This ACCV 2024 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ACCV 2024
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/accv

Do they Share the Same Tail? Learning Individual Compositional Attribute Prototype for Generalized Zero-Shot Learning

Yuyan Shi[1], Chenyi Jiang[1], Run Shi[1], and Haofeng Zhang[1⋆]

Nanjing University of Science and Technology, Nanjing 210000, China
{shiyuyan,jiangchenyi,shirun,zhanghf}@njust.edu.cn

Abstract. Attributes are considered fundamental in zero-shot learning. By incorporating the correspondences between classes and attributes as prior knowledge, the model is able to approximate a class prototype for numerous classes without the need for any visual samples of these classes. In the majority of prior research, attributes are considered primitives and are not subjected to further subdivision. While the only distinction between shared attributes across classes is the absolute magnitude of their values, this does not adequately reflect the more significant visual differences between these classes in natural images. To address this issue, we propose learning the **In**dividual **C**ompositional **A**ttribute **P**rototype (InCAP). Specifically, InCAP does not treat attributes as the sole primitives but uses attribute semantics as objects in compositions, while class semantics are introduced as a special kind of state description within these compositions. This approach allows attributes and classes to form the structure of the composition. To avoid information isolation between seen and unseen classes, these compositional attributes are not used for direct contrusting class prototypes. Instead, they serve as spatial composition bottlenecks to suppress potential overfitting caused by attribute-visual mismatches during training and provide advanced location guidance information during testing. Experiments demonstrate that InCAP achieves leading results on mainstream datasets, validating the full potential of this strategy.

Keywords: Generalized zero-shot learning · Image classification· Compositional attributes· Visual bottleneck

1 Introduction

A shared space connecting seen and unseen classes is considered an essential factor for a deep learning model to achieve zero-shot classification. Typically, mainstream zero-shot learning (ZSL) and generalized zero-shot learning (GZSL) methods [39,?,?] use attributes as priors to form this space. Attributes are specific descriptions of the classes to be classified, such as outline, color, and state.

⋆ Corresponding Author

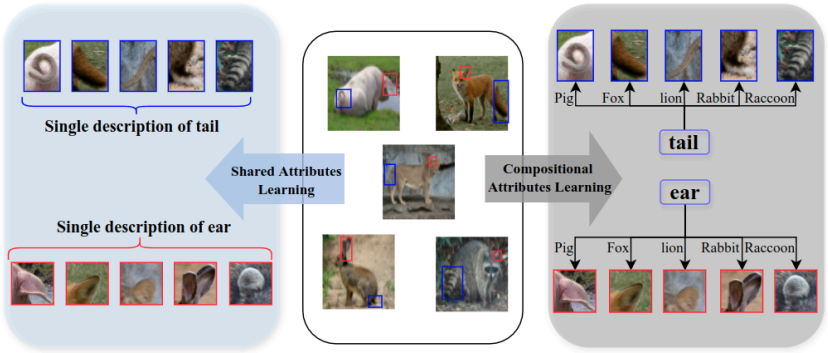

Fig. 1: Illustration of individual compositional attributes. Previous models trained on seen classes learn attribute prototypes and apply them to new classes for recognition and classification. However, the visual features of attributes can vary significantly between different animals, leading to what is known as inter-class variation of attributes. To address this issue, we propose a further subdivision of attributes. Instead of using generic attributes, we introduce more specific compositional attributes such as `pig's tail` and `rabbit's ears`. This approach enhances the model's ability to capture inter-class differences and improves recognition accuracy on unseen classes.

By using attributes as primitives and leveraging the correspondence between categories and attributes, the model can approximate class prototypes for unseen classes.

Based on the above knowledge, most current zero-shot learning (ZSL) methods use generative [31,27,41] or embedded structures [1,47,20] to achieve zero-shot image classification through attributes. These methods focus on generating synthetic features that closely resemble the real sample distribution or constructing clearer class boundaries in the common embedding space. While these approaches have made commendable progress, there is a lack of in-depth research on the structure of the attributes themselves.

In most of the aforementioned approaches, attributes are treated as the basic primitives that constitute a class description, differing only in their weight values. However, we pose two simple questions to the reader: (1) are there consistent visual representations of shared attributes across classes? and (2) are differences in the weight values of a single dimension sufficient to indicate these visual representational differences? We consider the answer to be no. For example, a rabbit's tail differs significantly from a horse's tail in terms of length, color, morphology, and other visual characteristics. A single-dimensional weighting value cannot encapsulate this distributional difference.

As depicted in Figure 1, it is challenging to find an accurate and unique description of an abstract attribute beyond a class, because the visual depiction of an attribute varies between classes. Consequently, a singular fixed attribute description is prone to causing visual-semantic misalignment.

To address the above issue, this paper proposes learning Individual Compositional Attribute Prototypes (InCAP) for GZSL. Unlike the previous approach, InCAP no longer uses the original attributes as minimal primitives for input. Instead, we further subdivide to form a compositional structure that includes both attribute semantics and class semantics, like `rabbit tail`. InCAP separates the original shared attributes into individual attributes for each class, using these as the minimal primitives, and we call them *compositional attributes*.

The translation of compositional attributes from text into embedding representations is typically realized through a text embedding model like CLIP [28]. Building on this, InCAP combines similar compositional attributes via clustering, facilitating the sharing of attribute information within a small scope. However, these enhancements could lead to the isolation of attributes shared between seen and unseen classes. Inspired by the IAB [16], we retain the original shared attributes at the time of classification but utilize a visual bottleneck to learn compositional attribute prototypes from the text embedding and furnish attribute region localization details for visual features within the common embedding space, instead of directly applying these to the classification within the same space. This approach enables InCAP to precisely compress regions in visual features that are independent of attributes, thereby successfully avoiding the erroneous allocation of shared attributes into semantically irrelevant areas of the embedding space.

In summary, our contributions are as follows:

- We propose new minimal primitives in semantic information, compositional attributes, for a more precise description of the attributes compared to the original shared attributes.
- Using the visual bottleneck with compositional attributes guides the model to accurately compress attribute-independent regions of visual features. This approach prevents attribute-visual matching bias and enhances the model's ability to generalize to unseen categories.
- State-of-the-art(SoTA) performance: our approach outperforms existing methods on three benchmark datasets. Comprehensive experimental analysis validates the model's effectiveness.

2 Related Work

2.1 Generalized Zero-Shot Learning

Image classification is a significant research direction in computer vision. Traditional image classification relies on supervised learning methods that require extensive labeled data. However, data labeling is a time-consuming and costly process, especially for images in specialized fields. Additionally, in many practical applications, data distribution is often long-tailed.

To enhance the utility and efficiency of image classification models, zero-shot image classification task is emerged [18,32]. These tasks require models to recognize and classify entirely new categories that are not present in the training data,

relying on transferring knowledge from seen classes to unseen classes. Generalized zero-shot image classification proposes a more realistic scenario, requiring models to recognize and categorize both seen and unseen classes during testing [4,36]. Depending on the strategy adopted, these methods can be categorized into generative methods [35,9,43,23] and embedding-based methods [37,19,38,15,22].

In generative approaches, abstract attribute information is utilized to enhance the quality of generated outputs [31,27]. However, some methods suggest that visual features generated based on generalized class-level attributes may lack realism [45,42,41]. Embedding-based methods often focus on learning localized attributes that can be transferred from seen classes to unseen classes during the image feature learning process [39,34]. Semantic information is frequently introduced into the semantic space as auxiliary information to guide the learning of visual features [38,21,6]. However, some approaches consider the still limitations of attribute localization and class-level attribute supervision [46,16,7].

2.2 Composition in Language

Composition is a fundamental concept in natural language processing, encompassing pairs [11,25,2] such as adjective-noun combinations [5,24,30] and verb-object interactions [29,44]. Simple combinations of two word vectors are insufficient for capturing the complexity of relationships and contextual dependencies between words [11]. In contrast, modeling two independent words as a new whole entity can effectively enhance text generation's accuracy and flexibility [26,30]. This approach personalizes the combined features according to the characteristics of the components, thus generating more precise and diverse linguistic descriptions. For instance, in attribute-object pairs, the object's characteristic representation varies according to the attribute's content.

Based on this premise, we propose that in GZSL, shared attributes and class names as textual information can form combinations similar to text pairs. Here, the class name serves as a description of the attribute, making it more specific and personalized. Introducing this combination of attributes in the model learning process can enhance the model's expressiveness and discriminative power.

3 Methodology

In this section, we define GZSL task and provide detailed information about the individual compositional attribute prototypes.

3.1 Problem Definition

In GZSL, the image set $\mathcal{X}$ is further divided into seen and unseen classes, *i.e.*, $\mathcal{X} = \mathcal{X}_s \cup \mathcal{X}_u$, and the corresponding labels are defined as $\mathcal{Y} = \mathcal{Y}_s \cup \mathcal{Y}_u$. The seen set is strictly disjoint from the unseen set, denoted as $\mathcal{Y}_s \cap \mathcal{Y}_u = \varnothing$. We define the train set as $\mathcal{S} = \{(x, y, \mathbf{z}) | x \in \mathcal{X}_s, y \in \mathcal{Y}_s, \mathbf{z} \in \mathcal{Z}\}$, where x denotes the images, y denotes the corresponding labels, and the class-level attribute vector

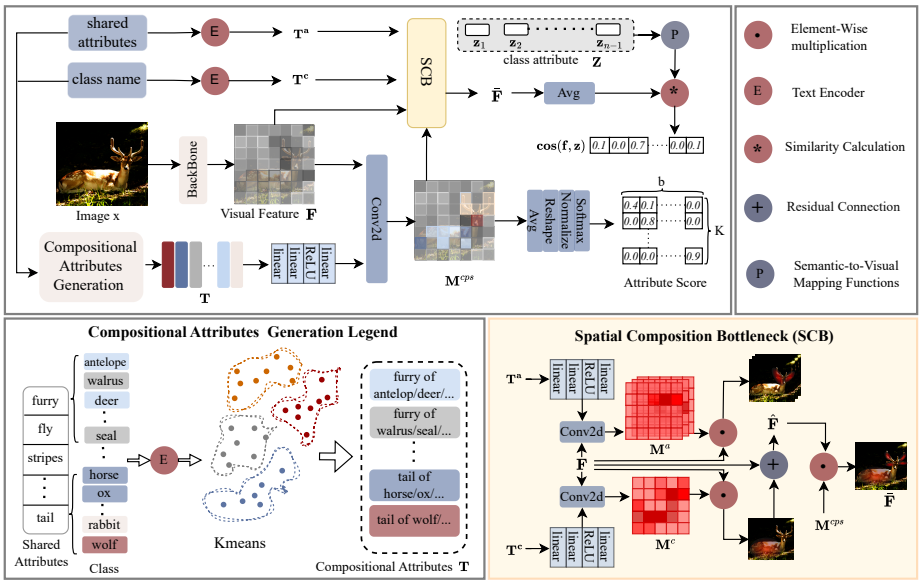

Fig. 2: Illustration of the proposed InCAP. The model achieves two primary tasks: first, it constructs more refined compositional attributes to distinguish attribute descriptions between different classes, addressing inter-class attribute differences; second, it integrates abstract semantic information (including original shared attributes, class names, and compositional attributes) for visual bottleneck, which compresses irrelevant attribute regions in the visual space, thereby enhancing the model's comprehension and representation of images.

$\mathbf{z} = \phi(y) \in \mathbb{R}^K$ represents K distinct attributes for class y. All class attributes form a collection $\mathcal{Z} = \{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{n-1}\}$, where n corresponds to the number of classes (including unseen classes). The primary objective of GZSL task is to predict the labels of images belonging to the unseen class and seen classes, *i.e.*, $\mathcal{X} \rightarrow \mathcal{Y}_s \cup \mathcal{Y}_u$.

3.2 Preliminaries

Attribute Correlation Map. Attributes constitute the sole learnable knowledge mutually exchanged between seen and unseen classes. APN [39] offers a direct framework for the prototype learning of attributes, *i.e.*, randomly initialize a set of learnable attribute prototypes $\mathcal{P} = \{p_1, p_2, \ldots, p_K\}$, the Attribute Correlation Map (AttrCM) from the visual features x can be obtained from the following equation:

$$\text{AttrCM}(x, \mathcal{P}) = \text{Conv}_{\mathcal{P}}(x). \tag{1}$$

$\text{Conv}_{\mathcal{P}}$ represents a convolution operation that is parameterized by employing the elements within $\mathcal{P}$ as the convolution kernel.

3.3 Model Structure

As shown in Figure 2, the shared attributes, class name information, and image x are jointly input to the model. Shared attributes and class names are combined to create new semantic primitives, compositional attributes. The image encoder extracts visual features $\mathbf{F}$ from an image x, which are then embedded in the visual space following a joint filtering process, and outputs $\bar{\mathbf{F}}$ that contains guidance for locating attribute regions. This process incorporates multiple levels of attribute-category compositions.

Consistent with preceding zero-shot methodologies [16,**?**], our classifier design employs nearest neighbors from cosine similarity in visual space for classification. The formula is as follows:

$$\mathbf{cos}(\mathrm{Avg}(\bar{\mathbf{F}}), \mathbf{z}) = \frac{\mathrm{Avg}(\bar{\mathbf{F}})}{\left\|\mathrm{Avg}(\bar{\mathbf{F}})\right\|_2} \otimes \frac{\vartheta(\mathbf{z})}{\|\vartheta(\mathbf{z})\|_2}, \tag{2}$$

where, $\otimes$ denotes matrix multiplication and $\mathbf{z}$ is the attribute feature corresponding to the image x. The symbols $\|\cdot\|_2$ and $\vartheta(\cdot)$ denote, respectively, the L2 norm normalization and the mapping function that projects class-level attributes into the visual space.

During training, the spatial composition bottleneck indirectly guides the learning of class attributes $\mathbf{z}$ mapping. This ensures that class-level attributes are projected to more appropriate locations in the visual space, achieving a more accurate visual-semantic correspondence.

3.4 Compositional Attributes

Predefined class-level attributes are crucial for knowledge transfer, but they only assign attribute scores to each class without describing the specific characteristics of the attributes. It assumes that shared attributes have the same characteristic representation across all classes, which is contrary to reality.

In our work, we no longer use the original shared attributes as minimal primitives. Instead, we decompose these shared attributes into separate attributes for each category, which we refer to as compositional attributes. This approach aims to expand attribute descriptions to more accurately characterize each category. As shown in Figure 2, we combine attributes and class names, leveraging natural language model to generate compositional attributes. There are many language models that can accomplish this task, such as FastText [3], BERT [8] and CLIP [28]. In this work, we choose the text encoder in CLIP.

Before processing, we construct the text input. The dataset provides shared attribute information, such as `tail` and class information, such as `lion`, which can be directly used as input text, denoted as $\{t_i^a\}_{i=1}^K$ and $\{t_i^c\}_{i=1}^n$. To provide more detailed and precise text information, we use fixed templates to combine textual information about attributes and class names, *e.g.*, $\{t_i\}_{i=1}^{n\times K}$ = {`class name`} `with the attribute` {`attribute`}.

We then use text encoder to generate the compositional attributes as shown in the following equation:

$$\mathbf{T} = \Lambda(E_{text}(t_i)), \tag{3}$$

where, $E_{text}(\cdot)$ indicates encoding of text, $\Lambda(\cdot)$ denotes stacking all tensors together and $\mathbf{T} \in \mathbb{R}^{n \times K \times m}$ can be viewed as a characterization of each attribute across different classes, m is the length of the vector characterizing each attribute. The same process can be applied to texts t^a and t^c to obtain their corresponding token tensors $\mathbf{T}^a \in \mathbb{R}^{K \times m}$ and $\mathbf{T}^c \in \mathbb{R}^{n \times m}$, which denote the attribute embedding and class name embedding, respectively.

Compositional attributes refines the inter-class distinctions of attributes. However, in the real world, some attributes do not significantly differ between similar classes. Excessive differentiation may lead to model overfitting rather than learn useful generalization capabilities. Therefore, we perform a clustering operation on the combined attributes to aggregate the feature representations of attributes in similar classes. The equation is shown below:

$$\hat{y} = \text{KMeans}(b, \mathbf{T}), \tag{4}$$

where, $\hat{y} \in \mathbb{R}^n$ stores the category labels after clustering and the function KMeans$(\cdot)$ denotes the clustering using the KMeans algorithm. b is the number of clusters specified as a hyperparameter to be discussed in detail in Section 4.4. For each cluster, the index of all classes belonging to the cluster is denoted as $\mathcal{I}_i = \{j | \hat{y}[j] = i\}$, $i = 1, 2, ..., b$.

We compute the maximum value of the tensor within the same cluster to generate a feature vector representing that cluster:

$$\bar{\mathbf{T}} = \max_{j \in \mathcal{I}_i} \mathbf{H}[j, :, :]. \tag{5}$$

This results in the final compositional attributes $\bar{\mathbf{T}} \in \mathbb{R}^{b \times K \times m}$. Subsequently, we map $\bar{\mathbf{T}}$ to the visual space and and compute the compositional attribute correlation map with $\mathbf{F}$. The specific operation is shown in the following equation:

$$\mathbf{M}^{\text{cps}} = \text{AttrCM}(\mathbf{F}, \delta_1(\bar{\mathbf{T}})), \tag{6}$$

where, $\delta_1(\cdot)$ is a map function.

Optimization by minimum entropy. On the basis of $\mathbf{M}^{\text{cps}}$, we compute the score of each attribute on the image x as shown in the following equation:

$$\text{Score} = \text{softmax}(\omega(\text{Avg}(\mathbf{M}^{\text{cps}}))), \tag{7}$$

where, $\omega(\cdot)$ restructures a one-dimensional tensor into a two-dimensional tensor of dimensions $K \times b$. Hence Score $\in \mathbb{R}^{K \times b}$ denotes the probability distribution of each attribute across each category.

Each row in Score denotes the probability distribution of attributes across various categories. We compute the minimum entropy for each row and construct a loss function by aggregating these entropy values. We constraint the model's

output by minimizing entropy during training, promoting a concentration of attribute probabilities within individual categories. The formula is shown below:

$$\mathcal{L}_{etp} = \sum_{i=0}^{K}(\min_{j}(-\operatorname{Score}_{i,j} \times \log(\operatorname{Score}_{i,j})). \tag{8}$$

3.5 Spatial Composition Bottleneck

Unlike abstract semantic descriptions, visual features usually contain more complexity and detail. To enhance attribute region guidance in the visual space, previous work often introduces shared attributes as a knowledge supplement for visual bottlenecks. This approach achieves better correspondence between semantics and vision, but the limited expressiveness of shared attributes restricts the model's performance.

Therefore, in our work, based on learning shared attribute prototypes and category prototypes we utilize the visual bottleneck to learn more precise combinatorial attribute prototypes from text embeddings. This process called spatial composition bottleneck is illustrated in the Figure 2.

First, the learnable shared attribute prototypes and category prototypes are initialized and introduced into the visual representation $\mathbf{F}$ respectively. This integration generates the corresponding shared attribute correlation map $\mathbf{M}^{\mathrm{a}} \in \mathbb{R}^{K\times h\times w}$ and class correlation map $\mathbf{M}^{\mathrm{c}} \in \mathbb{R}^{n\times h\times w}$ respectively. This process is illustrated by the following equation:

$$\begin{cases} \mathbf{M}^{\mathrm{a}} = \mathrm{AttrCM}(\mathbf{F}, \delta_2(\mathbf{T}^{a})), \\ \mathbf{M}^{\mathrm{c}} = \mathrm{AttrCM}(\mathbf{F}, \delta_3(\mathbf{T}^{c})), \end{cases} \tag{9}$$

where $\delta_2(\cdot)$ and $\delta_3(\cdot)$ denote the combination of linear layers and activation functions that map the token tensor $\mathbf{T}^{a}$ and $\mathbf{T}^{\mathrm{c}}$ to visual space, respectively.

Then, the learned prototype completes the attribute region localization in the image through the visual bottleneck. The localized image is then residually connected with the initial image embedding. This process enhances the model's ability to identify attributes and objects in vision while preserving the integrity of the original features. The computational formula is as follows:

$$\hat{\mathbf{F}} = \mathbf{F} \odot \sum_{i=0}^{K} \mathbf{M}^{\mathrm{a}} + \mathbf{F} \odot \sum_{i=0}^{n} \mathbf{M}^{\mathrm{c}} + \mathbf{F}, \tag{10}$$

where, $\odot$ indicates element-by-element multiplication.

Finally, the compositional attribute prototype is applied to the outputs of the initial two visual bottlenecks to achieve more precise attribute region localization guidance:

$$\bar{\mathbf{F}} = \hat{\mathbf{F}} \odot \sum_{s=0}^{bk}(\mathbf{M}^{\mathrm{cps}}{}_{sij}). \tag{11}$$

During training, we optimize the learning of combined attribute prototypes using minimum entropy loss. Gradient transfer indirectly influences the learning of shared attribute prototypes and category prototypes. This balanced integration of the three components enables the model to effectively implement the composition bottleneck.

3.6 Optimization and Inference

Optimization. During the training phase, the model is optimized using a composite loss function. After comparative analysis, we select a combination of minimum entropy loss, cross-entropy loss, and regularization loss to form the final loss function. The calculation of each component of the loss function is detailed below.

Cross-entropy loss is used for multi-class classification problems. It measures the discrepancy between the predicted probabilities of a classification model and the true labels. By minimizing this discrepancy, the model's predicted probabilities are brought closer to the true distribution. The calculation formula is as follows:

$$L_{ce} = -log\frac{exp(\mathbf{cos}(\mathrm{Avg}(\bar{\mathbf{F}}), \mathbf{z}))}{\sum_{\mathbf{z}_i \in \mathcal{Z}} exp(\mathbf{cos}(\mathrm{Avg}(\bar{\mathbf{F}}), \mathbf{z}_i))}. \tag{12}$$

Regularization loss enhances model performance by introducing constraints or penalties. In our approach, we minimize the Euclidean distance between visual feature and its corresponding attribute $\mathbf{z}$ to encourage the model to output higher confidence in predicting true labels, while simultaneously discouraging overfitting to irrelevant features. The calculation formula is as follows:

$$\mathcal{L}_{reg} = \left\| \mathrm{Avg}(\bar{\mathbf{F}}) - \vartheta(\mathbf{z}) \right\|_2^2. \tag{13}$$

Ultimately, combining multiple losses yields the model's final loss function $\mathcal{L}$, formulated as follows:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{reg} + \mathcal{L}_{etp}, \tag{14}$$

where λ represents hyperparameter that controls the effect of regularization loss on the overall loss function. We discuss them in detail in Section 4.4.

Inference. In GZSL, both seen and unseen classes are contained at inference time, but the model merely learns about the knowledge from seen classes during training. Therefore, we apply Calibrated Stacking (CS) [4] to jointly define the category. The classifier searches for the class embedding with the highest compatibility via:

$$\bar{y} = \underset{y=\in \mathcal{Y}_s \cup \mathcal{Y}_u}{\arg\max} (\mathbf{cos}(\mathrm{Avg}(\bar{\mathbf{F}}), \mathbf{z})) - \gamma \mathbb{I}(y \in \mathcal{Y}_s), \tag{15}$$

where, $\mathbb{I}(y \in \mathcal{Y}_s) = 1$ if y is a seen class and 0 otherwise, γ is the calibration factor for balancing the seen and unseen classes [4].

4 Experiments

4.1 Datases and Metrics

Datasets. AWA2 comprises 37,322 images that encompass a diverse range of animals. These images are distributed across 50 distinct animal categories, each is characterized by 85 attributes. There are 40 seen classes and 10 unseen classes.

CUB is a standard dataset for bird identification, featuring approximately 11,788 images drawn from 200 different bird classes, including 150 seen classes and 50 unseen classes. It includes 312 attributes, with around 30 images per bird.

SUN comprises roughly 14,340 images depicting a wide variety of natural scenes, spanning 717 different scene categories including 645 seen classes. It includes 102 attributes, encompassing indoor and outdoor environments, natural landscapes, and more.

Evaluation Metrics. Evaluation of GZSL performance employs the widely used harmonic mean A^H introduced by [36]: $A^H = \left(2 \times A^S \times A^U\right) \ / \ \left(A^S + A^U\right)$, where A^S and A^U denote the classification accuracy on seen and unseen classes, respectively. In our experiments, we also report A^S and A^U.

Implementation Details. Our model employs the pre-trained ResNet-101 on ImageNet as the backbone. The training process consists of two stages: initially learning parameters with the frozen backbone network, followed by unfreezing the backbone for fine-tuning to better adapt the pre-trained model to the task requirements. The entire training process is conducted on a 2080Ti GPU with a batch size of 32. For the dataset AWA2, the learning rate is set to 1×10^{-4} in the first stage and 1×10^{-7} in the second stage; for the dataset CUB, the learning rates are 1×10^{-5} and 5×10^{-7}, respectively; for the dataset SUN, the rates are 5×10^{-4} and 1×10^{-7}, respectively. Throughout the model, we configure two hyperparameters: λ and b. Here, λ assign weights in the loss function, while b represents the number of clusters in the KMeans algorithm. In section 4.4, we perform ablation experiments on these hyperparameters.

4.2 Comparison with State-of-the-Art Methods

As shown in Table 1, we present a comparative analysis of our model against two kinds of SoTA methods: embedding-based methods and generative methods. Generative methods utilize attribute information of unseen classes during training, whereas embedding methods do not require this prerequisite. Since our approach falls under the category of embedding-based methods, we include the results of some generative methods for reference in this section, but the primary comparison is made with embedding methods.

The experimental results demonstrate the superior performance of our model. The SoTA results of our model on the AWA2 and SUN datasets are 79.4% and 45.4% respectively. These results represent improvements of 3.1% and 0.7% over the IAB method, and 4.6% and 0.9% over the DPN method. It can be seen that our approach achieves excellent results on coarse-grained datasets. This is

Table 1: Results of GZSL on three classification benchmarks. We compare our model with state-of-the-art models on the CUB, AWA2, and SUN datasets. In this context, † and ‡ denote generative methods and embedding-based methods, respectively. We evaluate the model's performance using top-1 accuracy(A^S and A^U) as well as their harmonic mean (A^H) for both seen and unseen classes in GZSL. The table highlights the optimal A^H on each of the three datasets in red and the second-best A^H in blue. '-' means not reported.

	Method	Reference	CUB			AWA2			SUN		
			A^U	A^S	A^H	A^U	A^S	A^H	A^U	A^S	A^H
†	GCM-CF [42]	CVPR(2021)	61.0	59.7	60.3	60.4	75.1	67.0	47.9	37.8	42.2
	CE-GZSL [12]	CVPR(2021)	63.9	66.8	65.3	63.1	78.6	70.0	48.8	38.6	43.1
	ICCE [17]	CVPR(2022)	67.3	65.5	66.4	65.3	83.2	72.8	-	-	-
	SCE [13]	IJCV(2022)	66.5	68.6	67.6	64.3	77.5	70.3	45.9	41.7	43.7
	CMC-GAN [40]	IJCV(2023)	52.6	65.1	58.2	-	-	-	48.2	40.8	44.2
	DFCA [33]	TCSVT(2023)	70.9	63.1	66.8	66.5	81.5	73.3	48.9	38.8	43.3
‡	RGEN [38]	CVPR(2019)	73.5	60.0	66.1	76.5	67.1	71.5	31.7	44.0	36.8
	DAZLE [15]	CVPR(2020)	56.7	59.6	58.1	60.3	75.7	67.1	52.3	24.3	33.2
	DVBE [22]	CVPR(2020)	64.4	73.2	68.5	62.7	77.5	69.4	44.1	41.6	42.8
	APN [39]	NIPS(2020)	65.3	69.3	67.2	56.5	78.0	65.5	41.9	34.0	37.6
	GEM-ZSL [21]	CVPR(2021)	64.8	77.1	70.4	64.8	77.5	70.6	38.1	35.7	36.9
	TDCSS [10]	CVPR(2022)	44.2	62.8	51.9	59.2	74.9	66.1	-	-	-
	TransZero [6]	AAAI(2022)	69.3	68.3	68.8	61.3	82.3	70.2	52.6	33.4	40.8
	AS-ZSL [46]	PR(2023)	65.8	78.2	71.5	66.5	78.3	71.9	39.5	37.2	38.3
	DPN [14]	TCSVT(2024)	63.7	80.6	71.2	65.2	87.6	74.8	48.3	41.4	44.5
	IAB [16]	IJCV(2024)	70.1	78.5	74.0	70.0	83.8	76.3	46.9	42.7	44.7
	InCAP	ours	69.6	75.2	72.3	73.5	86.4	79.4	49.2	42.2	45.4

because these datasets contain classes with inherently greater variation, making the differences in attributes across classes more critical. The experimental results underscore the feasibility and importance of attribute refinement.

On the CUB dataset, our model achieves the second-best A^H of 72.3%, outperforming the DPN method by 1.1%. Our model performs slightly worse on the CUB dataset compared to the other two datasets. We believe this is due to two reasons: 1) CUB, as a fine-grained dataset, contains only different species of birds, resulting in negligible differences between most attributes; and 2) CUB contains 312 attributes, whereas AWA2 and SUN have only 85 and 102 attributes, respectively. Thus, the CUB dataset already provides a detailed description of attributes.

4.3 Ablation Experiment

We evaluate the contribution of various visual bottleneck combinations and minimum entropy loss functions to the model's performance through ablation experiments.

We establish a baseline to to provide a comparison. The baseline includes a simple linear layer that maps attribute information to the visual space to com-

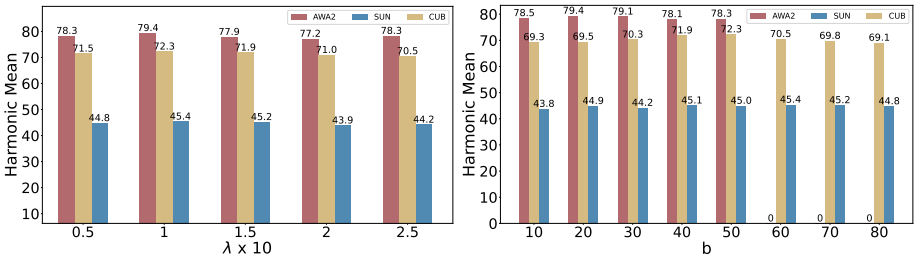

Fig. 3: Ablation results for hyperparameters λ and b. This figure describes ablation experiments involving the coefficients, λ and b. λ plays a crucial role in weighting the loss function, aimed at balancing the cross-loss and regularization loss. b is the number of cluster when clustering compositional attributes.

pute the similarity between semantic and visual embeddings for classification. And baseline is optimized using a cross-entropy loss and a regularization loss.

The experimental results are shown in Table 2. In the ablation experiments, different combinations of visual bottlenecks are incrementally added to the baseline to verify their effect on the model's classification performance. The visual bottleneck in the compositional attributes show a significant effect improving by 7.8%, 12.6% and 6.9% on the three datasets, respectively. And its combination with either t^c or t^a also demonstrate notable enhancement. On the CUB dataset, performance improvement is 8.2% and 8.8%; on the AWA2 dataset, performance improvement is 13.2% and 14%; and on the SUN dataset, performance improvement is 6.6% and 8.2%. However, the combined bottleneck involving all three elements achieves the optimal effect improving by 9.7%, 15.6% and 9.5% on the three datasets, as the model find a balance among them during the training process.

Furthermore, for the compositional attributes we introduce a minimum entropy loss. This loss function encourages the model to concentrate the probability distribution of each attribute within a single category, thereby reducing the ambiguity of the attribute representation and further improving the model's classification performance. We experimentally verify the impact of this loss on model optimization. As shown in Tabel 2, adding the minimum entropy loss to all combinations containing compositional attributes improves A^H.

In summary, this section provides an in-depth evaluation of the contribution of various visual bottleneck combinations and minimum entropy loss functions.

4.4 Hyper-Parameter Analysis

In the proposed model, we set two hyperparameters, λ and b, to balance the model's performance.

The hyperparameter λ assigns weights in the loss function and controls the effect of regularization loss on the overall loss function. By tuning λ, we can strike a balance between fitting the training data and preventing overfitting, thus

Table 2: Ablation study for InCAP. We experimentally verify the effect of different combinations of visual bottlenecks and conduct comparative experiments for the minimum entropy loss function. In our notation, t, t^a, and t^c represent composition attributes, shared attributes, and class names, respectively. The "+" symbol indicates the addition of corresponding visual bottlenecks in the module, while "$\surd$" denotes optimization using the corresponding loss function.

	Loss			CUB			AWA2			SUN		
	$\mathcal{L}_{ce}$	$\mathcal{L}_{reg}$	$\mathcal{L}_{etp}$	A^U	A^S	A^H	A^U	A^S	A^H	A^U	A^S	A^H
Baseline	$\surd$	$\surd$		51.8	78.8	62.5	52.7	79.2	63.3	46.1	29.0	35.6
$+t^a$	$\surd$	$\surd$		63.3	76.6	69.3	63.2	84.5	72.3	51.1	34.4	41.1
$+t^c$	$\surd$	$\surd$		60.8	77.6	68.2	66.2	76.1	70.8	52.2	30.7	38.6
$+t^a + t^c$	$\surd$	$\surd$		65.5	74.3	69.6	65.2	83.9	73.4	50.8	34.7	41.2
$+t$	$\surd$	$\surd$		66.4	74.7	70.3	68.3	85.3	75.9	49.2	37.4	42.5
$+t$	$\surd$	$\surd$	$\surd$	66.3	75.5	70.6	69.7	84.2	76.2	49.4	37.9	42.9
$+t + t^a$	$\surd$	$\surd$		67.6	75.5	71.3	71.6	84.0	77.3	51.2	38.3	43.8
$+t + t^a$	$\surd$	$\surd$	$\surd$	69.4	74.4	71.8	69.4	87.6	77.5	50.1	39.6	44.2
$+t + t^c$	$\surd$	$\surd$		67.3	74.3	70.7	68.3	87.1	76.5	51.0	36.0	42.2
$+t + t^c$	$\surd$	$\surd$	$\surd$	69.2	73.5	71.3	68.7	88.5	77.4	49.6	39.5	43.9
$+t + t^c + t^a$	$\surd$	$\surd$		70.3	74.1	72.2	71.9	87.4	78.9	48.1	42.5	45.1
InCAP	$\surd$	$\surd$	$\surd$	69.6	75.2	72.3	73.5	86.4	79.4	49.2	42.2	45.4

enhancing the model's generalization ability. In Figure 3, the harmonic mean A^H of the three datasets with different λ settings are prominently displayed. The figure shows that when λ is set to 0.1, the harmonic mean reaches its maximum value across all three datasets. This suggests that $\lambda = 0.1$ effectively controls the overfitting problem. Therefore, we conclude that 0.1 is the most appropriate value for λ in our model.

The parameter b indicates the number of clusters when clustering compositional attributes. Choosing an appropriate value for b effectively captures the differences between fine-grained attributes, improving the model's classification accuracy and generalization performance. Figure 3 visualizes the effect of the b value on classification accuracy. The optimal b value varies across the three datasets: 50 for CUB, 20 for AWA2, and 60 for SUN. This variation is due to the different characteristics of each dataset.

4.5 Visualization Analysis

Our approach proposes an attribute refinement method that focuses on inter-class differences in attributes. We verify the efficiency of this approach through visualization experiments. The control groups include the visualization results from the baseline model and the results from introducing initial shared attributes into the visual features. These experiments are conducted on the AWA2 dataset.

The visualization based solely on the baseline yields the poorest results; without the aid of textual information, the model's attention can not focus on the key information in the image. After introducing attribute information, the visual features learn some attribute localization. However, the visualized images

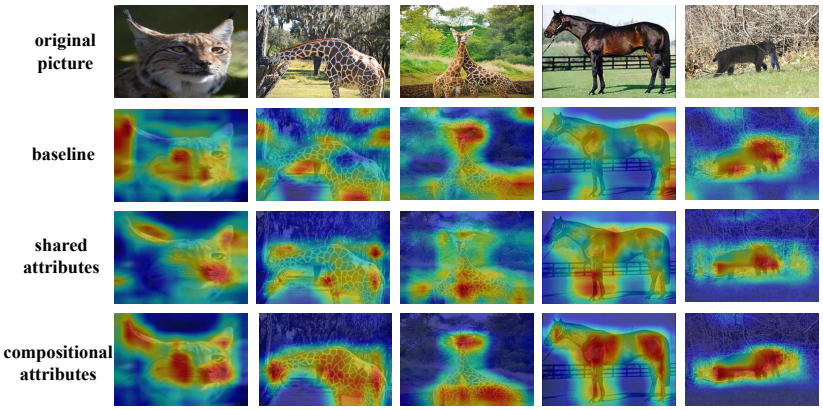

Fig. 4: Illustration of visualization comparison. We present visual comparison results to verify the effectiveness of compositional attributes in capturing key information in images. Initial, shared attributes are introduced into the visual features for comparison, with the baseline results serving as a reference.

still show unsatisfactory results, with the model's attention incorrectly focusing on some irrelevant attributes. We believe these results are due to the mismatch between the attribute text information provided by the dataset and the image content, as a single attribute text is often inconsistent with a variable visual image.

Our approach mitigates the problem of visual semantic mismatch due to inter-attribute differences Visualization results confirm that the compositional information better aligns with the image content, capturing critical information with minimal influence from irrelevant features.

5 Conclusion

In this paper, we introduce Individual Compositional Attribute Prototypes (InCAP) for GZSL, a method for constructing compositional attributes that distinguish attributes between attribute classes. We propose a spatial composition bottleneck that improves the accuracy of attribute-visual correspondence by compressing attribute-independent regions in the visual space. Extensive experiments conducted on three popular datasets yields SoTA results. The ablation study results also demonstrate that compositional attributes enhances the model's classification performance by focusing on inter-class differences compared to traditional shared attributes.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under the Grant Nos. 62371235, 62076132 and 62072246, and in part by the Key Research and Development Plan of Jiangsu Province (Industry Foresight and Key Core Technology Project) under Grant No. BE2023008-2.

References

1. Atzmon, Y., Chechik, G.: Adaptive confidence smoothing for generalized zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11671–11680 (2019)
2. Baroni, M., Zamparelli, R.: Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In: Proceedings of the 2010 conference on empirical methods in natural language processing. pp. 1183–1193 (2010)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the association for computational linguistics **5**, 135–146 (2017)
4. Chao, W.L., Changpinyo, S., Gong, B., Sha, F.: An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. pp. 52–68. Springer (2016)
5. Chen, C.Y., Grauman, K.: Inferring analogous attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 200–207 (2014)
6. Chen, S., Hong, Z., Liu, Y., Xie, G.S., Sun, B., Li, H., Peng, Q., Lu, K., You, X.: Transzero: Attribute-guided transformer for zero-shot learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 330–338 (2022)
7. Chen, Z., Zhang, P., Li, J., Wang, S., Huang, Z.: Zero-shot learning by harnessing adversarial samples. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 4138–4146 (2023)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Felix, R., Kumar, B., Reid, I., Carneiro, G.: Multi-modal cycle-consistent generalized zero-shot learning. Cornell University - arXiv,Cornell University - arXiv (Aug 2018)
10. Feng, Y., Huang, X., Yang, P., Yu, J., Sang, J.: Non-generative generalized zero-shot learning via task-correlated disentanglement and controllable samples synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9346–9355 (2022)
11. Guevara, E.R.: A regression model of adjective-noun compositionality in distributional semantics. In: Proceedings of the 2010 workshop on geometrical models of natural language semantics. pp. 33–37 (2010)
12. Han, Z., Fu, Z., Chen, S., Yang, J.: Contrastive embedding for generalized zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2371–2381 (2021)
13. Han, Z., Fu, Z., Chen, S., Yang, J.: Semantic contrastive embedding for generalized zero-shot learning. International Journal of Computer Vision **130**(11), 2606–2622 (2022)
14. Hu, Y., Feng, L., Jiang, H., Liu, M., Yin, B.: Domain-aware prototype network for generalized zero-shot learning. IEEE Transactions on Circuits and Systems for Video Technology **34**(5), 3180–3191 (2024)
15. Huynh, D., Elhamifar, E.: Fine-grained generalized zero-shot learning via dense attribute-based attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4483–4493 (2020)

16. Jiang, C., Shen, Y., Chen, D., Zhang, H., Shao, L., Torr, P.H.: Estimation of near-instance-level attribute bottleneck for zero-shot learning. International Journal of Computer Vision pp. 1–27 (2024)
17. Kong, X., Gao, Z., Li, X., Hong, M., Liu, J., Wang, C., Xie, Y., Qu, Y.: En-compactness: Self-distillation embedding & contrastive generation for generalized zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9306–9315 (June 2022)
18. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 951–958. IEEE (2009)
19. Li, K., Min, M.R., Fu, Y.: Rethinking zero-shot learning: A conditional visual classification perspective. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3583–3592 (2019)
20. Liu, Y., Guo, J., Cai, D., He, X.: Attribute attention for semantic disambiguation in zero-shot learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6698–6707 (2019)
21. Liu, Y., Zhou, L., Bai, X., Huang, Y., Gu, L., Zhou, J., Harada, T.: Goal-oriented gaze estimation for zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3794–3803 (2021)
22. Min, S., Yao, H., Xie, H., Wang, C., Zha, Z.J., Zhang, Y.: Domain-aware visual bias eliminating for generalized zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12664–12673 (2020)
23. Mishra, A., Krishna Reddy, S., Mittal, A., Murthy, H.A.: A generative model for zero shot learning using conditional variational autoencoders. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 2188–2196 (2018)
24. Misra, I., Gupta, A., Hebert, M.: From red wine to red tomato: Composition with context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1792–1801 (2017)
25. Mitchell, J., Lapata, M.: Vector-based models of semantic composition. In: proceedings of ACL-08: HLT. pp. 236–244 (2008)
26. Nagarajan, T., Grauman, K.: Attributes as operators: factorizing unseen attribute-object compositions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 169–185 (2018)
27. Narayan, S., Gupta, A., Khan, F.S., Snoek, C.G., Shao, L.: Latent embedding feedback and discriminative features for zero-shot classification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 479–495. Springer (2020)
28. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
29. Sadeghi, M.A., Farhadi, A.: Recognition using visual phrases. IEEE (2011)
30. Santa Cruz, R., Fernando, B., Cherian, A., Gould, S.: Neural algebra of classifiers. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 729–737. IEEE (2018)
31. Shen, Y., Qin, J., Huang, L., Liu, L., Zhu, F., Shao, L.: Invertible zero-shot recognition flows. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 614–631. Springer (2020)
32. Socher, R., Ganjoo, M., Manning, C.D., Ng, A.: Zero-shot learning through cross-modal transfer. Advances in neural information processing systems **26** (2013)

33. Su, H., Li, J., Lu, K., Zhu, L., Shen, H.T.: Dual-aligned feature confusion alleviation for generalized zero-shot learning. IEEE Transactions on Circuits and Systems for Video Technology **33**(8), 3774–3785 (2023)
34. Wang, C., Min, S., Chen, X., Sun, X., Li, H.: Dual progressive prototype network for generalized zero-shot learning. Advances in Neural Information Processing Systems **34**, 2936–2948 (2021)
35. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5542–5551 (2018)
36. 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. pp. 4582–4591 (2017)
37. Xie, G.S., Liu, L., Jin, X., Zhu, F., Zhang, Z., Qin, J., Yao, Y., Shao, L.: Attentive region embedding network for zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9384–9393 (2019)
38. Xie, G.S., Liu, L., Zhu, F., Zhao, F., Zhang, Z., Yao, Y., Qin, J., Shao, L.: Region graph embedding network for zero-shot learning. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 562–580. Springer (2020)
39. Xu, W., Xian, Y., Wang, J., Schiele, B., Akata, Z.: Attribute prototype network for zero-shot learning. Advances in Neural Information Processing Systems **33**, 21969–21980 (2020)
40. Yang, F.E., Lee, Y.H., Lin, C.C., Wang, Y.C.F.: Semantics-guided intra-category knowledge transfer for generalized zero-shot learning. International Journal of Computer Vision **131**(6), 1331–1345 (2023)
41. 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. pp. 14035–14044 (2020)
42. Yue, Z., Wang, T., Sun, Q., Hua, X.S., Zhang, H.: Counterfactual zero-shot and open-set visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15404–15414 (2021)
43. Zhang, C., Peng, Y.: Visual data synthesis via gan for zero-shot video classification. Cornell University - arXiv,Cornell University - arXiv (Apr 2018)
44. Zhang, H., Kyaw, Z., Chang, S.F., Chua, T.S.: Visual translation embedding network for visual relation detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5532–5540 (2017)
45. Zhao, X., Shen, Y., Wang, S., Zhang, H.: Boosting generative zero-shot learning by synthesizing diverse features with attribute augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 3454–3462 (2022)
46. Zhou, L., Liu, Y., Bai, X., Li, N., Yu, X., Zhou, J., Hancock, E.R.: Attribute subspaces for zero-shot learning. Pattern Recognition **144**, 109869 (2023)
47. Zhu, P., Wang, H., Saligrama, V.: Generalized zero-shot recognition based on visually semantic embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2995–3003 (2019)