Incremental Residual Concept Bottleneck Models

Supplementary Material

1. Further Discussion on CBMs

We discuss the geometric implications of the 3 challenges presented in this paper. As shown in Fig.1, by utilizing 2 sets of concepts, *Black* and *Furry*, we can categorize 4 classes of objects: *polar bear* (usually white and furry), *brown bear* (usually black and furry), *computer* (usually black and not furry) and *air conditioner* (usually white and not furry).

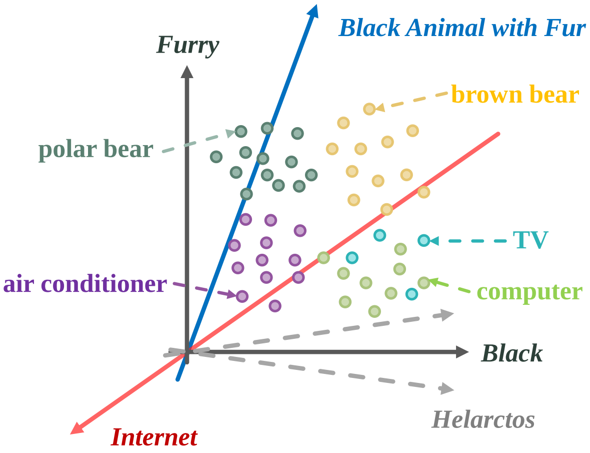

Figure 1. Further discussion on CBMs about 3 challenges.

- **Purity**: When concepts become overly complex and specific, their concept utilization efficiency is greatly reduced. For instance, the compound concept *Black Animal with Fur* can be entirely composed of the atomic concepts *Black* and *Furry*, without the need for introducing a new concept. In other words, the introduction of the new concept does not enrich the dimension of concept space, which remains the dimension of 2.

- **Precision**: When concepts are too difficult for the model to comprehend, the model may struggle to accurately determine the direction of concept vectors. For example, in the Fig.1, the 2 gray dashed lines represent a scenario where the model may be uncertain about the correct direction of the base vector of the concept *Helarctos*. This uncertainty can lead to variations in the length of the projection, resulting in a misunderstanding of the concept and leading to incorrect classification.

- **Completeness**: For the new category *TV* (also black and not furry), the existing concepts are unable to correctly differentiate between *TV* and *computer*. In such cases, it becomes necessary to introduce new concepts, such as *Internet* (computer has Internet, but TV does not), in order to increase the dimension of concept space and achieve correct classification.

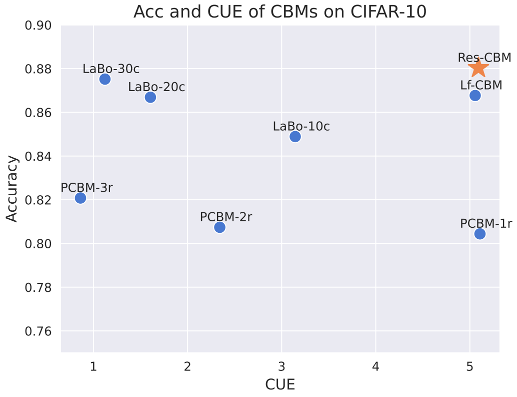

Figure 2. Accuracy and CUE of CBMs on CIFAR-10.

Our proposed method addresses the challenges of purity and precision by selecting atomic and generic concepts. Additionally, we tackle the challenge of completeness by employing the incremental concept discovery module to discover new concepts.

2. Concept Number and Performance

The concept utilization efficiency (CUE) is determined by both the length of each concept and the number of concepts. It reflects the challenges of purity and precision, as overly specific concepts may require a larger quantity of concepts, while complex compound concepts tend to have longer lengths per concept word. Completeness can be reflected by accuracy, where the results of CBMs closer to the performance of CLIP Linear Probing indicate that the concepts used are more complete. We visualize the accuracy and CUE of our method in Fig.2 and Fig.3 to highlight that our method better addresses the above 3 challenges.

A higher CUE and accuracy indicate better performance of the CBMs, which is represented by points in the upper-right corner of the figures.

3. Method Details

Concept Vector Initialization. The distribution of concept embeddings exhibits certain patterns. By using mean concept embedding as the initialization, we can introduce

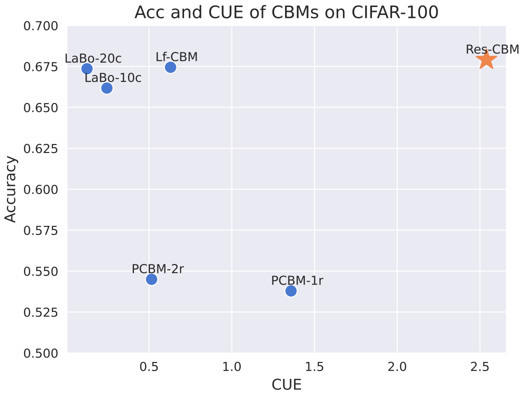

Figure 3. Accuracy and CUE of CBMs on CIFAR-100.

prior knowledge about the distribution of concept embeddings to the discovered concept vector, which accelerates the convergence process. Additionally, the reason for using the base concept bank is to avoid the discovered concept vector becoming too similar to some concept vector in the candidate concept bank so that a shortcut can be found to rapidly converge to a particular local optimum concept. The noise is also provided to increase randomness in the process.

Concept Similarity Loss. The concept similarity loss function guarantees that the discovered concept vector is similar to the concept embedding in the candidate concept bank. However, we find that although this approach restricts the meaning of the discovered concept vector, it can still be prone to converge to a particular concept and no longer exhibit any changes, leading to being trapped in a local optimum. Therefore, we choose to use the first M concepts for similarity constraint, and the resulting loss function adopts the average of these M concept similarity losses, thus converting the vector into a scalar value. It is noteworthy that which specific concept among these M concepts the model converges to will be determined by the optimization of the cross-entropy loss function.

Concept Utilization Efficiency. In addition to the quantity of concepts, we also pay attention to the average number of letters, based on the following empirical observation: compound concepts and high-level concepts tend to have a greater number of letters. Our intention is to obtain pure atomic concepts rather than complex compound concepts. Without considering the number of letters, a compound concept like *green leaf* would be encouraged due to its quantity being 1, rather than obtaining the separate concepts of *green* and *leaf* that we desire. Moreover, we expect to obtain concepts that can be precisely understood by CLIP, rather than complex concepts, which also enhances human interpretability. Frequently used words often have shorter lengths, such as *dog*, whereas less commonly used words, such as *canidae*, tend to have longer lengths. And when using word numbers, our results are more efficient, as shown in Tab.1.

Table 1. The results when using word number.

Methods	CIFAR-10	CIFAR-100
PCBM-1r	3.4823	0.9470
PCBM-2r	1.6682	0.3555
PCBM-3r	0.7758	N/A
Lf-CBM	2.6535	0.2860
LaBo-10c	1.9076	0.1183
LaBo-20c	0.9292	0.0600
LaBo-30c	0.6142	N/A
Res-CBM	**3.5072**	**1.6645**

Table 2. Base concept bank sizes and residual concept vector numbers of different datasets.

Dataset	Base Size	Residual Number
CIFAR-10	237	10
CIFAR-100	372	15
Tiny-ImageNet	634	30
CUB-200	177	30
Flower-102	186	20
Food-101	221	20
LAD-A	229	30
LAD-E	204	30
LAD-F	197	30
LAD-H	162	30
LAD-V	242	30

4. Experiment Details

To better replicate the results of our paper, we provide experimental details as follows. We used the Adam optimizer for all experiments. For the full data experiments, the batch size was set to 256. For the few shot tasks, the batch size was set to the number of classes.

For CLIP-based CBMs, the initial learning rate was set to 0.01, and it was decreased by a factor of 0.6 every 10 epochs. For Res-CBM and PCBM-h, another optimizer was used with an initial learning rate of 0.01, which was decreased by 0.6 every 10 epochs.

For the original independent CBMs, the initial learning rate of the concept extractor was set to 0.01, and it was decreased by a factor of 0.8 every 5 epochs. The initial learning rate of the concept classifier was set to 0.1 and decreased by a factor of 0.8 every 5 epochs.

For the incremental concept discovery module, the initial learning rate of the original concept classifier was set to 0.001, and it was decreased by a factor of 0.5 every 3 epochs. The initial learning rate of the residual concept classifier was set to 0.01, and it was decreased by a factor of 0.5 every 3 epochs.

The number of candidate concepts was set to 5, and the weight of the concept similarity loss function was set to 0.1. The number of residual vectors varied depending on the dataset, the specific numbers are presented in Tab.2.