An Empirical Study and Analysis of Generalized Zero-Shot Learning for Object Recognition in the Wild

Wei-Lun Chao*[1], Soravit Changpinyo*[1], Boqing Gong[2], and Fei Sha[3]

[1]Dept. of Computer Science, U. of Southern California, United States
[2]Center for Research in Computer Vision, U. of Central Florida, United States
[3]Dept. of Computer Science, U. of California, Los Angeles, United States
{weilunc, schangpi}@usc.edu, bgong@crcv.ucf.edu, feisha@cs.ucla.edu

Abstract. *Zero-shot learning* (ZSL) methods have been studied in the unrealistic setting where test data are assumed to come from unseen classes only. In this paper, we advocate studying the problem of *generalized zero-shot learning* (GZSL) where the test data's class memberships are unconstrained. We show empirically that naively using the classifiers constructed by ZSL approaches does not perform well in the generalized setting. Motivated by this, we propose a simple but effective calibration method that can be used to balance two conflicting forces: recognizing data from seen classes versus those from unseen ones. We develop a performance metric to characterize such a trade-off and examine the utility of this metric in evaluating various ZSL approaches. Our analysis further shows that there is a large gap between the performance of existing approaches and an upper bound established via *idealized* semantic embeddings, suggesting that improving class semantic embeddings is vital to GZSL.

1 Introduction

The availability of large-scale labeled training images is one of the key factors that contribute to recent successes in visual object recognition and classification. It is well-known, however, that object frequencies in natural images follow long-tailed distributions [1,2,3]. For example, some animal or plant species are simply rare by nature — it is uncommon to find alpacas wandering around the streets. Furthermore, brand new categories could just emerge with zero or little labeled images; newly defined visual concepts or products are introduced everyday. In this *real-world* setting, it would be desirable for computer vision systems to be able to recognize instances of those rare classes, while demanding minimum human efforts and labeled examples.

Zero-shot learning (ZSL) has long been believed to hold the key to the above problem of recognition in the wild. ZSL differentiates two types of classes: *seen* and *unseen*, where labeled examples are available for seen classes only. Without

* Equal contribution.

labeled data, models for unseen classes are learned by relating them to seen ones. This is often achieved by embedding both seen and unseen classes into a common semantic space, such as visual attributes [4,5,6] or WORD2VEC representations of the class names [7,8,9]. This common semantic space enables transferring models for the seen classes to those for the unseen ones [10].

The setup for ZSL is that once models for unseen classes are learned, they are judged based on their ability to discriminate among unseen classes, assuming the absence of seen objects during the test phase. Originally proposed in the seminal work of Lampert et al. [4], this setting has almost always been adopted for evaluating ZSL methods [10,11,12,13,14,15,8,16,17,18,19,20,21,22,23,24,25,26,27,28].

But, *does this problem setting truly reflect what recognition in the wild entails*? While the ability to learn novel concepts is by all means a trait that any zero-shot learning systems should possess, it is merely one side of the coin. The other important — yet so far under-studied — trait is the ability to *remember* past experiences, i.e., the *seen* classes.

Why is this trait desirable? Consider how data are distributed in the real world. The seen classes are often more common than the unseen ones; it is therefore unrealistic to assume that we will never encounter them during the test stage. For models generated by ZSL to be truly useful, they should not only accurately discriminate among either seen *or* unseen classes themselves but also accurately discriminate between the seen *and* unseen ones.

Thus, to understand better how existing ZSL approaches will perform in the real world, we advocate evaluating them in the setting of *generalized zero-shot learning* (GZSL), where test data are from both seen and unseen classes and we need to classify them into the joint labeling space of both types of classes. Previous work in this direction is scarce. See related work for more details.

Our contributions include an extensive empirical study of several existing ZSL approaches in this new setting. We show that a straightforward application of classifiers constructed by those approaches performs poorly. In particular, test data from unseen classes are almost always classified as a class from the seen ones. We propose a surprisingly simple yet very effective method called *calibrated stacking* to address this problem. This method is mindful of the two conflicting forces: recognizing data from seen classes and recognizing data from unseen ones. We introduce a new performance metric called Area Under Seen-Unseen accuracy Curve (AUSUC) that can evaluate ZSL approaches on how well they can trade off between the two. We demonstrate the utility of this metric by evaluating several representative ZSL approaches under this metric on three benchmark datasets, including the full ImageNet Fall 2011 release dataset [29] that contains approximately 21,000 unseen categories.

We complement our comparative studies in learning methods by further establishing an upper bound on the performance limit of ZSL. In particular, our idea is to use class-representative visual features as the *idealized* semantic embeddings to construct ZSL classifiers. We show that there is a large gap between existing approaches and this ideal performance limit, suggesting that improving class semantic embeddings is vital to achieve GZSL.

The rest of the paper is organized as follows. Section 2 reviews relevant literature. We define GZSL formally and shed lights on its difficulty in Section 3. In Section 4, we propose a method to remedy the observed issues in the previous section and compare it to related approaches. Experimental results, detailed analysis, and discussions are provided in Section 5, 6, and 7, respectively.

2 Related Work

There has been very little work on generalized zero-shot learning. [8,17,30,31] allow the label space of their classifiers to include seen classes but they only test on the data from the unseen classes. [9] proposes a two-stage approach that first determines whether a test data point is from a seen or unseen class, and then apply the corresponding classifiers. However, their experiments are limited to only 2 or 6 unseen classes. We describe and compare to their methods in Section 4.3, 5, and the Supplementary Material. In the domain of action recognition, [32] investigates the generalized setting with only up to 3 seen classes. [33] and [34] focus on training a zero-shot binary classifier for *each* unseen class (against seen ones) — it is not clear how to distinguish multiple unseen classes from the seen ones. Finally, open set recognition [35,36,37] considers testing on both types of classes, but treating the unseen ones as a single outlier class.

3 Generalized Zero-Shot Learning

In this section, we describe formally the setting of *generalized zero-shot learning*. We then present empirical evidence to illustrate the difficulty of this problem.

3.1 Conventional and Generalized Zero-shot Learning

Suppose we are given the training data $\mathcal{D} = \{(\boldsymbol{x}_n \in \mathbb{R}^{\mathsf{D}}, y_n)\}_{n=1}^{\mathsf{N}}$ with the labels y_n from the label space of *seen* classes $\mathcal{S} = \{1, 2, \cdots, \mathsf{S}\}$. Denote by $\mathcal{U} = \{\mathsf{S}+1, \cdots, \mathsf{S}+\mathsf{U}\}$ the label space of *unseen* classes. We use $\mathcal{T} = \mathcal{S} \cup \mathcal{U}$ to represent the union of the two sets of classes.

In the (conventional) zero-shot learning (ZSL) setting, the main goal is to classify test data into the *unseen* classes, assuming the absence of the seen classes in the test phase. In other words, each test data point is assumed to come from and will be assigned to one of the labels in $\mathcal{U}$.

Existing research on ZSL has been almost entirely focusing on this setting [4,10,11,12,13,14,15,8,16,17,18,19,20,21,22,23,24,25,26,27,28]. However, in real applications, the assumption of encountering data only from the unseen classes is hardly realistic. The seen classes are often the most common objects we see in the real world. Thus, the objective in the conventional ZSL does not truly reflect how the classifiers will perform recognition in the wild.

Motivated by this shortcoming of the conventional ZSL, we advocate studying the more general setting of *generalized zero-shot learning (GZSL)*, where we no longer limit the possible class memberships of test data — each of them belongs to one of the classes in $\mathcal{T}$.

3.2 Classifiers

Without the loss of generality, we assume that for each class $c \in \mathcal{T}$, we have a discriminant scoring function $f_c(\boldsymbol{x})$, from which we would be able to derive the label for $\boldsymbol{x}$. For instance, for an unseen class u, the method of synthesized classifiers [28] defines $f_u(\boldsymbol{x}) = \boldsymbol{w}_u^{\mathrm{T}}\boldsymbol{x}$, where $\boldsymbol{w}_u$ is the model parameter vector for the class u, constructed from its semantic embedding $\boldsymbol{a}_u$ (such as its attribute vector or the word vector associated with the name of the class). In ConSE [17], $f_u(\boldsymbol{x}) = \cos(s(\boldsymbol{x}), \boldsymbol{a}_u)$, where $s(\boldsymbol{x})$ is the predicted embedding of the data sample $\boldsymbol{x}$. In DAP/IAP [38], $f_u(\boldsymbol{x})$ is a probabilistic model of attribute vectors. We assume that similar discriminant functions for seen classes can be constructed in the same manner given their corresponding semantic embeddings.

How to assess an algorithm for GZSL? We define and differentiate the following performance metrics: $A_{\mathcal{U}\to\mathcal{U}}$ the accuracy of classifying test data from $\mathcal{U}$ into $\mathcal{U}$, $A_{\mathcal{S}\to\mathcal{S}}$ the accuracy of classifying test data from $\mathcal{S}$ into $\mathcal{S}$, and finally $A_{\mathcal{S}\to\mathcal{T}}$ and $A_{\mathcal{U}\to\mathcal{T}}$ the accuracies of classifying test data from either seen or unseen classes into the joint labeling space. Note that $A_{\mathcal{U}\to\mathcal{U}}$ is the standard performance metric used for conventional ZSL and $A_{\mathcal{S}\to\mathcal{S}}$ is the standard metric for multi-class classification. Furthermore, note that we do not report $A_{\mathcal{T}\to\mathcal{T}}$ as simply averaging $A_{\mathcal{S}\to\mathcal{T}}$ and $A_{\mathcal{U}\to\mathcal{S}}$ to compute $A_{\mathcal{T}\to\mathcal{T}}$ might be misleading when the two metrics are not balanced, as shown below.

3.3 Generalized ZSL is hard

To demonstrate the difficulty of GZSL, we report the empirical results of using a simple but intuitive algorithm for GZSL. Given the discriminant functions, we adopt the following classification rule

$$\hat{y} = \arg\max_{c\in\mathcal{T}} \quad f_c(\boldsymbol{x}) \tag{1}$$

which we refer to as *direct stacking*.

We use the rule on "stacking" classifiers from the following zero-shot learning approaches: DAP and IAP [38], ConSE [17], and Synthesized Classifiers (SynC) [28]. We tune the hyper-parameters for each approach based on class-wise cross validation [28,26,33]. We test GZSL on two datasets **AwA** [38] and **CUB** [39] — details about those datasets can be found in Section 5.

Table 1 reports experimental results based on the 4 performance metrics we have described previously. Our goal here is *not* to compare between methods. Instead, we examine the impact of relaxing the assumption of *the prior knowledge of* whether data are from seen or unseen classes.

We observe that, in this setting of GZSL, the classification performance for unseen classes ($A_{\mathcal{U}\to\mathcal{T}}$) drops significantly from the performance in conventional ZSL ($A_{\mathcal{U}\to\mathcal{U}}$), while that of seen ones ($A_{\mathcal{S}\to\mathcal{T}}$) remains roughly the same as in the multi-class task ($A_{\mathcal{S}\to\mathcal{S}}$). That is, *nearly all test data from unseen classes are misclassified into the seen classes*. This unusual degradation in performance highlights the challenges of GZSL; as we only see labeled data from seen classes

Table 1. Classification accuracies (%) on conventional ZSL ($A_{\mathcal{U}\to\mathcal{U}}$), multi-class classification for seen classes ($A_{\mathcal{S}\to\mathcal{S}}$), and GZSL ($A_{\mathcal{S}\to\mathcal{T}}$ and $A_{\mathcal{U}\to\mathcal{T}}$), on **AwA** and **CUB**. Significant drops are observed from $A_{\mathcal{U}\to\mathcal{U}}$ to $A_{\mathcal{U}\to\mathcal{T}}$.

	AwA				**CUB**			
Method	$A_{\mathcal{U}\to\mathcal{U}}$	$A_{\mathcal{S}\to\mathcal{S}}$	$A_{\mathcal{U}\to\mathcal{T}}$	$A_{\mathcal{S}\to\mathcal{T}}$	$A_{\mathcal{U}\to\mathcal{U}}$	$A_{\mathcal{S}\to\mathcal{S}}$	$A_{\mathcal{U}\to\mathcal{T}}$	$A_{\mathcal{S}\to\mathcal{T}}$
DAP [38]	51.1	78.5	2.4	77.9	38.8	56.0	4.0	55.1
IAP [38]	56.3	77.3	1.7	76.8	36.5	69.6	1.0	69.4
ConSE [17]	63.7	76.9	9.5	75.9	35.8	70.5	1.8	69.9
SynC$^{\text{o-vs-o}}$ [28]	70.1	67.3	0.3	67.3	53.0	67.2	8.4	66.5
SynC$^{\text{struct}}$ [28]	73.4	81.0	0.4	81.0	54.4	73.0	13.2	72.0

during training, the scoring functions of seen classes tend to dominate those of unseen classes, leading to biased predictions in GZSL and aggressively classifying a new data point into the label space of $\mathcal{S}$ because classifiers for the seen classes do not get trained on "negative" examples from the unseen classes.

4 Approach for GZSL

The previous example shows that the classifiers for unseen classes constructed by conventional ZSL methods should not be naively combined with models for seen classes to expand the labeling space required by GZSL.

In what follows, we propose a simple variant to the naive approach of *direct stacking* to curb such a problem. We also develop a metric that measures the performance of GZSL, by acknowledging that there is an inherent trade-off between recognizing seen classes and recognizing unseen classes. This metric, referred to as the Area Under Seen-Unseen accuracy Curve (AUSUC), balances the two conflicting forces. We conclude this section by describing two related approaches: despite their sophistication, they do not perform well empirically.

4.1 Calibrated stacking

Our approach stems from the observation that the scores of the discriminant functions for the seen classes are often greater than the scores for the unseen classes. Thus, intuitively, we would like to reduce the scores for the seen classes. This leads to the following classification rule:

$$\hat{y} = \arg\max_{c \in \mathcal{T}} \quad f_c(\boldsymbol{x}) - \gamma \mathbb{I}[c \in \mathcal{S}], \tag{2}$$

where the indicator $\mathbb{I}[\cdot] \in \{0, 1\}$ indicates whether or not c is a seen class and γ is a calibration factor. We term this adjustable rule as *calibrated stacking*.

Another way to interpret γ is to regard it as the prior likelihood of a data point coming from unseen classes. When $\gamma = 0$, the calibrated stacking rule reverts back to the direct stacking rule, described previously.

It is also instructive to consider the two extreme cases of γ. When $\gamma \to +\infty$, the classification rule will ignore all seen classes and classify all data points into

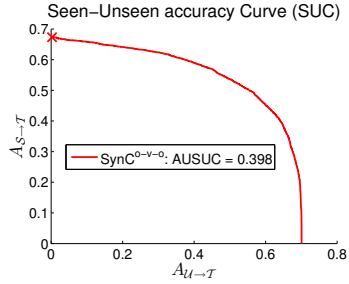

Fig. 1. The Seen-Unseen accuracy Curve (SUC) obtained by varying γ in the calibrated stacking classification rule eq. (2). The AUSUC summarizes the curve by computing the area under it. We use the method SynC$^{\text{O-vs-O}}$ on the **AwA** dataset, and tune hyperparameters as in Table 1. The red cross denotes the accuracies by direct stacking.

one of the unseen classes. When there is no new data point coming from seen classes, this classification rule essentially implements what one would do in the setting of conventional ZSL. On the other hand, when $\gamma \to -\infty$, the classification rule only considers the label space of seen classes as in standard multi-way classification. The calibrated stacking rule thus represents a middle ground between aggressively classifying every data point into seen classes and conservatively classifying every data point into unseen classes. Adjusting this hyperparameter thus gives a trade-off, which we exploit to define a new performance metric.

4.2 Area Under Seen-Unseen Accuracy Curve (AUSUC)

Varying the calibration factor γ, we can compute a series of classification accuracies ($A_{\mathcal{U}\to\mathcal{T}}$, $A_{\mathcal{S}\to\mathcal{T}}$). Fig. 1 plots those points for the dataset **AwA** using the classifiers generated by the method in [28] based on class-wise cross validation. We call such a plot the *Seen-Unseen accuracy Curve (SUC)*.

On the curve, $\gamma = 0$ corresponds to direct stacking, denoted by a cross. The curve is similar to many familiar curves for representing conflicting goals, such as the Precision-Recall (PR) curve and the Receiving Operator Characteristic (ROC) curve, with two ends for the extreme cases ($\gamma \to -\infty$ and $\gamma \to +\infty$).

A convenient way to summarize the plot with one number is to use the Area Under SUC (AUSUC)[1]. The higher the area is, the better an algorithm is able to balance $A_{\mathcal{U}\to\mathcal{T}}$ and $A_{\mathcal{S}\to\mathcal{T}}$. In Section 5, Section 6, and the Supplementary Material, we evaluate the performance of existing zero-shot learning methods under this metric, as well as provide further insights and analyses.

An immediate and important use of the metric AUSUC is for model selection. Many ZSL learning methods require tuning hyperparameters — previous work tune them based on the accuracy $A_{\mathcal{U}\to\mathcal{U}}$. The selected model, however, does not necessarily balance optimally between $A_{\mathcal{U}\to\mathcal{T}}$ and $A_{\mathcal{S}\to\mathcal{T}}$. Instead, we advocate using AUSUC for model selection and hyperparamter tuning. Models with higher

[1] If a single γ is desired, the "F-score" that balances $A_{\mathcal{U}\to\mathcal{T}}$ and $A_{\mathcal{S}\to\mathcal{T}}$ can be used.

values of AUSUC are likely to perform in balance for the task of GZSL. For detailed discussions, see the Supplementary Material.

4.3 Alternative approaches

Socher et al. [9] propose a two-stage zero-shot learning approach that first predicts whether an image is of seen or unseen classes and then accordingly applies the corresponding classifiers. The first stage is based on the idea of novelty detection and assigns a high novelty score if it is unlikely for the data point to come from seen classes. They experiment with two novelty detection strategies: Gaussian and LoOP models [40]. We briefly describe and contrast them to our approach below. The details are in the Supplementary Material.

Novelty detection The main idea is to assign a novelty score $N(\boldsymbol{x})$ to each sample $\boldsymbol{x}$. With this novelty score, the final prediction rule becomes

$$\hat{y} = \begin{cases} \arg\max_{c \in \mathcal{S}} & f_c(\boldsymbol{x}), \quad \text{if } N(\boldsymbol{x}) \leq -\gamma. \\ \arg\max_{c \in \mathcal{U}} & f_c(\boldsymbol{x}), \quad \text{if } N(\boldsymbol{x}) > -\gamma. \end{cases} \tag{3}$$

where $-\gamma$ is the novelty threshold. The scores above this threshold indicate belonging to unseen classes. To estimate $N(\boldsymbol{x})$, for the Gaussian model, data points in seen classes are first modeled with a Gaussian mixture model. The novelty score of a data point is then its negative log probability value under this mixture model. Alternatively, the novelty score can be estimated using the Local Outlier Probabilities (LoOP) model [40]. The idea there is to compute the distances of $\boldsymbol{x}$ to its nearest seen classes. Such distances are then converted to an outlier probability, interpreted as the likelihood of $\boldsymbol{x}$ being from unseen classes.

Relation to calibrated stacking If we define a new form of novelty score $N(\boldsymbol{x}) = \max_{u \in \mathcal{U}} f_u(\boldsymbol{x}) - \max_{s \in \mathcal{S}} f_s(\boldsymbol{x})$ in eq. (3), we recover the prediction rule in eq. (2). However, this relation holds only if we are interested in predicting one label $\hat{y}$. When we are interested in predicting a set of labels (for example, hoping that the correct labels are in the top K predicted labels, (i.e., the Flat hit@K metric, cf. Section 5), the two prediction rules will give different results.

5 Experimental Results

5.1 Setup

Datasets We mainly use three benchmark datasets: the **Animals with Attributes (AwA)** [38], **CUB-200-2011 Birds (CUB)** [39], and **ImageNet** (with full 21,841 classes) [41]. Table 2 summarizes their key characteristics.

Semantic spaces For the classes in **AwA** and **CUB**, we use 85-dimensional and 312-dimensional binary or continuous-valued attributes, respectively [38,39]. For **ImageNet**, we use 500-dimensional word vectors (WORD2VEC) trained by the skip-gram model [7,42] provided by Changpinyo et al. [28]. We ignore classes without word vectors, resulting in 20,345 (out of 20,842) unseen classes. We follow [28] to normalize all but binary embeddings to have unit ℓ_2 norms.

Table 2. Key characteristics of the studied datasets.

Dataset name	Number of seen classes	Number of unseen classes	Total number of images
AwA†	40	10	30,475
CUB‡	150	50	11,788
ImageNet§	1000	20,842	14,197,122

†: following the split in [38]. ‡: following [28] to report the average over 4 random splits. §: seen and unseen classes from ImageNet ILSVRC 2012 1K [41] and Fall 2011 release [29], respectively.

Visual features We use the GoogLeNet deep features [43] pre-trained on ILSVRC 2012 1K [41] for all datasets (all extracted with the Caffe package [44]). Extracted features come from the 1,024-dimensional activations of the pooling units, as in [20,28].

Zero-shot learning methods We examine several representative conventional zero-shot learning approaches, described briefly below. Direct Attribute Prediction (DAP) and Indirect Attribute Prediction (IAP) [38] are probabilistic models that perform attribute predictions as an intermediate step and then use them to compute MAP predictions of unseen class labels. ConSE [17] makes use of pre-trained classifiers for seen classes and their probabilitic outputs to infer the semantic embeddings of each test example, and then classifies it into the unseen class with the most similar semantic embedding. SynC [28] is a recently proposed multi-task learning approach that synthesizes a novel classifier based on semantic embeddings and base classifiers that are learned with labeled data from the seen classes. Two versions of this approach — $\text{SynC}^{\text{o-v-o}}$ and $\text{SynC}^{\text{struct}}$ — use one-versus-other and Crammer-Singer style [45] loss functions to train classifiers. We use binary attributes for DAP and IAP, and continuous attributes and WORD2VEC for ConSE and SynC, following [38,17,28].

Generalized zero-shot learning tasks There are no previously established benchmark tasks for GZSL. We thus define a set of tasks that reflects more closely how data are distributed in real-world applications.

We construct the GZSL tasks by composing test data as a combination of images from both seen and unseen classes. We follow existing splits of the datasets for the conventional ZSL to separate seen and unseen classes. Moreover, for the datasets **AwA** and **CUB**, we hold out 20% of the data points from the seen classes (previously, all of them are used for training in the conventional zero-shot setting) and merge them with the data from the unseen classes to form the test set; for **ImageNet**, we combine its validation set (having the same classes as its training set) and the 21K classes that are not in the ILSVRC 2012 1K dataset.

Evaluation metrics While we will primarily report the performance of ZSL approaches under the metric Area Under Seen-Unseen accuracy Curve (AUSUC)

developed in Section 4.1, we explain how its two accuracy components $A_{\mathcal{S}\to\mathcal{T}}$ and $A_{\mathcal{U}\to\mathcal{T}}$ are computed below.

For **AwA** and **CUB**, seen and unseen accuracies correspond to (normalized-by-class-size) multi-way classification accuracy, where the seen accuracy is computed on the 20% images from the seen classes and the unseen accuracy is computed on images from unseen classes.

For **ImageNet**, seen and unseen accuracies correspond to Flat hit@K (F@K), 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 unnormalized multi-way classification accuracy. Moreover, following the procedure in [8,17,28], we evaluate on three scenarios of increasing difficulty: (1) *2-hop* contains 1,509 unseen classes that are within two tree hops of the 1K seen classes according to the ImageNet label hierarchy[2]. (2) *3-hop* contains 7,678 unseen classes that are within three tree hops of the seen classes. (3) *All* contains all 20,345 unseen classes.

5.2 Which method to use to perform GZSL?

Table 3 provides an experimental comparison between several methods utilizing seen and unseen classifiers for generalized ZSL, with hyperparameters cross-validated to maximize AUSUC. Empirical results on additional datasets and ZSL methods are in the Supplementary Material.

The results show that, irrespective of which ZSL methods are used to generate models for seen and unseen classes, our method of *calibrated stacking* for generalized ZSL outperforms other methods. In particular, despite their probabilistic justification, the two novelty detection methods do not perform well. We believe that this is because most existing zero-shot learning methods are discriminative and optimized to take full advantage of class labels and semantic information. In contrast, either Gaussian or LoOP approach models all the seen classes as a whole, possibly at the cost of modeling inter-class differences.

Table 3. Performances measured in AUSUC of several methods for Generalized Zero-Shot Learning on **AwA** and **CUB**. The higher the better (the upper bound is 1).

Method	**AwA**			**CUB**		
	Novelty detection [9]		Calibrated	Novelty detection [9]		Calibrated
	Gaussian	LoOP	Stacking	Gaussian	LoOP	Stacking
DAP	0.302	0.272	0.366	0.122	0.137	0.194
IAP	0.307	0.287	0.394	0.129	0.145	0.199
ConSE	0.342	0.300	0.428	0.130	0.136	0.212
SynC$^{\text{o-vs-o}}$	0.420	0.378	0.568	0.191	0.209	0.336
SynC$^{\text{struct}}$	0.424	0.373	0.583	0.199	0.224	0.356

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

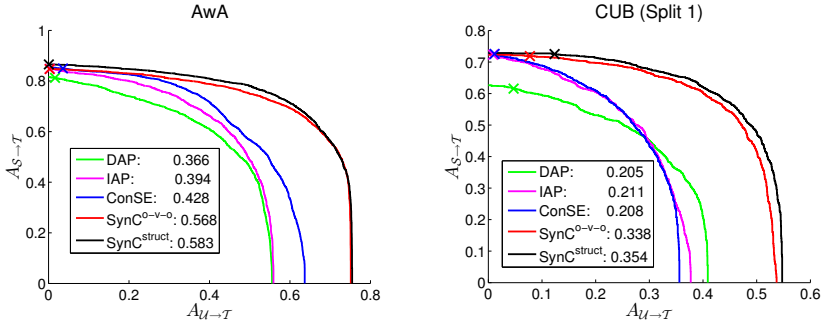

Fig. 2. Comparison between several ZSL approaches on the task of GZSL for **AwA** and **CUB**.

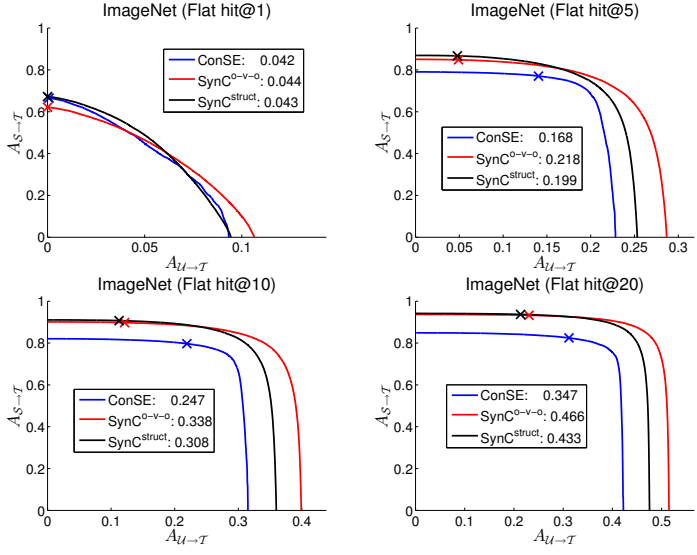

Fig. 3. Comparison between ConSE and SynC of their performances on the task of GZSL for **ImageNet** where the unseen classes are within 2 tree-hops from seen classes.

5.3 Which Zero-shot Learning approach is more robust to GZSL?

Fig. 2 contrasts in detail several ZSL approaches when tested on the task of GZSL, using the method of *calibrated stacking*. Clearly, the SynC method dominates all other methods in the whole ranges. The crosses on the plots mark the results of *direct stacking* (Section 3).

Fig. 3 contrasts in detail ConSE to SynC, the two known methods for large-scale ZSL. When the accuracies measured in Flat hit@1 (i.e., multi-class classification accuracy), neither method dominates the other, suggesting the different trade-offs by the two methods. However, when we measure hit rates in the top $K > 1$, SynC dominates ConSE. Table 4 gives summarized comparison in

Table 4. Performances measured in AUSUC by different zero-shot learning approaches on GZSL on **ImageNet**, using our method of *calibrated stacking*.

Unseen classes	Method	Flat hit@K			
		1	5	10	20
2-hop	ConSE	0.042	0.168	0.247	0.347
	$\text{SynC}^{\text{o-vs-o}}$	0.044	0.218	0.338	0.466
	$\text{SynC}^{\text{struct}}$	0.043	0.199	0.308	0.433
3-hop	ConSE	0.013	0.057	0.090	0.135
	$\text{SynC}^{\text{o-vs-o}}$	0.012	0.070	0.119	0.186
	$\text{SynC}^{\text{struct}}$	0.013	0.066	0.110	0.170
All	ConSE	0.007	0.030	0.048	0.073
	$\text{SynC}^{\text{o-vs-o}}$	0.006	0.034	0.059	0.097
	$\text{SynC}^{\text{struct}}$	0.007	0.033	0.056	0.090

AUSUC between the two methods on the **ImageNet** dataset. We observe that SynC in general outperforms ConSE except when Flat hit@1 is used, in which case the two methods' performances are nearly indistinguishable. Additional plots can be found in the Supplementary Material.

6 Analysis on (Generalized) Zero-shot Learning

Zero-shot learning, either in conventional setting or generalized setting, is a challenging problem as there is no labeled data for the unseen classes. The performance of ZSL methods depends on at least two factors: (1) how seen and unseen classes are related; (2) how effectively the relation can be exploited by learning algorithms to generate models for the unseen classes. For generalized zero-shot learning, the performance further depends on how classifiers for seen and unseen classes are combined to classify new data into the joint label space.

Despite extensive study in ZSL, several questions remain understudied. For example, given a dataset and a split of seen and unseen classes, what is the best possible performance of any ZSL method? How far are we from there? What is the most crucial component we can improve in order to reduce the gap between the state-of-the-art and the ideal performances?

In this section, we empirically analyze ZSL methods in detail and shed light on some of those questions.

Setup As ZSL methods do not use labeled data from unseen classes for training classifiers, one reasonable estimate of their best possible performance is to measure the performance on a multi-class classification task where annotated data on the unseen classes are provided.

Concretely, to construct the multi-class classification task, on **AwA** and **CUB**, we randomly select 80% of the data along with their labels from all classes (seen and unseen) to train classifiers. The remaining 20% will be used to assess both the multi-class classifiers and the classifiers from ZSL. Note that,

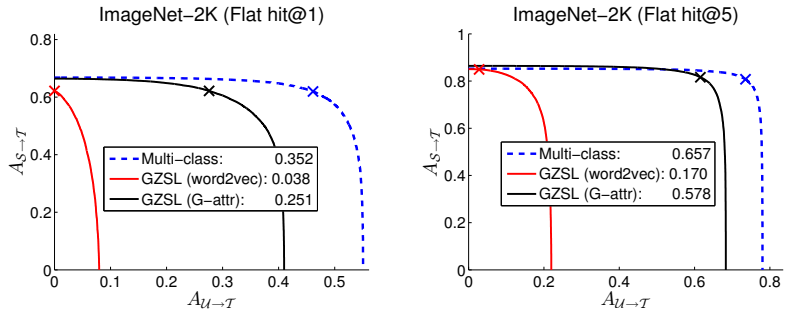

Fig. 4. We contrast the performances of GZSL to multi-class classifiers trained with labeled data from both seen and unseen classes on the dataset **ImageNet-2K**. GZSL uses WORD2VECTOR (in red color) and the idealized visual features (G-attr) as semantic embeddings (in black color).

for ZSL, only the seen classes from the 80% are used for training — the portion belonging to the unseen classes are not used.

On **ImageNet**, to reduce the computational cost (of constructing multi-class classifiers which would involve 20,345-way classification), we subsample another 1,000 unseen classes from its original 20,345 unseen classes. We call this new dataset **ImageNet-2K** (including the 1K seen classes from **ImageNet**). The subsampling procedure is described in the Supplementary Material and the main goal is to keep the proportions of difficult unseen classes unchanged. Out of those 1,000 unseen classes, we randomly select 50 samples per class and reserve them for testing and use the remaining examples (along with their labels) to train 2000-way classifiers.

For ZSL methods, we use either attribute vectors or word vectors (WORD2VEC) as semantic embeddings. Since SynC$^{\text{o-vs-o}}$ [28] performs well on a range of datasets and settings, we focus on this method. For multi-class classification, we train one-versus-others SVMs. Once we obtain the classifiers for both seen and unseen classes, we use the *calibrated stacking* decision rule to combine (as in generalized ZSL) and vary the calibration factor γ to obtain the Seen-Unseen accuracy Curve, exemplified in Fig. 1.

How far are we from the ideal performance? Fig. 4 displays the Seen-Unseen accuracy Curves for **ImageNet-2K** — additional plots on **ImageNet-2K** and similar ones on **AwA** and **CUB** are in the Supplementary Material. Clearly, there is a large gap between the performances of GZSL using the default WORD2VEC semantic embeddings and the ideal performance indicated by the multi-class classifiers. Note that the cross marks indicate the results of *direct stacking*. The multi-class classifiers not only dominate GZSL in the whole ranges (thus, with very high AUSUCs) but also are capable of learning classifiers that are well-balanced (such that *direct stacking* works well).

Table 5. Comparison of performances measured in AUSUC between GZSL (using WORD2VEC and **G-attr**) and multi-class classification on **ImageNet-2K**. Few-shot results are averaged over 100 rounds. GZSL with **G-attr** improves upon GZSL with WORD2VEC significantly and quickly approaches multi-class classification performance.

Method		Flat hit@K			
		1	5	10	20
GZSL	WORD2VEC	0.04	0.17	0.27	0.38
	G-attr from 1 image	0.08±0.003	0.25±0.005	0.33±0.005	0.42±0.005
	G-attr from 10 images	0.20±0.002	0.50±0.002	0.62±0.002	0.72±0.002
	G-attr from 100 images	0.25±0.001	0.57±0.001	0.69±0.001	0.78±0.001
	G-attr from all images	0.25	0.58	0.69	0.79
Multi-class classification		0.35	0.66	0.75	0.82

How much can idealized semantic embeddings help? We hypothesize that a large portion of the gap between GZSL and multi-class classification can be attributed to the weak semantic embeddings used by the GZSL approach.

We investigate this by using a form of *idealized* semantic embeddings. As the success of zero-shot learning relies heavily on how accurate semantic embeddings represent visual similarity among classes, we examine the idea of *visual features as semantic embeddings*. Concretely, for each class, semantic embeddings can be obtained by averaging visual features of images belonging to that class. We call them **G-attr** as we derive the visual features from GoogLeNet. Note that, for unseen classes, we only use the reserved training examples to derive the semantic embeddings; we do not use their labels to train classifiers.

Fig. 4 shows the performance of GZSL using **G-attr** — the gaps to the multi-class classification performances are significantly reduced from those made by GZSL using WORD2VEC. In some cases (see the Supplementary Material for more comprehensive experiments), GZSL can almost match the performance of multi-class classifiers without using any labels from the unseen classes!

How much labeled data do we need to improve GZSL's performance? Imagine we are given a budget to label data from unseen classes, how much those labels can improve GZSL's performance?

Table 5 contrasts the AUSUCs obtained by GZSL to those from mutli-class classification on **ImageNet-2K**, where GZSL is allowed to use visual features as embeddings — those features can be computed from a few labeled images from the unseen classes, a scenario we can refer to as "few-shot" learning. Using about (randomly sampled) 100 labeled images per class, GZSL can quickly approach the performance of multi-class classifiers, which use about 1,000 labeled images per class. Moreover, those G-attr visual features as semantic embeddings improve upon WORD2VEC more significantly under Flat hit@K = 1 than when K > 1.

We further examine on the whole **ImageNet** with 20,345 unseen classes in Table 6, where we keep 80% of the unseen classes' examples to derive **G-attr** and test on the rest, and observe similar trends. Specifically on Flat hit@1, the performance of G-attr from merely 1 image is boosted **threefold** of that

Table 6. Comparison of performances measured in AUSUC between GZSL with WORD2VEC and GZSL with **G-attr** on the full **ImageNet** with 21,000 unseen classes. Few-shot results are averaged over 20 rounds.

Method	Flat hit@K			
	1	5	10	20
WORD2VEC	0.006	0.034	0.059	0.096
G-attr from 1 image	0.018±0.0002	0.071±0.0007	0.106±0.0009	0.150±0.0011
G-attr from 10 images	0.050±0.0002	0.184±0.0003	0.263±0.0004	0.352±0.0005
G-attr from 100 images	0.065±0.0001	0.230±0.0002	0.322±0.0002	0.421±0.0002
G-attr from all images	0.067	0.236	0.329	0.429

by WORD2VEC, while G-attr from 100 images achieves over tenfold. See the Supplementary Material for details, including results on **AwA** and **CUB**.

7 Discussion

We investigate the problem of generalized zero-shot learning (GZSL). GZSL relaxes the unrealistic assumption in conventional zero-shot learning (ZSL) that test data belong only to unseen novel classes. In GZSL, test data might also come from seen classes and the labeling space is the union of both types of classes. We show empirically that a straightforward application of classifiers provided by existing ZSL approaches does not perform well in the setting of GZSL. Motivated by this, we propose a surprisingly simple but effective method to adapt ZSL approaches for GZSL. The main idea is to introduce a calibration factor to calibrate the classifiers for both seen and unseen classes so as to balance two conflicting forces: recognizing data from seen classes and those from unseen ones. We develop a new performance metric called the Area Under Seen-Unseen accuracy Curve to characterize this trade-off. We demonstrate the utility of this metric by analyzing existing ZSL approaches applied to the generalized setting. Extensive empirical studies reveal strengths and weaknesses of those approaches on three well-studied benchmark datasets, including the large-scale ImageNet with more than 20,000 unseen categories. We complement our comparative studies in learning methods by further establishing an upper bound on the performance limit of GZSL. In particular, our idea is to use class-representative visual features as the idealized semantic embeddings. We show that there is a large gap between the performance of existing approaches and the performance limit, suggesting that improving the quality of class semantic embeddings is vital to improving ZSL.

Acknowledgements B.G. is partially supported by NSF IIS-1566511. Others are partially supported by USC Graduate Fellowship, NSF IIS-1065243, 1451412, 1513966, 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. Sudderth, E.B., Jordan, M.I.: Shared segmentation of natural scenes using dependent pitman-yor processes. In: NIPS. (2008) 1
2. Salakhutdinov, R., Torralba, A., Tenenbaum, J.: Learning to share visual appearance for multiclass object detection. In: CVPR. (2011) 1
3. Zhu, X., Anguelov, D., Ramanan, D.: Capturing long-tail distributions of object subcategories. In: CVPR. (2014) 1
4. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR. (2009) 2, 3
5. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR. (2009) 2
6. Parikh, D., Grauman, K.: Relative attributes. In: ICCV. (2011) 2
7. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR Workshops. (2013) 2, 7
8. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: Devise: A deep visual-semantic embedding model. In: NIPS. (2013) 2, 3, 9
9. Socher, R., Ganjoo, M., Manning, C.D., Ng, A.: Zero-shot learning through cross-modal transfer. In: NIPS. (2013) 2, 3, 7, 9
10. Palatucci, M., Pomerleau, D., Hinton, G.E., Mitchell, T.M.: Zero-shot learning with semantic output codes. In: NIPS. (2009) 2, 3
11. Yu, X., Aloimonos, Y.: Attribute-based transfer learning for object categorization with zero/one training example. In: ECCV. (2010) 2, 3
12. Rohrbach, M., Stark, M., Schiele, B.: Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In: CVPR. (2011) 2, 3
13. Kankuekul, P., Kawewong, A., Tangruamsub, S., Hasegawa, O.: Online incremental attribute-based zero-shot learning. In: CVPR. (2012) 2, 3
14. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for attribute-based classification. In: CVPR. (2013) 2, 3
15. Yu, F.X., Cao, L., Feris, R.S., Smith, J.R., Chang, S.F.: Designing category-level attributes for discriminative visual recognition. In: CVPR. (2013) 2, 3
16. Mensink, T., Gavves, E., Snoek, C.G.: Costa: Co-occurrence statistics for zero-shot classification. In: CVPR. (2014) 2, 3
17. Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G.S., Dean, J.: Zero-shot learning by convex combination of semantic embeddings. In: ICLR. (2014) 2, 3, 4, 5, 8, 9
18. Jayaraman, D., Grauman, K.: Zero-shot recognition with unreliable attributes. In: NIPS. (2014) 2, 3
19. Al-Halah, Z., Stiefelhagen, R.: How to transfer? zero-shot object recognition via hierarchical transfer of semantic attributes. In: WACV. (2015) 2, 3
20. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: CVPR. (2015) 2, 3, 8
21. Fu, Y., Hospedales, T.M., Xiang, T., Gong, S.: Transductive multi-view zero-shot learning. TPAMI (2015) 2, 3
22. Fu, Z., Xiang, T., Kodirov, E., Gong, S.: Zero-shot object recognition by semantic manifold distance. In: CVPR. (2015) 2, 3
23. Li, X., Guo, Y., Schuurmans, D.: Semi-supervised zero-shot classification with label representation learning. In: ICCV. (2015) 2, 3
24. Romera-Paredes, B., Torr, P.H.S.: An embarrassingly simple approach to zero-shot learning. In: ICML. (2015) 2, 3

25. Kodirov, E., Xiang, T., Fu, Z., Gong, S.: Unsupervised domain adaptation for zero-shot learning. In: ICCV. (2015) 2, 3
26. Zhang, Z., Saligrama, V.: Zero-shot learning via semantic similarity embedding. In: ICCV. (2015) 2, 3, 4
27. Zhang, Z., Saligrama, V.: Zero-shot learning via joint latent similarity embedding. In: CVPR. (2016) 2, 3
28. Changpinyo, S., Chao, W.L., Gong, B., Sha, F.: Synthesized classifiers for zero-shot learning. In: CVPR. (2016) 2, 3, 4, 5, 6, 7, 8, 9, 12
29. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. (2009) 2, 8
30. Mensink, T., Verbeek, J., Perronnin, F., Csurka, G.: Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In: ECCV. (2012) 3
31. Tang, K.D., Tappen, M.F., Sukthankar, R., Lampert, C.H.: Optimizing one-shot recognition with micro-set learning. In: CVPR. (2010) 3
32. Gan, C., Yang, Y., Zhu, L., Zhao, D., Zhuang, Y.: Recognizing an action using its name: A knowledge-based approach. IJCV (2016) 1–17 3
33. Elhoseiny, M., Saleh, B., Elgammal, A.: Write a classifier: Zero-shot learning using purely textual descriptions. In: ICCV. (2013) 3, 4
34. Lei Ba, J., Swersky, K., Fidler, S., Salakhutdinov, R.: Predicting deep zero-shot convolutional neural networks using textual descriptions. In: ICCV. (2015) 3
35. Scheirer, W.J., de Rezende Rocha, A., Sapkota, A., Boult, T.E.: Toward open set recognition. TPAMI **35**(7) (2013) 1757–1772 3
36. Scheirer, W.J., Jain, L.P., Boult, T.E.: Probability models for open set recognition. TPAMI **36**(11) (2014) 2317–2324 3
37. Jain, L.P., Scheirer, W.J., Boult, T.E.: Multi-class open set recognition using probability of inclusion. In: ECCV. (2014) 393–409 3
38. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. TPAMI **36**(3) (2014) 453–465 4, 5, 7, 8
39. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology (2011) 4, 7
40. Kriegel, H.P., Kröger, P., Schubert, E., Zimek, A.: LoOP: local outlier probabilities. In: CIKM. (2009) 7
41. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV (2015) 7, 8
42. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. (2013) 7
43. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. (2015) 8
44. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACM Multimedia. (2014) 8
45. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. JMLR **2** (2002) 265–292 8