Semantically Aligned Bias Reducing Zero Shot Learning

Conference Paper · April 2019

DOI: 10.1109/CVPR.2019.00722

3 authors, including:

Akanksha Paul
Indian Institute of Technology Ropar
5 PUBLICATIONS **58** CITATIONS
SEE PROFILE

Prateek Munjal
Indian Institute of Technology Ropar
7 PUBLICATIONS **58** CITATIONS
SEE PROFILE

Semantically Aligned Bias Reducing Zero Shot Learning

Anonymous CVPR submission

Paper ID 3667

Abstract

Zero shot learning (ZSL) aims to recognize unseen classes by exploiting semantic relationships between seen and unseen classes. Two major problems faced by ZSL algorithms are the hubness problem and the bias towards the seen classes. Existing ZSL methods focus on only one of these problems in the conventional and generalized ZSL setting. In this work, we propose a novel approach, Semantically Aligned Bias Reducing (SABR) ZSL, which focuses on solving both the problems. It overcomes the hubness problem by learning a latent space that preserves the semantic relationship between the labels while encoding the discriminating information about the classes. Further, we also propose ways to reduce bias of the seen classes through a simple cross-validation process in the inductive setting and a novel weak transfer constraint in the transductive setting. Extensive experiments on three benchmark datasets suggest that the proposed model significantly outperforms existing state-of-the-art algorithms by ∼1.5-9% in the conventional ZSL setting and by ∼2-14% in the generalized ZSL for both the inductive and transductive settings.

1. Introduction

Zero-shot learning (ZSL) deals with the problem of recognizing the unseen class instances by relying on the labeled set of seen classes. It assumes that the semantic label embeddings of both seen and unseen classes are known apriori. ZSL thus learns to identify unseen classes by leveraging the semantic relationship between seen and unseen class embeddings.

On the basis of data available during the training phase, ZSL can be divided into two categories: **Inductive ZSL** and **Transductive ZSL**. In the inductive ZSL setting [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**], we are provided with only the labeled seen class instances along with the semantic embedding of unseen class labels during training. While in the transductive ZSL [**?**, **?**], in addition to the labeled seen class data and the semantic embedding of all labels, we are also provided with the unlabeled unseen class data. ZSL can also be categorized into **Conventional** and **Generalized ZSL** depending on the data that is presented to the model during testing. Conventional ZSL deals with the ZSL setting where data emerges only from unseen classes at test time. While generalized ZSL [**?**] is more realistic where data during testing can come from either seen or unseen classes.

Generally, ZSL approaches project the seen and unseen class data into a latent space that is robust for learning unseen class labels. One approach is to learn a latent space that is aligned towards the semantic label embedding [**?**, **?**, **?**, **?**, **?**, **?**]. The input data is transformed into this latent space for learning the classification models over seen and unseen classes. This approach leads to the well known **hubness problem** [**?**, **?**, **?**] where the transformed data become hubs for the nearby class embeddings leading to performance deterioration in both conventional and generalized ZSL. To alleviate the hubness problem, the other approaches [**?**, **?**, **?**, **?**] learn a latent visual space for recognizing the seen class labels by aligning the semantic class embeddings towards this latent space. Irrespective of the latent space for transforming the data, there is an inherent bias in the model towards seen classes, which we refer to as the **bias problem**. Due to this bias the models generally perform poorly on unseen classes.

Existing ZSL methods focus on addressing only one of these problems. In this work, we propose a novel method - **Semantically Aligned Bias Reducing (SABR)** ZSL to alleviate both the hubness and bias problems. To the best of our knowledge, this is the first approach that addresses both the problems simultaneously. We propose two versions of SABR - SABR-I and SABR-T for the inductive and the transductive ZSL settings respectively. Both these versions have a common first step that learns an intermediate representation for the seen and unseen class data. This intermediate latent space is learned to preserve both the semantic relationship between class embeddings and discriminating information among classes through a novel loss function.

After having learned the optimal latent space, both SABR-I and SABR-T learn generative adversarial networks (GAN) to generate the latent space representations. Specifically, SABR-I learns a conditional Wasserstein GAN for generating the latent space representations for the seen classes using only the seen class embeddings. As the label embeddings of seen and unseen classes exhibit semantic relationships that are being learned in first step, we utilize the generative network to synthesize unseen class representations for learning a classification model for ZSL and GZSL. Given that we only have labeled data for the seen classes, SABR-I reduces the bias by early stopping the training of conditional WGAN through a simulated ZSL problem induced on the seen class data.

SABR-T goes further to learn a different GAN for generating latent space instances for the unseen classes. This network is learned to minimize the marginal probability difference between the true latent space representations of the unlabeled unseen class instances and the synthetically generated representations. Further, the conditional probability distribution of the latent space representations given the semantic labels are weakly transferred from the conditional WGAN learned by SABR-I for the seen class labels. Specifically, we learn a Wasserstein GAN[**?**] for the unseen classes by constraining the amount of transfer from the seen classes as learned by SABR-I. Overall, the major contributions of the paper are as follows

- We propose a novel two-step solution for zero shot learning. In the first step, an appropriate latent space is learned by fine tuning a pre-trained model with semantic embedding to reduce the hubness problem. In the second step, generators for synthesizing unseen class representations are learned, whose bias towards the seen class is reduced by using an early stopping criterion in the inductive setting and a weak transfer criteria for the transductive setting.

- We introduce a loss function, which ensures that the embedding space is discriminative and semantically aligned with the semantic class embeddings.

- A novel adversarial generative transfer is proposed that tries to minimize both the conditional and marginal distributions of seen and unseen classes.

- Empirical evaluation across all the zero-shot learning datasets suggests that the proposed approach outperforms the state-of-the-art performance in both conventional and generalized ZSL in both the inductive and transductive settings.

2. Related Work

Zero-shot learning (ZSL) has been a well studied area in recent years. Early ZSL approaches [**?**, **?**, **?**, **?**, **?**, **?**] utilized the semantic label space for projecting the seen and unseen instances. DEVISE [**?**], ALE [**?**] and SJE [**?**] learned bilinear compatibility functions to model the relationship between visual and semantic space. ESZSL [**?**] added a regularizer to the bi-linear compatibility functions that bounded the norm of projected features and semantic attributes. All these methods were constrained by learning linear functions and was overcome by LATEM [**?**] and CMT [**?**] which learned non-linear functions.

Zhang *et al.* [**?**] were the first to demonstrate the hubness problem and suggested to use an intermediate visual space for projecting the seen and unseen class instances. Zhang *et al.* [**?**] and Ba *et al.* [**?**] transform both the semantic and visual space to a joint embedding space in which the visual representations are closer to their respective semantic representations. Annadani *et al.* [**?**] focused on utilizing the semantic structure while maintaining separability of classes. Our approach for the inductive ZSL setting (SABR-I) improves over the work of Annadani *et al.* [**?**], as in SABR-I we not only preserve semantic relations in visual space but also reduce the bias of seen classes.

Among the transductive ZSL approaches, Song *et al.* [**?**] leverage the conditional seen class data with unlabelled unseen class data to learn an unbiased latent embedding space. The bias towards seen classes is reduced by forcing a uniform prior over the output of the classifier for unseen class instances. Our approach differs from QFSL primarily in two ways. Firstly, we reduce the bias in both the inductive and transductive versions of our model. In the transductive version, SABR-T reduces the bias without enforcing the uniform prior on the output as this will reduce the unseen class conditional information in the latent space. Secondly, learning an optimal latent space helps to mitigate the hubness problem.

There also exists work on generative modeling for ZSL [**?**, **?**, **?**]. Verma *et al.* [**?**] model each class-conditional distribution as a Gaussian distribution whose parameters are learned by seen classes. It then predicts the parameters of class-conditional distribution for unseen classes. They further extend this work to incorporate the unseen class data and report results in transductive setting. Bucher *et al.* [**?**] and Xian *et al.* [**?**] generate pseudo instances of unseen classes by training a conditional generator for the seen classes. Our proposed approach for the inductive setting SABR-I differs from these approaches, as we learn a discriminative embedding space that preserves semantic relations which alleviates hubness problem. Further, we leverage the unlabeled unseen class data to reduce the bias of seen classes.

3. Methodology

3.1. Problem Definition

Let $D_s = \{x_i^s, y_i^s, c(y_i^s), i = 1, 2....N_s\}$ represent the set of seen class data instances, where x_i^s denotes the i^{th} seen class instance with the corresponding class label $y_i^s \in$

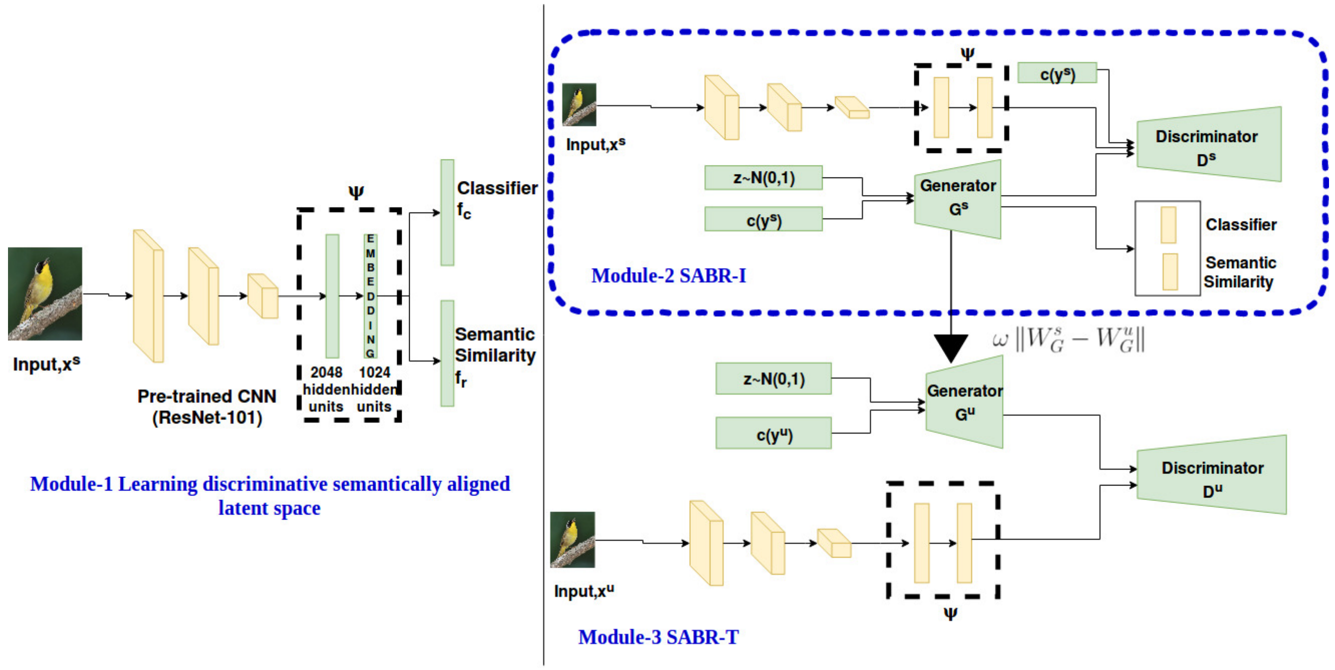

Figure 1. [Best viewed in color] An illustration of proposed Semantically Aligned Bias Reducing (SABR) model.

S (the set of seen classes). The semantic label embedding for each $y^s \in S$ is denoted by $c(y^s)$. In the inductive ZSL setting, we are provided only with the set of unseen labels $y^u \in U$ and the corresponding semantic label embedding $c(y^u)$. There is no overlap between the seen and unseen classes i.e., $S \cap U = \emptyset$. In the transductive ZSL setting, we also have unlabeled unseen class data represented by $D_u = \{x_i^u, y^u, c(y^u), i = 1, 2....N_u\}$ where x_i^u is the i^{th} unseen class instance. As the unseen class dataset is unlabeled, we do not have the labels y_i^u of x_i^u. The goal of conventional ZSL is to predict the label for each $x_i^u \in U$. In the generalized ZSL setting, the goal is to predict the label of a test sample, where the test sample can belong to either seen or unseen class.

3.2. Semantically Aligned Bias Reducing ZSL

In this section, we present our proposed two-tier model, SABR-I, for inductive ZSL (and a three-tier model, SABR-T for transductive ZSL) as shown in figure 1.

3.2.1 Learning the Optimal Latent Space

In the first step of SABR-I and SABR-T, we learn a latent space Ψ that preserves the semantic relations between classes while also learning the discriminating information for recognizing the classes. The semantic relations are essential as the learned latent space is later used for generating synthetic instances of unseen classes. The discriminating information is useful for learning the classifier and thus mitigating the hubness effect. We use pre-trained deep network Resnet-101 to extract features from the seen and unseen class images. For simplicity henceforth x_i^s and x_i^u refer to the features extracted from the pre-trained deep embedding models. These features are then used to learn a transformation, $\psi()$, that projects the seen and unseen class instances onto the latent space Ψ. $\psi()$ is modeled as a two layer fully connected network. The latent representations $\psi(x_i^s)$ are used to simultaneously learn a classifier (f_c) (for learning the discriminating information) and a regressor (f_r) (for preserving the semantic relationships among the labels). The classifier f_c outputs the one hot encoding of the class label of the instance and thus is trained minimizing the cross entropy loss,

$$\mathcal{L}_C = -\frac{1}{N_s}\sum_{i=1}^{N_s}\mathcal{L}(y_i^s, f_c(\psi(x_i^s))) \tag{1}$$

where $\mathcal{L}$ is cross entropy loss between true and predicted labels of seen class instance x_i^s.

The semantic relationships between the labels are preserved by ensuring that the output of the regressor f_r on the embedding of a seen instance $\psi(x_i^s)$ is closely related to the corresponding semantic embedding $c(y_i^s)$. We propose to use a similarity based cross-entropy loss, as defined in the equation below, between the predicted label embeddings of the regressor and the true semantic label embedding.

$$\mathcal{L}_S = -\sum_{i=1}^{N_s}\log\frac{\exp(\langle f_r(\psi(x_i^s)), c(y_i^s)\rangle)}{\Sigma_{y^s \in S}\exp(\langle f_r(\psi(x_i^s)), c(y_i^s)\rangle)} \tag{2}$$

where $\langle f_r(\psi(x_i^s)), c(y_i^s)\rangle$ refers to the similarity between predicted label embedding, $f_r(\psi(x_i^s))$, of each source instance x_i^s and its true semantic label embedding $c(y_i^s)$. This loss function ensures that the predicted label embeddings for all seen class instances belonging to a specific label form a cluster around the true semantic label embedding. The similarity could be defined using any measure such as Euclidean distance, cosine similarity or dot product.

The transformation function $\psi()$, as well as the classifier f_c and the regressor f_r are learned simultaneously by jointly minimizing the loss functions represented in equations 1 and 2 weighted by the factor γ

$$\mathcal{L}_{FS} = \min \mathcal{L}_C + \gamma * \mathcal{L}_S \tag{3}$$

Thus at the end of step 1, both the versions of SABR learn the transformation ψ to the latent space that is optimal in the sense that it possesses the discriminative information for classification and encodes the semantic relationship between the labels.

3.2.2 Bias Reducing Generator Network for SABR-I

The objective of inductive ZSL is to learn a classifier that can predict the labels of unseen class instances. As we do not have training instances of unseen classes, following the approach of Xian *et al.* [**?**], we learn a generator network that can generate synthetic unseen class instances as illustrated in the module 2 of Figure 1

Given the seen class embeddings $\psi(x_i^s) \in \Psi$, we first learn a conditional generator $G^s : \langle z, c(y^s)\rangle \rightarrow \Psi$. The generator takes as input a random noise vector z and a semantic label embedding $c(y^s)$ and outputs an instance $\tilde{x}^s$ in the latent space Ψ. As we know the labels associated with each seen class training instance, we train the conditional generator using the Wasserstein adversarial loss defined by

$$\begin{aligned}\mathcal{L}_G^s = &\ \mathbb{E}[D^s(\psi(x^s), c(y^s))] - \mathbb{E}[D^s(\tilde{x}^s, c(y^s))] \\ &- \lambda\, \mathbb{E}[(\|\nabla_{\hat{x}^s} D^s(\hat{x}^s, c(y^s))\| - 1)^2]\end{aligned} \tag{4}$$

where, D^s is the seen class conditional discriminator whose input is the seen class label embedding ($c(y^s)$) and the latent space instance ($\psi(x^s)$), $\hat{x}^s = \alpha\psi(x^s) + (1-\alpha)\tilde{x}^s$ with $\alpha \sim U(0,1)$ and λ is the gradient penalty coefficient. Thus, the objective for discriminator and generator pair is to

$$\min_{G^s} \max_{D^s} \mathcal{L}_G^s \tag{5}$$

We further want to encourage the generator to synthesize latent space embeddings of seen classes that are discriminative and encode the semantic similarity between the label embeddings. We achieve this by incorporating the loss functions defined in equation 1 and 2 to the overall optimization objective of the generator. We use the pre-trained classifier f_c and regressor f_r from the previous step while training the generator G^s. Thus the overall loss function for the generator-discriminator network can be defined as

$$\min_{G^s} \max_{D^s} \mathcal{L}_G^s + \beta(\mathcal{L}_C + \gamma\mathcal{L}_S) \tag{6}$$

This generator is then used to synthesize the latent space representations for the unseen classes. The semantic label embeddings encode relationships between the labels and therefore we expect the generator to synthesize meaningful latent representations of the unseen classes. However, the generator can be overly biased towards the seen classes due to the training set that is presented to it. This bias is mitigated using the principle of early stopping during training of the generator. The number of training epochs required to achieve the best performance is determined through a simple cross-validation set up on the seen classes.

3.2.3 Bias Reducing Generator Network for SABR-T

In the transductive setting the training process can benefit from modeling the unlabeled unseen class data. In particular, we model the marginal probability distribution of the unseen class unlabeled data via a GAN. We first obtain the latent representations of unseen class data x^u by transforming them using the function $\psi()$. Now, given the latent space representations of the unseen class instances $\psi(x^u)$, we learn a generator $G^u : \langle z, c(y^u)\rangle \rightarrow \Psi$ that takes noise z and semantic vector $c(y^u)$ as the input and outputs a synthetic instance $\tilde{x}^u$ in the latent space. G^u is trained as a conditional generator using the Wasserstein adversarial loss defined as follows

$$\begin{aligned}\mathcal{L}_G^u = &\ \mathbb{E}[D^u(\psi(x^u))] - \mathbb{E}[D^u(\tilde{x}^u)] \\ &- \lambda\, \mathbb{E}[(\|\nabla_{\hat{x}^u} D^u(\hat{x}^u)\| - 1)^2]\end{aligned} \tag{7}$$

were, D^u is the discriminator, $\hat{x}^u = \alpha\psi(x^u) + (1-\alpha)\tilde{x}^u$, α ∼U(0,1), and λ is the gradient penalty coefficient. Note that unlike D^s, D^u is not a conditional discriminator. Thus, the overall objective of the generator-discriminator pair for the unseen class instances can be defined as:

$$\min_{G^u} \max_{D^u} \mathcal{L}_G^u \tag{8}$$

The unlabeled class generator, G^u, trained in this fashion will produce synthetic unseen class latent representations that closely follow the true marginal distribution $P(\psi(x^u))$. However, it would not have learned the correct conditionals $P(\psi(x^u)|c(y^u))$. This is understandable as we do not have labeled unseen class data to train a conditional discriminator. On the other hand the seen class generator G^s also models the $P(\psi(x^s)|c(y^s))$. This is because of the seen class conditional discriminator D^s.

As the semantic label embeddings of the seen and unseen class share a common space and the latent representations of the seen and unseen class data are also from a common space, we hypothesize that generators of both the sets of classes must also be similar. Imposing this constraint allows us to transfer knowledge of the conditionals from the seen class generator to the unseen class generator.

Specifically, let W_G^s be the weights associated with the seen class generator, G^s and W_G^u be the weights associated with the unseen class generator, G^u. We propose a weak transfer constraint that forces W_G^u to be similar to that of W_G^s. We hypothesize that the unseen class generator learned using this constraint will encode the information on the conditionals. Thus, the overall objective of the generator network for the unseen transfer is formulated as:

$$\min_G \max_D \mathcal{L}_{WGAN}^u + \omega \left\| W_G^s - W_G^u \right\| \tag{9}$$

where, ω is a hyper-parameter controlling the importance of the similarity between the generators. When $\omega = 0$, the unseen class generator is completely independent of the seen class generator and there is no transfer of information between the two. This should result in synthetic unseen class instances that have very poor class conditional information in them. Large values of ω will force the unseen class generator to be identical to the seen class generator inducing high bias towards the seen classes; meaning the conditionals are biased towards the seen classes. This is also problematic as there is no overlap between the seen and unseen classes. Thus choosing an optimal hyper-parameter value that allows G^u to learn from G^s is important. This hyper-parameter is tuned through cross-validation on the set of seen classes.

3.3. Classification and Evaluation Metric

For the inductive setting, we generate synthetic unseen class representations using G^s. While for the transductive setting, we generate unseen class representations using G^u. As both G^s and G^u are conditional generators, we also have the unseen class labels associated with these synthetic instances. These are then used to train a softmax classifier to perform conventional ZSL. For the GZSL setting we combine the synthetically generated labeled unseen class representations with the representations of seen class labeled training instances to learn a softmax classifier.

We average the correct predictions for each class and report the average per class top accuracy [**?**, **?**] as below.

$$MCA_u = \frac{1}{|U|} \sum_{y^u \in U} acc_{y^u} \tag{10}$$

where, acc_{y^u} denotes top-1 accuracy on test unseen data for each class in U.

In the generalized ZSL setting, we compute average per class top accuracy for both seen and unseen classes and report the harmonic mean of seen and unseen accuracy [**?**, **?**] as defined below

$$H = \frac{2 * MCA_s * MCA_u}{MCA_s + MCA_u} \tag{11}$$

where, MCA_u, MCA_s denote the mean class accuracy on test unseen and seen classes respectively.

4. Experiments

4.1. Datasets

We evaluate the proposed methods using the following three benchmark datasets of ZSL. **Animals with Attributes2 (AWA2)** [**?**, **?**] that comprises of 37,322 images belonging to 50 classes where each class label is described using a 85-dimensional vector. We use 40 classes for training and the remaining 10 classes for testing. **Caltech-UCSDBirds-200-2011 (CUB)** [**?**] that contains 11,788 images from 200 different types of birds where the class label is represented using a 312 dimensional vector. We use 150 classes for training and the remaining 50 classes for testing. **SUN** [**?**] that consists of 14,340 images across 717 scenes with the class labels described using a 102 vector. Following [**?**, **?**], 580 classes out of 645 are used for training and the remaining 72 for testing.

We employ the proposed splits of Xian *et al.* [**?**, **?**] in all our experiments for fair comparison against prior approaches. For parameter tuning of the bias reducing models, we further divide the seen class set into train and validation splits preserving the original class distribution between train and test set. The best hyper-parameter values obtained through this cross-validation procedure is used to train the model using the entire training data.

4.2. Experimental Settings

4.2.1 Learning the optimal latent space

In all our experiments, we use Resnet-101[**?**] as the deep convolutional architecture. Prior literature has suggested using Resnet due to its on par or sometimes superior performance over other architectures such as VGG and Inception. The transformation function ψ is defined as two fully connected layers of 2048 and 1024 hidden units with ReLU activations acting on the final pooling layer of Resnet. The resulting 1024 dimensional space corresponds to the latent space, Ψ of our network and is trained by the loss proposed in equation 3. The classifier f_c is defined as a single fully connected layer output layer with softmax activation over embedding space Ψ. The semantically aligning regressor f_r is defined as a single output layer with the number of attributes in $c(y_s)$ over embedding space Ψ. This entire network is learned jointly. Learning rate is set to 0.001 and γ is 0.01 across all the datasets.

4.2.2 Learning the bias reducing generators

The seen and unseen class generators are modeled using a single fully connected hidden layer with 2048 leaky ReLU units. The discriminators contain a single hidden layer with 4096 leaky ReLU units that is connected to the output layer. Note that the SABR-I and SABR-T generators do not synthesize instances in the original data space, but the representations in the latent space Ψ. The gradient penalty coefficient of the Wasserstein generators, λ is set to 10 and β, the coefficient that enforces the generators to learn the representations of the latent space is set to 0.1 across all datasets. We first train a conditional WGAN for the seen classes and then adapt the WGAN for the unseen classes. The rest of hyper-parameters are fine tuned by stratified cross-validation over the seen class training set.

5. Results and Discussion

5.1. Bias reduction in SABR-I

In the inductive setting, as we do not have the unseen class data, the generator G^s tends to be overly biased towards the seen classes. We perform a three fold cross-validation experiment on the seen class training examples to validate this hypothesis. In each fold the seen class generator, G^s was allowed to train for a long time till the generator and discriminator converged. Samples for the unseen class in the validation set were synthesized from the generator at different epochs. These instances were used to train a classifier model for the GZSL setting. The average harmonic mean, H across the three folds as a function of the number of training epochs of the SABR-I generator is presented in Figure 2. It can be observed that as the number of training epochs increases, H increases initially but starts decreasing beyond an epoch. For detailed analysis, we plot the MCA_u, MCA_s and H values for AWA2. We observe that as the training progresses, MCA_s plateaus. However, the accuracy on the unseen classes in the cross-validation set MCA_u increases initially, but starts decreasing beyond epoch 45. Due to this, the overall H value decreases beyond this epoch. This behavior is very evident in the AWA2 dataset, and to a lesser extent in the CUB and SUN datasets due to significantly lesser number of training instances per class. This indicates that over training of the model based only on the seen class can bias the generator towards the seen classes and lose its capacity to generalize. This biasing is mitigated by stopping early at the best harmonic mean achieved *i.e.* at the 40, 45, and 95 epoch for AWA2, CUB, and SUN datasets respectively. In the transductive setting, this bias is subdued due to the presence of unlabeled data and there is no need for early stopping.

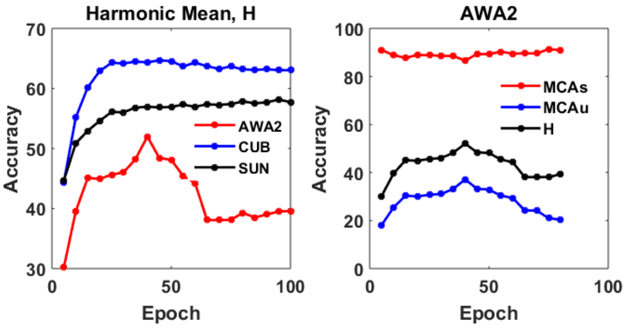

Figure 2. [Best viewed in color] Effect of bias measured by plotting average harmonic mean, H on the validation set (left). Detailed analysis on AWA2 by visualizing MCA_s, MCA_u and H.

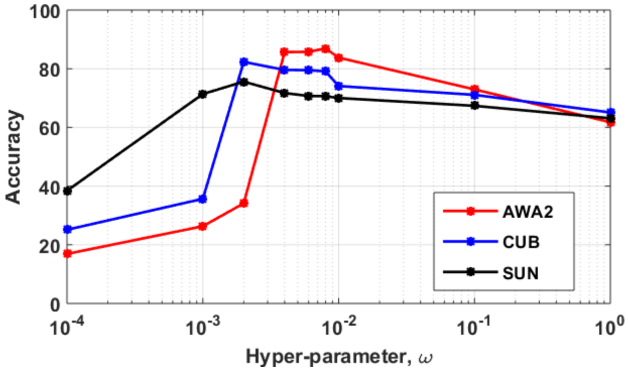

Figure 3. [Best viewed in color] Effect of the hyper-parameter ω on the cross-validation set.

5.2. Bias reduction in SABR-T

In the transductive setting, the hyper-parameter ω controls the bias in G^u towards the seen classes. We tune this hyper-parameter through cross-validation where the seen classes are divided into seen and unseen classes, similar to the inductive setting. Figure 3 illustrates the results obtained by the SABR-T while varying ω during the cross-validation process. We chose ω that performs best on this cross-validation set. For AWA2, CUB and SUN the model performs best at 0.008, 0.002 and 0.002 respectively. These values are fixed for the rest of the experiments.

5.3. Performance in the conventional ZSL setting

We compare our approach with existing state-of-the-art inductive and transductive ZSL methods. The results for the state of the art techniques reported in Table 1 are obtained from Xian *et al.* [**?**, **?**] and use the proposed splits on AWA2, CUB, and SUN datasets with Resnet-101 as the underlying CNN architecture.

The first observation from Table 1 is that no single state of the art method outperforms all the prior approaches on all the datasets for both the inductive and transductive settings. However, across all the inductive approaches, SABR-I performs better on AWA2 by 1.4%, CUB by 7.9% and by 1.4%

Type	Method	AWA2	CUB	SUN
I	DAP[?]	46.1	40	39.9
	IAP[?]	35.9	24	19.4
	CONSE[?]	44.5	34.3	38.8
	CMT[?]	37.9	34.6	39.9
	SSE[?]	61	43.9	51.5
	LATEM[?]	55.8	49.3	55.3
	ALE[?]	62.5	54.9	58.1
	DEVISE[?]	59.7	52	56.5
	SJE[?]	61.9	53.9	53.7
	ESZSL[?]	58.6	53.9	54.5
	SYNC[?]	46.6	55.6	56.3
	SAE[?]	54.1	33.3	4.3
	GFZSL[?]	63.8	49.4	60
	f-CLSWGAN[?]	-	57.3	60.8
	PSR[?]	63.8	56	61.4
	QFSL$^{\#}$[?]	63.5	58.8	56.2
I	SABR-I	**65.2**	**63.9**	**62.8**
T	ALE*[?]	70.6	54.4	55.5
	GFZSL*[?]	78.3	50.6	63.9
	DSRL*[?]	72.5	48.9	56.1
	QFSL$^{\#}$[?]	79.7	72.1	58.3
T	SABR-T	**88.9**	**74.0**	**67.5**

[1] I - inductive ZSL setting, T - transductive ZSl setting.
[2] Bold font denotes the best result while blue refers to the second best result.
[3] All results with * have been reported in Xian et al., [?] and are optimistically approximated by their graphs.
[#] Note that results reported on QFSL are on GoogleNet architecture.

Table 1. Performance in the conventional ZSL setting.

on SUN dataset. While in the transductive setting, SABR-T outperforms the current state-of-the-art by 9.2%, 1.9%, and 3.6% on the AWA2, CUB and SUN datasets respectively. We attribute this high performance gain to the reduced bias in the model by the better alignment of the marginals and weak conditional transfer from the seen class generator. The next best performing model on AWA2 and CUB in the transductive setting, QFSL, assumes a uniform distribution over the unseen class label space, which we suspect is confusing the classifier.

The significant performance jump in the transductive setting over the inductive setting suggests the importance of having the marginal probability distribution of the unseen classes through the unlabeled instances. We further discuss the robustness of our approach with the amount of unlabeled data in section 5.5.

5.4. Performance in the generalized ZSL setting

Table 2 presents the results for the generalized ZSL setting for various state-of-the-art approaches. SABR-I improves over DAP[?], IAP[?], CONSE[?], CMT[?], SSE[?], LATEM[?], ALE[?], DEVISE[?], SJE[?], ESZSL[?], SYNC[?], SAE[?] and GFZSL[?] by a large margin (over 10-20%) across all the benchmark datasets. These methods utilized semantic space for the embedding and thus faced the hubness problem. We observe that SABR-I outperforms PSR[?] by 14.65%, 22.89%, and 12.31% on the AWA2, CUB and SUN datasets respectively. This is because SABR-I not only preserves semantic relations in intermediate visual space like PSR but also, subdues the biasing effect by synthesizing unseen class instances and early stopping the training of the generator. SABR-I improves over f-CSLWGAN by 7.09% on CUB dataset. We attribute this improvement to learning an optimal latent representation for the dataset that is fine tuned using the seen class data. However, on the SUN dataset the performance of SABR-I marginally better than f-CLSWGAN by 2.1%. We speculate that the small number of training samples per class in the SUN dataset is insufficient to learn the optimal latent space and the seen class generator for SABR-I.

In the transductive setting, SABR-T performs better than QFSL by 7.61% and 9.38% on the AWA2 and SUN datasets respectively. There are two potential reasons for this significant improvement. Firstly, QFSL fine-tunes the entire ResNet-101 architecture, while SABR-T learns a latent space that preserves both the semantic relations and discriminative information using the ResNet representations without fine tuning the ResNet model. This mitigates the hubness problem. Secondly, QFSL tries to reduce the bias of the seen classes in the latent space by encouraging the output for unseen class instances to be uniformly distributed across all the unseen classes. This results in loss of class conditional information in the latent space. In contrast, SABR-T aligns the marginal distribution of the generated samples with that of the true unseen class instances, and learns the class conditionals from the seen class data. On the CUB dataset, while SABR-T yields superior performance over previous transductive approaches by 36.31%, except QFSL, where there is a marginal drop of 2.9%.

5.5. Amount of unlabeled data for transductive learning

Prior transductive ZSL approaches and SABR-T utilize all the available unlabeled unseen class instances for modeling the marginal probability distributions in both ZSL and GZSL settings. We hypothesize that the marginals of the unseen class instances can be learned from a smaller subset of unlabeled data. We conduct experiments using SABR-T on the AWA2 and SUN datasets as these contain maximum and minimum number of instances per class. SABR-T is trained with a randomly selected subset of unlabeled unseen class instances and the average performance over 5 trials for both ZSL and GZSL settings are reported in Figure 4. As expected the performance on both the datasets increases with increase in the number of unlabeled instances used dur-

Type	**Method**	**AWA2**			**CUB**			**SUN**		
		MCA$_u$	MCA$_s$	H	MCA$_u$	MCA$_s$	H	MCA$_u$	MCA$_s$	H
I	DAP[**?**]	0	84.7	0	1.7	67.9	3.3	4.2	25.1	7.2
	IAP[**?**]	0.9	87.6	1.8	0.2	72.8	0.4	1	37.8	1.8
	CONSE[**?**]	0.5	90.6	1	1.6	72.2	3.1	6.8	39.9	11.6
	CMT[**?**]	0.5	90	1	7.2	49.8	12.6	8.1	21.8	11.8
	SSE[**?**]	8.1	82.5	14.8	8.5	46.9	14.4	2.1	36.4	4
	LATEM[**?**]	11.5	77.3	20	15.2	57.3	24	14.7	28.8	19.5
	ALE[**?**]	14	81.8	23.9	23.7	62	34.4	21.8	33.1	26.3
	DEVISE[**?**]	17.1	74.7	27.8	23.8	53	32.8	16.9	27.4	20.9
	SJE[**?**]	8	73.9	14.4	23.5	59.2	33.6	14.7	30.5	19.8
	ESZSL[**?**]	5.9	77.8	11	12.6	63.8	21	11	27.9	15.8
	SYNC[**?**]	10.1	90.5	18	11.5	70.9	19.8	7.9	43.3	13.4
	SAE[**?**]	1.1	82.2	2.2	7.8	54	13.6	8.8	18	11.8
	GFZSL[**?**]	2.5	80.1	4.8	0	45.7	0	0	39.6	0
	f-CLSWGAN[**?**]	-	-	-	43.7	57.7	49.7	42.6	36.6	39.4
	PSR[**?**]	20.7	73.8	32.3	24.6	54.3	33.9	20.8	37.2	26.7
I	SABR-I	30.3	93.9	**46.9**	55.03	58.7	**56.8**	50.7	35.1	**41.5**
T	ALE*[**?**]			21.7			30.4			21.1
	GFZSL*[**?**]			40			33.5			0
	DSRL*[**?**]			32.3			28.9			20.5
	QFSL[**?**]	66.2	93.1	77.4	71.5	74.9	**73.2**	51.3	31.2	38.8
T	SABR-T	79.7	91.0	**85.0**	67.2	73.7	70.3	58.8	41.5	**48.6**

[1] I - inductive GZSL setting, T - transductive GZSL setting

[2] Bold font denotes the best result while blue refers to the second best result.

[3] All results with * only have H value reported by Xian et al., [**?**] and are optimistically approximated by the graphs.

Table 2. Performance in the Generalized ZSL Setting.

Figure 4. Change in the transductive ZSL and GZSL accuracy as a function of the number of unseen class instances.

ing training. Interestingly, on the AWA2 dataset, SABR-T achieves the best performance with using only 6.5% of the total unlabeled instances. We observe a similar trend on the SUN dataset with the performance peaking by using only 35% of the unlabeled data.

6. Summary and Future Work

In this work, we propose a novel approach, Semantically Aligned Bias Reducing (SABR) Zero Shot Learning, which focuses on solving both the hubness and bias problems that are commonly faced by zero shot learning (ZSL) techniques. SABR overcomes the hubness problem by learning a latent space that preserves the semantic relationship between the labels while encoding the discriminating information about the classes. Further, we also propose ways to reduce bias of the seen classes through a simple cross-validation process in the inductive setting and a novel weak transfer constraint in the transductive setting. Extensive experiments are conducted on three benchmark datasets (AWA2, CUB, and SUN) to investigate the efficacy of the proposed model. The results suggest that IDVAE significantly outperforms existing state-of-the-art zero shot learning algorithms by ∼1.5-9% in the conventional ZSL setting and by ∼2-14% in the generalized ZSL for both the inductive and transductive settings on most of the datasets. We also demonstrate that IDVAE reaches peak performance in the transductive setting by using only a fraction of the unlabeled unseen class instances for training.

In future, we would like to explore other semantic spaces like word vector embeddings or combination of different semantic label embeddings to accurately model the relationship between the seen and unseen class labels. It would also be beneficial to explore ways to extract semantic label embeddings from auxiliary sources as this would assist in extending ZSL to many real-world scenarios.

7. Acknowledgement