Supplementary Materials: Learning Visual-Semantic Hierarchical Attribute Space for Interpretable Open-Set Recognition

Zhuo Xu and Xiang Xiang*
National Key Lab of Multi-Spectral Information Intelligent Processing Technology,
School of Artificial Intelligence and Automation,
Huazhong University of Science and Technology, Wuhan, China
xex@hust.edu.cn

1. Prompting Multimodal Large Language Models for Visual Attributes

Recently, Multimodal Large Language Models (MLLM) have demonstrated powerful capabilities in image description. Given an image with a prompt, we can easily obtain attributes that describe the image. Lately, some work utilizes Large Language Model (LLM) to obtain attributes [9]. However, compared to using LLM, MLLMs utilize the information from the image modality to obtain more visual information, thus addressing the ambiguity issues. For example, the term 'mouse' can refer to an electronic device or an animal, and prompting it to LLM may not yield the desired results.

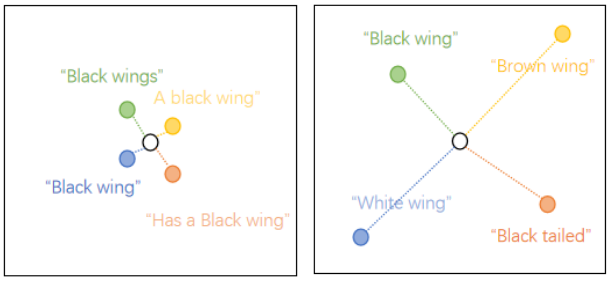
Formally, for any label c and its corresponding image x, we obtain a list of J attributes $attr_c = \mathcal{M}(class_c, x)$ using the MLLM, where $\mathcal{M}$ represents the MLLM. It is worth noting that the prompts provided to the MLLM are predefined; for example, "Describe the visual features for $class_c$ in the photo, list 6 pieces." We generate 100 attribute descriptors $attr$ for each category and transform these descriptors into binary attribute labels using clustering. For the j-th attribute descriptor of $class_c$, we use the text encoder UltraFastBERT [2] to convert $attr_c^j$, where j ranges from 1 to J, into word embeddings v_c^j. Subsequently, we use K-means to cluster these embeddings and assign labels. The i-th clustering center represents the i-th attribute. If the j-th attribute descriptor $attr_c^j$ of category c is assigned to category i, then the i-th attribute of category c receives a value of 1. Assuming there are K clustering centers, we finally obtain K-dimensional binary attributes for C categories. The resulting binary attribute vectors for each category, denoted as A_y where y ranges from 1 to C, can be referred to as attribute prototypes.

The density of attribute clusters reflects the quality of the corresponding attributes. A higher density of attribute clusters corresponds to a more explicit representation of the attribute semantics, while a lower density indicates a more ambiguous representation of semantics, as shown in Fig.1. Therefore, we clean the individual attribute clusters and filter out the sparsely clustered attribute clusters, which not only improves the accuracy of the model but also reduces the training burden. We use the Within-cluster Sum of Squares (WCSS) to represent the density of each attribute cluster:

$$WCSS = \frac{1}{N}\sum_{n=1}^{N} |x_n - c_i|^2 \tag{1}$$

where c_i represents the center of the i-th attribute cluster, and x_n represents the embedding of the n-th attribute descriptor within the cluster. A lower WCSS value indicates better better attrubute cluster.

We utilize four different MLLM templates, each generating several attributes. First, we provide an example question to the MLLM model, followed by a second query. For MLLM, we use the Qwen-VL-chat model [1]. During the generation process, we use 4 NVIDIA RTX 4090 (24 GB) GPU. Here are the templates we employ:



(a) Good attribute cluster (b) Bad attribute cluster

Figure 1. Good attribute clustering cluster: High attribute density, consistent and similar attributes representing the same concepts. Bad attribute clustering cluster: Low attribute density, confused meanings among attributes, unable to express the same concepts.

*Corresponding author. Xiang Xiang is also with Peng Cheng Lab.

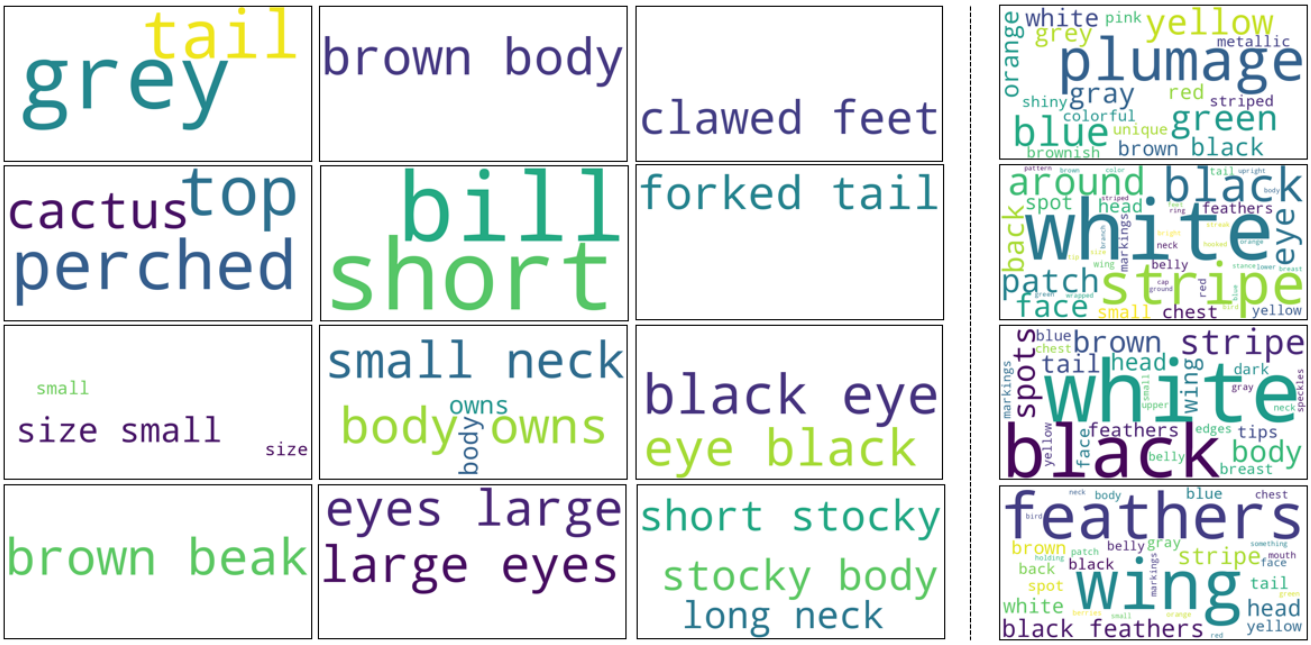


Figure 2. Word cloud visualization of attribute clusters corresponding to each binary attribute. The left side of dashed line represents 'good attributes' with small average intra-cluster distances, while the right side represents 'bad attributes' that have been filtered out due to large average intra-cluster distances.

Q: Describe the visual features for the {classname} in the photo, list 6 pieces.

Q: Describe what the {classname} looks like in the photo, list 6 pieces.

Q: Visually describe the {classname} in the photo, list 6 pieces.

Q: Describe the visual attributes for the {classname} in the photo, list 6 pieces.

Next, we apply the method mentioned above to cluster the attributes in the attribute pool and convert them into binary attributes. We set the number of clusters to be K = 2*C, where C is the total number of classes, resulting in binary attributes of length 2*C. In the sampling mechanism, we set a filter for the top 20% of maximum WCSS attributes, and further filter out the attributes that have a shared attribute count in the top 10% and bottom 10%.

Here, we present several examples of attributes generated by MLLM. Taking the example of the Black-footed Albatross from the CUB dataset, here is a partial display of the generated attributes: "has a long, narrow wingspan", "has a gray body", "has a white face", "has a gray beak", "has a gray tail", "Wet and hairless nose with curved nostrils". We also visualize the attributes corresponding to each attribute cluster after clustering. Each attribute cluster corresponds to a binary attribute, and we use word clouds to visualize all descriptors contained in this cluster to reflect the semantic information represented by this binary attribute, as shown in Figure 2.

2. Experimental Setting

2.1. Descriptions of Datasets

For open-set recognition (OSR) task, we follow the fine-grained semantic-shift open-set benchmark proposed by [11], including CUB-200-2011 [12], Stanford-Cars [4], and FGVC-Aircraft [6], and we use the same open-set splits. CUB-200-2011 is a fine-grained bird dataset containing 200 categories and 11,788 images. Stanford-Cars is a dataset that consists of 16,815 images of different vehicles belonging to 196 distinct categories. FGVC-Aircraft contains 10,200 images of aircraft, with 100 images for each of 102 different aircraft model variants. These datasets provide a comprehensive benchmark for our algorithm and span a diverse range of domains. Besides, we conduct experiments on attribute datasets including AWA2 [5] and LAD [14]. These datasets contain category-level attribute labels annotated by human experts. The AWA2 dataset comprises 37,322 images of 50 different animals with 85 attributes. We select the first 40 classes as known categories and the remaining 10 classes as unknown categories. The LAD dataset [14] comprises 78,017 images distributed across five major categories (animals, fruits, transportation, electrical appliances, and hair), 230 categories, and 359 semantic and visual attributes labeled at the category level. We conduct experiments on the vehicle subclasses within the LAD dataset, consisting of 50 classes.

Table 1. Results on the OOD detection task. AUR stands for AUROC and FPR represents FPR95.

ID Dataset	Method	iNaturalist		SUN		Places		Texture		Average	
		AUR ↑	FPR ↓	AUR ↑	FPR↓	AUR ↑	FPR ↓	AUR ↑	FPR ↓	AUR ↑	FPR ↓
CUB	MSP	97.23	9.08	94.55	16.01	90.80	25.08	96.46	12.48	94.76	15.66
	MCM	88.14	42.26	96.24	16.85	94.36	25.21	98.19	8.70	94.23	23.26
	MCAS(Ours)	99.55	2.24	98.84	5.18	98.99	4.43	98.99	4.56	99.09	4.10
Stanford-Cars	MSP	96.54	16.26	98.87	4.42	97.67	9.47	95.41	17.89	97.12	12.01
	MCM	99.52	0.73	99.80	0.34	99.58	1.22	99.81	1.24	99.68	0.88
	MCAS(Ours)	99.65	1.37	99.81	0.46	99.68	0.88	98.05	5.62	99.68	2.08
FGVC	MLS	91.80	33.11	93.88	21.73	90.47	82.08	91.40	28.28	91.89	41.30
	MCM	48.96	94.77	76.24	65.10	72.34	69.70	60.21	84.54	64.44	78.53
	MCAS(Ours)	96.22	14.60	97.83	4.24	97.30	7.88	94.81	12.34	96.54	9.77

2.2. Training Details

All the networks are trained on a system equipped with 1×NVIDIA RTX4090 (24G). For the semantic-shift open-set benchmark, we use ResNet-50 as the image encoder. For the remaining attribute datasets, we utilize ResNet-18 as the image encoder. The attribute branch and the classification branch are both composed of a single fully connected layer and a linear layer. The dimensionality of the fully connected layer is set to 512. For all the networks used in the fine-grained semantic shift open-set benchmark, a batch size of 32 samples is used with 8 workers for data loading. The images are resized to a size of 448. For the attribute datasets, a batch size of 128 is chosen, and the images are resized to a size of 224. Stochastic Gradient Descent (SGD) is employed with a weight decay of 1e-4 and a momentum of 0.9. The initial learning rate for all parameters is set to 5e-4. For all training data, we use RandAugment and set the parameters m=30 and n=6. For the parameter λ that controls the weighting of L_C and L_A, we set it to 0.5. The total number of training epochs is set to 200.

3. Additional Results

3.1. Experiments on OOD Detection Task

To further validate the effectiveness of our methods across different domains and datasets, we conduct experiments on the out-of-distribution (OOD) detection task. Unlike OSR, the unknown samples for OOD detection often come from different distributions. We use the known classes from the dataset we used for OSR in the main paper as the in-distribution (ID) data. Following [7, 8], we use several commonly used OOD datasets: iNaturalist [10], SUN [13], Places365 [15], and Texture [3]. We compare our method with MSP and MCM, as shown in Tab. 1. Our method demonstrates better OOD detection performance across various ID datasets, indicating its robustness to different domains and datasets.

3.2. Experiments on Different Attribute Generation Methods

We also test the OSR performance using attributes generated by LLM, as shown in Tab. 2. The results indicate that attributes generated using either LLMs or MLLMs can improve OSR performance. However, MLLMs provide more accurate attribute descriptions due to their ability to access images and address linguistic ambiguities. In addition, we also test the impact of using different numbers of prompts on the generated attributes. The content in parentheses in Tab. 2 indicates the number of prompts. Using more instructions can enhance the diversity of the generated attributes, which helps the model learn richer semantic information and improve OSR performance.

3.3. Experiments on Our Plug-and-play Attribute Learning Module

Considering that our proposed attribute learning module is plug-and-play, we also integrate our attribute learning module into other methods to test the impact of using attributes, as shown in Tab. 3. The results demonstrate that incorporating attributes into various methods can indeed lead to performance improvements for OSR.

3.4. Comparison with MLLM in Fine-Grained Classification Task

Considering that we use MLLM to generate attributes for the OSR task, we also test the classification accuracy of MLLM directly, as shown in Tab. 4. Our prompts include the names of all categories, asking MLLM to select the category that best matches the image. Here is the instruction we use for evaluation: 'Please complete the fine-grained image classification task. Below is a picture of a bird; please select the label that best matches the image from the options below: {cls0}, {cls1}...'. To avoid evaluation errors caused by improper output formatting, we only consider the classification accuracy of MLLM when its output falls within

Table 2. Performance comparison of using attributes form LLM and MLLM. The numbers indicate the number of prompts.

Method	CUB		Stanford-Cars		FGVC	
	AUROC↑	OSCR↑	AUROC↑	OSCR↑	AUROC↑	OSCR↑
LLM (4)	89.31 / 81.38	81.99 / 76.26	94.07 / 86.24	92.05 / 82.90	92.34 / 84.90	86.90 / 80.83
MLLM (1)	89.51 / 82.13	81.73 / 76.26	94.33 / 83.10	92.65 / 82.03	93.08 / 84.62	88.48 / 80.84
MLLM (4)	90.17 / 81.70	83.25 / 76.69	94.56 / 84.17	92.94 / 83.12	93.54 / 84.52	88.70 / 81.03

Table 3. The performance of different methods using MLLM-generated attributes or not.

Method	CUB		Stanford-Cars		FGVC	
	AUROC↑	OSCR↑	AUROC↑	OSCR↑	AUROC↑	OSCR↑
w/o MLLM						
MSP	89.68 / 81.04	82.57 / 75.78	94.38 / 85.66	91.83 / 84.06	91.95 / 82.25	84.58 / 76.79
ARPL	90.96 / 82.48	84.32 / 77.64	94.82 / 85.01	92.36 / 83.44	91.39 / 82.86	84.28 / 77.34
KPF	89.57 / 78.83	81.76 / 73.00	92.11 / 84.24	89.54 / 82.32	93.00 / 79.46	85.96 / 74.42
w/ MLLM						
MSP	90.17 / 81.70	83.25 / 76.69	94.56 / 84.17	92.94 / 83.12	93.54 / 84.52	88.70 / 81.03
ARPL	91.15 / 82.46	83.88 / 77.29	94.92 / 86.41	92.68 / 84.90	94.19 / 84.23	89.13 / 80.66
KPF	89.71 / 83.00	83.49 / 78.34	94.56 / 85.45	92.83 / 84.40	91.29 / 83.88	86.84 / 80.60

Table 4. Comparison of MLLM and our method for clasification.

Method	CUB	Stanford-Cars	FGVC
MLLM	24.72	72.92	8.14
Ours	89.04	89.51	96.94

the specified category range. However, the results indicate that MLLM performs unsatisfactorily in classification tasks. It only shows some effectiveness on Stanford Cars, falling short compared to supervised training models, and performs poorly on the other two datasets.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. 1

[2] Peter Belcak and Roger Wattenhofer. Exponentially faster language modelling. *arXiv preprint arXiv:2311.10770*, 2023. 1

[3] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. 3

[4] Afshin Dehghan, Syed Zain Masood, Guang Shu, Enrique Ortiz, et al. View independent vehicle make, model and color recognition using convolutional neural network. *arXiv preprint arXiv:1702.01721*, 2017. 2

[5] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pages 951–958. IEEE, 2009. 2

[6] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 2

[7] Yifei Ming, Ziyang Cai, Jiuxiang Gu, Yiyou Sun, Wei Li, and Yixuan Li. Delving into out-of-distribution detection with vision-language representations. *Advances in Neural Information Processing Systems*, 35:35087–35102, 2022. 3

[8] Yifei Ming and Yixuan Li. How does fine-tuning impact out-of-distribution detection for vision-language models? *International Journal of Computer Vision*, 132(2):596–609, 2024. 3

[9] Xinyu Tian, Shu Zou, Zhaoyuan Yang, and Jing Zhang. Argue: Attribute-guided prompt tuning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28578–28587, 2024. 1

[10] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 8769–8778, 2018. 3

[11] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need? *arXiv preprint arXiv:2110.06207*, 2021. 2

[12] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2

[13] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 3

[14] Bo Zhao, Yanwei Fu, Rui Liang, Jiahong Wu, Yonggang Wang, and Yizhou Wang. A large-scale attribute dataset for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[15] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 3