Language-Driven Multi-Label Zero-Shot Learning with Semantic Granularity

Supplementary Material

A. Semantic Granularity

Traditional multi-label classification methods focus more on the co-occurrence relationship of labels and pay little attention to the semantic granularity of labels. They excel at handling mutually competitive categories from object-centric datasets (*e.g.*, VOC [4] and MS-COCO [10]), such as cat and dog. Due to the demands of practical tasks, categories defined with rich semantic granularity should be identifiable. For instance, in image retrieval tasks, the system is often expected to retrieve related images at various semantic levels, such as "cat" and "animal."

Multi-label datasets with diverse semantic granularities, such as NUS-WIDE and Open Images, primarily consist of image data sourced from the web, such as Flickr. Since there is no unified category standard, users annotate images based on their own understanding. This results in the annotation of abstract classes, scene classes, and classes at varying semantic levels, posing significant challenges for image recognition, as illustrated in Fig. 1. To further illustrate this, we select four common superclasses—animal, vehicle, person, and building—from over 7,000 categories in Open Images and constructed the category hierarchy shown in Fig. 2. This hierarchy reveals multiple semantic levels and a large number of fine-grained categories. However, the actual hierarchies for all categories in NUS-WIDE and Open Images are not provided, meaning that the semantic granularity problem cannot be addressed by relying solely on the real hierarchical information of these datasets.

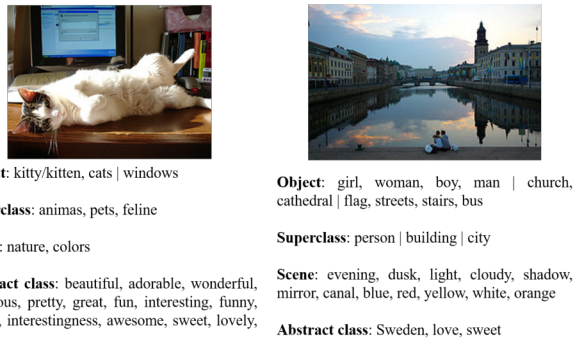

Figure 1. Examples annotated by the subset of 1006 classes from NUS-WIDE.

B. Text Description Generation Details

The collection of textual data is crucial to coping with various semantic granularities in our method, whose rich knowledge can be extracted into class names. Based on how humans identify novel classes, we collect textual descriptions of visual features, hierarchies, and co-occurrence scenes about categories. We expect language models to generate distinct and separate descriptors instead of integrating all elements together, which is beneficial to more detailed knowledge extraction when reconstructing class names. The GPT-4o mini model with a temperature parameter of 0.7 is used as our language model, which generates all text results via its public API.

Following the work [13], the prompt template with an example is used to generate individual attribute descriptors of visual features, as detailed below:

```
Q: What are useful visual features for
    distinguishing a television in a
    photo?
A: There are several useful visual
    features to tell there is a
    television in a photo:
- black or grey
- electronic device
- a large, rectangular screen
- a stand or mount to support the
    screen
- one or more speakers
- a power cord
- input ports for connecting to other
    devices
- a remote control

Q: What are useful features for
    distinguishing a {class_name} in a
    photo?
A: There are several useful visual
    features to tell there is a {
    class_name} in a photo: (Please
    generate it according to the example
     style above.)
-
```

To adapt the input form of the CLIP's text encoder, we further post-process each visual attribute to generate a complete text description. We integrate each visual attribute as a descriptor into the text template "`a {class_name}, which (is/has/etc) {descriptor}.`". For the above example, the attribute descriptor "a large, rectangular screen" is expanded to the text description "a television, which is a large, rectangular screen.". Attribute descriptors starting with "a" or "an" apply to "which is", while others

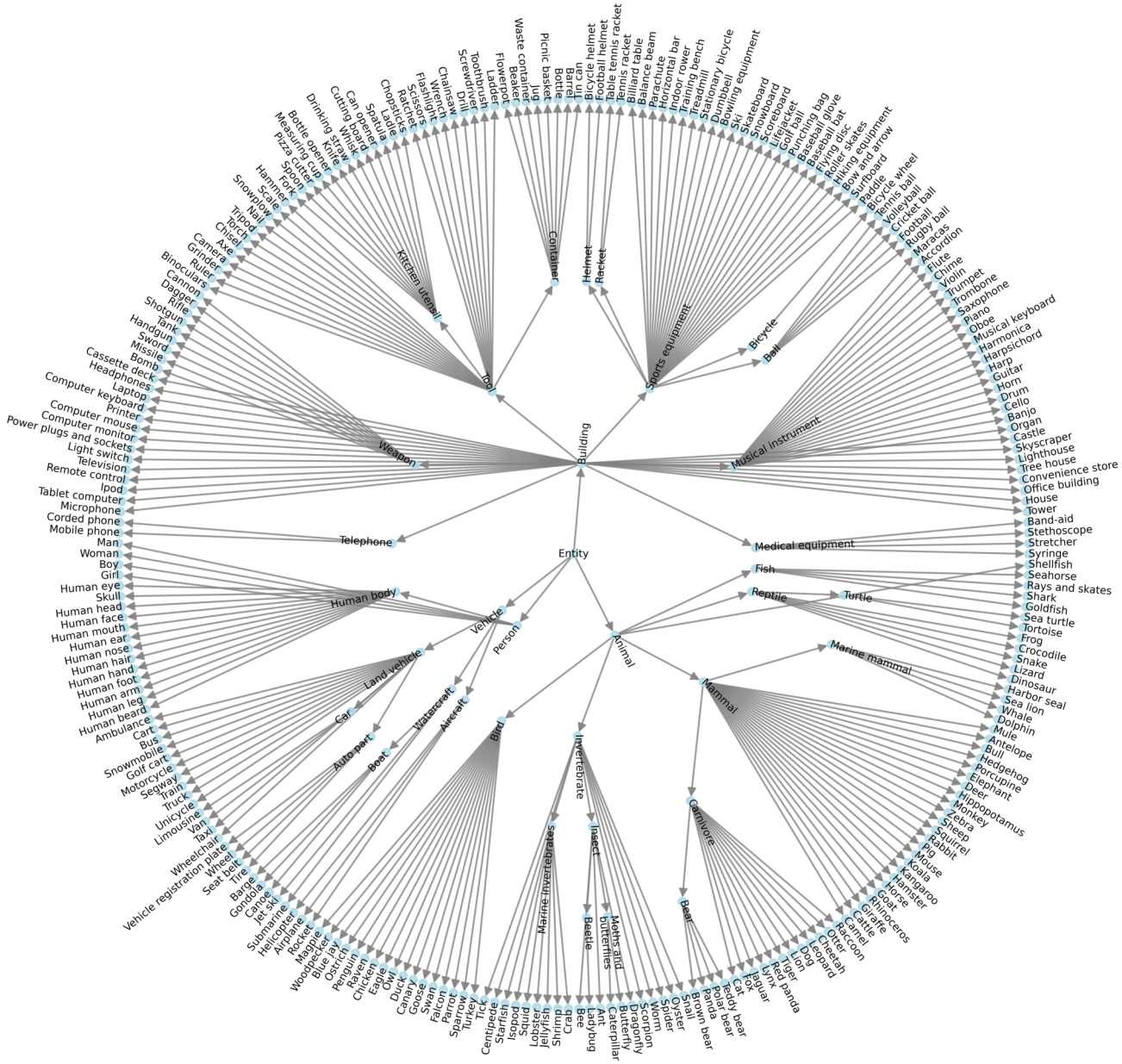

Figure 2. Hierarchical visualization of the partial categories from Open Images.

often apply to "which has", such as "a television, which has black or grey." for "black or grey".

The subclasses and superclasses of a class can be obtained by querying the language model through the following prompts [11]:

```
Generate a list of 10 types of the
    following category and output the
    list separated by '&' (without
    numbers): {class_name}
Generate a list of 3 super-categories
    that the following category belongs
    to and output the list separated by
    '&' (without numbers): {class_name}
```

For example, for the category "cat", the language model outputs its superclasses—pet, animal, and mammal—as well as its subclasses, such as Ragdoll, Sphynx, and others. As shown in Fig. 3, we build two 2-level hierarchies and a 3-level hierarchy. Elements of each level from the above hierarchies are retrieved to establish the relationship between a class and its sub/superclasses through

Dataset	Number of Classes	$\mathcal{D}^{vdes}$	$\mathcal{D}^{hier}$	$\mathcal{D}^{cos}$	Total	Avg. per class
MS-COCO	65	2,644	6,021	1,199	9,864	151.8
NUS-WIDE	1,006	43,607	103,521	18,384	165,512	164.5
Open Images	7,472	313,107	893,837	136,155	1,343,099	179.8

Table 1. Statistics on the number of text descriptions generated by GPT for different datasets.

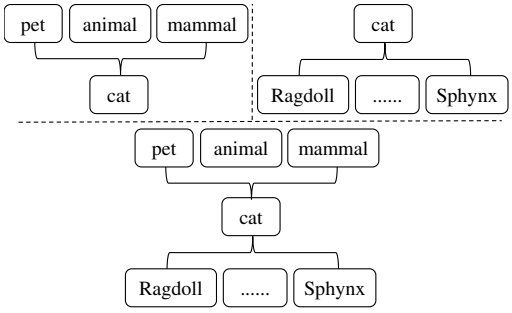

Figure 3. Semantic hierarchies of the category "cat".

Is-A [11], such as "`a {class_name}, which is a {superclass}.`". We show a text description example for each hierarchy, such as "a cat, which is a pet.", "a Ragdoll, which is a cat." and "a Ragdoll, which is a cat, which is a pet.".

We query the language model to get a list of co-occurrence scenes using the following prompt:

```
Please make a list of possible
    backgrounds where a {class_name}
    appears in a photo:
-
```

Scenes generated by LLM and the class name of a class are fused together to form scene descriptions, *i.e.*, "`a {class_name}, which appears together with the {scene}.`". For example, we get a list of scenes that co-occur with a cat: living room, garden, park, bedroom, pet store, and so on. Each scene is integrated into the text description, such as "a cat, which appears together with the garden." for "garden".

We try to keep the text descriptions of the above three types in a consistent format as much as possible, which helps to eliminate noise from different sentence forms and allows the text encoder to focus on the discriminative text data. The post-processed text descriptions, including visual attributes $\mathcal{D}^{vdes}$, hierarchical relationships $\mathcal{D}^{hier}$, and co-occurrence scenes $\mathcal{D}^{cos}$, are added to the text corpus as samples for training. As shown in Tab. 1, we count the number of samples in the three text description sets for each dataset and the average number of text samples in each category. The number of text samples in each category varies very little for different datasets.

Discussion: Differences in the Collection of Text Descriptions for Language-Driven Methods. TaI-DPT [5] tunes prompts by treating text data as images, whereas CoMC [12] uses text data to train a cross-modal classifier. We also use text descriptions as training samples to tackle the multi-label zero-shot learning task. In contrast, the text content used in our method is completely different from theirs. The text descriptions of TaI-DPT come from public image caption datasets(*e.g.*,MS-COCO, Opem Images). The captions in these datasets are annotated by humans and have high reliability with limited numbers. CoMC leverages GPT to generate text data in order to save labor and break through quantity limitations. The common point between the text descriptions of TaI-DPT and CoMC is that they both describe the content of photos containing specific categories, thereby deriving category labels for supervised learning. For example, the description "A cat perches curiously on the hood of a parked car." derives the labels "cat" and "car". To consider the diversity of text descriptions, synonym descriptions are added to the training corpus to derive target labels. For example, the description "A girl is sitting on the sofa with her puppy." derives the labels "person" and "dog". This requires the establishment of a synonym vocabulary that maps one-to-one with the target classes, which brings a lot of work to text post-processing. In addition, many image captions generated by GPT may not be real and need to be filtered manually.

Our method of collecting text descriptions is almost entirely automatic. After accessing GPT through the designed prompts and obtaining the results, the text descriptions are output according to the corresponding templates. GPT's output of category-related descriptions is relatively reliable and does not require manual filtering. Compared with the text descriptions of TaI-DPT and CoMC, the text descriptions of our method are based on the descriptions of the relationships and features for a class itself, such as "a cat, which is a pet." and "a cat, which has whiskers on the face.". Furthermore, the number of text descriptions trained in our method is significantly less than that required by CoMC. For example, for 81 unseen classes of NUS-WIDE, CoMC needs 40,000 sentences to train a classifier, while our method only uses 12,951 sentences to reconstruct class names.

Baseline	$\mathcal{D}^{vdes}$	$\mathcal{D}^{hier}$	$\mathcal{D}^{cos}$	Task	MS-COCO			NUS-WIDE		
					mAP	F1 (Top-3)	F1 (Top-5)	mAP	F1 (Top-3)	F1 (Top-5)
✓				ZSL	74.6	48.0	35.1	46.5	38.4	35.5
				GZSL	57.0	46.4	41.2	16.3	18.2	20.8
	✓			ZSL	80.4	50.9	36.9	51.3	45.2	42.9
				GZSL	64.1	52.2	46.8	17.5	19.3	22.1
		✓		ZSL	81.9	51.5	37.2	51.4	45.1	42.8
				GZSL	66.0	53.4	47.8	17.5	19.2	22.2
			✓	ZSL	81.5	50.6	36.9	51.5	45.4	43.1
				GZSL	65.1	51.9	46.6	17.7	19.4	22.2
	✓	✓	✓	ZSL	82.2	52.2	37.5	51.7	45.9	43.2
				GZSL	66.7	54.8	49.0	17.7	19.5	22.5

Table 2. Ablation study for different types of text descriptions. In the ZSL and GZSL tasks, mAP over all classes and F1 scores of Top-3 and Top-5 predictions on MS-COCO and NUS-WIDE are reported.

Methods	Task	MS-COCO			NUS-WIDE		
		mAP	F1 (Top-3)	F1 (Top-5)	mAP	F1 (Top-3)	F1 (Top-5)
Mean	ZSL	82.3	51.3	37.2	51.5	44.8	42.8
	GZSL	66.7	53.7	48.2	17.7	19.5	22.4
P-Eigen	ZSL	82.2	51.3	37.1	50.4	44.5	42.6
	GZSL	65.6	53.6	47.9	17.3	19.4	22.3
Ours	ZSL	84.1	52.8	37.4	51.7	47.4	44.4
	GZSL	69.6	59.4	51.0	17.9	19.5	22.5

Table 3. Comparison of text corpus integration methods. Mean and P-Eigen represent the average and the principal eigenvector, respectively.

C. Experiment Details

C.1. Dataset Details

MS-COCO is an object-centric dataset with no semantic granularity level. It is split into 48 seen classes and 17 unseen classes in the works [1, 2]. NUS-WIDE and Open Images (v4) contain a large number of categories and are rich in semantic granularity. The NUS-WIDE dataset is a web-based collection comprising 107,859 test images, encompassing 81 human-verified labels along with 925 labels obtained from Flickr user tags. As in LESA [7], the 925 labels and the remaining 81 labels are treated as seen and unseen, respectively. Open Images (v4) is a large-scale dataset including nearly 9 million training images and 125,456 test images. Following previous studies [6, 7], 7,186 labels with at least 100 images per class are designated as seen labels in the training set. The 400 most frequent test labels are selected as unseen labels, each being absent from the training set.

C.2. Implementation Details

The GPT-4o mini model [3] is used as our LLM to collect all text descriptions. We can access its public API to get the answer to a prompt containing a class name and call the API multiple times with the same prompt to get richer text results. Our backbone adopts the pre-trained CLIP with the image encoder ViT-B/16 and the corresponding text encoder. During training, the same frozen text encoder is used to encode both the class prompts and the text descriptions. The learnable class name vectors are randomly initialized. The only training parameters of our model are the class name vectors for all classes and they are optimized by the SGD optimizer with a batch size of 512 text descriptions in 30 epochs. Meanwhile, a cosine learning rate decay is used, and the initial learning is set to 1e-4. During inference, the images of input size 224×224 are fed into the image encoder. An image is cropped into $K \in \{4, 9\}$ snippets to perform semantic aggregation. The temperature coefficient τ_s is set to 0.05 for semantic aggregation of local tokens. Furthermore, we perform a sensitivity analysis for hyperparameters N_{c^*} and λ to choose the appropriate value, respectively. All experiments are conducted on Tesla V100 with a fixed random seed.

D. More Experimental Results

D.1. Ablation Study for Text Descriptions

To evaluate the effectiveness of each type of text description in reconstructing class names, we train class names separately using three different types of descriptions, including visual attributes $\mathcal{D}^{vdes}$, hierarchical relationships $\mathcal{D}^{hier}$, and co-occurrence scenes $\mathcal{D}^{cos}$. The results are shown in

Snippets(K)	Task	MS-COCO			NUS-WIDE		
		mAP	F1 (Top-3)	F1 (Top-5)	mAP	F1 (Top-3)	F1 (Top-5)
1 (1×1)	ZSL	84.1	52.8	37.4	51.7	47.4	44.4
	GZSL	69.6	59.4	51.0	17.9	19.5	22.5
4 (2×2)	ZSL	85.3	53.5	37.7	53.3	48.4	45.3
	GZSL	71.4	60.4	51.9	17.9	19.3	22.4
9 (3×3)	ZSL	86.1	53.6	37.7	53.6	48.7	45.6
	GZSL	72.9	60.4	52.0	17.5	18.8	22.0
16 (4×4)	ZSL	85.9	53.4	37.6	53.5	48.7	45.7
	GZSL	72.9	60.4	51.8	17.2	18.4	21.6

Table 4. Effect of the number of snippets on the performance of MSSA.

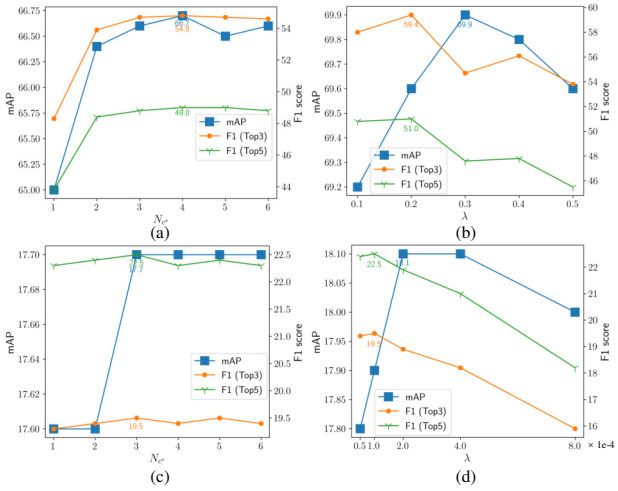

Figure 4. Effect of hyperparameters. The results of the GZSL task with respect to hyperparameters N_{c^*} and λ on MS-COCO (top row) and NUS-WIDE (bottom row) are shown.

Tab. 2. The baseline setup remains consistent with that in Tab.3 of the main paper, and no pair-based loss is applied to reconstructed class names.

Compared to the baseline, our approach significantly improves performance. For MS-COCO, class names reconstructed using hierarchical relationship descriptions achieve the best results, while combining all three types further improves performance. For NUS-WIDE, co-occurrence scene descriptions yield the best performance, though the differences among the three types of descriptions are relatively small. Nonetheless, integrating all three types still leads to performance gains.

D.2. Analysis of Text Integration Methods

We unify the embeddings of all class-related text descriptions into a single representation for each class using either the mean or the principal eigenvector. To verify the effectiveness of our class name reconstruction method, we compare it with these two approaches. For a fair comparison, our method does not include MSSA. As shown in Tab. 3, for MS-COCO, our method outperforms both the mean and principal eigenvector approaches in the ZSL task, improving mAP and F1 @ Top-3 and Top-5 by 1.8%, 1.5%, and 0.2%, and by 1.9%, 1.5%, and 0.3%, respectively. In the GZSL task, our method achieves even greater gains, surpassing the two approaches by 2.9%, 5.7%, and 2.8% and by 4.0%, 5.8%, and 3.1% for mAP and F1 @ Top-3 and Top-5. For NUS-WIDE, our method does not have a significant performance advantage in the GZSL task. In the ZSL task, our method outperforms both the mean and principal eigenvector approaches, improving mAP and F1 @ Top-3 and Top-5 by 0.2%, 2.6%, and 1.6%, and by 1.3%, 2.9%, and 1.8%, respectively.

D.3. Varying the Hyperparameters

We conduct a sensitivity analysis on the learnable class name token length N_{c^*} and balance hyperparameter λ across both datasets, evaluating their impact on mAP and F1 scores at Top-3 and Top-5 predictions. Since both seen and unseen classes are treated as novel classes, we focus on the influence of hyperparameters in the GZSL task. For MS-COCO, as shown in Fig. 4(a), mAP and F1 scores improve when N_{c^*} exceeds 1, reaching optimal performance at $N_{c^*} = 4$. A shorter class name token length limits the learning of semantic knowledge. Additionally, as λ varies, F1 scores fluctuate significantly in Fig. 4(b). We select λ based on the best F1 scores, setting it to 0.2. For NUS-WIDE, changes in N_{c^*} have minimal impact on the metrics, particularly F1 scores, as seen in Fig. 4(c), with the best performance observed at $N_{c^*} = 3$. In Fig. 4(d), λ significantly affects F1 scores when it exceeds $1e-4$, suggesting that an excessively large λ may hinder the optimization of the distance loss $\mathcal{L}_{MSE}$. The optimal choice is $\lambda = 1e-4$. All our losses are summed together. In the GZSL task, as the number of novel categories in the dataset increases, λ is progressively reduced to minimize the impact of $\mathcal{L}_D$.

In addition, we analyze the effect of the number of snippets (K) on the Multi-Snippet Semantic Aggregation (MSSA) module. The number of snippets directly affects

Dataset	Task	no merging			merging		
		mAP	F1 (Top-3)	F1 (Top-5)	mAP	F1 (Top-3)	F1 (Top-5)
NUS-WIDE	ZSL	51.7	47.4	44.4	51.7	47.4	44.4
	GZSL	17.9	19.5	22.5	32.1	25.8	31.0

Table 5. The results of PBL after synonym merging.

Method	Task	MS-COCO			NUS-WIDE		
		mAP	F1 (Top-3)	F1 (Top-5)	mAP	F1 (Top-3)	F1 (Top-5)
RCNn w/o MSSA	ZSL	84.1	52.8	37.4	51.7	47.4	44.4
	GZSL	69.6	59.4	51.0	17.9	19.5	22.5
RCNn w/ MSSA	ZSL	86.3	53.6	37.7	53.3	48.4	45.3
	GZSL	73.1	60.7	52.0	17.9	19.3	22.4
RCNn + IFT w/o MSSA	ZSL	85.0	53.8	37.9	51.4	46.4	43.4
	GZSL	78.1	67.5	57.5	19.8	23.3	27.2
RCNn + IFT w/ MSSA	ZSL	87.0	54.4	38.1	53.8	47.9	44.8
	GZSL	80.9	68.2	58.5	19.7	23.3	27.5

Table 6. The performance of RCNn integrated with IFT.

the balance between global and local semantic predictions. We explore this effect on the object-centric dataset MS-COCO and the multi-granularity semantic dataset NUS-WIDE. For each image, we crop it into 1×1, 2×2, 3×3, and 4×4 snippets, corresponding to $K = 1, 4, 9, 16$, respectively. The experimental results are shown in Tab. 4. For MS-COCO, all metrics improve as K increases, with mAP showing particularly significant gains. The best performance on both ZSL and GZSL tasks is achieved when $K = 9$. Both the seen and unseen classes in MS-COCO are object categories, and increasing K provides more fine-grained local semantic information. Considering the relatively small size of its test set, both $K = 4$ and $K = 9$ are selected. On NUS-WIDE, as K increases, all metrics on the ZSL task show improvements. However, on the GZSL task, the performance does not improve and even drops at $K = 4$, with a more significant decline observed at $K = 9$. This is mainly because the unseen classes in NUS-WIDE are primarily object categories, which require more fine-grained local semantic information. This trend is consistent with that observed on MS-COCO. In contrast, the seen classes in NUS-WIDE are numerous and semantically diverse, with abstract, scene, and parent classes appearing frequently. This leads the seen classes to favor global semantic information, resulting in the best GZSL performance when $K = 1$. Taking all factors into consideration, we select $K = 4$ for NUS-WIDE.

D.4. Merging of Synonyms

As discussed in Section 4.3 of the main paper, ranking-based evaluation metrics (*e.g.*, mAP and F1 score) may not accurately assess the predictions due to the limitations of existing ground-truth. To further validate that the ground-truth labels are insufficient for fairly evaluating our method, we adopt the evaluation protocol proposed in [9] to re-assess the predictions. Following its synonym matching approach, we identify and merge synonyms within the seen classes to avoid duplicate predictions. Specifically, we use the CLIP's text encoder to encode the reconstructed class names, and discover synonyms based on the similarity between class embeddings for subsequent merging. The predictions are then aggregated according to the synonym merging rules, which ensures that the Top-3 and Top-5 predictions for each image contain minimal redundant synonymous classes, thereby better matching the diverse classes in the ground-truth labels. The results of PBL after synonym merging are shown in Tab. 5, where all metrics on the GZSL task are significantly improved. This indicates that the ground-truth labels of seen classes for each image omit many synonymous annotations, making them insufficient for accurately evaluating the performance of semantic granularity-aware multi-label classification. In the ZSL task, the unseen classes do not contain synonyms, and thus the performance remains unchanged.

D.5. Image-based Fine-tuning

Our method (RCNn) uses textual descriptions to reconstruct class names under a fixed prompt context. It can further incorporate image-based fine-tuning (IFT) to learn the prompt context dynamically. In line with vision-driven ML-ZSL approaches, we fine-tune the prompts using image data from seen classes. Specifically, we adopt the prompt learning method of Independent V-L Prompting in [8], where the class names are those reconstructed from textual descriptions. The experiments are conducted using RCNn with (w/) and without (w/o) MSSA, and the results are presented

in Tab. 6. For MS-COCO, integrating IFT with RCNn leads to performance improvements on both the ZSL and GZSL tasks. The improvement is particularly significant for the GZSL task. For NUS-WIDE, incorporating IFT significantly improves performance on the GZSL task, while it leads to a decline in performance on the ZSL task.

D.6. Visualization of Attention Maps

As shown in Fig. 5, we visualize attention maps for several images from NUS-WIDE. Compared to CLIP, our method captures relevant regions with greater precision. For instance, in the last image, while CLIP highlights a broader area for "garden" and "grass," our method focuses on the most relevant regions. Additionally, our approach effectively detects small objects, such as "window." It also maintains semantic consistency across different granularity levels, as seen with "tiger" and "animal." Furthermore, scene categories like "sky," "reflection," "valley," and "garden" are well recognized.

E. Limitations

Our method relies on a large language model (LLM), and its performance is closely tied to the quality of the LLM's output. Given the vast number of categories in large-scale datasets (*e.g.*, Open Images) and their complex semantic hierarchies, the LLM's inherent knowledge limitations often prevent it from generating accurate attributes for many categories. Additionally, its hallucinations may introduce attributes that do not exist in certain classes, posing a potential risk of misleading our approach. The hierarchical relationships generated by LLM are generalized and may not align with the actual hierarchical relationships in a dataset, leading to inconsistencies in parent and child category predictions.

References

[1] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 384–400, 2018. 4

[2] Avi Ben-Cohen, Nadav Zamir, Emanuel Ben-Baruch, Itamar Friedman, and Lihi Zelnik-Manor. Semantic diversity learning for zero-shot multi-label classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 640–650, 2021. 4

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020. 4

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2010. 1

[5] Zixian Guo, Bowen Dong, Zhilong Ji, Jinfeng Bai, Yiwen Guo, and Wangmeng Zuo. Texts as images in prompt tuning for multi-label image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2808–2817, 2023. 3

[6] Sunan He, Taian Guo, Tao Dai, Ruizhi Qiao, Xiujun Shu, Bo Ren, and Shu-Tao Xia. Open-vocabulary multi-label classification via multi-modal knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 808–816, 2023. 4

[7] Dat Huynh and Ehsan Elhamifar. A shared multi-attention framework for multi-label zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8776–8786, 2020. 4

[8] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19113–19122, 2023. 6

[9] Chuang Lin, Yi Jiang, Lizhen Qu, Zehuan Yuan, and Jianfei Cai. Generative region-language pretraining for open-ended object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13958–13968, 2024. 6

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1

[11] Mingxuan Liu, Tyler L Hayes, Elisa Ricci, Gabriela Csurka, and Riccardo Volpi. Shine: Semantic hierarchy nexus for open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16634–16644, 2024. 2, 3

[12] Yicheng Liu, Jie Wen, Chengliang Liu, Xiaozhao Fang, Zuoyong Li, Yong Xu, and Zheng Zhang. Language-driven cross-modal classifier for zero-shot multi-label image recognition. In *Forty-first International Conference on Machine Learning*, 2024. 3

[13] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *the Eleventh International Conference on Learning Representations*, 2023. 1

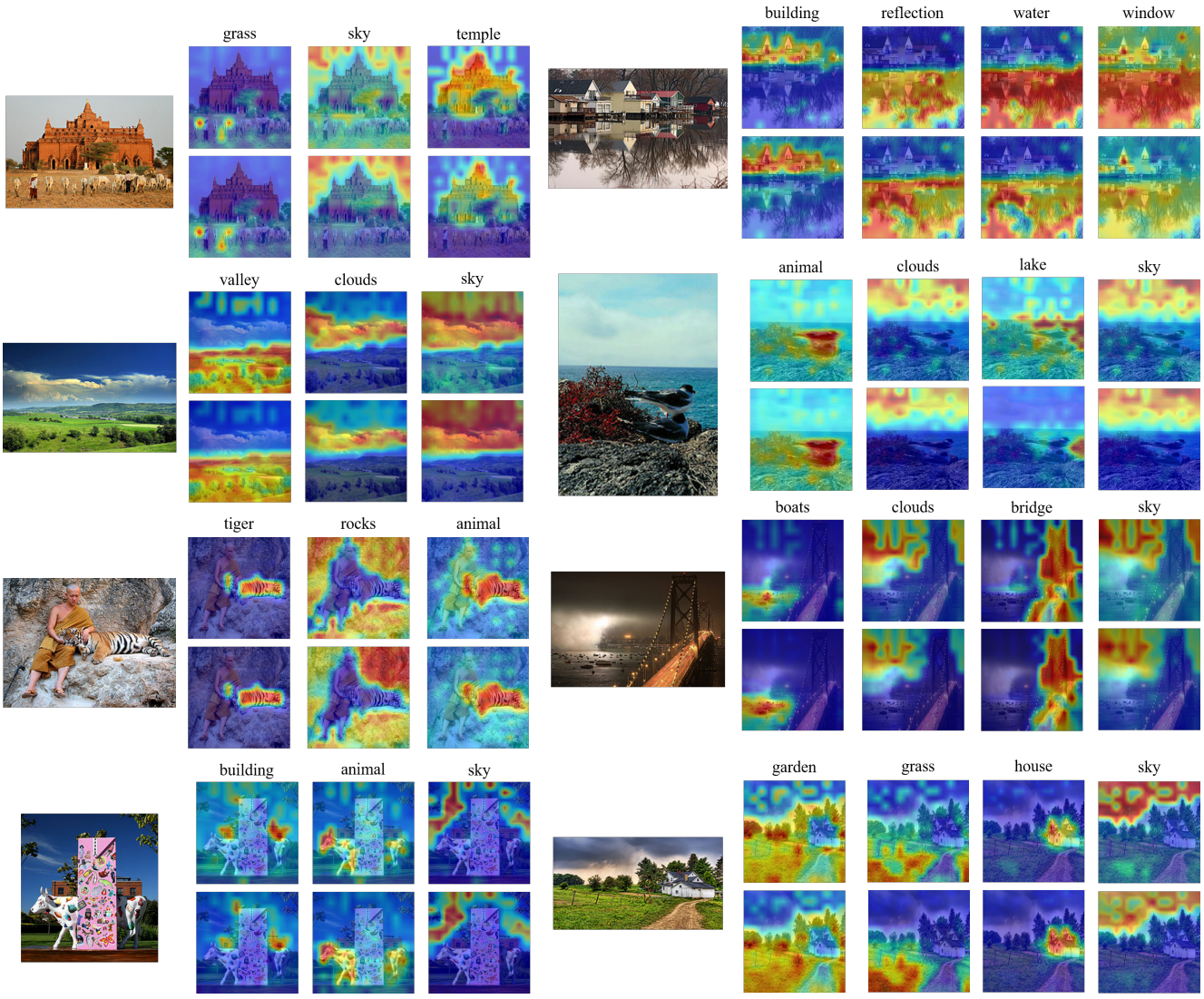

Figure 5. Visualization of attention maps for CLIP (top row) and our method (bottom row) on NUS-WIDE test images.