Attribute-formed Class-specific Concept Space: Endowing Language Bottleneck Model with Better Interpretability and Scalability

Supplementary Material

A. The implementation detail of DSS

A.1. The detail of summary step

This subsection introduces the implementation detail of the summary step in DSS. The diagram of the summary step is shown in Fig. A1. Considering the token length limitation of LLM and its unreliability in generating long texts, it is unable to directly ask LLM to summary the entire attribute set by giving LLM all the generated concepts $\{\boldsymbol{c}_i\}_{i=1}^{K}$ in the description step. Therefore, we query the attributes on a per-class basis. Furthermore, since quarrying LLM multiple times may lead to inconsistent outputs (e.g., nose & snout), we adopt an iterative approach to summarize the attribute set $\check{\mathbb{A}}$. That is, for each query, we encourage the LLM to use words from the existing attribute set $\check{\mathbb{A}}_{i-1}$ to summarize the attributes of $\boldsymbol{c}_i$, and only output new words when no suitable attributes are available in $\check{\mathbb{A}}_{i-1}$ based on the prompt q_{sum}, formally,

$$\check{\mathbb{A}}_i = \check{\mathbb{A}}_{i-1} + \text{LLM}\left(\boldsymbol{c}_i, \check{\mathbb{A}}_{i-1}, q_{sum}\right). \tag{A1}$$

The overall attribute set $\check{\mathbb{A}}$ is equal to $\check{\mathbb{A}}_K$.

To further avoid duplicate and synonymous attributes, we use LLM to resummarize the attribute set with the prompt q_{res}, formally,

$$\overline{\mathbb{A}} = \text{LLM}\left(\check{\mathbb{A}}, q_{res}\right). \tag{A2}$$

However, the attribute set summarized by the above prompt still has two limitations. The first is that some collected attributes describe the non-visual information of classes (e.g., the alternative name, smell, etc.). Predictions based on these non-visual attributes will disturb the interpretability of the inference process. And the other limitation is that some attributes are very sparse on categories, with only a few classes having corresponding descriptions. To address the above two limitations, we first use LLM to filter non-visual attributes using the prompt p_{vis}, formally,

$$\hat{\mathbb{A}} = \overline{\mathbb{A}} - \text{LLM}\left(\overline{\mathbb{A}}, q_{vis}\right), \tag{A3}$$

where $\hat{\mathbb{A}}$ represents the visual attribute set, $\text{LLM}\left(\overline{\mathbb{A}}, q_{vis}\right)$ is the non-visual attribute set summarized by LLM.

Then we count the number of descriptions corresponding to each attribute and remove attributes that occur with a frequency of less than $r\%$ across all classes to obtain the final attribute set $\mathbb{A}$.

A.2. The prompts used in DSS

This subsection introduces the prompts used in DSS. Since we directly use the concept sets collected by existing work [33] as the output of the Description Step, the prompts used in this step are identical to its released prompts. In the Summary Step, we first use q_{sum} to prompt LLM iteratively summarize the attributes by category, which is as follows:

> Your task is to extract attributes of different categories from the descriptions I gave you.
> Specially, you can complete the task by following the instructions:
> 1. You can select the noun related to the attribute form exsit attribute set, and if you think the attribute describe by the phrase is not among them, you can answer other words.
> 2. Each phrase corresponds to a description, and the number of the two should also be consistent.
> 3. Output a Python dictionary with the {attribute name} as the key, and no newline required between each description. PLEASE USE ":" AFTER the KEY.

Subsequently, we use LLM to remove duplicate attributes from the attribute set with the prompt q_{res}:

> Your task is to merge the attributes I give you into semantically consistent attribute groups.
> Specially, you can complete the task by following the instructions:
> 1. Only merge the attributes I give, and only merge semantically consistent attributes.
> 2. The semantics of the merged attributes should not be repeated.
> 3. The words representing an attribute group must be the words of the attributes I give, and the words in the same attribute group must all come from the attributes I give.
> 4. The sum of the words in all attribute groups should be equal to the attribute set I gave.
> 5. Output some python lists, each list represents a attribute group.
> ===
> Please merge semantically consistent attribute among the attributes attribute set:
> ===

Next, we use LLM to remove non-visual attributes with the prompt q_{vis}:

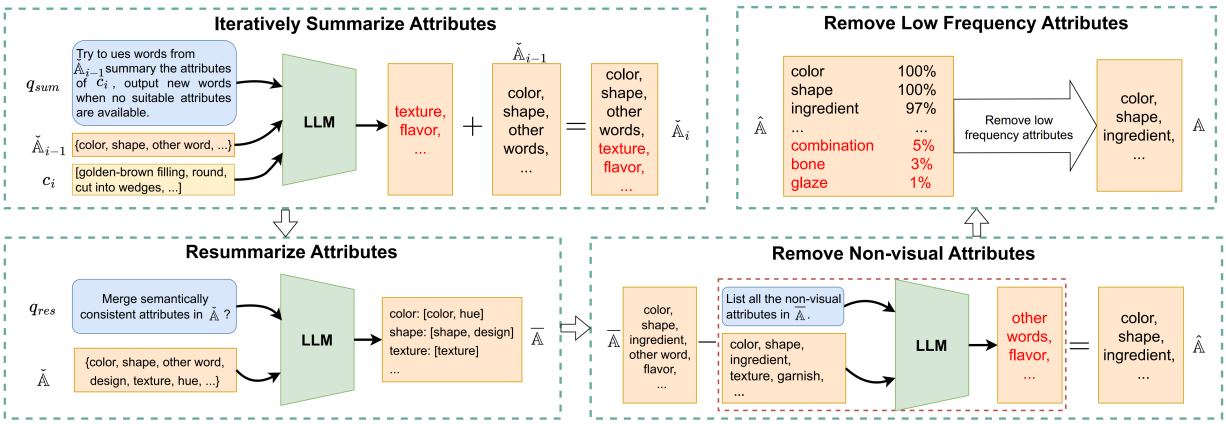


Figure A1. Illustration of the summary step in DSS strategy. Specifically, we summarize the attributes of class concepts through the following steps: first, iteratively summarize the attributes by category; next, remove duplicate attributes from the attribute set; then, eliminate non-visual attributes; and finally, remove sparse attributes.

	Approach	Aircraft	CUB	DTD	Flowers102	Food101	OxfordPets	CIFAR-10	CIFAR-100	ImageNet
Unexplainable	ZS-CLIP [25]	32.6	63.4	53.2	79.3	91.0	93.6	**86.0**	55.6	71.4
Training-free Language Bottleneck	VDCLIP [20]	-	63.5	54.4	-	92.4	92.3	-	-	71.5
	CuPL [24]	**36.7**	-	58.9	78.8	91.2	93.4	83.4	60.4	**74.1**
	CLIP-GPT [19]	34.5	64.8	56.4	77.8	91.1	92.8	-	-	71.8
	ALBM* (ours)	34.4	**66.5**	**59.9**	**79.9**	**91.6**	**93.9**	85.4	**61.5**	73.4

Table A1. Comparison with zero-shot CLIP and training-free language bottlenecks on the zero-shot setting, where class names are added in the concepts, ALBM* indicate zero-shot prediction based on our collected concept sets, and "-" indicates that the original approaches didn't collect the concept set for the dataset.

> Suppose you have some photos of {all class name}, please write down {attribute set} in order whether these attributes are the visual attributes of these pictures:

In the Supplement Step, we utilize LLM to supply the missing concepts with the prompt q_{sup} as follows:

> Your task is to describe a certain attribute of a certain class.
> Specially, you can complete the task by using short and precise descriptions. And no newline is required before each description.
> ===
> Please describe the attribute {attribute} of the class {class name} according to the following examples, and no newline required between each description:
> ===

where the content inside the curly braces represents the corresponding variables.

B. Additional analysis

B.1. Zero-shot performance with class names

In Tab. 3, we compared our approach with existing TfLB approaches under the setting where class descriptions only include visual concepts without class names, to rigorously evaluate the performance of interpretable image recognition. However, the performance of existing TfLB approaches suffers significantly under this setting. As a result, existing TfLB approaches [19, 24, 32] recommend including class names in the descriptions, such as "a photo of a class name, which has/is class concept," to achieve better classification performance. To further validate the effectiveness of our proposed method, we conducted comparative experiments under this setting as well, as shown in Tab. A1. From Tab. A1, it can be seen that our approach achieves the best performance on 6 out of 9 datasets, slightly underperforming the current state-of-the-art results on only 3 datasets. These results demonstrate the effectiveness of our proposed DSS strategy, which extracts more comprehensive visual information for each class by summarizing a cross-class shared attribute set.

	Approach	Aircraft	CUB	DTD	Flowers102	Food101	CIFAR-10	CIFAR-100	ImageNet	Average
Class-Shared Concept Space	Labo [33]	45.6	78.2	67.6	92.6	87.6	85.7	45.5	71.0	71.7
	ALBM (ours)	**53.5**	**83.4**	**68.6**	**98.1**	**89.1**	**86.0**	**62.4**	**76.5**	**77.2**
Class-Specific Concept Space	Labo [33]	**41.2**	69.5	66.3	**95.7**	82.9	80.9	49.5	69.2	69.4
	ALBM (ours)	40.0	**69.7**	**69.4**	92.4	**84.8**	**84.2**	**52.5**	**70.0**	**70.4**

Table A2. Comparison with existing LBM approach LaBo [33] in class-shared concept space and class-specific concept space under 16-shot few-shot learning setting. For fair comparison, we use CLIP's original visual representation instead of the feature of visual attribute prompt for our approach.

B.2. Comparison with existing LBM in class-shared and class-specific concept space

In Section 4.3, we analyzed the reason for our relatively worse performance on the base classes in the Aircraft and Food101 datasets compared with existing LBMs is that existing LBMs learn in a category-shared concept space, where they exploit explainable spurious cues to achieve better performance. To further verify this, we compared our ALBM with the existing LBM approach LaBo [33] in both class-shared concept space (where the concept classifier identifies classes based on concepts from all classes) and class-specific concept space (where the concept classifier identifies classes based on concepts specific to them), as shown in Tab. A2. It is worth noting that, for a fair comparison, we do not use visual attribute prompts here but instead use CLIP's original visual representation. Additionally, CLBM is not applicable to the category-specific concept space because its concept set does not provide a mapping between concepts and categories. From Tab. A2, it is clear that, in general, ALBM outperforms existing LBM methods in both class-shared and class-specific settings, demonstrating that the concept set we generate with a unified attribute set better reflects the visual information of classes. Furthermore, the performance of both LaBo and ALBM in the category-specific concept space is weaker than in the category-shared concept space, which highlights the trade-off between interpretability and performance. This is due to the insufficient interpretability of features extracted by the CLIP model. Therefore, we further propose VAPL to extract features on each fine-grade attribute.

B.3. Few-shot performance comparison

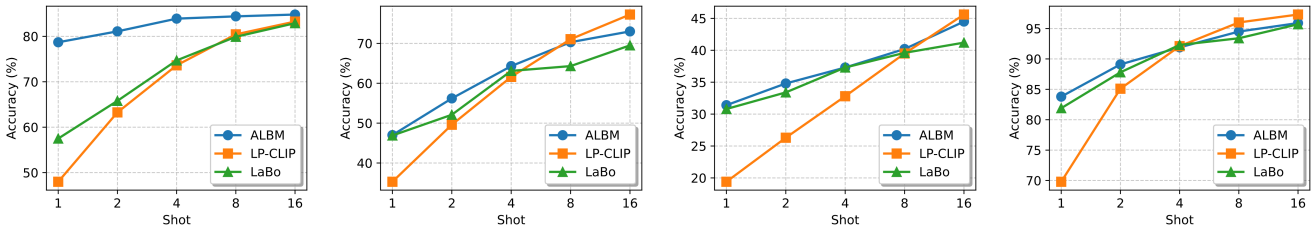
Yang et al. [33] found that compared to linear-probe CLIP, LBM achieves better few-shot performance by incorporating class concept information, which enhances the image recognition process. To evaluate the few-shot capability of our approach, we compare its performance with LaBo and LP-CLIP on Food101, CUB, Aircraft, and Flowers102 datasets, as shown in Fig. A2. It is clear that compared to LP-CLIP, ALBM demonstrates significant performance advantages, particularly when the number of training samples is extremely low. Additionally, it outperforms LaBo, further emphasizing its effectiveness. These results highlight that our collection strategy enhances few-shot learning by introducing more informative class concepts.

B.4. Interpretability verification via sparse prediction

To further verify the interpretability of ALBM, we evaluated the accuracy under diifferent NECs. Number of Effective Concepts (NEC) [28] is a newly proposed CBM interpretability metric that restricts the number of concepts with nonzero weights, which is motivated by the observation that when the number of concepts is very large, even those lacking interpretability can achieve high performance. Conversely, when concepts are sparse, only interpretable ones can provide sufficient information for recognition. Thus, by limiting NEC, different LBMs can be fairly compared in terms of interpretability and performance. As shown in Fig. A3, our approach achieves superior performances compared with LaBo and random concepts, further verifying the interpretability of ALBM.

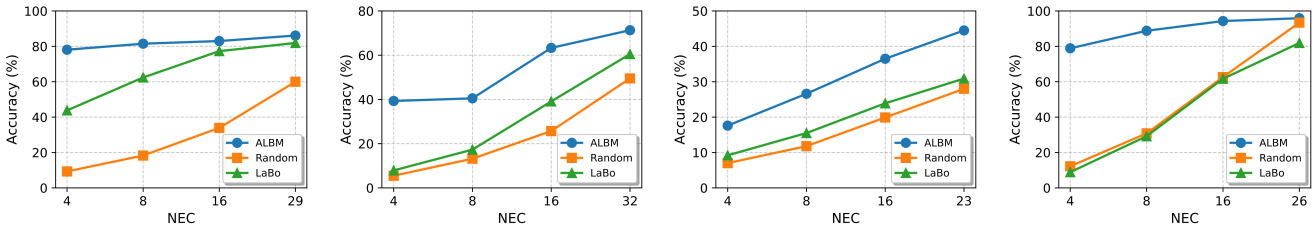
B.5. Relationship between interpretability and model size

In this subsection, we further analyze the relationship between interpretability and model size to provide guidance on model selection for the user of interpretable classification models. Therefore, we compare the zero-shot and base-to-novel classification performance of ALBM models using different versions of CLIP, as shown in Tab. A3 & A4. By comparing the first and second rows of Tab. A3 & A4, it can be observed that ViT-B/16 outperforms ViT-B/32 in all settings. This is due to that although these CLIP models have the same number of parameters, the smaller patches in the ViT-B/16 version preserve more local features, resulting in stronger interpretability. Furthermore, as shown in the third row of Tab. A3 & A4, CLIP with larger parameter size consistently outperforms its smaller versions, demonstrating a significant improvement in the ability to capture interpretable fine-grained attribute features. This aligns with the scaling law. Therefore, we recommend using larger models for interpretable image recognition whenever resources allow.



(a) Food101 (b) CUB (c) Aircraft (d) Flowers102

Figure A2. Few-shot performance comparison between our ALBM, LP-CLIP [25], and LaBo [33] on Food101, CUB, Aircraft, and Flowers102 datasets.



(a) Food101 (b) CUB (c) Aircraft (d) Flowers102

Figure A3. 16-shot performance comparison between our ALBM, LaBo [33], and randomly initialized concept bottleneck layer under different NECs. The experiments are conducted on Food101, CUB, Aircraft, and Flowers102 datasets.

CLIP Version	Aircraft	CUB	DTD	Flowers102	Food101	OxfordPets	CIFAR-10	CIFAR-100	ImageNet	Average
ViT-B/32 (88M)	12.5	16.9	38.9	30.3	56.4	28.5	66.6	30.6	51.0	38.1
ViT-B/16 (88M)	14.3	17.2	40.7	43.0	58.8	32.0	79.0	33.4	55.5	41.5
ViT-L/14 (304M)	**18.0**	**25.0**	**48.5**	**54.9**	**75.4**	**35.9**	**83.1**	**43.1**	**64.6**	**49.8**

Table A3. Zero-shot classification performance of ALBM with different CLIP versions, where the values in parentheses represent the parameter size of model.

CLIP Version	Aircraft		CUB		DTD		Flowers102		Food101		OxfordPets		CIFAR-10		CIFAR-100		ImageNet		Average	
	Base	Novel	Base	Novel	Base	Novel	Base	Novel	Base	Novel	Base	Novel	Base	Novel	Base	Novel	Base	Novel	Base	Novel
ViT-B/32	22.7	22.2	60.8	20.0	71.9	52.7	84.4	25.0	72.1	72.4	63.8	45.2	79.7	73.5	44.9	37.9	61.0	59.8	62.3	45.4
ViT-B/16	30.3	25.4	61.5	22.0	75.0	55.7	88.1	26.5	78.6	78.6	69.1	54.0	81.0	86.9	48.6	37.5	68.2	67.3	66.7	50.4
ViT-L/14	**38.7**	**33.0**	**91.9**	**27.8**	**78.6**	**60.5**	**91.7**	**32.4**	**88.5**	**86.8**	**79.2**	**61.1**	**90.8**	**93.6**	**59.3**	**55.1**	**75.0**	**73.9**	**77.0**	**58.3**

Table A4. Base-to-novel classification performance of ALBM with different CLIP versions.