A Plug-in Attribute Correction Module for Generalized Zero-shot Learning

Haofeng Zhang[a,*], Haoyue Bai[a], Yang Long[b], Li Liu[c], Ling Shao[c]

[a] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China*
[b] *School of Computer Science, Durham University, Durham, UK*
[c] *Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates*

Abstract

While Zero Shot Learning models can recognize new classes without training examples, they often fails to incorporate both seen and unseen classes together at the test time, which is known as the Generalized Zero-shot Learning (GZSL) problem. This paper identifies a bottleneck issue when attributes are not well-defined, reliable, inaccurate in quantitative representations, or suffering from the visual-semantic discrepancy. We propose a Generic Plug-in Attribute Correction (GPAC) module which can effectively accommodate conventional ZSL in GZSL tasks. Different from existing embedding-based approaches which often lose the favor of transparency in attributes, our key challenge is to fully preserve the original meaning of the attributes and make it complementary and interpretable to upgrade existing ZSL models. To this end, we propose a novel nonnegative constraint with iterative Stochastic Gradient Descent toolbox to effectively fit our GPAC module into previous ZSL models. Extensive experiments on five popular datasets show that our method can effectively correct attributes and make conventional ZSL can achieve state-of-the-art performance on GZSL tasks. It is also a good practice for future models when incorporating prior human knowledge.

*Corresponding author.
Email addresses: zhanghf@njust.edu.cn (Haofeng Zhang), baihy@njust.edu.cn (Haoyue Bai), yang.long@ieee.org (Yang Long), liuli1213@gmail.com (Li Liu), ling.shao@ieee.org (Ling Shao)

Keywords: Generalized Zero Shot Learning (GZSL), Attribute Correction, Orthogonal Constraint, Pluggable Module

1. Introduction

Conventional supervised learning-based image classification systems have achieved promising results due to the rapid development of deep learning technologies [31] and large-scale datasets of common categories. With the growth of digital technologies and daily increasing new items, the challenge now is to make pre-trained models can generalize to new categories without collecting new training examples with structured annotations. As a promising solution, Zero-Shot Learning (ZSL) [16, 7] can recognize unseen objects by transferring learned knowledge model from seen classes. Previous ZSL research has achieved promising results on the old setting that assume test images are from unseen classes only. A new challenging Generalized ZSL (GZSL) [5] has become the emerging problem in this research field. In particular, GZSL considers that the test image can come from both seen and unseen categories. Conventional ZSL approaches are proved to suffer from the prediction bias towards seen categories [39]. For example, a zebra can be correctly predicted by comparing its similarity to other unseen categories of *dogs* and *cats*. However, when considering training categories of *horses*, and *cows* together with unseen classes, most of zebra images will be misclassified as *horses*.

Most of existing GZSL work ascribe such a bias to the overlap between learned unseen distribution and that of seen classes [38]. In this paper, we investigate another important issue that has become the bottleneck issue to improve the performance of GZSL models. Since no training examples are available in ZSL, explicit prior knowledge is necessary to estimate the distribution of unseen images. There are mainly four popular knowledge representations in current work. The first is based on textual embeddings, *e.g.* Word2Vec [22], that are learned from large scale text dataset in unsupervised learning frameworks; the second is exploring the class relationship with ontology [11]; the third

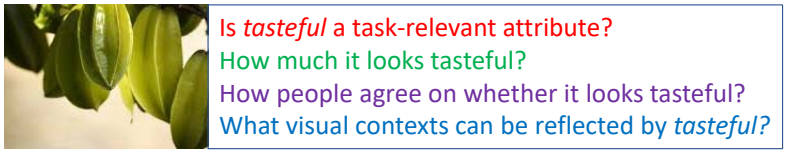


Figure 1: Designing and annotating attributes require appropriate guidelines and may not be accurate and discriminative.

is to associate unseen categories by similes in seen classes [20]; and the last is semantic attributes, which are manually defined and annotated by domain experts. Most of theoretical studies [34] adopt semantic attributes because of that each dimension of the attribute embedding has an explicit meaning. Using attributes can help qualitatively analysis of the ZSL model. More generally, such attribute-based predictions can benefit the model interpretation in deep learning research by large audience.

Currently, latent embedding has become one of the dominant frameworks for conventional ZSL problems. The learned latent space aims to mitigate the visual-semantic gap and make the representation more discriminative [39]. Such approaches have achieved state-of-the-art performance since the latent space can effectively preserve correlated visual-semantic information and remove the redundancy. However, the latent space fails to preserve the original meaning of each dimension in the attribute space, which makes the model difficult to interpret and understand. Guo *et al.* [8] select a subset of the attributes for learning different class distribution, variance, and entropy. Some recent work also attempts to synthesize samples of unseen classes from the attributes using Generative Adversarial Networks (GANs) [10], and then train a supervised model for all classes. However, these methods suffer from the same problem as the traditional supervised models, *i.e.* when a new unseen category is added, the model needs to be trained from scratch.

This paper proposes a new idea by correcting the attributes according to the visual contexts. Our key challenge and unique contribution is to preserve the original meanings of attributes. As the problems shown in Fig. 1, we pro-

pose a General Plug-in Attribute Correction (GPAC) algorithm to make the attributes more discriminative by two constraints: 1) different class attributes should be maximumly distinguishable, especially similar ones between seen and unseen classes in GZSL problems; 2) do not change the meaning of attributes and know how much and when not to make the correction. It is worth noting that GPAC is a plug-in module rather than a new ZSL framework. The corrected attributes can be well complementary to existing ZSL approaches. This paper adopts autoencoder-based [14] and label embedding-based [1] frameworks as examples. Furthermore, we provide an iterative optimization toolbox to effectively fit the GPAC module into ZSL models. Extensive evaluations are carried on five popular benchmarks, and the results show that GPAC can not only preserve the realistic meanings of the original attributes, but also significantly improve conventional ZSL models to state-of-the-art level in GZSL tasks. The contributions of our method are summarized as follows:

1) To our best knowledge, this is the first work that can explicitly correct attributes according to the visual contexts while preserving the original meaning of each attribute dimension;

2) To our best knowledge, GPAC is also the first plug-in module that aims to facilitate existing approaches and makes conventional ZSL models eligible or even state-of-the-art in GZSL tasks;

3) On five popular benchmarks, extensive quantitative and qualitative results manifest that the corrected attributes can better reflect the visual contexts without losing the integrability in attributes, which provides a good practice for future models when incorporating prior human knowledge.

The remaining part of this paper is organized as follows, Sec. 2 introduces the related works about ZSL and GZSL. Sec. 3 shows the detailed description of our method, which is followed by our experimental results and their analysis in Sec. 4. Sec. 5 makes a conclusion on this method.

2. Related Works

2.1. Zero Shot Learning

Zero Shot Learning (ZSL) has attracted an increasing number of attention by its powerful capability of recognizing new objects without training examples. Early frameworks such as DAP [15] estimated the labels by learning probabilistic attribute classifiers individually. In ALE [1], Akata *et al.* projected visual feature into semantic spaces via bilinear compatibility constraints with discriminative representation learning to disperse instances. Other conventional ZSL models such as CONvex combination of Semantic Embeddings (CONSE) [24] tried to automatically build unseen attributes from the instances of seen categories to reduce the effect of manual attributes. Kodirov *et al.* [14] adopted the idea of Auto-Encoder and directly used Euclidean distance to constrain the similarity of projected visual vectors and semantic embeddings.

There is a new trend of synthesis-based ZSL approaches. Long *et al.* [20] proposed to use the attributes of unseen classes to synthesize unseen visual features, and then train a fully supervised model with the seen data and the synthesized unseen visual features. An increasing number of generative methods have been proposed [10, 26], and many of them are based on GANs [32] or Variational Auto-Encoders (VAE). These methods all suffer from a serious problem when there comes a new category, retraining is unavoidable with the synthesized features of that unseen category.

The reliability of human-annotated attributes have been questioned since [12]. To the best of our knowledge, however, there is no attribute correction method has been proposed. On the similar purpose, attribute selection methods [8] have attempted to select part of the attributes by considering their class distribution, variance, and entropy. Such methods do not improve the quality of the attributes and cannot help with the GZSL tasks.

2.2. Generalized Zero Shot Learning

Different from conventional ZSL which assumes that all the test samples are only from the unseen categories, GZSL, firstly proposed by Chao *et al.*[5],

enlarges the query range to both seen and unseen classes. This is important in practice because we cannot guarantee whether the test image only belongs to unseen classes and hope the ZSL model can cooperate with the pre-trained model rather than discarding it. However, GZSL is more challenging. Due to there was no standard benchmarks particularly designed for GZSL, Xian *et al.* [35] defined a unified evaluation protocols and data splits on existing ZSL datasets. They evaluated a significant number of the state-of-the-art methods in depth, both in the conventional ZSL and GZSL settings. From then on, many methods have been proposed on this more realistic setting. For example, Liu *et al.* designed a Deep Calibration Network (DCN) to enable simultaneous calibration of deep networks on the confidence of source classes and uncertainty of target classes [17]. Pseudo distribution of seen samples on unseen classes is also employed to solve the domain shift problem on GZSL [37]. Besides, there are many other methods developed for this more realistic setting [10]. Significant performance gap has been reported between when applying ZSL approaches in GZSL tasks. A large proportion of misclassification is due to predicting unseen images into confusing seen classes. Despite some attempts on model development, there is no research attributing the GZSL problem with the attribute discriminativeness.

3. Methodology

3.1. Problem Definition

The dataset $\mathcal{C}$ consists of two completely separate splits, the seen $\mathcal{S}$ and the unseen classes $\mathcal{U}$, where, $\mathcal{S} = \{1, \cdots, s\}$, $\mathcal{U} = \{s+1, \cdots, s+u\}$, and $\mathcal{S} \cap \mathcal{U} = \varnothing$. In addition, each class of both $\mathcal{S}$ and $\mathcal{U}$ is associated with auxiliary attributes which are denoted as $\boldsymbol{S}_s \subset \mathcal{R}^{d_a \times s}$ and $\boldsymbol{S}_u \subset \mathcal{R}^{d_a \times u}$ respectively, where d_a is the dimension of the attribute space. Let $\boldsymbol{X}^s = \{\boldsymbol{x}_1^s, \cdots, \boldsymbol{x}_i^s, \cdots, \boldsymbol{x}_{N_s}^s\} \subset \mathbb{R}^{d_x \times N_s}$ denote the samples from seen classes $\mathcal{S}$, where N_s is the number of samples, d_x is the dimension of visual feature space, and each sample $\boldsymbol{x}_i^s$ is labeled with a single class in $\mathcal{S}$. Similarly, let $\boldsymbol{X}^u = \{\boldsymbol{x}_1^u, \cdots, \boldsymbol{x}_j^u, \cdots, \boldsymbol{x}_{N_u}^u\} \subset \mathbb{R}^{d_x \times N_u}$ denote

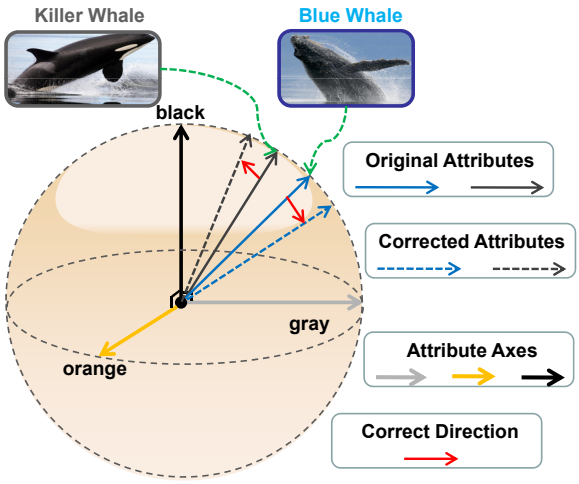


Figure 2: Illustration of the proposed GPAC, which tries to make the similar classes more discriminative while preserving their original meanings.

the samples from unseen classes $\mathcal{U}$, where N_u is the number of samples. In this paper, we aim to quantitatively correct the attributes $\boldsymbol{S} = [\boldsymbol{S}_s; \boldsymbol{S}_u]$ and learn a projection function $f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{W}$ from visual feature space to the attribute space to well classify these samples from both the seen and unseen classes $\mathcal{S} \cup \mathcal{U}$.

3.2. Attribute Correction

In this section, we introduce the concept of the proposed GPAC module. It is known that attributes are designed and annotated by experts and each entry has its realistic meaning. For example, the attribute of AWA [15] has an entry *"black"* with non-zero value denoting an animal has the black color on the body, *e.g. killer whale* and *blue whale*. Therefore, keeping the original meaning of attributes help interpret the prediction. In Fig. 2, "Killer Whale" and "Blue Whale" both have the attributes "black" and "gray", and neither of them has the attribute "orange", so it is necessary to preserve the "black" and "gray" and not introduce the "yellow" in the corrected attributes to keep their original meanings. Furthermore, attributes of both seen and unseen classes should be more discriminative for GZSL but human annotator would not be able

to provide accurate quantitative supervision or the quantity cannot reflect the visual contexts. Therefore, it is required to enlarge the gap between the similar classes, *e.g.*, the distance between "Killer Whale" and "Blue Whale" in Fig. 2 is increased to make them discriminative while preserving their original meanings. Therefore, we define the following loss function,

$$\begin{aligned} \mathcal{L}_I &= \alpha\|\boldsymbol{A}-\boldsymbol{S}\|_F^2+\beta\|\boldsymbol{A}^T\boldsymbol{A}-\boldsymbol{I}\|_F^2, \\ s.t. \quad & A_{ij}\geqslant 0, \text{and if } S_{ij}=0 \text{ then } A_{ij}=0. \end{aligned} \tag{1}$$

where, α and β are balancing coefficients for the two items, $\boldsymbol{S}=\boldsymbol{S}_s\cup\boldsymbol{S}_u$, $\boldsymbol{A}$ is the corrected attributes for both the seen and unseen classes, and A_{ij}, S_{ij} are the entries of the i^{th} row and j^{th} column of $\boldsymbol{A}$ and $\boldsymbol{S}$ respectively.

The former item keeps the originality of attributes while the later one disperse attributes and make the more discriminative to each other. Eq. 1 is independent of any ZSL models and thus can be effectively plugged into existing ZSL models. In the following, we adopt two conventional ZSL models as examples.

3.3. Plug in Autoencoder Frameworks

Autoencoder has become a main stream framework for two reasons: 1) it constrains both visual-semantic and the reverse embeddings to learn a optimal projection matrix $\boldsymbol{W}$; 2) can straightly fit to deep models, e.g. GANs. Without losing the generality, we adopt the earliest model SAE [14] in the example and plug in our GPAC module in the loss function $\mathcal{L}_I$:

$$\begin{aligned} \mathcal{L}_S =&\|\boldsymbol{X}^T\boldsymbol{W}-(\boldsymbol{A}\boldsymbol{B})^T\|_F^2+\gamma\|\boldsymbol{W}(\boldsymbol{A}\boldsymbol{B})-\boldsymbol{X}\|_F^2 \\ &+\alpha\|\boldsymbol{A}-\boldsymbol{S}\|_F^2+\beta\|\boldsymbol{A}^T\boldsymbol{A}-\boldsymbol{I}\|_F^2, \end{aligned} \tag{2}$$

where, γ is the balancing coefficient, $\boldsymbol{X}=\boldsymbol{X}^s$, and $\boldsymbol{B}$ is the one-hot label vector corresponding to $\boldsymbol{X}$. By applying the derivative of $\mathcal{L}_S$ with respect to $\boldsymbol{W}$ and setting it to zero, then we can obtain,

$$\begin{aligned} \frac{\partial\mathcal{L}_S}{\partial\boldsymbol{W}}=0 \Leftrightarrow & \boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W}+\gamma\boldsymbol{W}(\boldsymbol{A}\boldsymbol{B})(\boldsymbol{A}\boldsymbol{B})^T \\ &=\boldsymbol{X}(\boldsymbol{A}\boldsymbol{B})^T+\gamma\boldsymbol{X}(\boldsymbol{A}\boldsymbol{B})^T. \end{aligned} \tag{3}$$

Considering $\widehat{\boldsymbol{A}} = \boldsymbol{X}\boldsymbol{X}^T$, $\widehat{\boldsymbol{B}} = \gamma(\boldsymbol{A}\boldsymbol{B})(\boldsymbol{A}\boldsymbol{B})^T$, and $\widehat{\boldsymbol{C}} = \boldsymbol{X}(\boldsymbol{A}\boldsymbol{B})^T + \gamma\boldsymbol{X}(\boldsymbol{A}\boldsymbol{B})^T$, Eq. 3 gives the form in,

$$\widehat{\boldsymbol{A}}\boldsymbol{W} + \boldsymbol{W}\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{C}}, \tag{4}$$

and such Sylvester Equation can be solved in one line of implementation. Then, the derivative of $\mathcal{L}_S$ *w.r.t.* $\boldsymbol{A}$ can be calculated by

$$\begin{aligned}\frac{\partial \mathcal{L}_S}{\partial \boldsymbol{A}} =& -\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{B}^T + \boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T + \alpha(\boldsymbol{A}-\boldsymbol{S}) + 2\beta(\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{A}) \\ &+ \gamma(\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T - \boldsymbol{W}^T\boldsymbol{X}\boldsymbol{B}^T) \\ =&\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T + \alpha\boldsymbol{A} + 2\beta\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{A} + \gamma\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T \\ &- (\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{B}^T + \alpha\boldsymbol{S} + 2\beta\boldsymbol{A} + \gamma\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{B}^T).\end{aligned} \tag{5}$$

$\boldsymbol{A}$ can be effectively solved by iteratively updating with Stochastic Gradient Descent (SGD)

$$a_{ij}^{(t)} = a_{ij}^{(t-1)} - \eta\frac{\partial \mathcal{L}_S}{\partial a_{ij}^{(t-1)}} = a_{ij}^{(t-1)} - \Delta \tag{6}$$

where, $\Delta = \eta(\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T + \alpha\boldsymbol{A} + 2\beta\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{A} + \gamma\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T - (\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{B}^T + \alpha\boldsymbol{S} + 2\beta\boldsymbol{A} + \gamma\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{B}^T))_{ij}$, $a_{ij}^{(t)}$ is the result of t^{th} iteration in the row i and column j of $\boldsymbol{A}$, η is the learning rate, and all $\boldsymbol{A}$ in Eq. 6 is the abbreviation for $\boldsymbol{A}^{(t-1)}$. By setting $\eta = \frac{a_{ij}^{(t-1)}}{(\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T + \alpha\boldsymbol{A} + 2\beta\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{A} + \gamma\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T)_{ij}}$, Eq. 6 can be modified as

$$a_{ij}^{(t)} = a_{ij}^{(t-1)}\frac{(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{B}^T + \alpha\boldsymbol{S} + 2\beta\boldsymbol{A} + \gamma\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{B}^T)_{ij}}{(\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T + \alpha\boldsymbol{A} + 2\beta\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{A} + \gamma\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T)_{ij}}. \tag{7}$$

Since the attribute of each class should be nonnegative, we take the absolute value of Eq. 7. The denominator and numerator of Eq. 7 are both nonnegative, thus the updated a_{ij} is also kept to be nonnegative. Furthermore, if one entry of $\boldsymbol{a}_i$ equals zero, which means this class does not have such property, the updated $\boldsymbol{a}_i$ will also have zero value in this entry. This is important that our method does not change the absence/presence of an attribute, and just correct the values of attributes. The detailed algorithm of SAE+GPAC is described in Alg. 1.

3.4. Plug in Compatibility Models

Another widely adapted framework is based on compatibility functions, among which ALE [1] is the earliest ZSL model that exploits a max-margin

Algorithm 1: SAE+GPAC algorithm

Input :

The training data $\boldsymbol{X}$, and its corresponding labels $\boldsymbol{B}$ and attributes $\boldsymbol{S}$;

The hyper-parameters α, β, γ;

The outer iterative number $ITER$ and the inner iterative number $iter$.

Output:

The learned projection matrix $\boldsymbol{W}$, and the corrected attributes $\boldsymbol{A}$.

Initialize $\boldsymbol{W}$ with random value;

for $K = 1 \rightarrow ITER$ **do**

  Update $\boldsymbol{W}$ with $\boldsymbol{W} = Sylvester(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{B}}, \widehat{\boldsymbol{C}})$;

  for $k = 1 \rightarrow iter$ **do**

    Update $\boldsymbol{A}$ with $a_{ij}^{(t)} = a_{ij}^{(t-1)} \frac{(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{B}^T + \alpha\boldsymbol{S} + 2\beta\boldsymbol{A} + \gamma\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{B}^T)_{ij}}{(\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T + \alpha\boldsymbol{A} + 2\beta\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{A} + \gamma\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T)_{ij}}$;

    $\boldsymbol{A} = |\boldsymbol{A}|$;

  end

  $\boldsymbol{S} = \boldsymbol{A}$;

end

Return the learned $\boldsymbol{W}$ and $\boldsymbol{A}$.

strategy to learn a projection matrix from visual space to attribute space. Its difference to traditional max margin method is to utilize an inner product of attributes to replace the fixed margin value. By plugging in the GPAC module gives ALE+GPAC function,

$$\begin{aligned}\mathcal{L}_A =& \frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{j\in\mathcal{S}\setminus\ell(\boldsymbol{x}_i)}\max(\boldsymbol{x}_i^T\boldsymbol{W}(\boldsymbol{a}_j-\boldsymbol{a}_{\ell(\boldsymbol{x}_i)})+\boldsymbol{a}_j^T\boldsymbol{a}_{\ell(\boldsymbol{x}_i)},0)\\ &+\alpha\|\boldsymbol{A}-\boldsymbol{S}\|_F^2+\beta\|\boldsymbol{A}^T\boldsymbol{A}-\boldsymbol{I}\|_F^2+\gamma\|\boldsymbol{W}\|_{2,1},\end{aligned} \tag{8}$$

where, $\ell(\boldsymbol{x}_i)$ represents for the class of $\boldsymbol{x}_i$. $j\in\mathcal{S}\setminus\ell(\boldsymbol{x}_i)$ means j is selected from $\mathcal{S}$ except $\ell(\boldsymbol{x}_i)$. $\|\boldsymbol{W}\|_{2,1}$ is the $\mathcal{L}_{2,1}$ norm, which can be represented as $\sum_{i=1}^{d_x}\sqrt{\sum_{j=1}^{d_a}w_{ij}^2}$. Therefore, by fixing $\boldsymbol{A}$, the derivative of $\mathcal{L}_A$ with respect to $\boldsymbol{W}$ is,

$$\begin{aligned}\frac{\partial\mathcal{L}_A}{\partial\boldsymbol{W}} =& \frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{j\in\mathcal{S}}\mathbb{1}(\boldsymbol{x}_i^T\boldsymbol{W}(\boldsymbol{a}_j-\boldsymbol{a}_{\ell(\boldsymbol{x}_i)})+\boldsymbol{a}_j^T\boldsymbol{a}_{\ell(\boldsymbol{x}_i)}>0)\boldsymbol{x}_i(\boldsymbol{a}_j-\boldsymbol{a}_{\ell(\boldsymbol{x}_i)})^T\\ &+2\gamma\boldsymbol{D}\boldsymbol{W}\\ =&\frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{j\in\mathcal{S}}\mathbb{1}(\boldsymbol{x}_i^T\boldsymbol{W}(\boldsymbol{a}_j-\boldsymbol{a}_{\ell(\boldsymbol{x}_i)})+\boldsymbol{a}_j^T\boldsymbol{a}_{\ell(\boldsymbol{x}_i)}>0)\boldsymbol{x}_i\boldsymbol{a}_j^T\\ &-\frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{j\in\mathcal{S}}\mathbb{1}(\boldsymbol{x}_i^T\boldsymbol{W}(\boldsymbol{a}_j-\boldsymbol{a}_{\ell(\boldsymbol{x}_i)})+\boldsymbol{a}_j^T\boldsymbol{a}_{\ell(\boldsymbol{x}_i)}>0)\boldsymbol{x}_i\boldsymbol{a}_{\ell(\boldsymbol{x}_i)}^T\\ &+2\gamma\boldsymbol{D}\boldsymbol{W},\end{aligned} \tag{9}$$

where, $\boldsymbol{D}=diag([1/\|\boldsymbol{w}_1\|_2,\cdots,1/\|\boldsymbol{w}_{d_x}\|_2])$. $\mathbb{1}(\cdot)$ is the indicator function that when the condition is satisfied the result is one, otherwise zero. Thus, $\boldsymbol{W}$ can be updated with SGD,

$$\begin{aligned}\boldsymbol{W}^{(t)} &= \boldsymbol{W}^{(t-1)}-\eta_1\frac{\partial\mathcal{L}}{\partial\boldsymbol{W}_{ij}^{(t)}}\\ &= \boldsymbol{W}^{(t-1)}-\eta_1(\boldsymbol{P}1-\boldsymbol{P}_2+2\gamma\boldsymbol{D}\boldsymbol{W}),\end{aligned} \tag{10}$$

where, η_1 is the learning rate, $\boldsymbol{P_1}=\frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{j\in\mathcal{S}}\mathbb{1}(\boldsymbol{x}_i^T\boldsymbol{W}(\boldsymbol{a}_j-\boldsymbol{a}_{\ell(\boldsymbol{x}_i)})+\boldsymbol{a}_j^T\boldsymbol{a}_{\ell(\boldsymbol{x}_i)}>0)\boldsymbol{x}_i\boldsymbol{a}_j^T$, and $\boldsymbol{P}_2=\frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{j\in\mathcal{S}}\mathbb{1}(\boldsymbol{x}_i^T\boldsymbol{W}(\boldsymbol{a}_j-\boldsymbol{a}_{\ell(\boldsymbol{x}_i)})+\boldsymbol{a}_j^T\boldsymbol{a}_{\ell(\boldsymbol{x}_i)}>0)\boldsymbol{x}_i\boldsymbol{a}_{\ell(\boldsymbol{x}_i)}^T$. All the $\boldsymbol{W}$ in the second item of Eq. 10 is short for $\boldsymbol{W}^{(t)}$.

Since the derivative of $\boldsymbol{A}$ cannot be directly calculated, we split the first item of Eq. 8 into two parts according to the appearance of $\boldsymbol{a}_j$. The loss function $\mathcal{L}_A$ can be represented as the following,

$$\begin{aligned}\mathcal{L}_A = \frac{1}{N_s}(&\sum_{i=1}^{N_s \backslash \boldsymbol{a}_j} \sum_{k \in \mathcal{S}} \max(\boldsymbol{x}_i^T \boldsymbol{W}(\boldsymbol{a}_k - \boldsymbol{a}_{\ell(\boldsymbol{x}_i)}) + \boldsymbol{a}_k^T \boldsymbol{a}_{\ell(\boldsymbol{x}_i)}, 0) \\ &+ \sum_{i=1}^{\boldsymbol{a}_j} \sum_{k \in \mathcal{S}} \max(\boldsymbol{x}_i^T \boldsymbol{W}(\boldsymbol{a}_k - \boldsymbol{a}_j) + \boldsymbol{a}_k^T \boldsymbol{a}_j, 0)) \\ &+ \alpha \sum_{k \in \mathcal{S}} \|\boldsymbol{a}_k - \boldsymbol{s}_k\|^2 + \beta \|[\boldsymbol{a}_1, \cdots, \boldsymbol{a}_c]^T [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_c] - \boldsymbol{I}\|_F^2.\end{aligned} \tag{11}$$

where, $\sum_{i=1}^{N_s \backslash \boldsymbol{a}_j}$ means all the training samples except those belong to the $\boldsymbol{a}_j$ class, and $\sum_{i=1}^{\boldsymbol{a}_j}$ represents for the samples only belong to the $\boldsymbol{a}_j$ class.

Similar as Eq. 9, by fixing $\boldsymbol{W}$, the derivative of $\mathcal{L}$ with respect to each seen class $\boldsymbol{a}_j$ is,

$$\begin{aligned}\frac{\partial \mathcal{L}_{\boldsymbol{a}_j}}{\partial \boldsymbol{a}_j^{(s)}} = &\frac{1}{N_s} \sum_{i=1}^{N_s \backslash \boldsymbol{a}_j} \mathbb{1}(\boldsymbol{x}_i^T \boldsymbol{W}(\boldsymbol{a}_j - \boldsymbol{a}_{\ell(\boldsymbol{x}_i)}) + \boldsymbol{a}_j^T \boldsymbol{a}_{\ell(\boldsymbol{x}_i)} > 0) \\ &\qquad (\boldsymbol{W}^T \boldsymbol{x}_i + \boldsymbol{a}_{\ell(\boldsymbol{x}_i)}) \\ &- \frac{1}{N_s} \sum_{i=1}^{\boldsymbol{a}_j} \sum_{k \in \mathcal{S} \backslash j} \mathbb{1}(\boldsymbol{x}_i^T \boldsymbol{W}(\boldsymbol{a}_k - \boldsymbol{a}_j) + \boldsymbol{a}_k^T \boldsymbol{a}_j > 0) \boldsymbol{W}^T \boldsymbol{x}_i \\ &+ \frac{1}{N_s} \sum_{i=1}^{\boldsymbol{a}_j} \sum_{k \in \mathcal{S} \backslash j} \mathbb{1}(\boldsymbol{x}_i^T \boldsymbol{W}(\boldsymbol{a}_k - \boldsymbol{a}_j) + \boldsymbol{a}_k^T \boldsymbol{a}_j > 0) \boldsymbol{a}_k \\ &+ 2\alpha(\boldsymbol{a}_j - \boldsymbol{s}_j) + 4\beta(\boldsymbol{A}\boldsymbol{A}^T - \boldsymbol{I})\boldsymbol{a}_j.\end{aligned} \tag{12}$$

Since the unseen classes are independent of the first item in Eq. 8, the derivative of $\mathcal{L}_A$ with respect to each unseen class $\boldsymbol{a}_j$ is,

$$\frac{\partial \mathcal{L}_{\boldsymbol{a}_j}}{\partial \boldsymbol{a}_j^{(u)}} = 2\alpha(\boldsymbol{a}_j - \boldsymbol{s}_j) + 4\beta(\boldsymbol{A}\boldsymbol{A}^T - \boldsymbol{I})\boldsymbol{a}_j. \tag{13}$$

Similar as that in Eq. 10, $\boldsymbol{a}_j$ can also be implemented with SGD as,

$$a_{ij}^{(t)} = a_{ij}^{(t-1)} - \eta_2 \frac{\partial \mathcal{L}_{\boldsymbol{a}_j}}{\partial a_{ij}^{(t-1)}}. \tag{14}$$

By setting $\eta_2 = \frac{a_{ij}^{(t-1)}}{(\boldsymbol{q}_1 + \boldsymbol{q}_3 + 2\alpha \boldsymbol{a}_j + 4\beta \boldsymbol{A}\boldsymbol{A}^T \boldsymbol{a}_j)_i}$, where $\boldsymbol{q}_1 = \frac{1}{N_s} \sum_{i=1}^{N_s \backslash \boldsymbol{a}_j} \mathbb{1}(\boldsymbol{x}_i^T \boldsymbol{W}(\boldsymbol{a}_j - \boldsymbol{a}_{\ell(\boldsymbol{x}_i)}) + \boldsymbol{a}_j^T \boldsymbol{a}_{\ell(\boldsymbol{x}_i)} > 0)(\boldsymbol{W}^T \boldsymbol{x}_i + \boldsymbol{a}_{\ell(\boldsymbol{x}_i)})$ and $\boldsymbol{q}_3 = \sum_{i=1}^{\boldsymbol{a}_j} \sum_{k \in \mathcal{S} \backslash j} \mathbb{1}(\boldsymbol{x}_i^T \boldsymbol{W}(\boldsymbol{a}_k -$

$\boldsymbol{a}_j) + \boldsymbol{a}_k^T\boldsymbol{a}_j > 0)\boldsymbol{a}_k$, Eq. 14 for the seen classes can be further represented as,

$$a_{ij}^{(t)} = a_{ij}^{(t-1)} \frac{(\boldsymbol{q}_2 + 2\alpha\boldsymbol{s}_j + 4\beta\boldsymbol{a}_j)_i}{(\boldsymbol{q}_1 + \boldsymbol{q}_3 + 2\alpha\boldsymbol{a}_j + 4\beta\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{a}_j)_i}, \tag{15}$$

where $\boldsymbol{q}_2 = \frac{1}{N_s}\sum_{i=1}^{\boldsymbol{a}_j}\sum_{k\in\mathcal{S}\setminus j}\mathbb{1}(\boldsymbol{x}_i^T\boldsymbol{W}(\boldsymbol{a}_k - \boldsymbol{a}_j) + \boldsymbol{a}_k^T\boldsymbol{a}_j > 0)\boldsymbol{W}^T\boldsymbol{x}_i$. In addition, it is easy to compute the update function for the unseen classes,

$$a_{ij}^{(t)} = a_{ij}^{(t-1)} \frac{(\alpha\boldsymbol{s}_j + 2\beta\boldsymbol{a}_j)_i}{(\alpha\boldsymbol{a}_j + 2\beta\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{a}_j)_i}. \tag{16}$$

Similar as that in SAE to keep the attribute nonnegative, we also apply the same operation $a_{ij}^{(t)} = |a_{ij}^{(t)}|$ on Eq. 15 and Eq. 16. Furthermore, from Eq. 15 and Eq. 16, it is obvious to discover that when $a_{ij}^{(t-1)}$ equals zero, $a_{ij}^{(t)}$ is kept zero during the whole iteration, which guarantees that the existence of attribute will not be changed. The process of the algorithm for ALE+GPAC can be found in Alg. 2.

3.5. Discussion

In this subsection, we discuss the issue that why we apply the operation of making absolute value of $\boldsymbol{A}$ during iteration. Since it is known that the attribute has its realistic meaning, it is surely nonnegative. In Eq. 7, Eq. 15 and Eq. 16, if the numerators and denominators are nonnegative, the result of updated $\boldsymbol{A}$ is nonnegative, which need us to guarantee $\boldsymbol{W}$ is nonnegative. The nonnegative $\boldsymbol{W}$ can be realized by applying $w_{ij}\frac{(\boldsymbol{P2})_{ij}}{(\boldsymbol{P}_1+2\gamma\boldsymbol{D}\boldsymbol{W})_{ij}}$. However, since both nonnegative constraints of $\boldsymbol{W}$ and $\boldsymbol{A}$ can cause large concussion for loss value and finally lead to non-convergence, shown in Fig. 3 (a), we only constrain $\boldsymbol{A}$ to be nonnegative. In Fig. 3 (b), we show that the value of $\boldsymbol{A}$ (using $\|\boldsymbol{A}^{(t)} - \boldsymbol{A}^{(t-1)}\|_F^2$) can well converge without the nonnegative constraint of $\boldsymbol{W}$. In addition, we do not utilize the closed-form solution for $\boldsymbol{W}$ in ALE+GPAC, because the computational complexity of it can reach $\mathcal{O}(N \times C \times d_x \times d_a)$ for a single iteration, we use mini-batch SGD to reduce the computational complexity.

Algorithm 2: ALE+GPAC algorithm

Input :

The training data $\boldsymbol{X}$, and its corresponding attributes $\boldsymbol{S}$;

The hyper-parameters α, β, γ, and the mini-batch size t;

The iterative number $ITER$.

Output:

The learned projection matrix $\boldsymbol{W}$, and the corrected attributes $\boldsymbol{A}$.

Initialize $\boldsymbol{W}$ with random value;

for $K = 1 \rightarrow ITER$ **do**

  Randomly choose t samples as a mini-batch $\hat{\boldsymbol{X}}$ from the training set $\boldsymbol{X}$;

  Update $\boldsymbol{W}$ with $\boldsymbol{W}^{(t)} = \boldsymbol{W}^{(t-1)} - \eta_1(\boldsymbol{P}1 - \boldsymbol{P}_2 + 2\gamma\boldsymbol{D}\boldsymbol{W})$;

  for $\kappa = 1 \rightarrow$ *number of seen classes* **do**

    Update $\boldsymbol{a}_\kappa$ with $a_{ij}^{(t)} = a_{ij}^{(t-1)} \frac{(\boldsymbol{q}_2 + 2\alpha\boldsymbol{s}_j + 4\beta\boldsymbol{a}_j)_i}{(\boldsymbol{q}_1 + \boldsymbol{q}_3 + 2\alpha\boldsymbol{a}_j + 4\beta\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{a}_j)_i}$;

  end

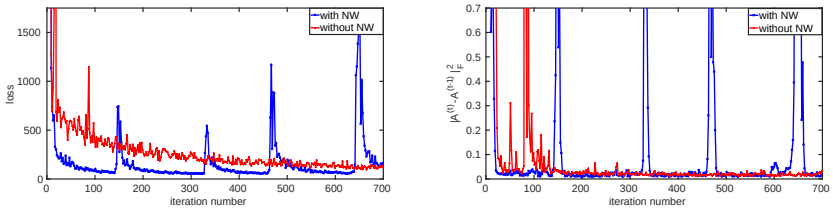
  for $\kappa = 1 \rightarrow$ *number of unseen classes* **do**

    Update $\boldsymbol{a}_\kappa$ with $a_{ij}^{(t)} = a_{ij}^{(t-1)} \frac{(\alpha\boldsymbol{s}_j + 2\beta\boldsymbol{a}_j)_i}{(\alpha\boldsymbol{a}_j + 2\beta\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{a}_j)_i}$;

  end

  $\boldsymbol{A} = |\boldsymbol{A}|$;

end



(a) Loss function (b) $\|\boldsymbol{A}^{(t)} - \boldsymbol{A}^{(t-1)}\|_F^2$

Figure 3: The convergence curves of loss function and $\boldsymbol{A}$ on AWA1, where 'NW' means Nonnegative $\boldsymbol{W}$.

4. Experiments

4.1. Datasets and settings

Datasets: To verify the effectiveness of our approach, we conduct experiments on five popular datasets, including SUN attribute (SUN) [25], Caltech-UCSD Birds-200-2011 (CUB-200) [30], Animal with Attribute 1 (AWA1) [15], Animal with Attribute 2 (AWA2) [34] and a Pascal & Yahoo attribute (aPY) [6]. The dataset splits for training and testing follow that used in [34].

Settings: The extracted features with ResNet [9] are exploited as our training data, and the same expert-annotate attributes employed in the evaluation in [34] are also utilized. Additionally, there are three hyper-parameters α, β, γ and learning rate η_1 in our method. Among these four hyper-parameters, we set $\gamma = 1 \times 10^{-3}$ and $\eta_1 = 0.1$ for ALE+GPAC. Besides, due to the fact that different hyper-parameters can lead to different performance on each dataset, we search our optimal parameters for α and β by employing a cross-validation strategy. To be specific, we randomly select 20% of the seen classes as validational unseen classes, and the parameters of best average performance of 5 executions are selected as the optimal hyper-parameters for each dataset. The source codes of all the three extensions can be found in the supplementary material.

4.2. Results on GZSL

Since our method focuses on the more realistic GZSL setting, the experiments are only conducted on GZSL. We follow the metrics proposed by Xian

Table 1: Comparison with state-of-the-art baselines on GZSL setting. '-' means not reported or not available.

	SUN			**CUB**			**AWA1**			**AWA2**			**aPY**		
Method	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**
DAP [15]	4.2	25.1	7.5	1.7	67.9	3.3	0.0	**88.7**	0.0	0.0	84.7	0.0	4.8	78.3	9.0
CONSE [24]	6.8	**39.9**	11.6	1.6	72.2	3.1	0.4	88.6	0.8	0.5	**90.6**	1.0	0.0	**91.2**	0.0
SSE [40]	2.1	36.4	4.0	8.5	46.9	14.4	7.0	80.5	12.9	8.1	82.5	14.8	0.2	78.9	0.4
LATEM [33]	14.7	28.8	19.5	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0	0.1	73.0	0.2
CVAE-ZSL [23]	-	-	26.7	-	-	34.5	-	-	47.2	-	-	-	-	-	-
CDL [13]	21.5	34.7	26.5	23.5	55.2	32.9	28.1	73.5	40.6	-	-	-	19.8	48.6	28.1
GFZSL [28]	0.0	39.6	0.0	0.0	45.7	0.0	1.8	80.3	3.5	2.5	80.1	4.8	0.0	83.3	0.0
LAGO [4]	18.8	33.1	23.9	21.8	**73.6**	33.7	23.8	67.0	35.1	-	-	-	-	-	-
PSEUDO [19]	19.0	32.7	24.0	23.0	51.6	31.8	22.4	80.6	35.1	-	-	-	15.4	71.3	25.4
KERNEL [36]	21.0	31.0	25.1	24.2	63.9	35.1	18.3	79.3	29.8	18.9	82.7	30.8	11.9	76.3	20.5
TVN [39]	18.2	28.9	22.3	21.6	47.5	29.7	18.2	87.5	30.2	-	-	-	8.8	59.1	15.4
VZSL [29]	15.2	23.8	18.6	17.1	37.1	23.8	22.3	77.5	34.6	-	-	-	8.4	75.5	15.1
RELNET [27]	11.1	20.0	14.3	14.0	35.7	20.1	22.9	76.9	35.3	18.6	87.3	30.6	11.5	60.9	19.4
PSR [2]	20.8	37.2	26.7	24.6	54.3	33.9	-	-	-	20.7	73.8	32.3	13.5	51.4	21.4
GAFE [18]	19.6	31.9	24.3	22.5	52.1	31.4	25.5	76.6	38.2	36.8	78.3	40.0	15.8	68.1	25.7
MSEA	12.3	23.1	16.1	11.1	54.0	18.4	1.9	86.1	3.7	2.2	88.5	4.3	1.5	83.7	2.9
MSEA+GPAC	**33.6**	23.7	**27.8**	**34.4**	37.5	35.9	36.3	45.2	40.3	33.0	48.0	39.1	**23.7**	45.9	**31.3**
ALE [1]	21.8	33.1	26.3	23.7	62.8	34.4	16.8	76.1	27.5	14.0	81.8	23.9	9.0	78.1	16.1
ALE+GPAC	23.5	31.3	26.8	30.0	47.5	**36.8**	**41.4**	60.4	**49.1**	**37.0**	77.5	**50.1**	19.4	59.4	29.3
SAE [14]	17.1	28.1	21.3	17.4	50.7	25.9	13.9	74.8	23.5	14.3	79.0	24.3	6.7	59.6	12.1
SAE+GPAC	33.1	21.9	26.3	33.4	39.0	36.0	36.1	45.4	40.2	31.1	52.6	39.1	20.0	50.3	28.6

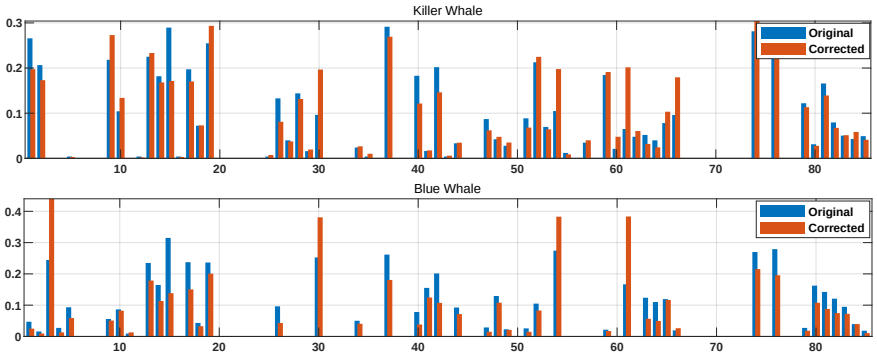


Figure 4: The original and the corrected attributes (ALE+GPAC) on AWA1, the upper figure is 'killer whale' from the seen classes, and the bottom one is 'blue whale' from the unseen classes.

et al. in [34], which uses the test unseen accuracy (ts), test seen accuracy (tr), and harmonic mean (H) to evaluate the performance.

In the first section of this paper, we have claimed that our GPAC can be effectively extended to many linear methods, so here we additionally implement the Mean Square Error of Attribute (MSEA) and apply our GPAC on it, and it can be represented as,

$$\begin{aligned}\mathcal{L}_M =&\|\boldsymbol{X}^T\boldsymbol{W} - (\boldsymbol{A}\boldsymbol{B})^T\|_F^2 + \gamma\|\boldsymbol{W}\|_{2,1} \\ &+ \alpha\|\boldsymbol{A} - \boldsymbol{S}\|_F^2 + \beta\|\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I}\|_F^2.\end{aligned} \tag{17}$$

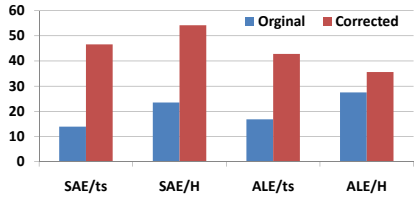
The iterative result of Eq. 17 can be solved similar as that in SAE+GPAC,

$$\begin{cases}\boldsymbol{W} =(\boldsymbol{X}\boldsymbol{X}^T + \gamma\boldsymbol{D})^{-1}\boldsymbol{X}\boldsymbol{A}\boldsymbol{B}^T \\ a_{ij} =a_{ij}|\dfrac{(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{B}^T + \alpha\boldsymbol{S} + 2\beta\boldsymbol{A})_{ij}}{(\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T + \alpha\boldsymbol{A} + 2\beta\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{A})_{ij}}|.\end{cases}, \tag{18}$$

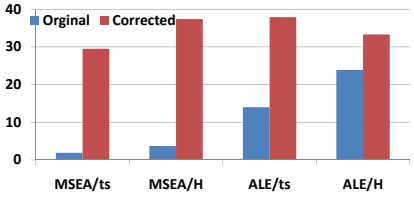
where, $\boldsymbol{D}$ is same as that in Eq. 9.

We compare our method with 17 recently proposed methods, including DAP [15], CONSE [24], SSE [40], LATEM [33], ALE [1], SAE [14], CVAE-ZSL [23], PRESERVE [3], CDL [13], GFZSL [28], LAGO [4], PSEUDO [19], KERNEL [36], TVN [39], VZSL [29], RELNET [27], PSR [2], and GAFE [18], and all the results on five datasets are recorded in Tab. 1. From this table, we can clearly

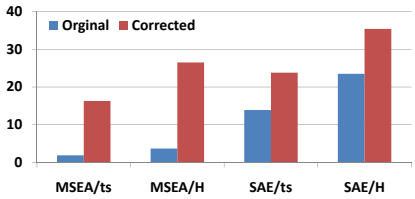
find that the extension of GPAC on MSEA and ALE can outperform all the listed state-of-the-art methods, especially on the datasets of AWA1, AWA2 and aPY. Concretely, MSEA extended our by GPAC can reach the best performance on SUN and aPY, and exceed 1.0% and 2.0% respectively compared to the second best method ALE+GPAC, which is also our GPAC based method. On other three datasets, our GPAC based ALE can achieve the best performance, and obtain 0.8%, 1.9%, and 11.0% improvement respectively in each dataset compared to the second best methods. In addition, from the bottom part of the Tab. 1, it is clearly observed that after the extension of our GPAC, all the methods including MSEA, ALE, and SAE can get improved significantly.



(a) MSEA+GPAC



(b) SAE+GPAC

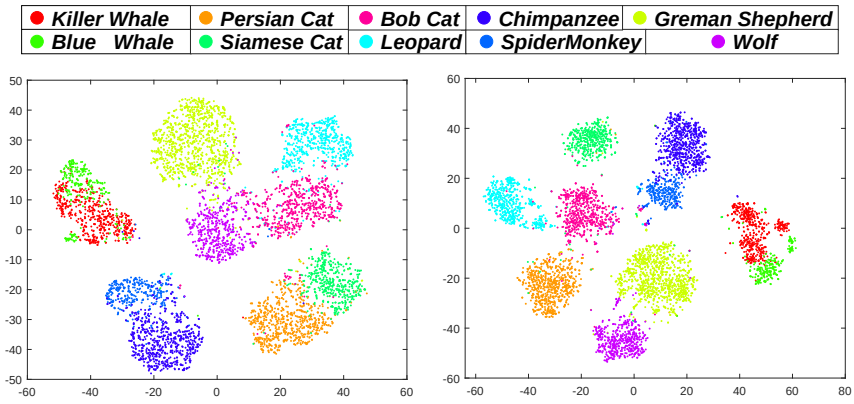


(c) ALE+GPAC

Figure 5: The effects of cross learning on AWA1.

4.3. Detailed Analysis

Visualization in Attribute Space. The objective of our GPAC in attribute space is to disperse all classes and make them more discriminative. Thus, in order to have a more intuitive understanding, we employ t-SNE [21] on AWA1 to illustrate the distributions of samples in this space. Specifically, we choose representative class pairs whose cosine similarities of prototypes in original at-



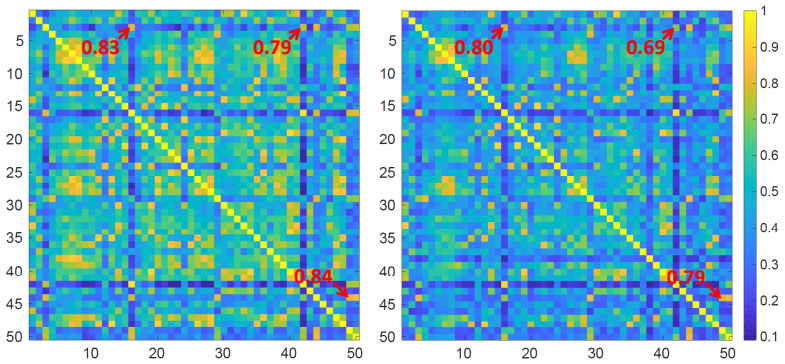
(a) Original ALE model (b) ALE+GPAC

Figure 6: The distribution of some selected similar classes in AWA1.

tribute space are about 0.8, *i.e.*, they are very similar and hard to be classified. After that, we finally get five pairs, including eight seen classes and two unseen classes, which can be found in the legend of Fig. 6. In Fig. 6, we illustrate the distribution of the samples from the selected classes with original ALE model and our ALE+GPAC. From this figure, it can be clearly seen that samples of 'Killer Whale (Seen) and 'Blue Whale (Unseen) are overlapped in original ALE model, while our GPAC+ALE can separate them effectively. This phenomenon can also be found in seen-seen pairs, *e.g.*, 'Persian Cat and 'Siamese Cat, which indicates our GPAC can perform well not only in the seen classes, but also in the unseen classes.

Attribute Correction. Since our GPAC focuses on the attribute correction, it is necessary to show what the corrected attribute is after the optimization. We select a pair of classes, 'killer whale' from the seen classes and 'blue whale' from the unseen classes, which are hard to be classified in original ALE model (shown in Fig. 6), and illustrate them with bar figures in Fig. 4. There are two phenomena worthy of attention. The first is that when the original attribute items equal zero, the corrected attribute items also equal zero, which indicates that our GPAC does not change the existence of the properties for each class. For example, when the class 'killer whale' does not have the prop-

erty 'stripe', the correct attribute also does not include it, which is reasonable for realistic world. The second phenomenon is when the two attribute values are similar for the two classes, the correction of GPAC is to make them change to opposite directions or vary with different amplitudes. For example, the 19^{th} item and the 74^{th} item have different change directions, which can guarantee the corrected attributes are more discriminative.



(a) Original attributes (b) Corrected attributes

Figure 7: The cosine similarities of class attributes on AWA1.

Similarity Check. Since it is known that the more different the class attributes are from each other, the easier the input samples can be classified, we further illustrate the change of similarities of class attributes in this section. We calculate the cosine similarity of each class attribute on AWA1, and visualize the similarity matrix in Fig. 7. Specifically, vectors from 0# to 40# in matrix are seen classes prototypes and the rest belongs to the unseen classes. Fig. 7(a) demonstrates the similarity of original attributes, while Fig. 7(b) illustrates that of the corrected attributes learned with ALE+GPAC. From the comparison of two figures, we can obviously found that the corrected attributes are much more discriminative from each other, which demonstrates the effectiveness of our model. Noted that not only seen classes become more discriminative against seen, seen against unseen, and unseen against unseen also become more discriminative. For example, we pick out three pairs, including the similarities of

'Killer Whale' (Seen) and *'Hamper-back Whale'* (Seen), *'Killer Whale'* (Seen) and *'Blue Whale'* (Unseen), *'Walrus'* (Unseen) and *'Seal'* (Unseen). These three pairs are very similar under the original attributes, around 0.8 in cosine similarity, which makes them hard to be classified, while the similarities of the corrected attributes are decreased, which shows the superiority of our GPAC.

Cross Learning. To verify whether the corrected attributes optimized with a single method are suitable for other methods, we conduct three cross learning experiments on AWA1, including the experiment with corrected attributes learned with MSEA+GPAC on traditional SAE and ALE, the experiments with corrected attributes learned with SAE+GPAC on traditional MSEA and ALE, and the experiments with corrected attributes learned with ALE+GPAC on traditional MSEA and SAE. The results are recorded in Fig. 5. From this figure, it can be clearly discovered that with the corrected attributes all these methods are significantly improved, which further demonstrates the effectiveness of the proposed method.

5. Conclusion

In this paper, we proposed a novel and effective GPAC module to accommodate conventional ZSL models in GZSL tasks. GPAC corrected the original expert-annotated attributes while preserving the realistic meanings of them. A similarity retention and an orthogonal constraints were introduced to make the corrected attribute make them more discriminative. The paper demonstrated two examples of plugging the GPAC module into typical ZSL frameworks, and an iterative optimization strategy was employed. Extensive experiments on five popular datasets were conducted and the results showed that the proposed GPAC can not only improve the traditional ZSL methods up to a state-of-the-art level in GZSL. Most importantly, detailed analysis on corrected attributes further validated the effectiveness of our GPAC. This paper has provided two good practices for future work. One is to seek an unified way to merge plug-in modules and models. The second is how to properly incorporate prior human

knowledge without losing the interpretability and eventually can give feedback and contribute to knowledge-level discovery and correction.

Acknowledgement

This work was supported in part by National Natural Science Foundation of China (No.61872187, No.61929104), and in part by Medical Research Council (MRC) Innovation Fellowship (MR/S003916/1), and in part by the "111" Program (No.B13022).

References

[1] Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C., 2013. Label-embedding for attribute-based classification, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition.

[2] Annadani, Y., Biswas, S., 2018a. Preserving semantic relations for zero-shot learning, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition.

[3] Annadani, Y., Biswas, S., 2018b. Preserving semantic relations for zero-shot learning, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition.

[4] Atzmon, Y., Chechik, G., 2018. Probabilitic and-or attribute grouping for zero-shot learning, in: The Conference on Uncertainty in Artificial Intelligence.

[5] Chao, W.L., Changpinyo, S., Gong, B., sha, F., 2016. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild, in: European Conference on Computer Vision.

[6] Farhadi, A., Endres, I., Hoiem, D., Forsyth, D., 2009. Describing objects by their attributes, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition.

[7] Geng, C., Tao, L., Chen, S., 2020. Guided cnn for generalized zero-shot and open-set recognition using visual and semantic prototypes. Pattern Recognition 102, 107263.

[8] Guo, Y., Ding, G., Han, J., Tang, S., 2018. Zero-shot learning with attribute selection, in: AAAI Conference on Artificial Intelligence, pp. 6870–6877.

[9] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition, pp. 770–778.

[10] Huang, H., Wang, C., Yu, P.S., Wang, C.D., 2019. Generative dual adversarial network for generalized zero-shot learning, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition, pp. 801–810.

[11] Huang, L., Ji, H., Cho, K., Dagan, I., Riedel, S., Voss, C.R., 2018. Zero-shot transfer learning for event extraction, in: Annual Meeting of the Association for Computational Linguistics, pp. 2160–2170.

[12] Jayaraman, D., Grauman, K., 2014. Zero-shot recognition with unreliable attributes, in: Advances in Neural Information Processing Systems.

[13] Jiang, H., Wang, R., Shan, S., Chen, X., 2018. Learning class prototypes via structure alignment for zero-shot recognition, in: European Conference on Computer Vision, pp. 118–134.

[14] Kodirov, E., Xiang, T., Gong, S., 2017. Semantic autoencoder for zero-shot learning, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition.

[15] Lampert, C.H., Nickisch, H., Harmeling, S., 2009. Learning to detect unseen object classes by between-class attribute transfer, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition.

[16] Li, Z., Yao, L., Chang, X., Zhan, K., Sun, J., Zhang, H., 2019. Zero-shot event detection via event-adaptive concept relevance mining. Pattern Recognition 88, 595 – 603.

[17] Liu, S., Long, M., Wang, J., Jordan, M.I., 2018. Generalized zero-shot learning with deep calibration network, in: Advances in Neural Information Processing Systems 31, pp. 2005–2015.

[18] Liu, Y., Xie, D., Gao, Q., Han, J., Wang, S., Gao, X., 2019. Graph and autoencoder based feature extraction for zero-shot learning, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, pp. 3038–3044.

[19] Long, T., Xu, X., Li, Y., Shen, F., Song, J., Shen, H., 2018. Pseudo transfer with marginalized corrupted attribute for zero-shot learning, in: ACM Conference on Multimedia, pp. 1802–1810.

[20] Long, Y., Liu, L., Shao, L., Shen, F., Ding, G., Han, J., 2017. From zero-shot learning to conventional supervised classification: Unseen visual data synthesis, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition, pp. 1627–1636.

[21] Maaten, L.v.d., Hinton, G., 2008. Visualizing data using t-sne. Journal of Machine Learning Research 9, 2579–2605.

[22] Mikolov, T., Chen, K., Corrado, G.S., Dean, J., 2013. Efficient estimation of word representations in vector space, in: International Conference on Learning Representations.

[23] Mishra, A., Reddy, S.K., Mittal, A., Murthy, H.A., 2018. A generative model for zero shot learning using conditional variational autoencoders, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition.

[24] Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G.S., Dean, J., 2014. Zero-shot learning by convex combination

of semantic embeddings, in: International Conference on Learning Representations.

[25] Patterson, G., Xu, C., Su, H., Hays, J., 2014. The sun attribute database: Beyond categories for deeper scene understanding. International Journal of Computer Vision 108, 59–81.

[26] Schonfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., Akata, Z., 2019. Generalized zero-and few-shot learning via aligned variational autoencoders, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition, pp. 8247–8255.

[27] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M., 2018. Learning to compare: Relation network for few-shot learning, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition.

[28] Verma, V.K., Rai, P., 2017. A simple exponential family framework for zero-shot learning, in: The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Springer. pp. 792–808.

[29] Wang, W., Pu, Y., Verma, V.K., Fan, K., Zhang, Y., Chen, C., Rai, P., Carin, L., 2018. Zero-shot learning via class-conditioned deep generative models, in: AAAI Conference on Artificial Intelligence.

[30] Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P., 2010. Caltech-UCSD birds 200. Technical Report. California Institute of Technology.

[31] Wu, F., Jing, X., Dong, X., Hu, R., Yue, D., Wang, L., Ji, Y., Wang, R., Chen, G., 2020a. Intraspectrum discrimination and interspectrum correlation analysis deep network for multispectral face recognition. IEEE Transactions on Cybernetics 50, 1009–1022.

[32] Wu, F., Jing, X.Y., Wu, Z., Ji, Y., Dong, X., Luo, X., Huang, Q., Wang, R., 2020b. Modality-specific and shared generative adversarial network for cross-modal retrieval. Pattern Recognition 104, 107335.

[33] Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B., 2016. Latent embeddings for zero-shot classification, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition.

[34] Xian, Y., Lampert, C.H., Schiele, B., Akata, Z., 2018. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. IEEE Transactions on Pattern Analysis and Machine Intelligence 41, 2251–2265.

[35] Xian, Y., Schiele, B., Akata, Z., 2017. Zero-shot learning-the good, the bad and the ugly, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition.

[36] Zhang, H., Koniusz, P., 2018. Zero-shot kernel learning, in: Proceedings of the IEEE Conference on Computer Visiona and Pattern Recognition, pp. 7670–7679.

[37] Zhang, H., Liu, J., Yao, Y., Long, Y., 2020a. Pseudo distribution on unseen classes for generalized zero shot learning. Pattern Recognition Letters 135, 451 – 458.

[38] Zhang, H., Liu, L., Long, Y., Zhang, Z., Shao, L., 2020b. Deep transductive network for generalized zero shot learning. Pattern Recognition 105, 107370.

[39] Zhang, H., Long, Y., Guan, Y., Shao, L., 2019. Triple verification network for generalized zero-shot learning. IEEE Transactions on Image Processing 28, 506–517.

[40] Zhang, Z., Saligrama, V., 2015. Zero-shot learning via semantic similarity embedding, in: International Conference on Computer Vision.