Bias-Awareness for Zero-Shot Learning the Seen and Unseen

William Thong
w.e.thong@uva.nl
Cees G. M. Snoek
cgmsnoek@uva.nl

University of Amsterdam
Science Park 904
Amsterdam
The Netherlands

Abstract

Generalized zero-shot learning recognizes inputs from both seen and unseen classes. Yet, existing methods tend to be biased towards the classes seen during training. In this paper, we strive to mitigate this bias. We propose a bias-aware learner to map inputs to a semantic embedding space for generalized zero-shot learning. During training, the model learns to regress to real-valued class prototypes in the embedding space with temperature scaling, while a margin-based bidirectional entropy term regularizes seen and unseen probabilities. Relying on a real-valued semantic embedding space provides a versatile approach, as the model can operate on different types of semantic information for both seen and unseen classes. Experiments are carried out on four benchmarks for generalized zero-shot learning and demonstrate the benefits of the proposed bias-aware classifier, both as a stand-alone method or in combination with generated features.

1 Introduction

Zero-shot recognition [15, 23] considers if models trained on a given set of seen classes $\mathcal{S}$ can extrapolate to a distinct set of unseen classes $\mathcal{U}$. In generalized zero-shot learning [8, 38], we also want to remember the seen classes and evaluate over the union of the two sets of classes $\mathcal{T} = \mathcal{S} \cup \mathcal{U}$. Nevertheless, when evaluating existing models in the generalized scenario, the seminal work of Chao *et al*. [8] highlights that predictions tend to be biased towards the seen classes observed during training. In this paper, we consider the challenge of mitigating this inherent bias present in classifiers by proposing a bias-aware model.

An effective remedy to remove the bias towards seen classes is to calibrate their predictions during inference. Chao *et al*. [8] propose to reduce the scores for the seen classes, which in return improves the generalized zero-shot learning performance. Yet, the bias towards seen classes should also be tackled while training classifiers, and not only during the evaluation phase, to address the bias from the start. Towards this goal, seen and unseen classes can be addressed separately during training. Liu *et al*. [17] define two separate training objectives to calibrate the confidence of seen classes and the uncertainty of unseen classes. Atzmon and Chechik [4] break the classification into two separate experts, with one model for seen classes and another one for unseen classes. Their COSMO approach provides compelling results at the expense of a third additional expert to combine results. As generalized zero-shot

learning considers both seen and unseen classes simultaneously, learners should benefit from mitigating the bias in both directions by considering both sets jointly rather than separately.

The main objective of this paper is to mitigate the bias towards seen classes by considering predictions of seen and unseen classes simultaneously during training. To achieve this, we propose a simple bias-aware learner that maps inputs to a semantic embedding space where class prototypes are formed by real-valued representations. We address the bias by introducing (*i*) a calibration for the learner with temperature scaling, and (*ii*) a margin-based bidirectional entropy term to regularize seen and unseen probabilities jointly. We show that the bias towards seen classes is also dataset-dependent, and every dataset does not suffer to the same extent. Finally, we illustrate the versatility of our approach. By relying on a real-valued embedding space, the model can handle different types of prototype representation for both seen and unseen classes, and operate either on real features, akin to compatibility functions, or leverage generated unseen features. Comparisons on four datasets for generalized zero-shot learning show the effectiveness of bias-awareness. All source code and setups are released[1].

2 Related Work

Generalized zero-shot learning has been introduced to provide a more realistic and practical setting than zero-shot learning, as models are evaluated on both seen and unseen classes [8]. This change in evaluation has a large impact on existing compatibility functions designed for zero-shot learning, as they do not perform well in the generalized setting [7, 8, 38]. Indeed, whether they are based on a ranking loss [1, 2, 11, 27, 37] or synthesis [5, 6, 7], compatibility functions empirically exhibit a very low accuracy for unseen classes. As identified by Chao *et al*. [8], this indicates a strong inherent bias in all classifiers towards the seen classes. To overcome the low accuracy for unseen classes, both Kumar Verma *et al*. [14] and Xian *et al*. [39] learn a conditional generative model to generate image features. Once trained, image features of unseen classes are sampled by changing the conditioning of the generative models. Classification then consists of training a one-hot softmax classifier on both real and sampled image features. Having access during training to generated unseen features leads to an increase in unseen class accuracy. Among the different generative models used in generalized zero-shot learning are generative adversarial networks [10, 16, 39], variational autoencoders [14, 29] or a combination of both [40]. Still, a classifier trained on generated features suffers from a bias towards seen classes because generative models do not fully match the true distribution of unseen classes. In this paper, we strive for a bias-aware classifier, which can behave as a stand-alone model like compatibility functions and also leverage unseen features sampled from a generative model.

Addressing the bias in classifiers remains an open challenge for generalized zero-shot learning. Although Chao *et al*. [8] identify the critical bias towards seen classes, only a few works try to address it during training. Related works separate the seen and unseen classifications. Liu *et al*. [17] map both features and semantic representations to a common embedding space. Probabilities are then calibrated separately in this common space to make seen class probabilities confident and reduce the uncertainty of unseen class probabilities. Atzmon and Chechik [4] train expert models separately for seen and unseen class predictions. Their predictions are further combined in a soft manner with a third expert to produce the final decision. In this paper, we strive to address the bias by considering seen and unseen

[1] Source code is available at https://github.com/twuilliam/bias-gzsl

class probabilities jointly rather than separately. Having access during training to the joint class probabilities lets the bias-aware model learn how to balance them from the start.

3 Method

During training, a generalized zero-shot learner $G : X \rightarrow \mathcal{T}$ is given a training set $\mathcal{D}^{\mathcal{S}} = \{(x_n, y_n), y_n \in \mathcal{S}\}_{n=1}^{N}$, where $x_n \in \mathbb{R}^D$ is an image feature of dimension D and y_n comes from the set $\mathcal{S}$ of seen classes, with $\mathcal{S} \subset \mathcal{T}$. For each $c \in \mathcal{S}$ there exists a corresponding semantic class representation $\phi(c) \in \mathbb{R}^A$ of dimension A. At testing time, G predicts for each sample in the testing set $\mathcal{D}^{\mathcal{T}} = \{x_n\}_{n=1}^{M}$ a label that belongs to $\mathcal{T}$ by exploiting the joint set of seen and unseen semantic class representations. This problem formulation can be extended with an auxiliary dataset $\widetilde{\mathcal{D}}^{\mathcal{U}} = \{(\widetilde{x}_n, y_n), y_n \in \mathcal{U}\}_{n=1}^{\widetilde{N}}$, where y_n comes from the set of unseen classes $\mathcal{U}$. $\widetilde{\mathcal{D}}^{\mathcal{U}}$ mimics image features from unseen classes, and is typically sampled from a generative model. The joint set $\{\mathcal{D}^{\mathcal{S}}, \widetilde{\mathcal{D}}^{\mathcal{U}}\}$ now covers both seen and unseen classes.

In this paper, we propose a bias-aware generalized zero-shot learner $f(\cdot)$, which can operate during training with only $\mathcal{D}^{\mathcal{S}}$ similar to compatibility functions (Section 3.1) or the joint set $\{\mathcal{D}^{\mathcal{S}}, \widetilde{\mathcal{D}}^{\mathcal{U}}\}$ similar to classifiers in the generative approach (Section 3.2). In both scenarios, the learner includes mechanisms to mitigate the bias towards seen classes. Learning consists of mapping inputs x to their corresponding semantic class representations $\phi(c)$. In other words, the model regresses to a real-valued vector, which describes a class prototype. We denote the set of seen class prototypes as $\Phi^{\mathcal{S}} = \{\phi(c), c \in \mathcal{S}\}$, unseen class prototypes as $\Phi^{\mathcal{U}} = \{\phi(c), c \in \mathcal{U}\}$, and their union as $\Phi^{\mathcal{T}} = \Phi^{\mathcal{S}} \cup \Phi^{\mathcal{U}} = \{\phi(c), c \in \mathcal{T}\}$. Usually, the semantic knowledge used for class prototypes corresponds to semantic attributes [9, 15], word vectors of the class name [11, 23], hierarchical representations [1, 2, 37], or sentence descriptions [26, 39]. To exploit this diversity in semantic knowledge, we propose to swap the representation types for seen and unseen prototypes (Section 3.3).

3.1 Stand-alone classification with seen classes only

We design the bias-aware generalized zero-shot learner as a probabilistic model with two key principles. First, it is calibrated towards seen classes such that inputs from unseen classes yield a low confidence prediction at testing time. In return, this reduces the bias towards seen classes for unseen class inputs. Second, it maps inputs to class prototypes in the semantic embedding space. Following these two principles, we propose:

$$p(c|x, \mathcal{S}) = \exp\left(\frac{s(f(x), \phi(c))}{T}\right) \Big/ \sum_{c' \in \mathcal{S}} \exp\left(\frac{s(f(x), \phi(c'))}{T}\right), \tag{1}$$

where $s(\cdot, \cdot)$ is the cosine similarity and $T \in \mathbb{R}_{>0}$ is the temperature scale. When $T = 1$, it acts as the normal softmax function. When $T > 1$, probabilities are spreading out. When $T < 1$, probabilities tend to concentrate similar to a Dirac delta function. Contrary to knowledge distillation [13], we seek to concentrate the probabilities with a low temperature scale for discriminative purposes. Learning the probabilistic model is done via minimizing the cross-entropy loss function over the training set of seen examples $\mathcal{D}^{\mathcal{S}}$:

$$\mathcal{L}_{\mathrm{s}} = -\frac{1}{N} \sum_{n=1}^{N} \log p(y_n|x_n, \mathcal{S}). \tag{2}$$

This probabilistic model behaves like a compatibility function, because it only sees samples from seen classes during training. At testing, the evaluation simply measures the similarity in the embedding space with respect to the union of seen and unseen prototypes $\Phi^{\mathcal{T}}$.

Variants of this prototype-based learner have been proposed in image retrieval [18, 20, 35, 40] or image classification [17, 31, 36]. We differ by (*i*) fixing the prototypes to be semantic class representations rather than learning them; (*ii*) learning a mapping from the inputs to the class representations rather than learning a common embedding space; (*iii*) applying a softmax function to provide a probabilistic interpretation of cosine similarities; and (*iv*) calibrating the model with the same temperature scaling for both training and testing.

3.2 Classification with both seen and unseen classes

In the generative approach for generalized zero-shot learning, samples from unseen classes are generated. We can then use the generated data $\widetilde{\mathcal{D}}^{\mathcal{U}}$ as an auxiliary dataset for calibration and for entropy regularization. In this context, given an input x the probabilistic model learns to predict a class from the union of both seen and unseen classes:

$$p(c|x,\mathcal{T}) = \exp\left(\frac{s\left(f(x),\phi(c)\right)}{T}\right) \Big/ \sum_{c'\in\mathcal{T}} \exp\left(\frac{s\left(f(x),\phi(c')\right)}{T}\right). \tag{3}$$

The only and major difference with eq. 1 resides in the class prototypes that are considered to produce the prediction, while $f(\cdot)$ remains the same model. $p(c|x,\mathcal{S})$ only evaluates over the set of seen class prototypes $\Phi^{\mathcal{S}}$, while $p(c|x,\mathcal{T})$ evaluates over the union of seen and unseen class prototypes $\Phi^{\mathcal{T}}$. In this case, the temperature scaling ensures the model is confident for both seen and unseen classes. This difference also makes the learning distinctive from related works (*i.e.*, DCN [17] or COSMO [4]), as they consider seen and unseen classifications separately rather than jointly. Akin to eq. 2, we minimize the cross-entropy loss function on the joint set $\{\mathcal{D}^{\mathcal{S}}, \widetilde{\mathcal{D}}^{\mathcal{U}}\}$ of seen and unseen classes:

$$\mathcal{L}_{\text{s+u}} = -\frac{1}{N}\sum_{n=1}^{N} \log p(y_n|x_n,\mathcal{T}) - \frac{1}{\widetilde{N}}\sum_{n=1}^{\widetilde{N}} \log p(y_n|\widetilde{x}_n,\mathcal{T}). \tag{4}$$

This probabilistic model behaves like a classifier used in generative approaches, because it sees samples from both seen and unseen classes at both training and testing times, and the partition function normalizes over the union of seen and unseen sets of classes. Having a classification over the union enables regularization in both seen and unseen directions.

Bidirectional entropy regularization. Intuitively, when an image from an unseen class is fed to the classifier, probabilities for seen classes should yield a high entropy, while probabilities for unseen classes should result in a low entropy. In other words, the evaluation over seen classes of an unseen class input should be uncertain, because the image comes from a class the classifier has never encountered during training. Conversely, when an image from a seen class is fed to the classifier, the entropy of the probabilities for unseen classes should be high, while the entropy for seen classes should be low. To encourage this effect, given an image x, we compute the normalized Shannon entropy [30] of the probabilistic model $p(c|x,\mathcal{T})$ for both seen and unseen class directions:

$$\mathcal{H}_{\text{s}}(x) = \frac{-1}{|\mathcal{S}|}\sum_{c\in\mathcal{S}} p(c|x,\mathcal{T})\log p(c|x,\mathcal{T}), \text{ and } \mathcal{H}_{\text{u}}(x) = \frac{-1}{|\mathcal{U}|}\sum_{c\in\mathcal{U}} p(c|x,\mathcal{T})\log p(c|x,\mathcal{T}), \tag{5}$$

where $\mathcal{H}_\text{s}$ and $\mathcal{H}_\text{u}$ are the average entropy for seen and unseen classes, and $|\cdot|$ is the cardinality of the set. For training, we derive a margin-based regularization for both seen and unseen class directions:

$$R_\text{s} = \left[m + \frac{1}{N} \sum_{n=1}^{N} \mathcal{H}_\text{s}(x_n) - \frac{1}{\widetilde{N}} \sum_{n=1}^{\widetilde{N}} \mathcal{H}_\text{s}(\widetilde{x}_n) \right]_+, \tag{6}$$

$$R_\text{u} = \left[m + \frac{1}{\widetilde{N}} \sum_{n=1}^{\widetilde{N}} \mathcal{H}_\text{u}(\widetilde{x}_n) - \frac{1}{N} \sum_{n=1}^{N} \mathcal{H}_\text{u}(x_n) \right]_+, \tag{7}$$

where $[\cdot]_+ = \max(0, \cdot)$. R_s ensures a margin of at least m between the average seen class entropy of seen inputs x_n and generated unseen inputs $\widetilde{x}_n$. In other words, this formulation seeks to minimize $\mathcal{H}_\text{s}(x_n)$ and maximize $\mathcal{H}_\text{s}(\widetilde{x}_n)$. R_u has a corresponding effect on the unseen class entropy. The final loss function for training then becomes:

$$\mathcal{L}_\text{f} = \mathcal{L}_\text{s+u} + \lambda_\text{Ent}(R_\text{s} + R_\text{u}), \tag{8}$$

where $\lambda_\text{Ent} \in \mathbb{R}_{\geq 0}$ is a hyper-parameter to control the contribution of the bidirectional entropy.

3.3 Swapping seen and unseen class representations

As presented above, relying on a real-valued embedding space allows mechanisms to mitigate the bias in two scenarios. It also enables to swap class representations to less biased representations. Consider now the case where there exist multiple types of semantic information, which differ by their type of representation and by how expensive it is to collect them. For example, attribute descriptions require expert knowledge, while sentence descriptions can be crowd-sourced to non-expert workers. Practically, sentences tend to be less biased than attributes and perform better [39], but do not offer a comprehensive expert-based explanation [26]. One could then train a model for seen classes on attributes as they rely on expert-based explanations and rely for unseen classes on sentences as they are easier to collect. This results in different representation types for seen and unseen classes.

Formally, we assume that we have access to seen prototypes $\{\Phi_A^\mathcal{S}, \Phi_B^\mathcal{S}\}$ with representations from domain A and B. For evaluation, we have access to unseen prototypes $\Phi_A^\mathcal{U}$ of domain A, but $\Phi_B^\mathcal{U}$ of domain B is absent. The objective is then to learn a mapping β from $\Phi_A^\mathcal{S}$ to $\Phi_B^\mathcal{S}$, in order to regress $\hat{\Phi}_B^\mathcal{U}$ from $\Phi_A^\mathcal{U}$ at testing time. We define the mapping as a linear least squares regression problem with Tikhonov regularization, which corresponds to:

$$\min_\beta \|\Phi_B^\mathcal{S} - \beta \Phi_A^\mathcal{S}\|_2 + \lambda_\beta \|\beta\|_2. \tag{9}$$

where λ_β controls the amount of regularization. Relying on a linear transformation prevents overfitting, as the mapping involves a limited set of class prototypes. During evaluation, we apply β to unseen prototypes of domain A to regress their values in domain B: $\hat{\Phi}_B^\mathcal{U} = \beta \Phi_A^\mathcal{U}$. Swapping representations then corresponds to regressing from one domain to another.

4 Experimental Details

Datasets. We report experiments on four datasets commonly used in generalized zero-shot learning, *e.g.*, [7, 8, 26, 38]. For all datasets, we rely on the train and test splits proposed

by Xian *et al*. [38]. ***Caltech-UCSD-Birds 200-2011 (CUB)*** [34] contains 11,788 images from 200 bird species. Every species is described by a unique combination of 312 semantic attributes to characterize the color, pattern and shape of their specific parts. Moreover, every bird image comes along with 10 sentences describing the most prominent characteristics [26]. 150 species are used as seen classes during training, and 50 distinct species are left out as unseen classes during testing. ***SUN Attribute (SUN)*** [25] contains 14,340 images from 717 scene types. Every scene is also described by a unique combination of 102 semantic attributes to characterize material and surface properties. 645 scene types are used as seen classes during training, and 72 distinct scene types are left out as unseen classes during testing. ***Animals with Attributes (AWA)*** [15] contains 30,475 images from 50 animals. Every animal comes with a unique combination of 85 semantic attributes to describe their color, shape, state or function. 40 animals are used as seen classes during training, and 10 distinct animals are left out as unseen classes during testing. ***Oxford Flowers (FLO)*** [22] contains 8,189 images from 102 flower plants. Every flower plant image is described by 10 different sentences describing the shape and appearance [26]. 82 flowers are used as seen classes during training, and 20 distinct flowers are left out as unseen classes during testing.

Features extraction. For all datasets, we rely on the features extracted by Xian *et al*. [38]. Image features x come from ResNet101 [12] trained on ImageNet [28] and sentence representations are extracted from a 1024-dimensional CNN-RNN [26]. As established by Xian *et al*. [38], parameters of ResNet101 and the CNN-RNN are frozen and are not fine-tuned during the training phase. No data augmentation is performed either.

Evaluation. We evaluate experiments with calibration stacking as proposed by Chao *et al*. [8], which penalizes the seen class probabilities to reduce the bias during evaluation. Following Xian *et al*. [38], we compute the average per-class top-1 accuracy of seen classes (denoted as $\mathbf{s}$) and unseen classes (denoted as $\mathbf{u}$), as well as their harmonic mean $\mathbf{H} = (2 \times \mathbf{s} \times \mathbf{u})/(\mathbf{s}+\mathbf{u})$. We report the 3-run average.

Implementation details. In our model, $f(\cdot)$ corresponds to a multilayer perceptron with 2 hidden layers of size 2048 and 1024 to map the features x to the joint visual-semantic embedding space of size A. The output layer has a linear activation, while hidden layers have a ReLU activation [21] followed by a Dropout regularization ($p = 0.5$) [32]. We train $f(\cdot)$ using stochastic gradient descent with Nesterov momentum [33]. We set the following hyper-parameters for all datasets: learning rate of 0.01 with cosine annealing [19], initial momentum of 0.9, batch size of 64, temperature of 0.05, and an entropy regularization term of 0.1 with a margin of 0.2. For AWA, we reduce the learning rate to 0.0001 and increase the entropy regularization to 0.5 while keeping the same margin. When relying on sentence representations, we double the capacity of $f(\cdot)$ with twice the number of hidden units in each layer. We set hyper-parameters on a hold-out validation set and re-train on the joint training and validation sets. The source code uses the Pytorch framework [24].

5 Results

Bias variation. To verify whether the bias towards seen classes is dataset-dependent, we measure the average linkage between seen and unseen representations. Concretely, we compute the average of the pairwise cosine similarity between $\Phi^{\mathcal{S}}$ and $\Phi^{\mathcal{U}}$. A high average linkage then refers to a high similarity between seen and unseen representations. Intuitively, a high average linkage is not desirable as unseen representations can easily be confused with

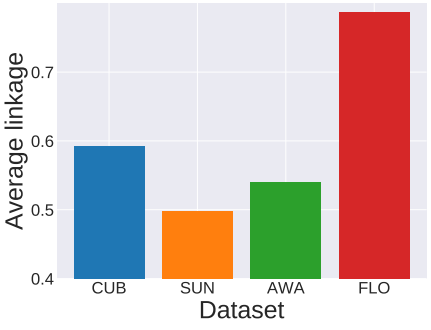

Figure 1: **Bias variation** across datasets. When measuring the average linkage between seen and unseen representations, FLO is the most affected while SUN is the least. Thus, the bias towards seen classes differs across datasets.

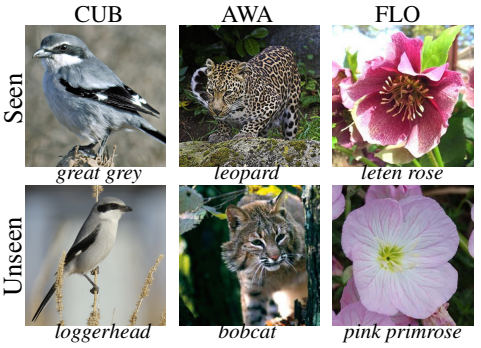

Figure 2: **Seen and unseen class samples**. Visual differences arise from the global shape (CUB, AWA) or colors (FLO). Yet, their semantic class representation yields a very high pairwise similarity, which creates a high bias.

seen ones, which makes the generalized zero-shot learning problem harder. Figure 1 depicts the average linkage per dataset. FLO exhibits the highest average linkage while SUN the lowest, with a 1.6 times difference. In other words, classifiers trained on FLO are highly affected by the bias towards seen classes. Figure 2 illustrates seen and unseen class samples with a very high pairwise similarity on CUB, AWA and FLO. Visually, these classes can be differentiated by their color or shape. Though, their semantic representations are very similar, which creates a high bias. Now that we have established that the bias towards seen classes differs across datasets, we can address the bias within generalized zero-shot learners.

Temperature scaling. Figure 3 varies the scale of the temperature in eq. 1. Following related metric learning works (*e.g.*, [36, 41]), we consider the temperature as a hyper-parameter. When treated as a latent parameter, the optimization diverges as its value goes down to zero to satisfy the loss function. The highest $\mathbf{H}$ score occurs when $T = 0.05$ on the validation set of all datasets. Performance starts to degrade substantially after $T > 0.1$. A temperature lower than $T < 0.05$ can yield even higher scores, but is usually prone to numerical errors. As such, we set $T = 0.05$ in all our experiments when training the model with only seen samples (eq. 2) or in combination with generated unseen samples (eq. 4). We also evaluate modifying T between training and testing phases. Setting it to 1 during training and testing, as in a normal softmax, drops $\mathbf{H}$ by 43.3% on AWA. And changing it to 0.05 when testing, drops the score by 25.6%. Keeping a fixed temperature value ensures $f(\cdot)$ maps inputs to prototypes similarly in training and testing. The temperature value should also be low to promote a more confident and discriminative model that yields narrow probabilities. Hence, the model reduces the bias by having a lower likelihood to classify an unseen class input as part of a seen class.

Entropy regularization. Figure 4 ablates the direction of the margin-based entropy term in eq. 8. For this experiment, we rely on unseen class features generated from Cycle-CLSWGAN [10]. When using a unidirectional entropy regularization, the improvement is either very low, or even negative, over a model without any regularization. Interestingly, this negative effect does not depend on the direction, as both $\mathcal{H}_s$ and $\mathcal{H}_u$ are affected when considered individually. Regularizing in only one direction forces the model to compensate for the other direction. Only the bidirectional regularization provides a consistent benefit

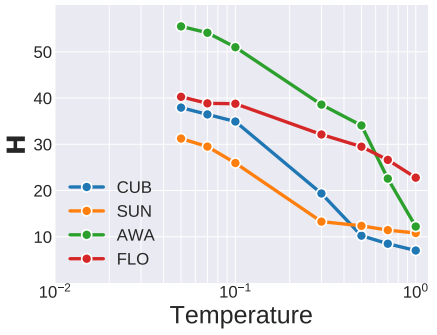

Figure 3: **Temperature scaling** ablation from $T = 0.05$ to $T = 1$. Temperature values over 0.1 degrade the performance because probabilities start to spread which makes the model less confident.

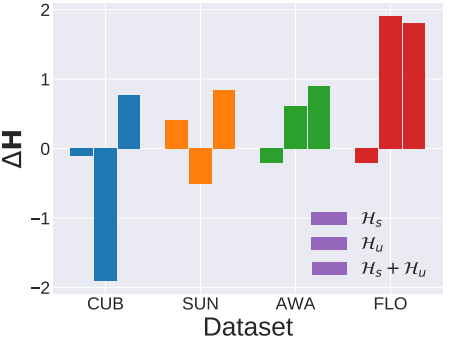

Figure 4: **Entropy regularization** in one ($\mathcal{H}_s$ or $\mathcal{H}_u$, hatched) and two ($\mathcal{H}_s + \mathcal{H}_u$, not hatched) directions compared with models without. Regularizing in only one direction can result in a negatively effect. Including both directions consistently improves results by creating a better bias trade-off.

Seen	Unseen	H
Att	Att	48.5
Sen	Att	47.4
Att	Sen	49.7
Sen	Sen	50.3

Table 1: **Swapping attribute (`Att`) and sentence (`Sen`) representations**. While `Att-Att` and `Sen-Sen` are the usual non-swapped evaluation settings, our method can also swap them. When using sentences for unseen classes, it always improves upon attributes in swapped and non-swapped evaluations as they are less biased and more discriminative.

for all datasets. This positive effect indicates the importance of balancing out both seen and unseen probabilities when mitigating the bias. Regularizing in both directions jointly helps the model learn a correct bias trade-off.

Swapping representations. Table 1 presents the different combinations of attribute (`Att`) and sentence (`Sen`) representations for training and evaluation. `Att-Att` and `Sen-Sen` are the common non-swapped settings. `Sen-Sen` forms an upper-bound as sentences provide better class representations over attributes. Indeed, sentence descriptions exhibit a lower average linkage than attribute descriptions. In a swapped setting, the unseen representations are regressed from representations in another domain based on eq. 9. A model trained on `Att` can be improved by 1.2 points at testing time when using `Sen` to regress the unseen representations. However, a model trained on `Sen` degrades when using `Att` to regress unseen representations. Indeed, `Sen-Att` requires to map low-dimensional attribute representations of unseen classes to a high-dimensional space of sentence representations on which the classifier has been trained. `Sen-Att` then involves dimensionality expansion, which is a harder problem than dimensionality compression in `Att-Sen`. In the scenario where a model is trained on attributes for seen classes derived from experts, it is possible to leverage sentences for unseen classes derived from crowd-sourcing to further improve the results.

Comparison with the state of the art. Table 2 compares our bias-aware prototype learner with eight other classifiers. Scores from other classifiers correspond to the performance as reported by the authors in their original paper. First, we consider stand-alone classifiers, which

Method	CUB			SUN			AWA			FLO		
	u	s	H	u	s	H	u	s	H	u	s	H
DeViSE [11]	23.8	53.0	32.8	16.9	27.4	20.9	13.4	68.7	22.4	9.9	44.2	16.2
w/ f-CLSWGAN [39]	52.2	42.4	46.7	38.4	25.4	30.6	35.0	62.8	45.0	45.0	38.6	41.6
SJE [1]	23.5	59.2	33.6	14.7	30.5	19.8	11.3	74.6	19.6	13.9	47.6	21.5
w/ f-CLSWGAN [39]	48.1	37.4	42.1	36.7	25.0	29.7	37.9	70.1	49.2	52.1	56.2	54.1
LATEM [37]	15.2	57.3	24.0	14.7	28.8	19.5	7.3	71.7	13.3	6.6	47.6	11.5
w/ f-CLSWGAN [39]	53.6	39.2	45.3	42.4	23.1	29.9	33.0	61.5	43.0	47.2	37.7	41.9
ESZSL [27]	12.6	63.8	21.0	11.0	27.9	15.8	6.6	75.6	12.1	11.4	56.8	19.0
w/ f-CLSWGAN [39]	36.8	50.9	43.2	27.8	20.4	23.5	31.1	72.8	43.6	25.3	69.2	37.1
ALE [2]	23.7	62.8	34.4	21.8	33.1	26.3	16.8	76.1	27.5	13.3	61.6	21.9
w/ f-CLSWGAN [39]	40.2	59.3	47.9	41.3	31.1	35.5	47.6	57.2	52.0	54.3	60.3	57.1
DCN [17]	28.4	60.7	38.7	25.5	37.0	30.2	25.5	84.2	39.1	–	–	–
One-hot softmax [39]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
w/ f-CLSWGAN [39]	43.7	57.7	49.7	42.6	36.6	39.4	57.9	61.4	59.6	59.0	73.8	65.6
w/ Cycle-CLSWGAN [10]†	45.7	61.0	52.3	49.4	33.6	40.0	56.9	64.0	60.2	72.5	59.2	65.1
w/ CADA-VAE [29]	51.6	53.5	52.4	47.2	35.7	40.6	57.3	72.8	64.1	–	–	–
w/ f-VAEGAN-D2 [40]†	48.4	60.1	53.6	45.1	38.0	**41.3**	57.6	70.6	63.5	56.8	74.9	64.6
w/ LisGAN [16]	46.5	57.9	51.6	42.9	37.8	40.2	52.6	76.3	62.3	57.7	83.9	68.3
COSMO [4]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
w/ f-CLSWGAN [39]	60.5	41.0	48.9	35.3	40.2	37.6	64.8	51.7	57.5	59.6	81.4	68.8
w/ LAGO [3]	44.4	57.8	50.2	44.9	37.7	41.0	52.8	80.0	63.6	n/a	n/a	n/a
This paper	45.1	52.5	48.5	41.0	30.1	34.7	55.2	70.5	61.9	42.6	66.6	52.0
w/ f-CLSWGAN [39]	50.7	49.9	50.3	41.1	31.6	35.7	57.7	68.4	62.5	53.8	76.0	63.0
w/ Cycle-CLSWGAN [10]†	57.4	58.2	**57.8**	44.8	32.7	37.8	61.3	69.2	**65.0**	69.3	79.9	**74.2**

† Method relies on sentence representations instead of attribute representations for CUB.

Table 2: **Comparison with the state of the art**, where classifiers are delimited by a horizontal rule and their combination with a generative model is in `teletype` font. "n/a" denotes a non-applicable setting to the method while "–" refers to non-reported results in the original paper. Compared with one-hot softmax and COSMO, our proposal is a stand-alone method that can also operate with seen class samples only. Compared with the other compatibility functions that also operate in this similar stand-alone setting, it achieves the best results (underlined). When extended with generated unseen class samples, we also improve over other classifiers (**bold**), leading to state-of-the-art results on the three most biased datasets out of four (see Figure 1).

only observe the seen class inputs during training, *i.e.*, without using any generated features. In this setting, our bias-aware formulation outperforms existing compatiblity functions [1, 2, 11, 17, 27, 37] on all datasets. It is also interesting to note that recent formulations with one-hot softmax [39] or COSMO [4] cannot operate in this setting. Indeed, they rely on a discrete label space for classification while we rely on a real-valued embedding space. This enables our formulation to incorporate new unseen classes easily and at near zero cost, similar to compatibility functions. Second, our approach is easily extended with existing generative models to include an auxiliary dataset $\widetilde{\mathcal{D}}^{\mathcal{U}}$ for unseen classes. We select f-CLSWGAN [39]

and Cycle-CLSWGAN [10] as the authors provide source code to evaluate on all four datasets. Reproducing the models from their original source code yields results within a reasonable range, *i.e.*, less than a 2-point difference in the **H** score. We obtain better results with Cycle-CLSWGAN [10] than f-CLSWGAN [39], which highlights the importance of the quality of the generated unseen class features. Moreover, our method profits more when generated samples better reflect the true distribution. When switching from f-CLSWGAN [39] to Cycle-CLSWGAN [10] on CUB, a one-hot softmax classifier leads to a 2.6% increase while our bias-aware classifier with a joint entropy regularization yields a 7.5% increase. We achieve state-of-the-art results on CUB, AWA and FLO. Only on the SUN dataset the one-hot softmax [39] and COSMO [4] provide higher scores. This originates from a lower bias towards seen classes in the SUN dataset (see Figure 1), which makes a bias-aware model less beneficial. When a dataset exhibits a low bias, separating the model for seen and unseen classes is preferred for equal treatment. Conversely, when a dataset exhibits a high bias, the training of the model should consider seen and unseen classes jointly to balance out their probabilities from the start. Overall, we produce competitive results in both scenarios, especially compared with classifiers without any bias-awareness.

6 Conclusion

The classification of seen and unseen classes in generalized zero-shot learning requires models to be aware of the bias towards seen classes. In this paper, we present such a model which calibrates the probabilities of seen and unseen classes jointly during training, and ensures a margin between the average entropy of both seen and unseen class probabilities. Learning consists of regressing inputs to real-valued representations. Relying on a mapping to a real-valued embedding space enables to swap seen and unseen representation types, and to evaluate the model in a stand-alone scenario or in combination with generated unseen features. Overall, our proposed bias-aware learner provides an effective alternative to separate classification approaches or classifiers without bias-awareness.

Acknowledgements The authors thank Zeynep Akata for helpful initial discussions, and Hubert Banville for feedback. William Thong is partially supported by an NSERC scholarship.

References

[1] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015.

[2] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE TPAMI*, 38(7), 2016.

[3] Yuval Atzmon and Gal Chechik. Probabilistic and-or attribute grouping for zero-shot learning. In *UAI*, 2018.

[4] Yuval Atzmon and Gal Chechik. Adaptive confidence smoothing for generalized zero-shot learning. In *CVPR*, 2019.

[5] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016.

[6] Soravit Changpinyo, Wei-Lun Chao, and Fei Sha. Predicting visual exemplars of unseen classes for zero-shot learning. In *ICCV*, 2017.

[7] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Classifier and exemplar synthesis for zero-shot learning. *IJCV*, 128(1), 2020.

[8] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, 2016.

[9] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[10] Rafael Felix, Vijay BG Kumar, Ian Reid, and Gustavo Carneiro. Multi-modal cycle-consistent generalized zero-shot learning. In *ECCV*, 2018.

[11] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS-W*, 2014.

[14] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *CVPR*, 2018.

[15] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE TPAMI*, 36(3), 2014.

[16] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. In *CVPR*, 2019.

[17] Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Generalized zero-shot learning with deep calibration network. In *NeurIPS*, 2018.

[18] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

[19] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[20] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017.

[21] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[22] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

[23] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In *NeurIPS*, 2009.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[25] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.

[26] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016.

[27] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015.

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115 (3), 2015.

[29] Edgar Schönfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *CVPR*, 2019.

[30] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[31] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1), 2014.

[33] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

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

[35] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

[36] Zhirong Wu, Alexei A Efros, and Stella X Yu. Improving generalization via scalable neighborhood component analysis. In *ECCV*, 2018.

[37] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016.

[38] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE TPAMI*, 2018.

[39] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018.

[40] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, 2019.

[41] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. In *BMVC*, 2019.