From Classical to Generalized Zero-Shot Learning: A Simple Adaptation Process

Yannick Le Cacheux, Hervé Le Borgne, Michel Crucianu

From Classical to Generalized Zero-Shot Learning: a Simple Adaptation Process

Yannick Le Cacheux CEA LIST
Hervé Le Borgne CEA LIST
Michel Crucianu CEDRIC Lab – CNAM

September 27, 2018

Abstract

Zero-shot learning (ZSL) is concerned with the recognition of previously *unseen* classes. It relies on additional semantic knowledge for which a mapping can be learned with training examples of *seen* classes. While classical ZSL considers the recognition performance on unseen classes only, generalized zero-shot learning (GZSL) aims at maximizing performance on both seen and unseen classes. In this paper, we propose a new process for training and evaluation in the GZSL setting; this process addresses the gap in performance between samples from unseen and seen classes by penalizing the latter, and enables to select hyper-parameters well-suited to the GZSL task. It can be applied to any existing ZSL approach and leads to a significant performance boost: the experimental evaluation shows that GZSL performance, averaged over eight state-of-the-art methods, is improved from 28.5 to 42.2 on CUB and from 28.2 to 57.1 on AwA2.

1 Introduction

Zero-shot learning (ZSL) [1, 2, 3] aims to recognize classes for which no training example is available. This is often achieved by relying on additional semantic knowledge, consisting for example in vectors of attributes. During training, a relation between visual features and semantic attributes is learned from training examples belonging to the *seen* classes, for which both modalities (visual and semantic) are available. This model is then applied in the testing phase on examples from *unseen* classes, for which no visual instance was available during training. Predictions on these classes can thus be made on the basis of the inferred relation between visual and semantic features.

In classical ZSL, the test set only contains examples from the novel, unseen classes, and these classes alone can be predicted. Although this setting has enabled significant progress in methods linking visual content to semantic

information in the last few years [4], it is hardly realistic. It seems much more reasonable to assume that objects which are to be classified can belong to either a seen class or an unseen class, since in real-life use-cases one could legitimately want to recognize both former and novel classes. This setting is usually referred to as generalized zero-shot learning (GZSL).

However, recent work shows that a direct use of a ZSL model in a GZSL setting usually leads to unsatisfactory results. Indeed, in addition to the number of candidate classes being higher due to the presence of the seen classes among them, most samples from unseen classes are incorrectly classified as belonging to one of the seen classes [5]. Different methods have been proposed to measure this discrepancy, such as the area under the curve representing all the possible trade-offs between the accuracies on samples from seen classes versus samples from unseen classes [5], or their harmonic mean [4] to penalize models with strong imbalance between the two. While these proposals only measure the extent of the problem, we aim to explicitly address this issue in addition to quantifying its impact.

The main contribution of this paper is a new process for training and evaluating models in a GZSL setting. In accordance with recent studies, we show that the application of a ZSL model "out of the box" gives results that are far from optimal in the GZSL context. We demonstrate how two simple techniques – the calibration of similarities and the use of appropriately balanced regularization – can dramatically improve the performance of most models. The final score for the GZSL task can thus be increased up to a factor of two, with no change regarding the underlying hypotheses of the GZSL task or the data available at any given time, which means that our process is applicable to any ZSL model. We also provide new insights on the reasons why these two techniques are relevant and on the fundamental differences between samples from seen and unseen classes.

We extensively evaluate these techniques on several recent ZSL methods. For sanity-check, we independently reproduce results obtained in the literature before applying our process. We find that some models show a variability in performance with respect to their random initialization, so measures averaged over several runs should be preferred. We eventually find that, with fair comparison under unbiased conditions as enabled by our process, a regularized linear model can give results close to or even better than the state-of-the-art.

2 Related work

An early rigorous definition and evaluation of GZSL was put forward in [5]. The authors argue that this setting is more realistic than ZSL and highlight the gap between accuracies on seen and unseen classes when labels from *all* classes can be predicted (denoted respectively $A_{\mathcal{U}\to\mathcal{C}}$ and $A_{\mathcal{S}\to\mathcal{C}}$, and formally defined in Sec. 3.1). They also introduce the idea of calibration to address this issue and suggest a new metric for GZSL, *Accuracy Under Seen-Unseen Curve (AUSUC)*, which measures the trade-off between the two accuracies but does not directly

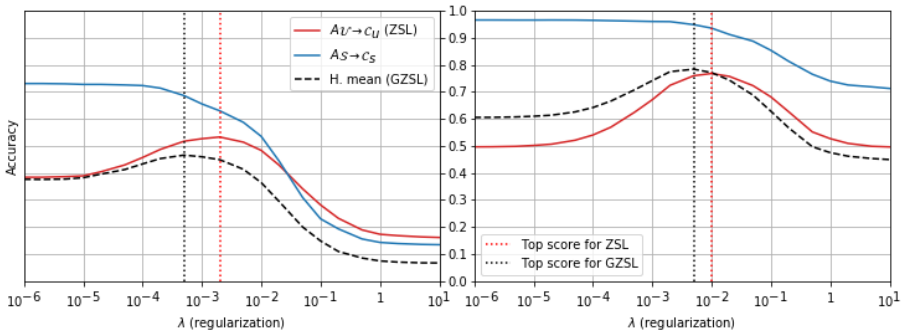


Figure 1: Illustration of how the regularization parameter λ affects the accuracies on samples from seen and unseen classes $A_{\mathcal{U}\to\mathcal{C}_u}$ and $A_{\mathcal{S}\to\mathcal{C}_s}$ (see Sec. 3.1) as measured on CUB [6] (left) and AwA2 [7] (right). Optimal regularization is not the same in a ZSL setting, where performance is measured by $A_{\mathcal{U}\to\mathcal{C}_u}$ (red dotted line), and in a GZSL setting, where it is measured by the harmonic mean of $A_{\mathcal{U}\to\mathcal{C}}$ and $A_{\mathcal{S}\to\mathcal{C}}$ (black dotted line).

provide the expected performance in real use-cases.

An extensive evaluation of recent ZSL methods with a common protocol is provided in [4], both in ZSL and GZSL settings. The authors use a different metric for GZSL, the harmonic mean between $A_{\mathcal{U}\to\mathcal{C}}$ and $A_{\mathcal{S}\to\mathcal{C}}$, which does not directly quantify the trade-off between accuracies but better estimates the practical performance of a given model. However, they do not explicitly address the gap between similarities evaluated on seen and unseen classes [5], which has a significant impact on the final performance as we show in Sec. 4.3.

Further GZSL results based on the harmonic mean metric are provided in [8, 9, 10]. All three methods rely on generators of artificial training examples from unseen classes. However, these methods assume that a semantic description of all unseen classes is available during training. This assumption is not necessarily met in practice and makes the inclusion of additional unseen classes more difficult.

Transductive ZSL methods [11, 12, 13] also assume that additional information, taking the form of unlabeled samples from unseen classes, is available during training. This can naturally lead to improved performance. In this article, we make none of these assumptions and consider that *no* information regarding unseen classes is available at training time.

3 Proposed approach

3.1 Problem statement

We denote by $\mathcal{C}_s$ the set of classes *seen* during training and by $\mathcal{C}_u$ the set of *unseen* classes. We define $\mathcal{C} = \mathcal{C}_s \cup \mathcal{C}_u$, with $\mathcal{C}_s \cap \mathcal{C}_u = \emptyset$. During the

training phase, we consider N^{tr} training samples consisting of D-dimensional visual features $\mathbf{X}^{tr} = (\mathbf{x}_1^{tr}, \ldots, \mathbf{x}_{N^{tr}}^{tr})^\top \in \mathbb{R}^{N^{tr} \times D}$ and corresponding labels $\mathbf{y}^{tr} = (y_1^{tr}, \ldots, y_{N^{tr}}^{tr})^\top \in {\mathcal{C}_s}^{N^{tr}}$, as well as K-dimensional semantic *class prototypes* noted by $\mathbf{S}^{tr} = (\mathbf{s}_1^{tr}, \ldots, \mathbf{s}_{|\mathcal{C}_s|}^{tr})^\top \in \mathbb{R}^{|\mathcal{C}_s| \times K}$. We seek to learn a function $f : \mathbb{R}^D \times \mathbb{R}^K \to \mathbb{R}$ assigning a similarity score to each pair composed of a visual feature vector and a semantic representation so as to minimize the following regularized loss:

$$\frac{1}{N^{tr}} \sum_{n=1}^{N^{tr}} \sum_{c=1}^{|\mathcal{C}_s|} L(f(\mathbf{x}_n^{tr}; \mathbf{s}_c^{tr}), y_n^{tr}) + \lambda \Omega[f] \tag{1}$$

where L is the loss function and Ω the regularization term weighted by λ. During the testing phase, we consider N^{te} unlabeled visual samples $\mathbf{X}^{te} = (\mathbf{x}_1^{te}, \ldots, \mathbf{x}_{N^{te}}^{te})^\top \in \mathbb{R}^{N^{te} \times D}$ and class prototypes for *candidate classes*. In a ZSL setting, the candidate classes $\mathcal{C}^{te}$ are the unseen classes such that $\mathbf{S}^{te} \in \mathbb{R}^{|\mathcal{C}_u| \times K}$. In a GZSL setting, classes to be predicted can be in either $\mathcal{C}_u$ or $\mathcal{C}_s$, such that $\mathcal{C}^{te} = \mathcal{C}$ and the class prototypes are $\mathbf{S}^{te} = (\mathbf{s}_1^{te}, \ldots, \mathbf{s}_{|\mathcal{C}_s|}^{te}, \mathbf{s}_{|\mathcal{C}_s|+1}^{te}, \ldots, \mathbf{s}_{|\mathcal{C}_s|+|\mathcal{C}_u|}^{te})^\top \in \mathbb{R}^{|\mathcal{C}| \times K}$. In both cases, given a function $\hat{f}$ learned in the training phase, we want to estimate a prediction $\hat{y}$ for a visual testing sample $\mathbf{x}$ such that:

$$\hat{y} = \underset{c \in \mathcal{C}^{te}}{\operatorname{argmax}} \hat{f}(\mathbf{x}; \mathbf{s}_c^{te}) \tag{2}$$

In classical ZSL, performance is measured by the *accuracy of unseen classes among unseen classes*, noted $A_{\mathcal{U} \to \mathcal{C}_u}$, while in GZSL we are interested in the *accuracy of unseen classes among all classes* and the *accuracy of seen classes among all classes*, noted respectively $A_{\mathcal{U} \to \mathcal{C}}$ and $A_{\mathcal{S} \to \mathcal{C}}$ as in [5]. $A_{\mathcal{S} \to \mathcal{C}_s}$ is similarly defined.

3.2 Calibration and GZSL split

As evidenced by [5], when a ZSL model is applied in a GZSL setting, $A_{\mathcal{S} \to \mathcal{C}}$ is usually significantly higher than $A_{\mathcal{U} \to \mathcal{C}}$. This is because most samples from unseen classes are incorrectly classified into one of the seen classes. To address this, a *calibration factor* γ is added in [5] to penalize seen classes. Eq. (2) then becomes:

$$\hat{y} = \underset{c \in \mathcal{C}^{te}}{\operatorname{argmax}} \left(\hat{f}(\mathbf{x}; \mathbf{s}_c^{te}) - \gamma \mathbb{1}[c \in \mathcal{C}_s] \right) \tag{3}$$

where $\mathbb{1}[\cdot]$ is an indicator function.

The *Accuracy Under Seen-Unseen Curve (AUSUC)* metric also proposed in [5] is defined as the area under the curve representing $A_{\mathcal{S} \to \mathcal{C}}$ versus $A_{\mathcal{U} \to \mathcal{C}}$ when γ varies from $-\infty$ to $+\infty$, which shows the trade-off between the two.

Instead of computing a metric involving all possible trade-offs between $A_{\mathcal{U} \to \mathcal{C}}$ and $A_{\mathcal{S} \to \mathcal{C}}$, we look for a single specific value of γ, corresponding to the best compromise between the two as measured by the harmonic mean of $A_{\mathcal{U} \to \mathcal{C}}$ and $A_{\mathcal{S} \to \mathcal{C}}$ [4]. We propose to determine the optimal value of γ with a cross-validation specific to GZSL. Usually in machine learning a dataset is divided at

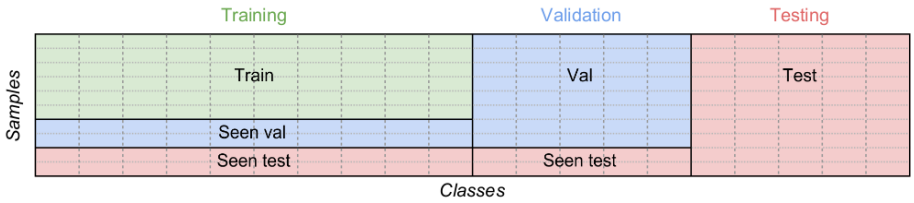


Figure 2: Illustration of the different splits. Each column is a class and each cell is a sample. In this example there are 20 different classes with 10 samples per class. Five classes are used for testing, five other for validation and the remaining ten for training. Among the samples from the validation and training classes, 20% are kept for testing (*seen test set*) and 20% more samples from training classes are kept for validation (*seen validation set*).

random into three parts: a training, a validation and a testing set. In classical ZSL, this splitting process is done with respect to the classes as opposed to the samples: a set of classes is used for training, a disjoint set for validation and a final mutually disjoint set for testing. In GZSL, a fraction (usually 20%) of the samples from the validation and training sets are kept for testing time to be used as test samples from seen classes. We refer to this set as the *seen test set*. Note that here *seen* only indicates that these samples belong to seen classes, not that they have been used during training. To be able to cross-validate parameters for GZSL, we further keep an additional 20% of the remaining training set to be used as samples from seen classes when cross-validating parameters; we refer to this set as the *seen validation set*. Fig. 2 illustrates this partitioning.

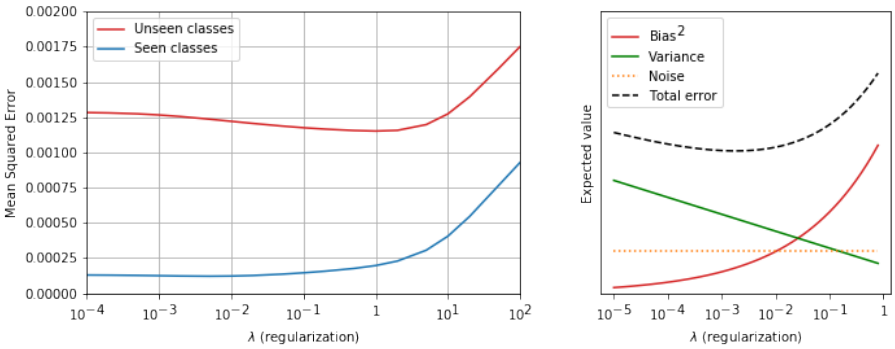
To determine the optimal value of γ we first train a model on the GZSL training set. We then use the (GZSL) validation set and the seen validation set to compute the GZSL metric (the harmonic mean) and keep the value γ^* that maximizes this metric. The ZSL model is subsequently re-trained on the training, validation and seen validation sets, then class similarities are computed for the test set. The value γ^* is subtracted from the similarities of seen classes and the resulting similarities are used to compute the final GZSL score.

3.3 Regularization for GZSL

The usual approach to optimize the regularized loss (Eq. (1)) in GZSL consists in using the value of λ determined on the ZSL task. We argue here that this is unlikely to be optimal and provide some insight to justify our position. Then, we propose a simple method to determine a better value of λ to improve performance in GZSL.

Figure 1 shows $A_{\mathcal{U}\to\mathcal{C}_u}$ and $A_{\mathcal{S}\to\mathcal{C}_s}$ as a function of λ for a regularized linear model (ridge regression [14, 15]), measured on the first validation splits of the *proposed splits* of [4] on CUB [6] and AwA2 [7].

In each case, there is a value of λ that maximizes the ZSL score $A_{\mathcal{U}\to\mathcal{C}_u}$, indicated by the red dotted vertical line, that we note λ^*_{ZSL}. The overall ten-



(a) (b)

Figure 3: (a) MSE of predicted attributes (averaged over attributes and samples) as a function of the regularization parameter λ; (b) Illustration of the bias-variance decomposition.

dency for $A_{\mathcal{S}\to\mathcal{C}_s}$ is to decrease as λ increases. This is not a concern for the ZSL task, since it only considers samples from unseen classes. However, for the GZSL task, we want the best trade-off between $A_{\mathcal{U}\to\mathcal{C}}$ and $A_{\mathcal{S}\to\mathcal{C}}$. Note that $A_{\mathcal{U}\to\mathcal{C}} \leq A_{\mathcal{U}\to\mathcal{C}_u}$ and $A_{\mathcal{S}\to\mathcal{C}} \leq A_{\mathcal{S}\to\mathcal{C}_s}$, with equality only if we are able to perfectly distinguish samples from seen and unseen classes. It follows that λ^*_{GZSL}, the value of λ that maximizes the GZSL score, is not necessarily the same as λ^*_{ZSL}: a small decrease from λ^*_{ZSL} can significantly increase $A_{\mathcal{S}\to\mathcal{C}_s}$ while only slightly penalizing $A_{\mathcal{U}\to\mathcal{C}_u}$. This has a similar impact on the maximum values obtainable by $A_{\mathcal{S}\to\mathcal{C}}$ and $A_{\mathcal{U}\to\mathcal{C}}$, and can ultimately improve the GZSL score. We quantify in Sec. 4.3 the gains attributed to the use of λ^*_{GZSL}.

The reason why λ affects $A_{\mathcal{U}\to\mathcal{C}_u}$ and $A_{\mathcal{S}\to\mathcal{C}_s}$ in this way can be explained with the bias-variance decomposition. For regression, we generally assume that we are given a dataset $\mathcal{D} = (\mathbf{X}, \mathbf{t})$, with samples $(\mathbf{x}_n, t_n)$ independently drawn from a joint distribution $p(\mathbf{x}, t)$, such that $p(t|\mathbf{x}) = \mathcal{N}(t|h(\mathbf{x}), \sigma^2)$, where h is the true dependence. For a prediction function $\hat{h}$ estimated from $\mathcal{D}$ we can then write the expected loss on a new pair $(\mathbf{x}, t)$ as:

$$\mathbb{E}_{\mathcal{D},\mathbf{x},t}[(t - \hat{h}(\mathbf{x}))^2] = \sigma^2 + \mathbb{E}_{\mathcal{D},\mathbf{x}}[(h(\mathbf{x}) - \hat{h}(\mathbf{x}))^2] + \mathrm{var}_{\mathcal{D},\mathbf{x}}[\hat{h}(\mathbf{x})] \tag{4}$$

where the first term is the intrinsic noise of the dataset, the second is the (squared) bias of the predictor and the third is the variance in the estimation of the predictor. It can be shown [14, 15] that for ridge regression the bias increases and the variance decreases with the regularization parameter λ, as illustrated in Fig. 3(b).

In the case of ZSL, $\mathbf{x}$ corresponds to visual samples and t to attribute(s) to be estimated from $\mathbf{x}$. The variance comes from both the differences between samples from the same class (intra-class variance) and from the differences between classes (inter-class variance). Intra-class variance is usually significantly

smaller than inter-class variance in ZSL. Therefore, most of the variance in Eq. (4) can be attributed to the choice of training classes $\mathcal{C}_s$. For samples from unseen classes, the bias-variance decomposition applies and there exists a λ corresponding to the best trade-off between the two. This is evidenced in Fig. 3(a), where the red curve shows the Mean Squared Error (MSE) in the predictions of attributes from unseen classes as a function of λ, for a regularized linear model on the first validation split of AwA2 [7].

For a sample from a seen class, the variance attributable to the choice of the training classes is much smaller since, by definition, the seen class is present in the training dataset. This allows to better estimate attributes from seen classes and most of the expected error therefore comes from the intrinsic noise and the bias. Thus, the expected error mostly increases with λ, as evidenced by the blue curve in Fig. 3(a). If we plausibly assume that the accuracy of predictions for samples from a given class depends on how well we estimate their attributes, this explains both why predictions are better for samples from seen classes than from unseen classes and why their behavior with respect to λ is different.

We then suggest the following procedure to select the optimal value of λ: we repeat the protocol described in Sec. 3.2 for selecting γ^* and we take the value of λ which gives the best result for the harmonic mean between $A_{\mathcal{U}\to\mathcal{C}}$ and $A_{\mathcal{S}\to\mathcal{C}}$ on the validation set *after* having subtracted γ^* from the similarities of seen classes. The rest of the process is identical: we retrain the ZSL model on the training, validation and seen validation sets with the hyperparameter λ^*_{GZSL} that we just determined, we compute similarities for the test set, subtract γ^* from the similarities of seen classes and compute the resulting GZSL score.

4 Experimental evaluation

4.1 Methods

We independently reimplemented six methods frequently cited in the literature to evaluate them with our protocol: ALE [16], DeViSE [17], SJE [18], Sync [19], ESZSL [20] and SAE [21].

In addition, we also evaluate two simple linear models. $\text{Linear}_{\mathcal{V}\to\mathcal{S}}$ applies a linear mapping $\mathbf{W} \in \mathbb{R}^{K\times D}$ from the visual space $\mathcal{V}$ to the semantic space $\mathcal{S}$ to minimize standard MSE. With $\mathbf{T}^{tr} = (\mathbf{s}^{tr}_{y^{tr}_1}, ..., \mathbf{s}^{tr}_{y^{tr}_N})^\top \in \mathbb{R}^{N^{tr}\times K}$ the matrix whose rows correspond to the class prototypes associated to each training sample based on its label, the loss function can be formulated as:

$$\frac{1}{N^{tr}}\|\mathbf{X}^{tr}\mathbf{W}^\top - \mathbf{T}^{tr}\|^2_F + \lambda\|\mathbf{W}\|^2_F \tag{5}$$

$\text{Linear}_{\mathcal{S}\to\mathcal{V}}$ is based on [22] where the authors argue that using the semantic space as the embedding space reduces the variance of the projected points and thus aggravates the hubness problem [23]. They suggest instead to project semantic class prototypes onto the visual space and to compute similarities in this space. Keeping $\mathbf{W} \in \mathbb{R}^{K\times D}$ as our linear mapping, we formulate the loss

function as:

$$\frac{1}{N^{tr}}\|\mathbf{X}^{tr} - \mathbf{T}^{tr}\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_F^2 \tag{6}$$

We can easily obtain closed-form solutions for the two models from the objective functions (5) and (6). For the $\text{Linear}_{\mathcal{V}\to\mathcal{S}}$ model we have $\mathbf{W} = {\mathbf{T}^{tr}}^\top\mathbf{X}^{tr}({\mathbf{X}^{tr}}^\top\mathbf{X}^{tr} + \lambda N^{tr}\mathbf{I}_D)^{-1}$ (7) and for the $\text{Linear}_{\mathcal{S}\to\mathcal{V}}$ model $\mathbf{W} = ({\mathbf{T}^{tr}}^\top\mathbf{T}^{tr} + \lambda N^{tr}\mathbf{I}_K)^{-1}{\mathbf{T}^{tr}}^\top\mathbf{X}^{tr}$ (8).

4.2 Experimental setting

4.2.1 Datasets

We perform our experiments on two standard datasets for ZSL: Caltech-UCSD-Birds 200-2011 (CUB) [6] and Animals with Attributes2[1] (AwA2) [7]. Results for two additional datasets (SUN Attribute Database [25] and Attributes Pascal and Yahoo [26]) are available in the supplementary material. CUB is a fine-grained dataset composed of 11788 pictures of birds from 200 species (*black footed albatross*, ..., *common yellowthroat*). It comes with 312-dimensional binary attributes for each picture, that are averaged by class to obtain semantic class prototypes. AwA2 is a coarse-grained dataset comprising 37322 pictures of 50 animal species (*antelope*, ..., *zebra*). For each class, 85-dimensional attributes are provided.

4.2.2 Splits

The best performing ZSL methods usually rely on visual features obtained with deep neural networks pre-trained on ImageNet [27], such as GoogLeNet [28] or ResNet [29]. As evidenced by [4], this induces a huge bias for ZSL datasets whose classes are not disjoint from categories of ImageNet, as is the case with AwA2, since test classes cannot be considered truly unseen. We therefore adopt the approach of [4] and use their terms *Standard Split* (S.S.) for the split widely used in the literature and *Proposed Split* (P.S.) for the split they introduce. The training and validation splits are further divided for GZSL as described in Sec. 3.2. Statistics regarding the (GZSL) splits are given in the supplementary material.

4.2.3 Settings

Attributes are normalized such that each class prototype has unit $\ell 2$ norm. We use the 101-layered ResNet [29] pre-trained on ImageNet [27] as visual features extractor, keeping the $D = 2048$ activations of the last pooling units.

[1] AwA2 was recently proposed in [7] as a replacement for the Animals with Attributes (AwA) dataset [24] whose images are not publicly available.

Table 1: ZSL score: per-class accuracy $A_{\mathcal{U}\to\mathcal{C}_u}$, as reported in [4] and independently reproduced. S.S.: Standard Split, P.S.: Proposed Split [4]. Averaged over 5 runs.

Method	CUB [6]				AwA2 [7]			
	Reported in [4]		Reproduced		Reported in [4]		Reproduced	
	S.S.	P.S.	S.S.	P.S.	S.S.	P.S.	S.S.	P.S.
Linear$_{\mathcal{V}\to\mathcal{S}}$	*n/a*	*n/a*	41.0 ± 0.0	41.8 ± 0.0	*n/a*	*n/a*	68.2 ± 0.0	49.7 ± 0.0
Linear$_{\mathcal{S}\to\mathcal{V}}$	*n/a*	*n/a*	56.0 ± 0.0	53.5 ± 0.0	*n/a*	*n/a*	**85.5** ± 0.0	**68.9** ± 0.0
ALE [16]	53.2	54.9	54.8 ± 0.8	54.0 ± 1.2	80.3	**62.5**	80.3 ± 2.2	62.9 ± 2.3
DeViSE [17]	53.2	52.0	52.5 ± 0.9	52.6 ± 1.3	68.6	59.7	76.6 ± 1.6	62.1 ± 1.6
SJE [18]	**55.3**	53.9	53.8 ± 2.3	49.2 ± 1.4	69.5	61.9	80.4 ± 2.9	62.2 ± 1.2
ESZSL [20]	55.1	53.9	34.9 ± 0.0	34.9 ± 0.0	75.6	58.6	70.5 ± 0.0	50.8 ± 0.0
Sync [19]	54.1	**55.6**	**56.4** ± 0.9	**54.8** ± 0.6	71.2	46.6	65.6 ± 0.8	58.1 ± 0.8
SAE [21]	33.4	33.3	56.2 ± 0.0	53.3 ± 0.0	**80.7**	54.1	81.1 ± 0.0	62.8 ± 0.0

4.2.4 Metrics

For ZSL, we evaluate the accuracy of samples from unseen classes among unseen classes $A_{\mathcal{U}\to\mathcal{C}_u}$. There are two possible ways to define accuracy: most of the literature uses *per sample* accuracy, defined as $100 \cdot \frac{1}{N^{te}} \sum_{n=1}^{N^{te}} \mathbb{1}[\hat{y}(\mathbf{x}_n^{te}) = y_n^{te}]$, while in [4] it is argued that *per class* accuracy, defined as $100 \cdot \frac{1}{|\mathcal{C}^{te}|} \sum_{c\in\mathcal{C}^{te}} \frac{1}{|\{n|y_n=c\}|} \sum_{y_n=c}^{n} \mathbb{1}[\hat{y}(\mathbf{x}_n^{te}) = y_n^{te}]$, better takes class imbalance into account.

We report per class accuracy for fair comparison with the extensive results of [4]. Nonetheless, to enable comparison with the rest of the literature, we also provide per sample accuracy results in Table 3. For GZSL we compute the harmonic mean between $A_{\mathcal{U}\to\mathcal{C}}$ and $A_{\mathcal{S}\to\mathcal{C}}$, defined as $\frac{2\cdot A_{\mathcal{U}\to\mathcal{C}}\cdot A_{\mathcal{S}\to\mathcal{C}}}{A_{\mathcal{U}\to\mathcal{C}}+A_{\mathcal{S}\to\mathcal{C}}}$.

Accuracy is again assumed to be per class unless otherwise stated.

4.3 Results

We first evaluate the performances of the different methods in a classical ZSL setting. Table 1 shows the average per class accuracy measured on testing sets of the Standard Splits (S.S.) and the Proposed Splits (P.S.) [4] of CUB [6] and AwA2 [7]. We report the average score and the standard deviation over 5 runs with different random initializations. We also report the results from [4]. We see that some methods such as SJE [18] have high variability with respect to the initialization; for such methods, it is good practice to report average results since a single test run may not be representative of the true performance of the model. On the other hand, methods with closed-form or deterministic solutions such as the Linear$_{\mathcal{V}\to\mathcal{S}}$, Linear$_{\mathcal{S}\to\mathcal{V}}$, ESZSL [20] or SAE [21] are not dependent on the initialization and thus have a standard deviation of 0.

Most of the reproduced scores are consistent with [4], with two notable exceptions: first, a significant increase in performance is observed with SAE [21] and can be explained by the fact that similarities are computed in the visual

Table 2: GZSL score (harmonic mean of $A_{\mathcal{U}\to\mathcal{C}}$ and $A_{\mathcal{S}\to\mathcal{C}}$, per class accuracy) with and without calibration and GZSL regularization. On Proposed Split [4], averaged over 5 runs.

Method	**CUB** [6]				**AwA2** [7]			
	Reported in [4]	**Ours**			**Reported in** [4]	**Ours**		
with **calibration**	-	-	✓	✓	-	-	✓	✓
with $\boldsymbol{\lambda^*_{GZSL}}$	-	-	-	✓	-	-	-	✓
$\text{Linear}_{\mathcal{V}\to\mathcal{S}}$	*n/a*	18.2	34.3	35.5	*n/a*	8.3	47.3	48.1
$\text{Linear}_{\mathcal{S}\to\mathcal{V}}$	*n/a*	32.5	41.9	43.5	*n/a*	44.3	62.7	64.0
ALE [16]	34.4	35.6	45.1	46.2	23.9	26.9	55.8	55.8
DeViSE [17]	32.8	35.1	43.6	43.4	27.8	17.4	54.6	54.6
SJE [18]	33.6	29.7	41.2	44.2	14.4	28.9	58.2	59.0
ESZSL [20]	21.0	17.9	33.7	33.9	11.0	39.9	53.6	53.7
Sync [19]	19.8	33.2	46.2	47.6	18.0	30.6	61.0	61.0
SAE [21]	13.6	25.7	43.1	43.1	2.2	29.5	60.2	60.2
Average	25.9	28.5	41.1	**42.2**	16.2	28.2	56.7	**57.1**

Table 3: ZSL and GZSL scores with 10-crop features, evaluated with per class (p.c.) and per sample (p.s.) accuracies. With calibration and λ^*_{GZSL}. On P.S. [4], averaged over 5 runs.

Method	**CUB** [6]				**AwA2** [7]			
	ZSL		**GZSL**		**ZSL**		**GZSL**	
	Acc. p.c.	**Acc. p.s.**	**H. p.c.**	**H. p.s.**	**Acc. p.c.**	**Acc. p.s.**	**H. p.c.**	**H. p.s.**
$\text{Linear}_{\mathcal{V}\to\mathcal{S}}$	45.6	45.6	39.8	39.8	51.0	43.6	49.0	45.6
$\text{Linear}_{\mathcal{S}\to\mathcal{V}}$	57.1	57.2	47.7	48.0	**70.4**	**69.3**	**65.1**	**68.7**
ALE [16]	57.4	57.5	**49.2**	**49.3**	63.0	61.1	56.9	55.5
DeViSE [17]	52.9	52.9	42.4	42.5	63.1	62.2	55.0	50.6
SJE [18]	51.9	52.1	46.7	46.9	63.8	61.6	59.4	57.6
ESZSL [20]	39.0	38.8	38.7	38.6	52.6	51.9	54.4	57.9
Sync [19]	**57.5**	**57.6**	48.9	49.1	59.3	56.1	62.6	63.2
SAE [21]	56.1	56.2	46.3	46.6	63.5	65.4	62.3	63.6

space, with results close to those of the $\text{Linear}_{\mathcal{S}\to\mathcal{V}}$ model (results are close to those of $\text{Linear}_{\mathcal{V}\to\mathcal{S}}$ when similarities are computed in the semantic space). Second, the score for ESZSL [20] is significantly lower than reported in [4]. We found that the use of non-normalized attributes enables to reach performances comparable with [4], but we could not reproduce the reported results for ES-ZSL [20] with normalized attributes. For the sake of consistency, we chose to report results obtained with normalized attributes.

Table 2 shows results for GZSL. We measure the harmonic mean between per class accuracies $A_{\mathcal{U}\to\mathcal{C}}$ and $A_{\mathcal{S}\to\mathcal{C}}$ on the testing set of the Proposed Split [4]. We evaluate three settings: a ZSL model applied directly in a GZSL setting, i.e. with no calibration and a regularization specific to ZSL (λ^*_{ZSL}) as opposed to GZSL (λ^*_{GZSL}); a ZSL model with calibration and ZSL regularization λ^*_{ZSL}; and a ZSL model with calibration and regularization λ^*_{GZSL} specific to the

GZSL problem. We report the average score over 5 runs; standard deviations are available in the supplementary material. We also report the results from [4], which correspond to the setting with no calibration and no λ^*_{GZSL}. We can see that the calibration process significantly improves GZSL performance: in our experiments, the average score for all models improves from 28.5 with no calibration to 41.1 with calibration on CUB, and from 28.2 to 56.7 on AwA2. It is worth noting that the lowest score with calibration is close to or higher than the highest score without. The use of a regularization parameter specific to the GZSL task can lead to an additional improvement in performance. In some cases, the optimal λ is the same for the ZSL task and the GZSL task on the validation set, leading to no additional improvement over the score with calibration. However, every time they are different, λ^*_{GZSL} is smaller than λ^*_{ZSL}, as expected from the results in Sec. 3.3. The only exception is with DeViSE [17] on CUB: a λ^*_{GZSL} higher than λ^*_{ZSL} was selected during cross-validation, probably due to random noise, resulting in a slightly lower final GZSL score.

Table 3 shows results with improved visual features; each original 256×256 image is cropped into ten 224×224 images: one in each corner and one in the center for both the original image and its horizontal symmetry. The ResNet features of the resulting images are averaged to obtain a 2048-dimensional vector. We report results for ZSL ($A_{\mathcal{C}\rightarrow\mathcal{C}_u}$, abbreviated *Acc.*) and GZSL (using the harmonic mean metric, abbreviated *H.*) on the testing set of the Proposed Split [4]. In order to facilitate fair comparison with the rest of the literature, both per class *(p.c.)* and per sample *(p.s.)* metrics are reported. Additional metrics like AUSUC[5] are available in the supplementary material. Results with 10-cropped visual features are almost always better than the results with standard visual features in Table 2. The per sample metrics are on average not very different from the per class metrics. This is not surprising since classes in both CUB and AwA2 are fairly balanced.

5 Conclusion

We proposed a simple process for applying ZSL methods in a GZSL setting. This process is based on the empirical observation that ZSL models perform differently on samples from seen and unseen classes. We provided insights about why this should be expected and suggested steps to overcome these problems. Through extensive experiments, we showed that this process enables significant improvements in performance for many existing ZSL methods. Finally, we provided results under optimal conditions for these methods with different metrics to support fair comparison with the rest of the state-of-the-art.

References

[1] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. CVPR 2009*, pp. 951–958, IEEE, 2009.

[2] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks," in *AAAI*, vol. 1, p. 3, 2008.

[3] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Proc. NIPS 2009*, pp. 1410–1418, 2009.

[4] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning – the good, the bad and the ugly," in *Proc. CVPR 2017*, pp. 3077–3086, IEEE, 2017.

[5] 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 *Proc. ECCV 2016*, pp. 52–68, Springer, 2016.

[6] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 dataset," 2011.

[7] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning – a comprehensive evaluation of the good, the bad and the ugly," *arXiv preprint arXiv:1707.00600*, 2017.

[8] M. Bucher, S. Herbin, and F. Jurie, "Generating visual representations for zero-shot classification," in *ICCV Workshops: TASK-CV*, IEEE, 2017.

[9] V. Kumar Verma, G. Arora, A. Mishra, and P. Rai, "Generalized zero-shot learning via synthesized examples," in *Proc. CVPR 2010*, pp. 4281–4289, IEEE, 2018.

[10] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proc. CVPR 2018*, IEEE, 2018.

[11] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Transductive multi-view zero-shot learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2332–2345, 2015.

[12] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *Proc. CVPR 2015*, pp. 2452–2460, IEEE, 2015.

[13] M. Rohrbach, S. Ebert, and B. Schiele, "Transfer learning in a transductive setting," in *Proc. NIPS 2013*, pp. 46–54, 2013.

[14] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[15] W. N. van Wieringen, "Lecture notes on ridge regression," *arXiv preprint arXiv:1509.09169*, 2015.

[16] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1425–1438, 2016.

[17] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, *et al.*, "Devise: A deep visual-semantic embedding model," in *Proc. NIPS 2013*, pp. 2121–2129, 2013.

[18] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Proc. CVPR 2015*, pp. 2927–2936, IEEE, 2015.

[19] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *Proc. CVPR 2016*, pp. 5327–5336, IEEE, 2016.

[20] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proc. ICML 2015*, pp. 2152–2161, 2015.

[21] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *Proc. CVPR 2017*, pp. 4447–4456, IEEE, 2017.

[22] Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, and Y. Matsumoto, "Ridge regression, hubness, and zero-shot learning," in *Proc. ECML PKDD 2015*, pp. 135–151, Springer, 2015.

[23] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Hubs in space: Popular nearest neighbors in high-dimensional data," *Journal of Machine Learning Research*, vol. 11, no. Sep, pp. 2487–2531, 2010.

[24] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, 2014.

[25] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *Proc. CVPR 2012*, pp. 2751–2758, IEEE, 2012.

[26] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, p. Proc. CVPR 2009, IEEE, 2009.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR 2009*, pp. 248–255, IEEE, 2009.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. CVPR 2015*, pp. 1–9, IEEE, 2015.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR 2016*, pp. 770–778, IEEE, 2016.

6 Supplementary Material – From Classical to Generalized Zero-Shot Learning: a Simple Adaptation Process

6.1 Details regarding the compared methods

ALE [16], DeViSE [17] and SJE [18] employ bilinear similarity functions of the form $f(\mathbf{x};\mathbf{s}) = \mathbf{s}^\top \mathbf{W}\mathbf{x}$ with $\mathbf{W} \in \mathbb{R}^{K\times D}$ combined with a hinge rank loss of the form $\max(0, M + f(\mathbf{x}_n;\mathbf{s}_c) - f(\mathbf{x}_n;\mathbf{s}_{y_n}))$ for $c \neq y_n$, where M is a margin usually set to a fraction of the expected value of $f(\mathbf{x};\mathbf{s})$. For a given visual sample $\mathbf{x}_n$, DeViSE simply sums the hinge range losses for all values of $c \in \mathcal{C}_s$ such that $c \neq y_n$. ALE sums these terms and adds a multiplicative factor to lessen the weight of samples for which many classes contribute to the hinge rank loss. SJE only keeps the loss of the class c corresponding to the maximum of the hinge loss. We add a regularization term of the form $\lambda\|\mathbf{W}\|_F^2$ to all these models, $\|\cdot\|_F$ being the Frobenius norm.

Sync [19] uses *"phantom" objects* located in both the visual and semantic spaces to generate classifiers for unseen classes. We employ the *structured loss* described in Sec. 3.2 of [19] as it yields better results. ESZSL [20] is a simple linear model regularized with respect to the projected visual features of class prototypes, the projected semantic features of the visual samples and the projection matrix. One of its advantages is the existence of a closed form solution. SAE [21] learns an autoencoder on visual features aiming to balance a reconstruction loss and a representation loss so that its internal representation corresponds to the semantic attributes of visual features.

6.2 Details regarding datasets and splits

In addition to CUB [6] and AwA2 [7], we provide results for two other datasets, the SUN Attribute Database (SUN) [25] and Attributes Pascal and Yahoo (aPY) [26]. SUN is a fine-grained dataset comprising 717 categories of scenes (*abbey*, ..., *zoo*) with 20 images per category, for a total of 14340 images. It comes with 102-dimensional vectors of attributes for each picture; they are averaged by class to obtain semantic class prototypes. aPY is a coarse-grained dataset composed of 15339 images from 3 broad categories (animals, objects and vehicles), further divided into a total of 32 subcategories (*aeroplane*, ..., *zebra*). It is rather imbalanced, with the category *person* representing a third of all samples. Moreover, with 21 out of 32 classes present in the ImageNet [27] dataset, it is also highly biased when used in a ZSL context with deep networks pre-trained on ImageNet [27] as visual features extractors. [4] also provides Proposed Splits in addition to the Standard Splits for these two other datasets; we use the same protocol as with CUB [6] and AwA2 [7], described in Sec. 3.2 of the main text, to further divide these splits for GZSL.

We thus have a total of four datasets, each with two different testing sets, with three different validation sets for each testing set. Statistics regarding

Table 4: Statistics of the dataset splits for each of the four datasets, with two testing splits (Standard Splits or S.S. and Proposed Splits or P.S. as defined in [4]) per dataset and three validation splits per testing split. Each cell contains a number of samples on the left and the number of corresponding classes on the right.

Split	Total	Training			Validation		Testing
		Train	Seen val	Seen test	Val	Seen test	Test
CUB [6]							
S.S. val1	11788 / *200*	3773 / *100*	943 / *100*	1178 / *100*	2369 / *50*	592 / *50*	2933 / *50*
S.S. val2		3798 / *100*	949 / *100*	1186 / *100*	2338 / *50*	584 / *50*	
S.S. val3		3784 / *100*	946 / *100*	1182 / *100*	2355 / *50*	588 / *50*	
P.S. val1		3760 / *100*	940 / *100*	1175 / *100*	2357 / *50*	589 / *50*	2967 / *50*
P.S. val2		3760 / *100*	940 / *100*	1175 / *100*	2357 / *50*	589 / *50*	
P.S. val3		3806 / *100*	951 / *100*	1189 / *100*	2300 / *50*	575 / *50*	
AwA2 [7]							
S.S. val1	37322 / *50*	13284 / *27*	3321 / *27*	4151 / *27*	7665 / *13*	1916 / *13*	6985 / *10*
S.S. val2		13041 / *27*	3260 / *27*	4075 / *27*	7969 / *13*	1992 / *13*	
S.S. val3		13908 / *27*	3476 / *27*	4345 / *27*	6887 / *13*	1721 / *13*	
P.S. val1		12940 / *27*	3235 / *27*	4043 / *27*	7353 / *13*	1838 / *13*	7913 / *10*
P.S. val2		13404 / *27*	3351 / *27*	4188 / *27*	6773 / *13*	1693 / *13*	
P.S. val3		12965 / *27*	3241 / *27*	4051 / *27*	7322 / *13*	1830 / *13*	
SUN [25]							
S.S. val1	14340 / *717*	7437 / *581*	1859 / *581*	2324 / *581*	1024 / *64*	256 / *64*	1440 / *72*
S.S. val2		7437 / *581*	1859 / *581*	2324 / *581*	1024 / *64*	256 / *64*	
S.S. val3		7437 / *581*	1859 / *581*	2324 / *581*	1024 / *64*	256 / *64*	
P.S. val1		7424 / *580*	1856 / *580*	2320 / *580*	1040 / *65*	260 / *65*	1440 / *72*
P.S. val2		7424 / *580*	1856 / *580*	2320 / *580*	1040 / *65*	260 / *65*	
P.S. val3		7424 / *580*	1856 / *580*	2320 / *580*	1040 / *65*	260 / *65*	
aPY [26]							
S.S. val1	15339 / *32*	6424 / *13*	1606 / *13*	2007 / *13*	2127 / *7*	531 / *7*	2644 / *12*
S.S. val2		6608 / *13*	1652 / *13*	2064 / *13*	1897 / *7*	474 / *7*	
S.S. val3		6644 / *13*	1660 / *13*	2076 / *13*	1852 / *7*	463 / *7*	
P.S. val1		3896 / *15*	973 / *15*	1217 / *15*	1064 / *5*	265 / *5*	7924 / *12*
P.S. val2		3716 / *15*	928 / *15*	1161 / *15*	1288 / *5*	322 / *5*	
P.S. val3		3271 / *15*	817 / *15*	1021 / *15*	1845 / *5*	461 / *5*	

these splits are available in Table 4. For a given dataset and testing set, the average score over the three different validation splits is employed for selecting hyperparameters by cross-validation.

Intra-class and inter-class variances for each dataset are reported in Table 5.

6.3 Additional results

Table 6 is an extended version of Table 1 in the main text. It reports the per class accuracy $A_{\mathcal{U}\to\mathcal{C}_u}$ measured on the Standard Splits (S.S.) and the Proposed Splits (P.S.) [4] of CUB [6] and AwA2 [7], with the addition of the two datasets SUN [25] and aPY [26]. Results are averaged over five runs with different random initializations.

Results on SUN [25] are mostly consistent with what has been observed on CUB [6] and AwA2 [7]. The addition of a regularization parameter to ALE [16], DeViSE [17] and SJE [18] enables a slight increase in the ZSL score. The $\text{Linear}_{\mathcal{S}\to\mathcal{V}}$ model reaches performances close to or even better than the state-

Table 5: Intra-class and inter-class variance for each dataset.

Dataset	Intra-class variance	Inter-class variance
CUB [6]	138.0	231.9
AwA2 [7]	226.4	379.0
SUN [25]	239.9	397.3
aPY [26]	262.4	370.5

of-the-art. A significant increase in performance is observed with SAE [21] and can be explained by the fact that similarities are computed in the visual space, with results close to those of the Linear$_{\mathcal{S} \rightarrow \mathcal{V}}$ model (results are close to those of Linear$_{\mathcal{V} \rightarrow \mathcal{S}}$ when similarities are computed in the semantic space). The score for ESZSL [20] is significantly lower than reported in [4]. We found that the use of non-normalized attributes enables to reach performances comparable with [4], but we could not reproduce the reported results for ESZSL [20] with normalized attributes. For the sake of consistency, we chose to report results obtained with normalized attributes.

Results on aPY [26] are more mixed: important differences between reported and reproduced scores exist on at least one split for most methods. This can be explained by the nature of this dataset: the low number of classes, the class imbalance and the high bias due to the presence of most of its classes in ImageNet [27] imply that the validation sets can be significantly different from the test sets. For example, the average validation score on the three validation splits is 55.2 for DeViSE [17], but this score drops to an average of 29.0 on the testing set. It is therefore difficult to find relevant hyperparameters.

Table 7 is an extended version of Table 2 in the main text. It reports the average GZSL score, as measured by the harmonic mean between $A_{\mathcal{U} \rightarrow \mathcal{C}}$ and $A_{\mathcal{S} \rightarrow \mathcal{C}}$, for the evaluated methods on the four datasets with and without our process. The results on SUN [25] are again mostly consistent with what has been observed on CUB [6] and AwA2 [7], while results on aPY [26] differ due to the nature of the dataset. In particular, due to the difficulty in selecting good hyperparameters on this dataset, the regularization parameter selected for the GZSL task is not always relevant and does not lead to an increase in the final score.

Table 8 is an extended version of Table 3 in the main text. It reports results for the ZSL and the GZSL tasks when visual features are improved with 10-crop. The AUSUC metric [5] has been added for GZSL; the accuracies used for computing the Seen-Unseen Curve are the per sample accuracies as is usually the case. Visual features obtained with 10-crop almost always lead to better results on CUB [6], AwA2 [7] and SUN [25], while again no clear pattern is visible for aPY [26]. It is worth noting that although the harmonic mean score and the AUSUC score are visibly correlated, a better AUSUC does not necessarily imply a better harmonic mean. For example, among the bilinear compatibility models ALE [16], DeViSE [17] and SJE [18] on the AwA2 [7] dataset, DeViSE has the highest AUSUC score with 0.538 (compared to respectively 0.512 and

0.529 for ALE and SJE) but the lowest harmonic mean score with 55.0 (per class accuracy, compared to respectively 56.9 and 59.4 for ALE and SJE).

Table 6: Previously reported and independently reproduced ZSL per-class accuracy. S.S.: Standard Split, P.S.: Proposed Split [4]. Averaged over 5 runs.

Method	**Reported in** [4]		**Reproduced**	
	S.S.	**P.S.**	**S.S.**	**P.S.**
CUB [6]				
$\text{Linear}_{\mathcal{V} \to \mathcal{S}}$	*n/a*	*n/a*	41.0 ± 0.0	41.8 ± 0.0
$\text{Linear}_{\mathcal{S} \to \mathcal{V}}$	*n/a*	*n/a*	56.0 ± 0.0	53.5 ± 0.0
ALE [16]	53.2	54.9	54.8 ± 0.8	54.0 ± 1.2
DeViSE [17]	53.2	52.0	52.5 ± 0.9	52.6 ± 1.3
SJE [18]	**55.3**	53.9	53.8 ± 2.3	49.2 ± 1.4
ESZSL [20]	55.1	53.9	34.9 ± 0.0	34.9 ± 0.0
Sync [19]	54.1	**55.6**	**56.4** ± 0.9	**54.8** ± 0.6
SAE [21]	33.4	33.3	56.2 ± 0.0	53.3 ± 0.0
AwA2 [7]				
$\text{Linear}_{\mathcal{V} \to \mathcal{S}}$	*n/a*	*n/a*	68.2 ± 0.0	49.7 ± 0.0
$\text{Linear}_{\mathcal{S} \to \mathcal{V}}$	*n/a*	*n/a*	**85.5** ± 0.0	**68.9** ± 0.0
ALE [16]	80.3	**62.5**	80.3 ± 2.2	62.9 ± 2.3
DeViSE [17]	68.6	59.7	76.6 ± 1.6	62.1 ± 1.6
SJE [18]	69.5	61.9	80.4 ± 2.9	62.2 ± 1.2
ESZSL [20]	75.6	58.6	70.5 ± 0.0	50.8 ± 0.0
Sync [19]	71.2	46.6	65.6 ± 0.8	58.1 ± 0.8
SAE [21]	**80.7**	54.1	81.1 ± 0.0	62.8 ± 0.0
SUN [25]				
$\text{Linear}_{\mathcal{V} \to \mathcal{S}}$	*n/a*	*n/a*	50.8 ± 0.0	46.7 ± 0.0
$\text{Linear}_{\mathcal{S} \to \mathcal{V}}$	*n/a*	*n/a*	62.8 ± 0.0	**61.5** ± 0.0
ALE [16]	59.1	**58.1**	63.5 ± 0.5	59.4 ± 0.3
DeViSE [17]	57.5	56.5	61.6 ± 0.3	58.4 ± 0.3
SJE [18]	57.1	53.7	59.9 ± 0.6	55.3 ± 1.1
ESZSL [20]	57.3	54.5	21.0 ± 0.0	16.0 ± 0.0
Sync [19]	**59.1**	56.3	58.5 ± 0.5	56.6 ± 1.8
SAE [21]	42.4	40.3	**63.7** ± 0.0	61.0 ± 0.0
aPY [26]				
$\text{Linear}_{\mathcal{V} \to \mathcal{S}}$	*n/a*	*n/a*	30.0 ± 0.0	31.6 ± 0.0
$\text{Linear}_{\mathcal{S} \to \mathcal{V}}$	*n/a*	*n/a*	**42.2** ± 0.0	**40.0** ± 0.0
ALE [16]	30.9	39.7	16.0 ± 5.7	30.4 ± 0.9
DeViSE [17]	35.4	**39.8**	30.3 ± 3.2	29.0 ± 1.1
SJE [18]	32.0	32.9	18.9 ± 2.7	33.5 ± 1.9
ESZSL [20]	34.4	38.3	31.6 ± 0.0	16.3 ± 0.0
Sync [19]	**39.7**	23.9	34.1 ± 0.5	35.5 ± 0.1
SAE [21]	8.3	8.3	31.7 ± 0.0	29.0 ± 0.0

Table 7: GZSL score (harmonic mean of $A_{\mathcal{U}\to\mathcal{C}}$ and $A_{\mathcal{S}\to\mathcal{C}}$, per class accuracy) with and without calibration and GZSL regularization. On Proposed Split [4], averaged over 5 runs.

Method	**Reported in** [4]	**Ours**		
with **calibration**	-	-	✓	✓
with $\boldsymbol{\lambda^*_{\mathbf{GZSL}}}$	-	-	-	✓
CUB [6]				
Linear$_{\mathcal{V}\to\mathcal{S}}$	*n/a*	18.2 ± 0.0	34.3 ± 0.0	35.5 ± 0.0
Linear$_{\mathcal{S}\to\mathcal{V}}$	*n/a*	32.5 ± 0.0	41.9 ± 0.0	43.5 ± 0.0
ALE [16]	34.4	**35.6** ± 1.3	45.1 ± 1.3	46.2 ± 1.0
DeViSE [17]	32.8	35.1 ± 0.6	43.6 ± 1.1	43.4 ± 0.7
SJE [18]	33.6	29.7 ± 1.7	41.2 ± 0.6	44.2 ± 1.0
ESZSL [20]	21.0	17.9 ± 0.0	33.7 ± 0.0	33.9 ± 0.0
Sync [19]	19.8	33.2 ± 1.0	**46.2** ± 0.8	**47.6** ± 0.2
SAE [21]	13.6	25.7 ± 0.0	43.1 ± 0.0	43.1 ± 0.0
Average (CUB)	25.9	28.5	41.1	**42.2**
AwA2 [7]				
Linear$_{\mathcal{V}\to\mathcal{S}}$	*n/a*	8.3 ± 0.0	47.3 ± 0.0	48.1 ± 0.0
Linear$_{\mathcal{S}\to\mathcal{V}}$	*n/a*	**44.3** ± 0.0	**62.7** ± 0.0	**64.0** ± 0.0
ALE [16]	23.9	26.9 ± 1.5	55.8 ± 1.7	55.8 ± 1.7
DeViSE [17]	27.8	17.4 ± 4.1	54.6 ± 1.2	54.6 ± 1.2
SJE [18]	14.4	28.9 ± 2.3	58.2 ± 0.9	59.0 ± 1.9
ESZSL [20]	11.0	39.9 ± 0.0	53.6 ± 0.0	53.7 ± 0.0
Sync [19]	18.0	30.6 ± 0.3	61.0 ± 0.5	61.0 ± 0.5
SAE [21]	2.2	29.5 ± 0.0	60.2 ± 0.0	60.2 ± 0.0
Average (AwA2)	16.2	28.2	56.7	**57.1**
SUN [25]				
Linear$_{\mathcal{V}\to\mathcal{S}}$	*n/a*	16.3 ± 0.0	23.3 ± 0.0	25.2 ± 0.0
Linear$_{\mathcal{S}\to\mathcal{V}}$	*n/a*	24.4 ± 0.0	34.2 ± 0.0	34.8 ± 0.0
ALE [16]	26.3	**26.9** ± 0.3	33.6 ± 0.2	33.6 ± 0.2
DeViSE [17]	20.9	25.2 ± 0.2	31.4 ± 0.4	31.4 ± 0.4
SJE [18]	19.8	25.5 ± 0.6	**34.9** ± 0.4	**35.3** ± 0.4
ESZSL [20]	15.8	6.7 ± 0.0	11.1 ± 0.0	11.1 ± 0.0
Sync [19]	13.4	20.6 ± 0.8	27.9 ± 1.0	27.9 ± 1.0
SAE [21]	11.8	25.1 ± 0.0	34.0 ± 0.0	34.7 ± 0.0
Average (SUN)	18.0	21.3	28.8	**29.2**
aPY [26]				
Linear$_{\mathcal{V}\to\mathcal{S}}$	*n/a*	4.0 ± 0.0	32.7 ± 0.0	32.7 ± 0.0
Linear$_{\mathcal{S}\to\mathcal{V}}$	*n/a*	**21.5** ± 0.0	**39.1** ± 0.0	**39.2** ± 0.0
ALE [16]	8.7	19.4 ± 0.5	30.1 ± 0.9	29.6 ± 1.1
DeViSE [17]	9.2	20.7 ± 0.7	28.0 ± 1.0	27.4 ± 1.1
SJE [18]	6.9	16.1 ± 1.3	36.3 ± 1.2	36.3 ± 1.2
ESZSL [20]	4.6	13.3 ± 0.0	20.6 ± 0.0	20.4 ± 0.0
Sync [19]	***13.3***	13.8 ± 0.1	36.4 ± 0.1	36.4 ± 0.1
SAE [21]	0.9	16.7 ± 0.0	33.8 ± 0.0	32.1 ± 0.0
Average (aPY)	7.3	15.7	**32.1**	31.8

Table 8: ZSL and GZSL scores with 10-crop features, evaluated with per class (p.c.) and per sample (p.s.) accuracies. With calibration and λ^*_{GZSL}. On P.S. [4], averaged over 5 runs.

Method	**ZSL**		**GZSL**		
	Acc. p.c.	**Acc. p.s.**	**H. p.c.**	**H. p.s.**	**AUSUC**
CUB [6]					
Linear$_{\mathcal{V}\to\mathcal{S}}$	45.6 ± 0.0	45.6 ± 0.0	39.8 ± 0.0	39.8 ± 0.0	0.246 ± 0.000
Linear$_{\mathcal{S}\to\mathcal{V}}$	57.1 ± 0.0	57.2 ± 0.0	47.7 ± 0.0	48.0 ± 0.0	0.315 ± 0.000
ALE [16]	57.4 ± 0.3	57.5 ± 0.3	**49.2** ± 0.6	**49.3** ± 0.6	**0.347** ± 0.005
DeViSE [17]	52.9 ± 1.0	52.9 ± 1.0	42.4 ± 0.8	42.5 ± 0.7	0.270 ± 0.007
SJE [18]	51.9 ± 1.7	52.1 ± 1.7	46.7 ± 0.9	46.9 ± 1.0	0.308 ± 0.009
ESZSL [20]	39.0 ± 0.0	38.8 ± 0.0	38.7 ± 0.0	38.6 ± 0.0	0.224 ± 0.000
Sync [19]	**57.5** ± 1.8	**57.6** ± 1.8	48.9 ± 1.2	49.1 ± 1.2	0.337 ± 0.018
SAE [21]	56.1 ± 0.0	56.2 ± 0.0	46.3 ± 0.0	46.6 ± 0.0	0.307 ± 0.000
AwA2 [7]					
Linear$_{\mathcal{V}\to\mathcal{S}}$	51.0 ± 0.0	43.6 ± 0.0	49.0 ± 0.0	45.5 ± 0.0	0.357 ± 0.000
Linear$_{\mathcal{S}\to\mathcal{V}}$	**70.4** ± 0.0	**69.3** ± 0.0	**65.1** ± 0.0	**68.7** ± 0.0	**0.598** ± 0.000
ALE [16]	63.0 ± 1.8	61.1 ± 1.7	56.9 ± 1.8	55.5 ± 2.2	0.521 ± 0.014
DeViSE [17]	63.1 ± 1.8	62.2 ± 0.6	55.0 ± 1.7	50.6 ± 0.8	0.538 ± 0.005
SJE [18]	63.8 ± 2.0	61.6 ± 2.8	59.4 ± 1.2	57.6 ± 1.8	0.529 ± 0.022
ESZSL [20]	52.6 ± 0.0	51.9 ± 0.0	54.4 ± 0.0	57.9 ± 0.0	0.434 ± 0.000
Sync [19]	59.3 ± 0.2	56.1 ± 0.4	62.6 ± 0.1	63.2 ± 0.2	0.511 ± 0.003
SAE [21]	63.5 ± 0.0	65.4 ± 0.0	62.3 ± 0.0	63.6 ± 0.0	0.585 ± 0.000
SUN [25]					
Linear$_{\mathcal{V}\to\mathcal{S}}$	48.6 ± 0.0	48.6 ± 0.0	26.7 ± 0.0	25.8 ± 0.0	0.107 ± 0.000
Linear$_{\mathcal{S}\to\mathcal{V}}$	62.0 ± 0.0	62.0 ± 0.0	36.3 ± 0.0	35.4 ± 0.0	0.184 ± 0.000
ALE [16]	62.2 ± 0.2	62.2 ± 0.2	34.9 ± 0.2	34.4 ± 0.3	0.183 ± 0.002
DeViSE [17]	61.2 ± 0.3	61.2 ± 0.3	32.5 ± 0.4	32.1 ± 0.4	0.167 ± 0.003
SJE [18]	58.2 ± 0.8	58.4 ± 0.8	**36.8** ± 0.6	**36.0** ± 0.4	**0.194** ± 0.002
ESZSL [20]	17.2 ± 0.0	17.2 ± 0.0	11.8 ± 0.0	11.6 ± 0.0	0.169 ± 0.000
Sync [19]	56.6 ± 1.8	56.6 ± 1.8	27.9 ± 1.0	27.2 ± 1.1	0.123 ± 0.008
SAE [21]	**62.7** ± 0.0	**62.7** ± 0.0	35.9 ± 0.0	35.1 ± 0.0	0.185 ± 0.000
aPY [26]					
Linear$_{\mathcal{V}\to\mathcal{S}}$	28.6 ± 0.0	12.5 ± 0.0	33.3 ± 0.0	16.2 ± 0.0	0.081 ± 0.000
Linear$_{\mathcal{S}\to\mathcal{V}}$	**40.6** ± 0.0	**23.9** ± 0.0	**38.4** ± 0.0	21.0 ± 0.0	0.098 ± 0.000
ALE [16]	31.3 ± 1.0	16.8 ± 2.7	31.5 ± 0.9	21.1 ± 1.0	0.101 ± 0.005
DeViSE [17]	30.9 ± 0.9	16.4 ± 1.6	28.7 ± 1.2	19.8 ± 0.8	0.096 ± 0.005
SJE [18]	34.1 ± 0.8	16.4 ± 1.6	36.9 ± 0.7	**23.0** ± 1.2	**0.110** ± 0.009
ESZSL [20]	12.2 ± 0.0	7.2 ± 0.0	17.1 ± 0.0	11.6 ± 0.0	0.077 ± 0.000
Sync [19]	34.9 ± 0.2	16.8 ± 0.2	37.4 ± 0.1	19.4 ± 0.1	0.102 ± 0.001
SAE [21]	29.5 ± 0.0	11.1 ± 0.0	32.7 ± 0.0	13.0 ± 0.0	0.058 ± 0.000