for test class accuracy but also for train class accuracy when evaluating zero-shot learning. Our final observation from Figure 3 is that CMT* is better than CMT in all cases which supports the argument that a simple novelty detection scheme helps to improve results.

6. Conclusion

In this work, we evaluated a significant number of state-of-the-art zero-shot learning methods on several datasets within a unified evaluation protocol both in zero-shot and generalized zero-shot settings. Our evaluation showed that compatibility learning frameworks have an edge over learning independent object or attribute classifiers and also over hybrid models. We discovered that some standard zero-shot splits may treat feature learning disjoint from the training stage and accordingly proposed new dataset splits. Moreover, disjoint training and validation class split is a necessary component of parameter tuning in zero-shot learning setting. Including training classes in the search space while evaluating the methods, i.e. generalized zero-shot learning, provides an interesting playground for future research. In summary, our work extensively evaluated the good and bad aspects of zero-shot learning while sanitizing the ugly ones.

References

[1] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele. Multi-cue zero-shot learning with strong supervision. In *CVPR*, 2016.

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

[3] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *TPAMI*, 2016.

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

[5] A. Bendale and T. E. Boult. Towards open set deep networks. In *CVPR*, 2016.

[6] M. Bucher, S. Herbin, and F. Jurie. Improving semantic embedding consistency by metric learning for zero-shot classiffication. In *ECCV*, 2016.

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

[8] 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.

[9] 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.

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

[11] 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 *NIPS*, 2013.

[12] Y. Fu and L. Sigal. Semi-supervised vocabulary-informed learning. In *CVPR*, 2016.

[13] Z. Fu, T. Xiang, E. Kodirov, and S. Gong. Zero-shot object recognition by semantic manifold distance. In *CVPR*, June 2015.

[14] S. Garcia and F. Herrera. An extension on"statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons. *JLMR*, 9:2677–2694, 2008.

[15] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning (2nd Ed.)*. Springer Series in Statistics. Springer, 2008.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] S. Huang, M. Elhoseiny, A. M. Elgammal, and D. Yang. Learning hypergraph-regularized attribute predictors. In *CVPR*, 2015.

[18] S. B. J. Weston and N. Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. In *ECML*, 2010.

[19] L. Jain, W. Scheirer, and T. Boult. Multi-class open set recognition using probability of inclusion. In *ECCV*, 2014.

[20] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*. ACM, 2002.

[21] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, 2015.

[22] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. In *TPAMI*, 2013.

[23] H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *AAAI*, 2008.

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

[25] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[26] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, 2014.

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

[28] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.

[29] 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 *CVPR*, 2016.

[30] M. Rohrbach, M. Stark, and B.Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*, 2011.

[31] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps here – and why? Semantic relatedness for knowledge transfer. In *CVPR*, 2010.

[32] B. Romera-Paredes and P. H. Torr. An embarrassingly simple approach to zero-shot learning. *ICML*, 2015.

[33] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult. Towards open set recognition. *TPAMI*, 36, 2013.

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

[35] 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.

[36] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 2005.

[37] N. Usunier, D. Buffoni, and P. Gallinari. Ranking with ordered weighted pairwise classification. In *ICML*, 2009.

[38] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, Caltech, 2010.

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

[40] X. Yu and Y. Aloimonos. Attribute-based transfer learning for object categorization with zero or one training example. In *ECCV*, 2010.

[41] H. Zhang, X. Shang, W. Yang, H. Xu, H. Luan, and T.-S. Chua. Online collaborative learning for open-vocabulary visual classifiers. In *CVPR*, 2016.

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

[43] Z. Zhang and V. Saligrama. Zero-shot learning via joint semantic similarity embedding. In *CVPR*, 2016.

[44] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.