ISAE-GZSL: Interpolated Incomplete Semantic Attribute Enhancement for Generalized Zero-Shot Learning

Xiaomeng Zhang[1] and Zhi Zheng[1,2,*]

[1] College of Computer and Cyber Security, Fujian Normal University, Fuzhou, 350117, China
[2] College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, China
*Corresponding author
E-mail: zhengz@fjnu.edu.cn

Abstract. Generalized zero-shot learning (GZSL) is focused on recognizing classes, both seen and unseen, without the need for labeled data specifically for the unseen classes. GZSL has attracted much attention by transforming the traditional GZSL into a fully supervised learning task. Most GZSL methods use a single semantic attribute(each category can only correspond to a specific semantic attribute) plus Gaussian noise to generate visual features, assuming a one-to-one correspondence between these visual features and single semantic attributes. However, in practice, there may be cases of attribute missingness in images, leading to visual features that lack certain attributes, thus failing to achieve a good mapping between semantic attributes and visual features. Therefore, visual features of the same class should have diverse semantic attributes. To address this issue, we propose a new method for enhancing semantic attributes called "Interpolated Semantic Attribute Enhancement for Generalized Zero-Shot Learning(ISAE-GZSL)." This method uses interpolation to deal with the problem of semantic attribute missingness in real-world situations, thereby enhancing semantic diversity and generating more realistic and diverse visual features. We assess the performance of the proposed model across four benchmark datasets, and the findings demonstrate substantial enhancements over current state-of-the-art methods, especially in handling categories with severe attribute missingness in the datasets.

Keywords: Generalized zero-shot learning(GZSL) · semantic attribute · feature generation · incomplete attribute.

1 Introduction

In recent years, deep learning has rapidly advanced, but it often relies on large amounts of annotated image data to effectively train and generalize models. However, in practical scenarios, new classes frequently arise, often with few or

no training samples. Training with such scarce data can lead to poor model generalization. Hence, Zero-Shot Learning (ZSL) has emerged. ZSL aims to identify new classes by leveraging auxiliary semantic knowledge as supplementary information, mimicking the human cognitive process [1].

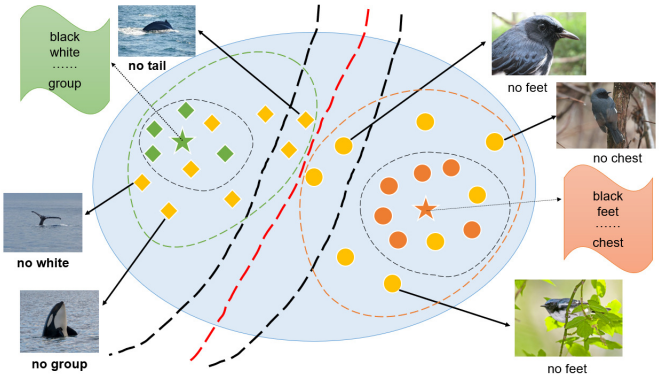


Fig. 1. ⋆ represents class-level semantic attributes, yellow ⋄ and ○ represent real visual features, green ⋄ and orange ○ represent visual features generated by a single semantic attribute plus Gaussian noise, red dashed lines represent correct classification boundaries, and two black dashed lines represent incorrect classification boundaries.

ZSL aims to classify unseen classes by establishing a mapping between the visual and semantic domains. To facilitate the transfer of knowledge from seen training data to unseen test data [2], category-level semantic attributes must be utilized as supplementary information [2,3] to bridge the knowledge gap between seen and unseen classes. These attributes play a crucial role as they are the sole basis for inferring the visual features of unseen classes. Based on the scope of classification, ZSL methods are divided into two categories: Traditional ZSL and GZSL [4]. Traditional ZSL focuses on predicting unseen classes, while GZSL extends this capability to include predictions for both unseen and seen classes. In recent years, GZSL has garnered increasing attention due to its practicality and increased difficulty. Therefore, this paper also adopts the GZSL framework.

To address the issue of seen class bias [5], GZSL methods have emerged, focusing on generating visual features for unseen classes. Specifically, Generative Adversarial Networks (GANs) [6] and Variational Autoencoders (VAEs) [7] are commonly used to generate features, thereby transforming the GZSL problem into a traditional supervised classification task. In this paper, the classic generative model framework "Feature Generating Networks for Zero-Shot Learning" (abbreviated as f-CLSWGAN) [8] is used.

In GZSL, auxiliary semantic attributes are typically associated with specific classes, indicating that each class is characterized by a single semantic attribute. The majority of GZSL approaches [9,10,11] synthesize visual features by directly combining the semantic attribute of a single class with Gaussian noise. However,

in practical scenarios, attribute incompleteness in images often occurs, causing visual features generated from a single semantic attribute to inadequately represent the issue of attribute incompleteness in reality. This mismatch leads to an inauthentic distribution of generated visual features, where the single semantic attribute fails to map accurately to the generated features, thus creating unstable classification boundaries. Consequently, the model's ability to handle samples with missing attributes is compromised. Illustrated in Fig. 1, the visual features of some images may not contain all attribute information, the distribution of visual features from images with incomplete attributes is present within the real visual feature distribution but is absent from the distribution of features generated from a single semantic attribute, adversely affecting classification performance.

To better simulate real-world scenarios (cases of attribute incompleteness) and enhance the diversity of generated visual features, we propose a new method called "Interpolated Semantic Attribute Enhancement for Generalized Zero-Shot Learning" (abbreviated as ISAE-GZSL). This method aims to enhance the diversity of semantic attributes by interpolating the values of incomplete semantic attributes, thereby improving the diversity of generated visual features. Initially, the Expectation-Maximization (EM) algorithm is applied to cluster attributes using the extracted attribute word vectors. Then, within each clustered group, we use an interpolation method based on the correlation of semantic attributes to set the values of incomplete semantic attributes. Next, to further enhance the diversity of generated visual features, we introduce a diversity loss function. Finally, we add a self-supervised learning module to enable the model to learn how to generate visual features that are more diverse and realistic under incomplete semantic attribute conditions. Through these improvements, our model has been optimized for handling missing attributes, thereby enhancing the quality and diversity of generated samples.

The main contributions of this work are outlined as follows.

- We propose a novel approach that uses interpolation to address the issue of missing attributes in practice. This method considers scenarios where attributes are missing in real-world situations, enhancing the realism and diversity of the generated visual features, thereby improving accuracy.
- Following the stringent evaluation protocol introduced in [4], our model is assessed across four well-known zero-shot learning datasets, delivering competitive performance across all of them.

Here's the breakdown of the remaining sections: Section 2 delves into related research, Section 3 offers a comprehensive outline of the ISAE-GZSL approach, Section 4 showcases experimental outcomes, and Section 5 encapsulates our research discoveries.

2 Related Work

In recent years, with the continuous development of generative models, semantic-based ZSL generation methods have emerged. For example, f-CLSWGAN

[8] used Generative Adversarial Networks (GAN) to generate visual features of unseen classes for classification. Additionally, the LisGAN method [12] utilized semantic descriptions to constrain the generation of features for unseen classes, ensuring that they are derived from randomly generated noise. The combination of GAN and Variational Autoencoder (VAE) in F-VAEGAN-D2 [13] resulted in the generation of more realistic and diverse visual features for unseen classes. Tf-vaegan [14] further introduced a Semantic Embedding Decoder, which enforces cycle-consistency constraints on semantic embeddings during training, feature synthesis, and classification stages. CE-GZSL [9], based on f-CLSWGAN, used contrastive learning to achieve supervision over instances within and between classes. Traditional generation methods typically synthesize visual features by directly using single-class semantic attributes along with Gaussian noise to enrich the visual features. These generated visual features are assumed to contain all semantic attributes (i.e., the generated visual features contain complete semantic attributes). However, in reality, since images themselves may lack some attributes, the extracted visual features often suffer from semantic attribute incompleteness. Therefore, when there is a mismatch between visual features and semantic attributes, the model fails to correctly identify the class, resulting in a one-to-many scenario (where one semantic attribute corresponds to multiple visual features within the same class). We believe that the semantic description of visual features of the same class should be diverse. To achieve a many-to-many scenario (where one semantic attribute corresponds to multiple visual features and vice versa within the same class), this paper proposes an interpolation-based method to enhance semantic attributes, addressing the issue of semantic attribute incompleteness in generation methods.

3 Method

Here we initially present certain symbols and problem definitions. For training, we utilize s visible classes, while u unseen classes are employed for testing. $\mathcal{X}$ represents visual features, $\mathcal{Y}$ represents class labels, and a represents class semantic attributes. The sample set of seen classes is $\mathcal{X}^s = \left\{x^1, x^2, ..., x^s\right\}$, and the corresponding label set of seen classes is $\mathcal{Y}^s = \left\{y^1, y^2, ..., y^s\right\}$, where s represents the number of samples in the seen classes. The sample set of unseen classes is $\mathcal{X}^u = \left\{x^{s+1}, x^{s+2}, ..., x^{s+u}\right\}$ and the corresponding label set of unseen classes is $\mathcal{Y}^u = \left\{y^{s+1}, y^{s+2}, ..., y^{s+u}\right\}$, where u represents the number of samples in the unseen classes. The set of attributes of seen classes is represented as $\mathcal{A}^s = \left\{a^1, a^2, ..., a^s\right\}$, where s is the count of semantic attributes of the seen classes, and the set of attributes of unseen classes is represented as $\mathcal{A}^u = \left\{a^{s+1}, a^{s+2}, ..., a^{s+u}\right\}$, where u is the count of semantic attributes of the unseen classes. In this study, we opt for the more realistic and challenging scenario of GZSL. The training dataset tuple $\mathcal{T}^{train}$ can be represented as $\mathcal{T}^{train} = \{\mathcal{X}^s, \mathcal{Y}^s, \mathcal{A}\}$; the testing dataset tuple $\mathcal{T}^{test}$ can be represented as $\mathcal{T}^{test} = \{\mathcal{X}, \mathcal{A}\}$, where $\mathcal{X} = \mathcal{X}^s \cup \mathcal{X}^u$, $\mathcal{Y} = \mathcal{Y}^s \cup \mathcal{Y}^u$, $\mathcal{A} = \mathcal{A}^s \cup \mathcal{A}^u$ and $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$. Therefore, the main goal of GZSL is to train a classifier $\boldsymbol{F}$: $\mathcal{X} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u$, where

the search space includes the entire label space $\mathcal{Y}^s \cup \mathcal{Y}^u$. The process of ISAE-GZSL is shown in Fig. 2.

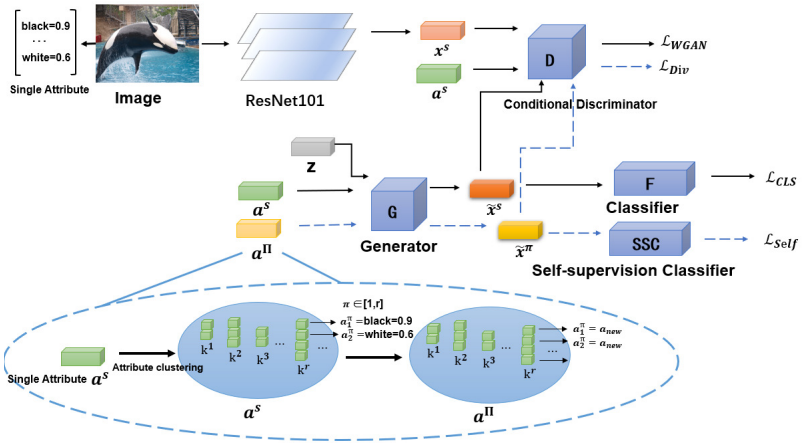


Fig. 2. ISAE-GZSL Architecture Diagram. ISAE-GZSL consists of a feature generation module f-CLSWGAN, a classifier (***F***), and a self-supervised classifier (**SSC**). Given a visual feature and attribute pair (x^s, a^s), $\mathcal{L}_{WGAN}$ trains the conditional discriminator ***D*** by using real and synthetic features as inputs. Additionally, incomplete visual features $\tilde{x}^\pi$ and incomplete attribute a_{new} are introduced and optimized using the diversity loss $\mathcal{L}_{Div}$. Meanwhile, **SSC** further improves model performance through self-supervised loss optimization. We optimized the parameters of the ***G***, ***D***, ***F***, and **SSC** modules during training.

3.1 Reviewing f-CLSWGAN Method

The generative model used in this paper is the f-CLSWGAN model, which aims to improve the performance of the classifier in the GZSL task by generating realistic features to compensate for the missing unseen classes in the training set using a generative adversarial network (GAN). In the f-CLSWGAN method, there are three key components: the conditional generator $G(z, a))$ (also denoted as ***G***), the discriminator $D(x^s, a^s)$ (also denoted as ***D***), and the classifier ***F***. ***G*** generates realistic visual features based on the semantic attributes of the classes. The input semantic attributes a include semantic descriptions of both seen and unseen classes, and the output is the generated visual features $(\tilde{x}^s)$. ***D*** takes as input the features generated by the generator and real features, combined with the semantic attributes of the classes, to learn to distinguish between them and encourage the generator to produce more realistic features. The output of the discriminator is a real value between 0 and 1, indicating the authenticity of the input features. ***F*** ensures that the features can be correctly classified into the corresponding classes. Both ***G*** and ***D*** are conditioned on embeddings using the Wasserstein GAN (WGAN) loss function.

The WGAN loss ($\mathcal{L}_{WGAN}$) is:

$$\mathcal{L}_{WGAN} = \mathbb{E}\left[D\left(x^s, a^s\right)\right] - \mathbb{E}\left[D(\tilde{x}^s, a^s)\right] - \lambda\mathbb{E}\left[\left(|| \bigtriangledown D\left(x', a^s\right)||_2 - 1\right)^2\right], \tag{1}$$

where $x' = \tau x^s + (1-\tau)\,\tilde{x}^s$, with τ sampled from a uniform distribution $U\,(0,1)$, represents the blending of visual feature x^s and a synthesized feature $\tilde{x}^s$ to produce an intermediate feature x', λ is a penalty coefficient.

The f-CLSWGAN model minimizes the classification loss of the generated features by using the negative log likelihood function to ensure that the generated features are suitable for training the discriminator.

The classification loss ($\mathcal{L}_{CLS}$) is:

$$\mathcal{L}_{CLS} = -\mathbb{E}_{\tilde{x}\sim p\tilde{x}}\left[\log P(y^s)|\,\tilde{x}^s; \theta\right], \tag{2}$$

where $\tilde{x}^s = G(a^s, z)$ represents the generation of $\tilde{x}^s$ from z(z is random noise) and a^s, with $\tilde{y}^s$ denoting the class label of $\tilde{x}^s$. θ is the parameter set of a linear softmax classifier that is pre-trained on the real features of seen classes. The term $(P(y^s|(\tilde{x}^s); \theta)$ signifies the classifier's prediction of label $\tilde{y}^s$ based on the feature $\tilde{x}^s$.

Therefore, the total loss of f-CLSWGAN is:

$$\mathcal{L}_{CLSWGAN} = \min_{G}\max_{D}\mathcal{L}_{WGAN} + \nu\mathcal{L}_{CLS}, \tag{3}$$

where the parameter ν is specific to the classifier's weighting.

Through the optimization of these loss functions, the f-CLSWGAN model ensures that the generated features are indistinguishable from real features to the discriminator, thereby enhancing the realism and diversity of the generated features.

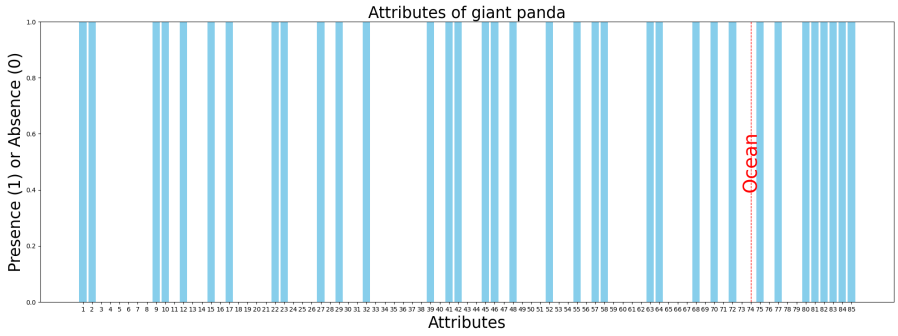


Fig. 3. Attribute of giant panda in the AWA1 dataset.

3.2 ISAE-GZSL Model

Attribute clustering. Traditional methods for generating visual features often rely on a single semantic attribute per class (assuming that a single semantic attribute already contains all relevant information for the class). However,

in practical applications, the visual features of some images may not contain all attribute information. For example, a photo of a bird perched on a tree, its feet hidden by foliage, may not exhibit attribute information regarding its claws, or a photo of a shark may lack many attribute details due to the difficulty of capturing the entire body. Therefore, the visual features of these images may not encompass all relevant semantic attribute information. Visual features generated based on a single semantic attribute may not adequately simulate real-world attribute incompleteness, leading to a lack of realism in the distribution of generated visual features and unstable classification boundaries. Consequently, the performance of the model in handling samples with missing attributes may also be affected.

To address this situation, we borrow the concepts of complete and incomplete attributes proposed in "Boosting Generative Zero-Shot Learning by Synthesizing Diverse Features with Attribute Augmentation" (abbreviated as SDFA^2) [15], which has achieved good results, to generate visual features. The specific steps are as follows: First, set the dimensions of certain semantic attributes to 0 to indicate that the attribute is missing (e.g., setting the dimension for the feet of a bird to 0 indicates missing attribute information for the feet). When defining semantic attributes, some attribute dimensions do not exist or are not meaningful for certain classes. Therefore, in these classes, the values of these attributes are set to 0 (e.g., the "ocean" attribute dimension in the giant panda class is set to 0 because giant panda do not have this attribute, as depicted in Fig. 3, where attributes are represented in an 85-dimensional space, with blue indicating presence and white indicating absence of the attribute. It can be observed that the position marked by the red dashed line corresponds to the attribute "ocean" with a value of 0). To handle high-dimensional attribute data, we use the Expectation-Maximization (EM) algorithm [16] to estimated mean μ_π, and then cluster the attributes into r clusters (i.e., $k = \left\{k^1, k^2, ..., k^r\right\}$). Optimal clustering is achieved by minimizing the squared error $\mathcal{L}_{EM}$:

$$\mathcal{L}_{EM} = \sum_{\pi=1}^{r} \sum_{v \epsilon k^\pi} \left\| v - \mu_\pi \right\|, \tag{4}$$

where μ_π is represents the average vector of cluster k^π , and the word vectors $v = \left\{v^1, v^2, ..., v^j\right\} = \text{Word2vec}\left\{a^1, a^2, ..., a^j\right\}$, where Word2vec is a technique that converts words into vectors, j is the number of attribute dimensions.

In the context of attribute grouping into r categories, SDFA^2 defined a^{Π} as the incomplete attribute, where π ranges from 1 to r, setting the π-th group attribute of the semantic attribute a to 0. Setting the attributes of the π-th group to 0 simulates incomplete attribute features. This method can more realistically simulate the incompleteness of visual features and enhance the performance of the model in handling samples with missing attributes.

Interpolation Based on Semantic Attribute Correlation. Nevertheless, we recognize that directly zeroing out semantic attributes might result in generated samples deviating considerably from reality. In practical scenarios, it is rare for all semantic attributes of an object to be 0. This setting may result in unnatural generation outcomes, akin to forcibly removing certain attributes,

which is not quite realistic in real-world scenarios. Therefore, we consider using an interpolation method based on attribute correlation. Through interpolation, we can learn more information from similar attributes(such as estimating 'brown' based on the presence of 'black' and 'gray'), better estimate the values of missing attributes, and thus more naturally simulate the situation of missing attributes. This results in samples that better fit real-world scenarios, making the generated samples more diverse and realistic.

- For each cluster $k^{\pi}, \pi\epsilon[1,r]$ in a^{Π}, calculate the average value A^{π}_{avg} of all non-zero attribute values a' :

$$A^{\pi}_{avg} = \frac{1}{L}\sum_{n=1}^{L} a'_n, \tag{5}$$

  where a'_n represents the n-th non-zero attribute value of a', L is the length of a'.
- For each attribute a^{π}_i in the cluster k^{π}, calculate the intra-cluster attribute similarity ω:

$$\omega = \mathrm{Cos}(a^{\pi}_{b\times j}, a^{t}_{b\times j}) = \frac{a^{\pi}_i \times a^{\pi}_t}{\|a^{\pi}_i\| \, \|a^{\pi}_t\|}, \tag{6}$$

  where b is the batch size, j is the number of attribute dimensions, a^{π}_i is current attribute, and $a^{\pi}_t \epsilon (k^{\pi} - a^{\pi}_i)$. Both a^{π}_i and a^{π}_t are two-dimensional matrices of shape $(b \times j)$.
- Assuming the current attribute is a^{π}_i , and a^{ϖ}_i is the most similar attribute within the cluster that meets the similarity threshold, excluding a^{π}_i, based on L, we set the incomplete attribute values in the following three cases:

$$a_{new} = \begin{cases} \omega \times a^{\pi}_i + (1-\omega) \times a^{\varpi}_i \ , & \text{if } L > 1 \ \textit{and} \ \omega >= \Phi \\ A^{\pi}_{avg} \ , & \text{if } L > 1 \ \textit{and} \ \omega < \Phi \\ a^{\pi}_i \ , & \text{if } L = 1 \\ 0 \ , & \text{if } L = 0 \end{cases}, \tag{7}$$

  when $L > 1$ and $\omega \geq \Phi$ (where Φ is the set threshold, which is taken as 0.9 in this paper.), select the most similar attribute a^{ϖ}_i within the current cluster for linear interpolation. When $L > 1$ and $\omega < \Phi$, use A^{π}_{avg} for interpolation. When $L = 1$, meaning there is only one non-zero value among all attribute values of the current attribute a^{π}_i, set $a_{\mathrm{new}} = a^{\pi}_i$. When $L = 0$, meaning all attribute values of the current attribute a^{π}_i are zero, indicating that this attribute does not exist in the category, directly set the attribute value to 0.

Through the above steps, we can obtain the interpolated attribute values, i.e., the incomplete attribute values(such as setting the foot attribute of a bird to a_{new} when its feet are covered by leaves). Then, use these incomplete attribute values to generate visual features, which simulates the situation of missing attributes in the original samples, thereby making the distribution of generated visual features more realistic.

3.3 Diversity Loss

To increase the diversity of generated visual features and broaden their distribution in the attribute space, it is necessary to ensure greater differences between the generated features corresponding to different attributes (both incomplete and complete semantic attributes). If there are larger differences between attributes, then the generated visual features will also be more diverse. In order to let the model learn to generate more diverse and distinguishable features based on different attributes, thereby enhancing the diversity and authenticity of the generated samples, we introduce a diversity loss L_{Div} for optimization:

$$\mathcal{L}_{Div} = \mathbb{E}\left[D\left(x^s, a^s\right)\right] - \frac{1}{r}\sum_{\pi=1}^{r}\{\mathbb{E}\left[D(\tilde{x}^\pi, a^{\Pi})\right] - \lambda\mathbb{E}\left[\left(|| \bigtriangledown D\left(\check{x}^\pi, a^s\right)||_2 - 1\right)^2\right]\}, \tag{8}$$

where a^s is the complete semantic attribute, a^{Π} is the incomplete semantic attribute, and $\tilde{x}^\pi = G(a^{\Pi}, z)$ is the visual feature generated by a^{Π} and z. The last term is the gradient penalty, where $\check{x}^\pi = \tau x + (1-\tau)\tilde{x}^\pi$, and τ follows a uniform distribution $U(0,1)$. λ is the penalty coefficient, r is the number of clusters into which the attributes are divided.

3.4 Self-supervised Loss

To teach the model how to generate more diverse and realistic visual features based on incomplete semantic attributes, we need the model to recognize which attribute information is missing from the generated visual features. To achieve this, we train a learnable incomplete attribute identification classifier (**SSC** : $x \rightarrow h^\pi$), allowing the model to learn a new mapping relationship between attributes and labels. For incomplete semantic attributes a^{Π} that are missing a certain group of attributes, we use interpolation based on attribute correlation to set all attributes of the π-th group to a new value a_{new} , making the attribute values more realistic and diverse. At the same time, we set a self-supervised label h^π for the generated visual features, indicating that these features lack this group of attributes. Specifically, h^π is initialized as a one-dimensional vector starting from 1. Depending on which group of attributes is missing (e.g., if the r-th group is missing), the corresponding positions in h^π are set to r, resulting in h^π being set as $[r, r, ..., r]$. This is optimized through self-supervised loss $\mathcal{L}_{Self}$:

$$\mathcal{L}_{Self} = -\mathbb{E}\left[\log P(h^\pi|\tilde{x}^\pi;\xi)\right], \tag{9}$$

where $\tilde{x}^\pi = G(a^{\Pi}, z)$ represents the visual feature obtained from the incomplete attribute a^{Π} and z, h^π is the corresponding self-supervised label, and ξ represents the set of learnable parameters of **SSC**.

The overall loss is:

$$\mathcal{L}_{total} = \mathcal{L}_{CLSWGAN} + \kappa\mathcal{L}_{Div} + \eta\mathcal{L}_{Self}, \tag{10}$$

where κ and η are hyperparameters.

Algorithm 1 Interpolated Incomplete Semantic Attribute Enhancement for GZSL(ISAE-GZSL)

Input: Seen data $x^s \in \mathcal{X}$, Unseen data $x^u \in \mathcal{X}$, Random noise z ,Complete semantic attributes $a^s \cup a^u \in \mathcal{A}$, Incomplete semantic attributes $a^{\Pi} \in \mathcal{A}$ and Number of training epochs T.
Training:
for $i = 1$ to T **do**
    $\boldsymbol{G}$ generates visual features by $\tilde{x}^s = G(a^s, z)$ and $\tilde{x}^{\pi} = G(a^{\Pi}, z)$.
    $\boldsymbol{D}$ discriminates between real and generated visual features by Eq.(8).
    Calculate the $\mathcal{L}_{Self}$ by solving Eq.(9).
    Calculate the $\mathcal{L}_{CLS}$ using $\tilde{x}^s$, $\tilde{x}^{\pi}$ and x^s by solving Eq.(2).
    Calculate the $\mathcal{L}_{total}$ by solving the Eq.(10).
end for
Output(Testing): Using $\boldsymbol{F}$ for prediction: $\mathcal{X} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u$.

3.5 Training

Our method utilizes both complete and incomplete semantic attributes when generating visual features. We generate visual features based on different proportions of complete and incomplete semantic attributes. These features are then used to train the classifier, thereby enhancing the diversity and realism of the model. The detailed algorithm for ISAE-GZSL is provided in Algorithm 1.

4 Experiments

Datasets and Experimental settings. We evaluated our method on four benchmark ZSL/GZSL datasets: Apy [3], AWA1 [17], CUB [18] and SUN [19]. Similar to most popular methods, all datasets in this paper were extracted using ResNet101 to obtain 2048-dimensional visual features, without further fine-tuning. Additionally, all datasets followed the consistent visible/invisible class data partitioning and class embedding methods described in [4]. For detailed information, please refer to Table 1. We implemented the proposed method in the PyTorch framework and conducted experiments using an NVIDIA GeForce RTX 3080 GPU.

Evaluation Protocols. During the evaluation phase, for consistency, we compared using the evaluation metrics proposed in [7]. For GZSL, it is necessary to separately calculate the seen class accuracy (S) and the unseen class accuracy (U), where S represents the average per-class Top-1 accuracy for the seen classes, and U represents the average per-class Top-1 accuracy for the unseen classes. The performance of GZSL is evaluated by their harmonic mean: $H = (2 \times S \times U) / (S + U)$. The reason H is used as the key evaluation metric is that it balances the performance between the U and S metrics. A higher H value indicates higher accuracy for both U and S.

Implementation Details. The network optimization employs the Adam algorithm [20], with the parameters β_1 and β_2 set to 0.5 and 0.999, respectively.

Table 1: Evaluation dataset characteristics.

	train images	test images(S/U)	class(S/U)	Granularity	Att
Apy [3]	5932	7924/1483	20/12	Coarse	64
AWA1 [17]	19832	5685/4958	40/10	Coarse	85
CUB [18]	7057	2679/1764	150/50	Fine	312
SUN [19]	10320	1440/2580	645/72	Fine	102

Table 2: State-of-the-art comparisons on four datasets. The best results are highlighted in red and the symbol "*" represents baseline. Here, U represents the Top-1 accuracy for unseen classes, S represents the Top-1 accuracy for seen classes, and H is the harmonic mean of both.

Methods	Apy			AWA1			CUB			SUN		
	U	S	H	U	S	H	U	S	H	U	S	H
f-CLSWGAN* [8]	32.9	61.7	42.9	57.9	61.4	59.6	43.7	57.7	49.7	42.6	36.6	39.4
SDFA^2[15]	38.0	62.8	47.4	59.1	72.8	65.2	51.5	57.5	54.3	48.7	36.9	42.0
ISAE-GZSL(Ours)	**37.2**	**68.9**	**48.3**	**59.6**	**73.4**	**65.8**	**52.2**	**57.1**	**54.5**	**48.1**	**37.1**	**42.2**

Similar to the method described in reference [21], the coefficient λ for the gradient penalty starts from an initial value of 10, which is employed during WGAN training. For consistency, we use the same number of clustering groups r as in [15] and maintain consistency in the ratio of complete semantic attributes to incomplete semantic attributes. For more information, please refer to reference [8].

4.1 Comparison with Baseline

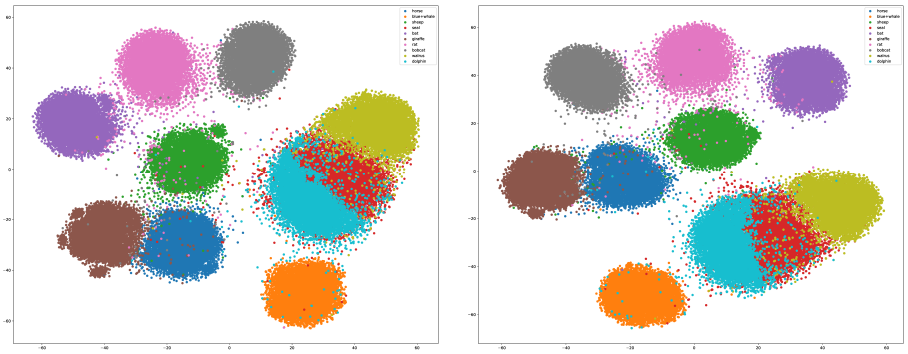
The main baseline model in this paper is f-CLSWGAN, which enhances ZSL performance by generating high-quality sample features using the conditional WGAN-GP framework based on category attributes. For detailed information about it, please refer to Section 3.1.

From Table 2, it can be seen that our method achieves the highest H values. Compared to the f-CLSWGAN method, our method improves the H values by 2.8%, 4.8%, 5.4%, and 6.2% on the SUN, CUB, Apy, and AWA1 datasets, respectively. Compared to the SDFA^2 method, our method improves the H values by 0.2%, 0.2%, 0.6%, and 0.9% on the CUB, SUN, AWA1, and Apy datasets, respectively. Additionally, from Table 2, it can be observed that on the Apy and SUN datasets, the U values have a slight decrease, while on the Apy, AWA1, and CUB datasets, the S values have a significant increase. The reason for the decrease in U values may be that our method fails to better capture the feature

distribution of unseen classes, and the diversity features generated by interpolation fail to fully cover the feature space of unseen classes. At the same time, it can also be observed that the improvement in fine-grained datasets (CUB and SUN) is not as significant as in coarse-grained datasets (Apy and AWA1), possibly because the fine-grained datasets have small differences between categories, so the diversity of attribute changes generated by interpolation is not obvious.

4.2 Visualizations

Our method and the SDFA^2 method both generate visual features through diverse attributes, which is different from traditional methods that typically only introduce Gaussian noise to increase diversity. To further validate the effectiveness of our method, we conducted t-SNE visualization [22] on the unseen classes of the AWA and CUB datasets and compared them with the SDFA^2 method. From Fig.4 and 5, it can be seen that our method has more stable classification boundaries. Additionally, our method also exhibits some diversity in feature distribution, with feature points of the same class having a wider distribution range in the visualization.

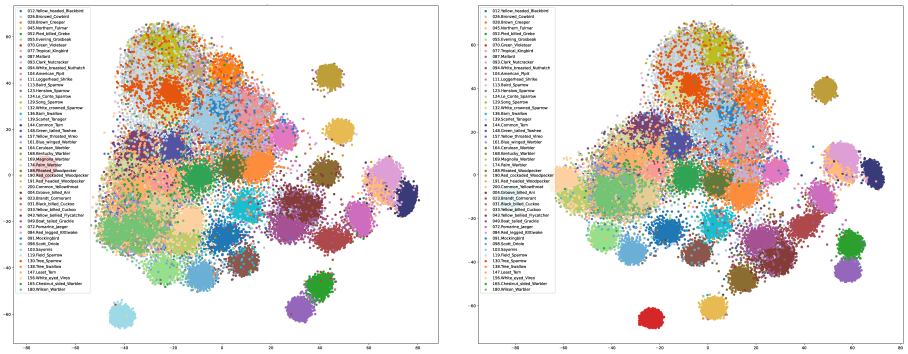


(a) SDFA^2. (b) ISAE-GZSL.

Fig. 4. t-SNE visualization of unseen class images synthesized by SDFA^2(a) and ISAE-GZSL(b) in the GZSL on the AWA1 datasets.

4.3 Comparison of Attribute Missingness

Since the proportion of datasets with missing attributes in our four datasets is relatively small, the improvement of our method in Table 2 is not significant. To further demonstrate the effectiveness of our method in enhancing attributes through interpolation, generating diverse visual features, and mitigating attribute missingness in datasets, we conducted further analysis. Fig. 6 shows a comparison of the class accuracies between the SDFA^2 method and our method on the AWA dataset. From the figure, we can see that our method has higher accuracy in 21 classes compared to the SDFA^2 method, while the SDFA^2 method



(a) SDFA^2. (b) ISAE-GZSL.

Fig. 5. t-SNE visualization of unseen class images synthesized by SDFA^2(a) and ISAE-GZSL (b) in the GZSL on the CUB datasets.

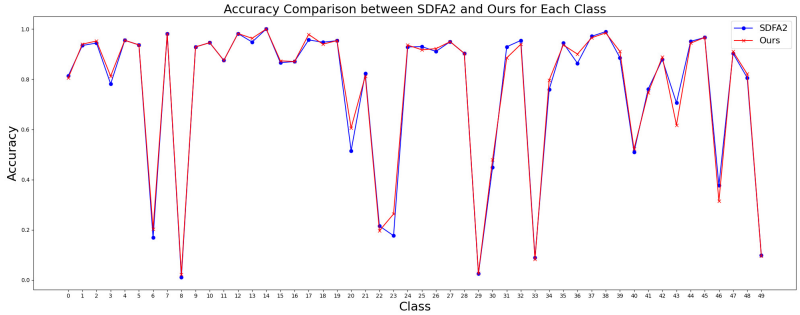


Fig. 6. Comparison of Class Accuracy on the AWA1 Dataset.

(a) blue whale. (b) kill whale.

Fig. 7. The classes with higher classification accuracy under the ISAE-GZSL method.

(a) fox. (b) sheep.

Fig. 8. The classes with higher classification accuracy under the SDFA^2 method.

has higher accuracy in 16 classes. Especially in the classes where our method performs better (such as blue whale and kill whale, according to Fig. 7) compared to those where SDFA^2 performs better (such as fox and sheep, according to Fig. 8), most images in these classes suffer from severe attribute missingness issues. This problem of attribute absence is more pronounced in these datasets. Based on these observations, our method demonstrates greater effectiveness in handling attribute missingness compared to SDFA^2.

5 Conclusion

In this work, we propose a novel method to enhance the generation of semantic attributes. By using interpolation to address the issue of semantic attribute incompleteness in practical scenarios, we enhance the diversity of semantic attributes. This approach makes the distribution of generated visual features closer to the distribution of real visual features, while also making the generated visual features more diverse. Experimental results show that compared to state-of-the-art methods, our approach performs better when dealing with categories in datasets with severe attribute missingness, significantly improving performance. In future studies, we aim to explore advanced interpolation methods to enhance diverse semantic attribute representation and address attribute missingness, thereby improving GZSL applications.

6 Acknowledgements

The work is supported by the National Key Research and Development Program of China (No.2022YFB4201603), the National Natural Science Foundation of China (No.61873033) and the Natural Science Foundation of Fujian Province (No. 2024H0012).

References

1. Palatucci, M., Pomerleau, D., Hinton, G.E., Mitchell, T.M.: Zero-shot learning with semantic output codes. In: Proceedings of the Annual Conference on Neural Information Processing Systems(NeurIPS). pp. 1410–1418 (2009)
2. Socher, R., Ganjoo, M., Manning, C.D., Ng, A.: Zero-shot learning through cross-modal transfer. In: Proceedings of the Annual Conference on Neural Information Processing Systems(NeurIPS). pp. 1–10 (2013)
3. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR). pp. 1778–1785 (2009)
4. Xian, Y., Schiele, B., Akata, Z.: Zero-shot learning - the good, the bad and the ugly. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1410–1418 (2017)

5. Shigeto, Y., Suzuki, I., Hara, K., Shimbo, M., Matsumoto, Y.: Ridge regression, hubness, and zero-shot learning. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases(ECML-PKDD). pp. 135–151 (2015)
6. Yu, Y., Ji, Z., Han, J., Zhang, Z.: Episode-based prototype generating network for zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14032–14041 (2020)
7. Verma, V.K., Arora, G., Mishra, A., Rai, P.: Generalized zero-shot learning via synthesized examples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4281–4289 (2018)
8. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). pp. 5542–5551 (2018)
9. Z. Han, Z. Fu, S.C.J.Y.: Contrastive embedding for generalized zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2731–2381 (2021)
10. Wang, Z., Hao, Y., Mu, T., Li, O., Wang, S., He, X.: Bi-directional distribution alignment for transductive zero-shot learning. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19893–19902 (2023)
11. Liu, Y., Tao, K., Tian, T., Gao, X., Han, J., Shao, L.: Transductive zero-shot learning with generative model-driven structure alignment. Pattern Recognition(PR) **153**, 110561 (2024)
12. Li, J., Jing, M., Lu, K., Ding, Z., Zhu, L., Huang, Z.: Leveraging the invariant side of generative zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7402–7411 (2019)
13. Xian, Y., Sharma, S., Schiele, B., Akata, Z.: F-vaegan-d2: A feature generating framework for any-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10267–10276 (2019)
14. Narayan, S., Gupta, A., Khan, F.S., Snoek, C.G.M., Shao, L.: Latent embedding feedback and discriminative features for zero-shot classification. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. pp. 479–495 (2020)
15. X. Zhao, Y. Shen, S.W.H.Z.: Boosting generative zero-shot learning by synthesizing diverse features with attribute augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 1–9 (2022)
16. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
17. Lampert, C. H.; Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). p. 951–958 (2009)
18. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S.J., Perona, P.: Caltech-ucsd birds 200. Technical Report CNS-TR-2010-001, Caltech (2010)
19. Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2751–2758 (2012)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017)
21. M. Arjovsky, S. Chintala, L.B.W.g.a.n.: Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning(ICML). pp. 214–223 (2017)
22. Maaten, L.V.D., Hinton, G.E.: Visualizing data using t-sne. Journal of Machine Learning Research. **9**, 2579–2605 (2008)