Incremental Residual Concept Bottleneck Models

Chenming Shang[12*]   Shiji Zhou[1]   Hengyuan Zhang[1]   Xinzhe Ni[1]   Yujiu Yang[1†]   Yuwang Wang[12†]

[1]Tsinghua University   [2]Shanghai AI Laboratory

{scm22@mails.,zhoushiji@mail.,zhang-hy22@mails.}tsinghua.edu.cn

{nxz22@mails.,yang.yujiu@sz.,wang-yuwang@mail.}tsinghua.edu.cn

Abstract

*Concept Bottleneck Models (CBMs) map the black-box visual representations extracted by deep neural networks onto a set of interpretable concepts and use the concepts to make predictions, enhancing the transparency of the decision-making process. Multimodal pre-trained models can match visual representations with textual concept embeddings, allowing for obtaining the interpretable concept bottleneck without the expertise concept annotations. Recent research has focused on the concept bank establishment and the high-quality concept selection. However, it is challenging to construct a comprehensive concept bank through humans or large language models, which severely limits the performance of CBMs. In this work, we propose the Incremental **Res**idual **C**oncept **B**ottleneck **M**odel (Res-CBM) to address the challenge of concept completeness. Specifically, the residual concept bottleneck model employs a set of optimizable vectors to complete missing concepts, then the incremental concept discovery module converts the complemented vectors with unclear meanings into potential concepts in the candidate concept bank. Our approach can be applied to any user-defined concept bank, as a post-hoc processing method to enhance the performance of any CBMs. Furthermore, to measure the descriptive efficiency of CBMs, the **C**oncept **U**tilization **E**fficiency (CUE) metric is proposed. Experiments show that the Res-CBM outperforms the current state-of-the-art methods in terms of both accuracy and efficiency and achieves comparable performance to black-box models across multiple datasets.*

1. Introduction

Deep neural networks (DNNs) [31, 45] have achieved unprecedented success in a wide range of machine learning tasks, including computer vision [16, 32], natural language processing [14, 59], and speech recognition [34, 41].
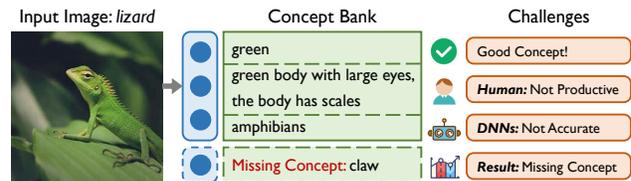

Figure 1. Challenges in CBMs. When a concept contains many atomic attributes, making it too complex for human to comprehend. When a concept is too high-level, DNNs may struggle to provide accurate prediction. Additionally, there is a risk of human-designed concept bank missing some important concepts.

However, due to their complex and deep structures, they are often regarded as black-box models [7, 49], which are too difficult to understand and interpret. In fields that demand high levels of trustworthiness, such as medicine [44], healthcare [21], education [60], and finance [39], model interpretability has become increasingly crucial. It indicates whether we can trust the DNNs' decisions, and how we can rectify the errors when the DNNs make mistakes. Making deep learning models more interpretable is a significant yet challenging research topic.

A promising approach for achieving interpretability in deep learning is through concept-based models [46], which leverage human-generated high-level concepts to explain the black-box features of DNNs. Among these approaches, Concept Bottleneck Models (CBMs) [26] map the visual representations to a set of concept values that are understandable to humans. These interpretable concepts are then used to make the final decision by a linear function, greatly enhancing our understanding of the decision-making process. Of greater significance, humans can repair models' faults by directly editing or intervening on the concept bottleneck [3, 48].

Due to the considerable cost of the fine-grained and precise annotation for each concept in CBMs, multimodal pre-trained model-based CBMs [58] have recently emerged as a research hotspot. CLIP [40] demonstrates the capacity to establish correspondences between visual information and

textual descriptions. Accordingly, we can utilize the CLIP text encoder to encode human-understood concepts into concept embeddings and use the visual encoder to encode images into visual representations. By projecting visual representations to each concept embedding, we can obtain the concept bottleneck. The recent innovations have primarily concentrated on comprehensive concept bank establishment and efficient high-quality concept selection. However, these methods face three critical challenges as shown in Fig.1:

- **Purity:** The essence of concepts is to abstract complex information into a combination of simple foundational elements, which can generalize to unseen data through the infinite combinations of finite concepts, offering an efficient way to describe the world. Long and complex descriptions like *small, black insect with six legs* or *long, thin tool with a wooden handle*, can be composed entirely of combinations of simple concepts such as *small*, *black*, *insect*, *leg*, etc., which may increase the productivity of concept utilization. Since these complicated concepts are excessively detailed, they may not be shared across different classes, which will limit their ability to capture shared knowledge between different categories.

- **Precision:** Each concept should be clearly understandable for both humans and multimodal pre-trained models when annotating concepts. Specifically, concepts should not be markedly more difficult than classes. For example, high-level concepts like *amphibians* or *perennial*, are relatively harder for humans to understand and exhibit weaker interpretability. Moreover, we cannot ascertain whether CLIP possesses the capability to understand these high-level concepts, which can potentially lead to errors in concept annotation.

- **Completeness:** Concept bank should strive to include as much relevant visual information that is beneficial for downstream classification tasks as possible. It is essential to avoid losing too much information in the process. However, it is challenging to establish a comprehensive concept bank directly [57]. The exhaustive capability of language descriptions is limited, making it difficult to accomplish a fully thorough concept bank through manual design and selection.

In this work, we propose the Incremental **Res**idual **C**oncept **B**ottleneck **M**odel (Res-CBM) to address the three challenges above. First, we establish a simple and pure base concept bank and candidate concept bank to ensure that the CLIP can accurately recognize each concept. Next, we initialize a set of optimizable vectors as filling concepts and optimize this set of filling concepts through the residual concept bottleneck model, to make up for the inadequacy in the base concept bank. Finally, the incremental concept discovery module is utilized to translate these complemented vectors with unclear meanings into potential concepts in the

candidate concept bank, improving the performance while preserving the interpretability of the CBM. Our main contributions can be summarized as follows:

1. We systematically reviewed the research on the CLIP-based CBMs, clarifying the importance of the concepts' purity, precision, and completeness.
2. We proposed the residual concept bottleneck model to make up for the insufficiency in the base concept bank and converted these complemented unknown vectors into potential concepts in the candidate concept bank through the incremental concept discovery module. It should be emphasized that our approach can be applied to any user-defined concept bank, as a post-hoc processing method to enhance the performance of any CBMs.
3. To measure the descriptive efficiency of CBMs, we introduced the metric of the concept utilization efficiency. We performed extensive experiments to validate our methodology and elaborate ablation studies to illustrate the effectiveness of each component.

2. Related Work

Deep neural networks (DNNs) can be explained through pixels [8, 47], samples [15, 22], weights [54], individual neurons [5, 13], subnetworks [2, 12], representations [4, 19, 20], etc [43]. Concept-based models [23, 33, 52, 53] are representation-level interpretable approaches, whose core idea is to map representations with unknown meanings extracted by the black-box DNNs to a set of concepts that can be comprehended by humans [9, 11, 35, 36]. Concept-based models can be broadly categorized into Concept Bottleneck Models (CBMs) [26], which describe concepts as independent values annotated by experts, and Concept Activation Vectors (CAVs) [24], which represent concepts as normal vectors of the decision boundaries that distinguish positive and negative samples of a concept. CBMs involve the datasets labeled with fine-grained and precise attributes by expert knowledge, such as CUB-200 [18], OAI [10], and LAD [61], whereas CAVs require learning the concept activation vectors on additional probe datasets, and the directions of the concept activation vectors are strictly correlated with the probe datasets [42]. These limitations severely restrict the development of concept-based models.

Recently, there has been a rapid development of multimodal pre-trained models [17], among which CLIP does an excellent job of relating visual information to text through contrastive learning, giving researchers confidence for unsupervised concept labeling with CLIP. PCBM [58] generates a concept bank via ConceptNet [50] and calculates the projection distance between the CLIP textual concept embeddings and the CLIP visual representations of images, analogous to the CAVs, to obtain the concept bottleneck. PCBM-h [58] introduces an uninterpretable residual linear layer to compensate for incomplete concept extraction by
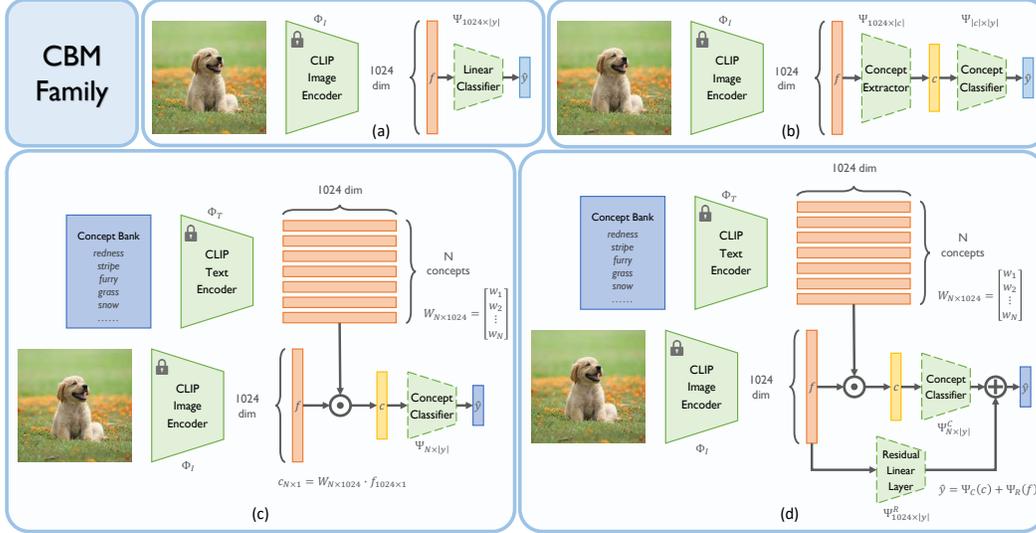
Figure 2. Different CBMs' structures. (a): CLIP linear probing [40]. (b): Original Concept Bottleneck Model (CBM) [26]. (c) Post-hoc Concept Bottleneck Model (PCBM) [38, 56, 58]. (d): Hybrid Post-hoc Concept Bottleneck Model (PCBM-h) [58].

fitting the difference between the PCBM's result and the ground truth through the original visual representations of CLIP. Label-free CBM [38] only leverages CLIP to concept annotation, maximizing the similarity between the concept bottleneck of the features extracted by arbitrary visual backbone and the concept annotations labeled by CLIP. LaBo [56] generates a set of candidate concepts through large language models and designs a submodular optimization block to select the concepts.

3. Method

3.1. Problem Formulation

Consider a dataset of image-label pairs $\mathcal{D} = \{(x, y)\}$, where $x \in \mathcal{X}$ is the image and $y \in \mathcal{Y}$ is the label. We have N human-selected concepts to describe the essential information of these images, which can be denoted as discrete tokens $\mathcal{E} = \{e_1, e_2, ..., e_N\}$. Multimodal pre-trained alignment model (e.g., CLIP [40]) has an image encoder $\Phi_I : \mathcal{X} \to \mathbb{R}^d$ and a text encoder Φ_T, which can map images and text into a shared d-dimensional feature space respectively. We encode the discrete tokens with the CLIP text encoder then perform L_2 normalization[1] to obtain the concept embeddings $\{w_1, w_2, ..., w_N | w_i = \Phi_T(e_i), i = 1, 2, ..., N\}$ with the length of 1 and the dimension of d. We concatenate these concept embeddings to a concept projection matrix $W_{N \times d} : \mathbb{R}^d \to \mathbb{R}^N$ in arbitrary order, also identified as a concept bank.

Correspondingly, we utilize the CLIP image encoder to get the visual representations $f = \Phi_I(x)$. Considering that

CLIP has aligned the images with the textual data when pre-training, the visual representations share the feature space with any concept embedding, and the projection length $\|f\|_2 \cdot \cos\langle w_i, f \rangle$ can reflect the presence of a particular concept in the image. Since undergoing L_2 normalization on w_i, the concept probability can be directly formulated in terms of the dot product $c = W \cdot f$.

Fig.2 illustrates the structure of common CBMs. For **CLIP Linear Probing** [29], an uninterpretable linear classifier $\Psi : \mathbb{R}^d \to \mathcal{Y}$ is trained directly from visual representations:

$$\hat{y} = \Psi(f) = \Psi(\Phi_I(x)) \quad (1)$$

Original CBM [26] first extracts concepts using a concept extractor $\Psi_{\text{cpt}} : \mathbb{R}^d \to \mathbb{R}^N$ and then linearly unites the extracted concepts for label prediction by concept classifier $\Psi_{\text{cls}} : \mathbb{R}^N \to \mathcal{Y}$.

$$\hat{y} = \Psi_{\text{cls}}(\Psi_{\text{cpt}}(f)) \quad (2)$$

Post-hoc CBM [58] stands for a category of CLIP-based CBMs, including Label-free CBM [38], LaBo [56], etc. They attain concept bottleneck by projecting visual representations f directly to a concept bank W, rather than by learning a concept extractor. After getting concepts, only a linear concept classifier $\Psi_c : \mathbb{R}^N \to \mathcal{Y}$ needs to be learned.

$$\hat{y} = \Psi_c(c) = \Psi_c(W \cdot f) \quad (3)$$

PCBM-h [58] adds a linear layer $\Psi_r : \mathbb{R}^d \to \mathcal{Y}$ between the original visual representation and the labels on the basis of PCBM, which is used to predict the difference between the PCBM result and the ground truth.

$$\hat{y} = \Psi_c(c) + \Psi_r(f) = \Psi_c(W \cdot f) + \Psi_r(f) \quad (4)$$

---

[1]The symbol of L_2 normalization is omitted for writing convenience.

3.2. Concept Bank Establishment

Candidate Concept Bank. To ensure precision, we need to make sure that CLIP is capable of recognizing each concept, and one straightforward way is to construct the concept bank from the CLIP pre-trained dataset. For purity, it is desirable for each concept to be at the atom level rather than the compound level, thus maintaining the clarity and specificity. We find that the scene graph [27] consisting of visual concepts to be an appropriate foundation, in which an atom is defined as a singular visual concept, corresponding to a single scene graph node. Atoms are subtyped into *objects*, *relationships*, and *attributes*. We pick the nouns and adjectives in them as the candidate concept bank $\mathcal{E}^*$.

Base Concept Bank. Although our method can recover missing concepts from a diverse base concept bank, a high-quality base concept bank can remarkably improve the efficiency of our approach. A base concept bank is expected to consist of two parts. Firstly, it should include general concepts that are applicable across various scenarios. We select the pre-defined concept library proposed by [24], which has been widely used in various tasks [1, 55], as our general concept bank $\mathcal{E}_{\text{ge}}$. Secondly, it is supposed to include concepts relevant to the specific classification task. Following the PCBM [58], we collect all concepts from ConceptNet [50] that have relations of *hasA*, *isA*, *partOf*, *HasProperty*, and *MadeOf* with the classes in each classification task to build the associated concept bank $\mathcal{E}_{\text{as}}$. Both concept banks should be included in the candidate concept bank, from which we can derive the base concept bank:

$$\mathcal{E}_0 = (\mathcal{E}_{\text{ge}} \cap \mathcal{E}^*) \cup (\mathcal{E}_{\text{as}} \cap \mathcal{E}^*) \quad (5)$$

We encode the discrete concept tokens with the CLIP text encoder then perform L_2 normalization to obtain the base and candidate concept embedding matrices:

$$W_0 = \Phi_T(\mathcal{E}_0), \quad W^* = \Phi_T(\mathcal{E}^*) \quad (6)$$

3.3. Residual Concept Bottleneck Model

Previous methods have typically involved pre-generating a set of relevant concepts and then deleting some of them to get the final concept bank. However, on the one hand, the pre-generated concepts might be insufficient, as their scale has often limitations. On the other hand, manual filtering without leveraging dataset information may result in the omission of important concepts. Therefore, we propose the residual concept bottleneck model to learn the missing concepts from the dataset content in an incremental approach compared to other decremental methods as shown in Fig.3.

We first randomly initialize D optimizable vectors $U_{D \times d} = \{u_1, u_2, ..., u_D\}$ with dimensions consistent with those of the existing concept embeddings. We likewise
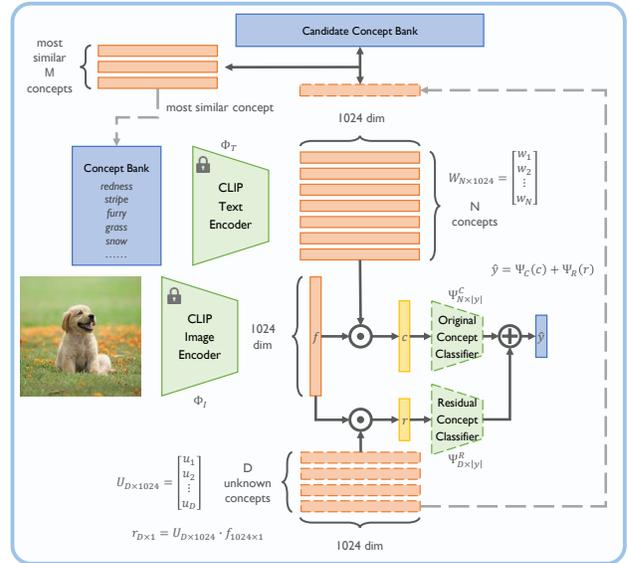

Figure 3. Incremental Residual Concept Bottleneck Model.

project the visual representations onto these vectors, resulting in residual concepts $r = U \cdot f$ with unclear meanings. The prediction result of using the residual concepts to complement the original concepts is as follows:

$$\hat{y} = \Psi_c(c) + \Psi_r(r) = \Psi_c(W \cdot f) + \Psi_r(U \cdot f) \quad (7)$$

where $\Psi_c : \mathbb{R}^N \to \mathcal{Y}$ is the original concept linear classifier using existing concepts and $\Psi_r : \mathbb{R}^D \to \mathcal{Y}$ is the residual concept linear classifier using unknown residual concepts.

For one thing, we intend to maximize the accuracy when using only the original concepts, which can be formulated as the following optimization problem:

$$\min_{\Psi_c} \mathop{\mathbb{E}}_{(x,y)\in\mathcal{D}} \left[\mathcal{L}\big(\Psi_c(W \cdot \Phi_I(x)), y\big) + \lambda \cdot \Omega(\Psi_c) \right] \quad (8)$$

where $\mathcal{L}(\hat{y}, y)$ is the cross-entropy loss function, $\Omega(\Psi_c)$ is a complexity measure to regularize the model, and λ is the regularization strength.

From another perspective, we aim to improve the performance after incorporating the residual concepts, which can be solved by the following optimization problem:

$$\min_{\Psi_r, U} \mathop{\mathbb{E}}_{(x,y)\in\mathcal{D}} \left[\mathcal{L}\big(\Psi_c(W \cdot f) + \Psi_r(U \cdot f), y\big) \right] + \lambda \cdot \Omega(\Psi_r) \quad (9)$$

where $f = \Phi_I(x)$.

To simultaneously address these two problems, we perform two forward and backward propagations in each batch to update the parameters. When performing the first propagation, we utilize only the original concepts to optimize the original concept classifier Ψ_c. During the second propagation, we fix the parameters of the original concept classifier

Ψ_c and optimize the residual concept classifier Ψ_r as well as the learnable unknown concept vectors U using both the original and the residual concepts.

To avoid the influence of extreme values from unknown residual concepts, we employ cosine similarity instead of projection distance as the concept bottleneck and standardize the activations of each concept to have a mean of 0 and a standard deviation of 1 to speed up convergence [38]. These operations are reversible and do not affect the relative order of concepts or introduce additional information.

3.4. Incremental Concept Discovery

Through the aforementioned steps, the residual concepts effectively compensate for the incompleteness of the original concepts. However, these randomly learned concepts lack human interpretability, so we next translate them into human-understandable concepts.

Parameters Initialization. We first initialize another optimizable vector v_d, denoted as the discovered concept vector, whose value is the mean concept embeddings of the base concept bank with added noise. Subsequently, we append the discovered concept vector to the base concept bank and remove one of the learned unknown residual concept vectors, whose residual concept embedding matrix changes to $U_{-v_d} = U_{(D-1) \times d}$ and residual classifier changes to $\Psi_{r-v_d} : \mathbb{R}^{D-1} \to \mathcal{Y}$. After performing L_2 normalization and projection, the visual representations will also obtain a similarity score $c_d = v_d \cdot f$ to the discovered concept vector, and there will be a corresponding discovered concept linear classifier $\Phi_d : \mathbb{R} \to \mathcal{Y}$. We adopt the language priors as the initial weights to ensure stable updating of the discovered concept vector. In particular, the weights of the discovered concept classifier are initialized as the cosine similarity between the discovered concept embedding and the text embeddings for each class name, while other parameters and weights are kept untouched.

Concept Similarity Loss Function. After adding the discovered concept vector to the base concept bank, the prediction under the original and residual strategy is as follows:

$$\hat{y}_{\text{org}} = \Psi_c(c) + \Psi_d(c_d) = \Psi_c(W \cdot f) + \Psi_d(v_d \cdot f) \quad (10)$$
$$\hat{y}_{\text{res}} = \hat{y}_{\text{org}} + \Psi_{r-v_d}(U_{-v_d} \cdot f) \quad (11)$$

To further guarantee that the discovered concept vector is interpretable, we propose the concept similarity loss function $\mathcal{L}_{\text{sim}}$, which utilizes the candidate concept bank W^* to constrain the discovered concept embedding:

$$\mathcal{L}_{\text{sim}} = 1 - \mathbb{E}\left[\text{Rank}\left(W^* \cdot \frac{v_d^T}{\|v_d\|_2}, M \right) \right] \quad (12)$$

where $\text{Rank}(v, M)$ denotes sorting v from largest to smallest and selecting the first M of them.

Referring to the update strategy of the residual concept bottleneck model, we just add the discovered concept with its classifier during the first pass propagation:

$$\min_{\Psi_c, \Psi_d, v_d} \mathbb{E}\left[\mathcal{L}(\hat{y}_{\text{org}}, y) \right] + \alpha \cdot \mathcal{L}_{\text{sim}} + \lambda \cdot \Omega(\Psi_c, \Psi_d) \quad (13)$$

where α is the weight of the concept similarity loss.

When performing the second propagation, we fix the parameters of the original concept classifier Ψ_c, the discovered concept classifier Ψ_d, and the discovered concept embedding v_d, optimizing the residual concept classifier Ψ_{r-v_d} and the unknown residual concept vectors U_{-v_d}:

$$\min_{\Psi_{r-v_d}, U_{-v_d}} \mathbb{E}\left[\mathcal{L}(\hat{y}_{\text{res}}, y) \right] + \lambda \cdot \Omega(\Psi_{r-v_d}) \quad (14)$$

We recover concepts in an incremental way until all the learned unknown residual concept vectors are transformed into pure, accurate, and human-interpretable concepts. At this point, the base concept library is fully updated, and experiments confirm its superior completeness.

4. Experiments

4.1. Experimental Setup

Datasets. We conduct comprehensive experiments on 7 datasets, including: CIFAR-10, CIFAR-100 [28], Tiny-ImageNet [30], LAD [61], CUB-200 [51], Food-101 [6], Flower-102 [37]. For a fair comparison, we apply CLIP-RN50 when comparing with other CBMs and CLIP-ViT-L/14 on few-shot classification tasks as the backbone, and all linear classifiers are optimized with the Adam [25] optimizer. For more detailed information, please refer to the supplementary material.

Baselines. To evaluate our proposed approach, we compare it with the following baseline methods:

- CLIP Zero-shot[40]: the cosine similarities between the visual representations and the textual embeddings of each class name are calculated directly without any training. The class with the highest cosine similarity score is selected as the final classification result.
- Linear Probing (LP)[40]: We exclusively utilize the CLIP visual encoder and the obtained visual representations are then used to predict the label by training a linear classifier.
- Original CBM[26]: The original CBM relies on fine-grained concept annotations, so we compare our method with it only on LAD and CUB with available attribute annotations. In addition, CBMs may suffer from concept leakage, which can only be alleviated by CBMs trained independently [36]. Hence, we choose CBM-independent as the baseline for comparison.
- PCBM[58]: PCBM uses the ConceptNet[50], a knowledge graph dataset, to conduct a concept bank. It treats classes as nodes and incorporates surrounding neighboring nodes in the concept bank. More concepts can be included by expanding multiple layers outwards.

Table 1. Comparison with other CLIP-based CBMs. Bold indicates the best interpretable result, underline indicates the 2nd-best result, italic indicates the black-box methods.

Method	Interpretability	Length	CIFAR 10			CIFAR 100		
			Accuracy	Number	CUE	Accuracy	Number	CUE
Zero-shot	Yes	N/A	0.6796	N/A	N/A	0.2620	N/A	N/A
LP	No	N/A	0.8858	N/A	N/A	0.6995	N/A	N/A
PCBM-1r	Yes	9	0.8044	175	**5.1073**	0.5379	440	1.3583
PCBM-2r	Yes	9	0.8074	383	2.3423	0.5450	1175	0.5153
PCBM-3r	Yes	9	0.8208	1058	0.8620	N/A	N/A	N/A
PCBM-h	No	9	0.8757	175	N/A	0.6683	440	N/A
Lf-CBM	Yes	12	0.8677	143	5.0565	<u>0.6745</u>	892	<u>0.6301</u>
LaBo-10c	Yes	27	0.8489	100	3.1441	0.6618	1000	0.2451
LaBo-20c	Yes	27	0.8669	200	1.6054	0.6736	2000	0.1247
LaBo-30c	Yes	26	<u>0.8752</u>	300	1.1221	N/A	N/A	N/A
Res-CBM	Yes	7	**0.8803**	237+10	<u>5.0914</u>	**0.6791**	372+15	**2.5396**

Table 2. Comparison with CBMs with labeled concept annotations. The labeling significance is consistent with Tab.1.

Method	LAD-A	LAD-E	LAD-F	LAD-H	LAD-V	CUB	**Mean**
LP	0.9432	0.8661	0.7393	0.4924	0.8671	0.7214	0.7714
Original CBM	<u>0.8964</u>	<u>0.7826</u>	0.5535	<u>0.3388</u>	<u>0.7907</u>	**0.6513**	<u>0.6689</u>
PCBM-annotation	0.8934	0.7711	<u>0.5982</u>	0.2982	0.7814	0.6363	0.6631
Res-CBM-before	0.7897	0.7709	0.4356	0.1429	0.7548	0.5987	0.5812
Res-CBM-after	**0.8979**	**0.8045**	**0.6624**	**0.3862**	**0.8301**	<u>0.6243</u>	**0.7009**

- PCBM-h[58]: After training the PCBM, PCBM-h is applied as a post-processing method that connects the visual representation and the class labels result through short-cuts. It utilizes a linear classifier to predict the difference between the result of PCBM and the ground truth.
- Lf-CBM[38]: In contrast to PCBM, Lf-CBM generates concepts through a large language model (LLM). It solely utilizes CLIP as the concept annotator, allowing it to act on any visual backbone. We directly select CLIP as the visual backbone, thus omitting the process of computing the similarity matrix.
- LaBo[56]: LaBo similarly applies the LLM to generate candidate concepts. However, the quantity of concepts in LaBo is specific to each class. So we generate 10, 20, and 30 concepts per class to compare the performance. In addition, LaBo's linear classifier differs from the above baselines. To evaluate the quality of concepts equitably, we choose the same classifier as the above methods.

Metrics. Besides accuracy, we also concentrate on concepts' efficiency, including both the average length of each concept and quantity of concepts. Specifically, we define the accuracy improvement brought by all concepts' length to measure the **C**oncept **U**tilization **E**fficiency (CUE):

$$\text{CUE} = \frac{10000 \times \text{Acc}}{N \times \bar{L}} \quad (15)$$

where Acc is the classification accuracy, N is the quantity of concepts and $\bar{L}$ is the average number of letters included in each concept. Larger CUE means higher efficiency.

4.2. Accuracy and Efficiency

We compare our method with other CBMs in terms of accuracy and efficiency. Tab.1 provides the results of our method compared to other CLIP-based CBMs.[2] Res-CBM attains higher accuracy than PCBM by 5.95% on CIFAR-10 and 13.41% on CIFAR-100 and even outperforms the inexplicable PCBM-h. Despite Lf-CBM and LaBo being powerful baselines, our methodology still achieves state-of-the-art results. Particularly, it should be noted that our method considerably exceeds both Lf-CBM and LaBo in terms of efficiency, as evidenced by the CUE metric.

On LAD and CUB-200 with attribute annotations, we still use CLIP-50 with frozen parameters as the visual backbone to train Original CBM according to formula (2), as the results are shown in Tab.2.[3] The close performance of the Original CBM and PCBM-annotation suggests that the CBMs' performance heavily relies on the expert-selected concepts more than the model structure. Although Res-CBM-before performs poorly, a remarkable improvement is observed in Res-CBM-after after applying our unsupervised incremental concept discovery, which even surpasses supervised CBMs with concept annotations on most datasets.

---

[2]PCBM-#r represents expanding outward to include neighboring nodes # times. LaBo-#c represents selecting # concepts per class.

[3]PCBM-annotation refers to PCBM that utilizes the textual embeddings of the annotated attributes as the concept bank. Res-CBM-before indicates the performance on the initial base concept bank. Res-CBM-after represents the performance after incremental concept discovery.
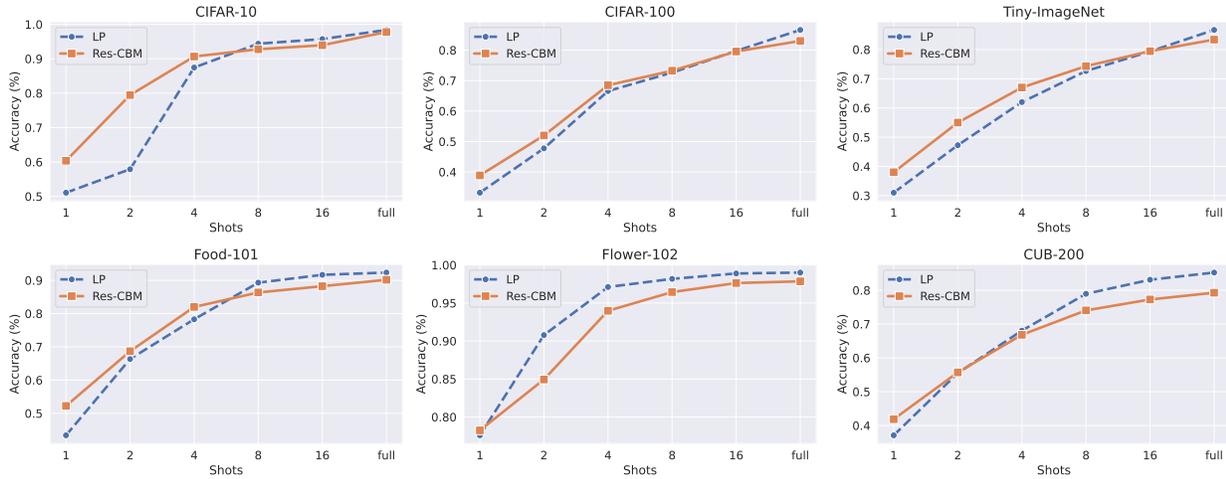
Figure 4. Test accuracy (%) comparison between Res-CBM and LP on 6 datasets. The x-axis represents the number of labeled images.
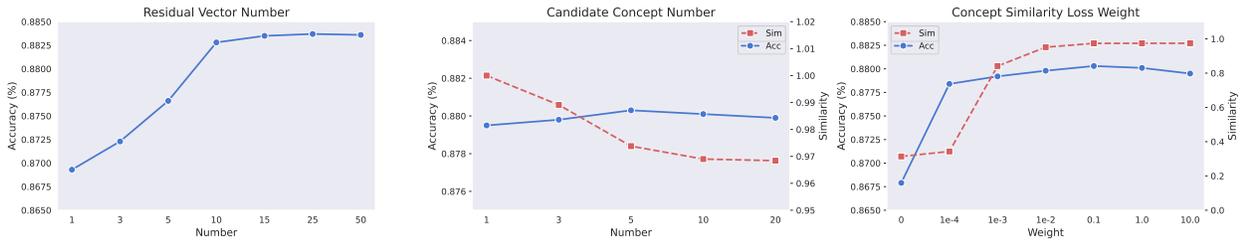

Figure 5. Ablation results on residual vector numbers, concept similarity loss weights and candidate concept numbers.

4.3. Interpretability

In this subsection, we visualize the mechanism of Res-CBM and demonstrate how the newly discovered concepts assist in advancing classification performance. For each image, we display the top 5 concepts with the highest activation scores in the concept bottleneck, as shown in Fig.6. Through a sparse concept classifier, we can obtain predictions, which signifies predicting the class by a linear combination of concepts.

However, due to the incompleteness of the base concept bank, it is challenging to achieve accurate classifications by solely relying on these concepts in several images, e.g., *dragonfly* and *honeybee*, *peach* and *hawthorn*, both of which possess concepts from the base concept bank. With the help of Res-CBM, we have discovered 2 concepts, *nectar* and *rough*, from the candidate concept bank. Unlike dragonflies, honeybees often gather nectar from the center of flowers, thus associating them with the concept of nectar. Moreover, hawthorns have a rougher surface compared to peaches. By incorporating these discovered concepts into the base concept bank, these initially misclassified samples are now correctly categorized, achieving automatic model debugging to a certain extent.

4.4. Few-shot Capability

Fig.4 illustrates the performance comparison under different data volume settings between Res-CBM and LP. Compared to the end-to-end black-box model, our method achieves superior performance when little data is available, and exhibits a slight performance gap with larger amounts of sample sizes, which indicates that our method has maintained the accuracy without sacrificing interpretability.

For common object datasets, our approach demonstrates excellent performance with 1, 2, 4, and 8 labeled images per category. We credit its success to the incorporation of human concept information, which helps the model extract vital aspects relevant to the categories from the representations of the black-box backbone.

For fine-grained object datasets, our method outperforms the black-box model when the data volume is extremely low. As the data volume increases, the unexplainable black-box model's performance becomes more prominent, especially in expert datasets such as Flower-102 and CUB-200. It might be attributed to common concepts struggling to distinguish the subtle differences between these fine-grained objects. A feasible solution is to enhance the specificity of concepts by a tailored specialized concept candidate bank.
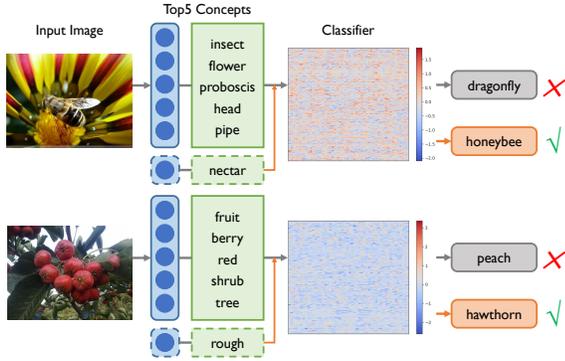
Figure 6. Concepts applied in Res-CBM on LAD-A (top) and LAD-F (bottom). Solid boxes represent concepts from the base concept bank, dashed boxes indicate the discovered concepts. Heatmaps refer to learned linear concept classifiers, in which the x-axis represents classes, the y-axis represents concepts, and the values represent the contribution of concepts to categories.

4.5. Ablation Study

Residual Vector Number. To investigate the impact of residual vector numbers on performance improvement and demonstrate that our method can be applied as a post-processing approach to any CBMs, we record the results of attaching different residual vector numbers for Lf-CBM on CIFAR-10, as shown in Fig.5-left. As the quantity of vectors increases, the performance of Lf-CBM continuously improves and gradually saturates at around 15 vectors.

Candidate Concept Number. We fix the residual vector number at 10 and set the concept similarity loss weight to 0.1, then record the final accuracy and the average similarity between the discovered concepts and the most similar concepts in the candidate concept bank during the discovery process in Fig.5-middle. When the candidate concept number is small, the model tends to converge to this unique concept, making it difficult to escape local optima. When it is large, the optimizing vectors easily oscillate between different concepts, resulting in lower accuracy and similarity. Overall, setting the candidate concept number to 5 seems to be a good hyperparameter choice.

Concept Similarity Loss Weight. With a fixed candidate concept number at 5, we also record the final accuracy and average similarity for different loss weights as shown in Fig.5-right. When the weight is low, the similarity of the discovered concepts is extremely low, which resembles a random selection strategy, leading to poorer performance. When the weight is large, the model tends to converge to a local optimum only satisfing the concept similarity loss, sacrificing the ability to explore more appropriate concepts. Since Res-CBM has already saturated after the completion of unknown concept vectors, the cross-entropy loss value is small. Therefore, a weight of 0.1 realizes optimal results.

5. Discussion and Conclusion

Rethinking CBMs. Indeed CBMs are valuable attempts towards transparency of the decision-making process, one important and fundamental problem may be still unclear, which is: how many concepts are good? especially when lacking a well-accepted and rigorous definition of concepts. In this paper, we have tried to explore this problem. Firstly, the number of concepts should not be the larger the better. From the optimization perspective, the operation of CBMs to project visual representations into concept space is a linear transformation. If its rank is greater than or equal to the dimension d of the concept vector, this process is invertible. Therefore, it is straightforward to re-estimate the inverse transformation matrix to preserve classification accuracy in subsequent linear classifiers. The number of concepts larger than d may decrease the efficiency, making the interpretability less effective.

On the other hand, the concept should be an efficient way to express categories. Humans can describe objects in the world exponentially using a finite set of concepts. Theoretically, from informatics, when each concept weight is binary, the concept quantity needed to distinguish n categories will be enough approximately $\log_2 n$. This is the reason for the poor CUE of LaBo [56], which finds concepts for each category in a way that the CUE complexity is $\mathcal{O}(n)$ rather than $\mathcal{O}(\log_2 n)$. This suggests that the CUE is crucial in CBMs, and a reasonable number of concepts should lie in the interval of $[\lceil \log_2 n \rceil, d)$.

Conclusion. In this paper, we emphasize the significance of purity, precision, and completeness in CBMs and introduce the CUE metric to evaluate the descriptive efficiency of CBMs. Our proposed Res-CBM complements the missing concepts in any CBM with an incremental concept discovery approach, which achieves optimal accuracy with the best concept utilization efficiency.

Limitation and Future Work. For our proposed Res-CBM, the candidate concept bank construction for fine-grained datasets remains a challenge. Additionally, the sequential concept discovery approach increases computational time costs, while more efficient concept similarity calculation and parallel concept discovery techniques are needed to develop. For the CBM structure, existing CBMs represent all concepts into a single bottleneck, disregarding the hierarchical nature of concepts, where simple concepts can be combined to form more complicated ones.

Acknowledgements

References

[1] Abubakar Abid, Mert Yuksekgonul, and James Zou. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *International Conference on Machine Learning*, pages 66–88. PMLR, 2022. 4

[2] Mohammed Amer and Tomás Maul. A review of modularization techniques in artificial neural networks. *Artificial Intelligence Review*, 52:527–561, 2019. 2

[3] Mohammad Taha Bahadori and David E Heckerman. Debiasing concept-based explanations with causal analysis. *arXiv preprint arXiv:2007.11500*, 2020. 1

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 2

[5] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2: 129–146, 2020. 2

[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014. 5

[7] Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016. 1

[8] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. 2

[9] Jonathan Crabbé and Mihaela van der Schaar. Concept activation regions: A generalized framework for concept-based explanations. *Advances in Neural Information Processing Systems*, 35:2590–2607, 2022. 2

[10] Felix Eckstein, Wolfgang Wirth, and Michael C Nevitt. Recent advances in osteoarthritis imaging—the osteoarthritis initiative. *Nature Reviews Rheumatology*, 8(10):622–630, 2012. 2

[11] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, et al. Concept embedding models: Beyond the accuracy-explainability trade-off. *Advances in Neural Information Processing Systems*, 35:21400–21413, 2022. 2

[12] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 2

[13] Amirata Ghorbani and James Y Zou. Neuron shapley: Discovering the responsible neurons. *Advances in neural information processing systems*, 33:5922–5932, 2020. 2

[14] Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers, 2024. 1

[15] Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: A survey. *arXiv preprint arXiv:2212.04612*, 2022. 2

[16] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022. 1

[17] Xue Han, Yi-Tong Wang, Jun-Lan Feng, Chao Deng, Zhan-Heng Chen, Yu-An Huang, Hui Su, Lun Hu, and Peng-Wei Hu. A survey of transformer-based multimodal pre-trained modals. *Neurocomputing*, 515:89–106, 2023. 2

[18] Xiangteng He and Yuxin Peng. Fine-grained visual-textual representation learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(2):520–531, 2019. 2

[19] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 3–19. Springer, 2016. 2

[20] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 264–279, 2018. 2

[21] Junjia Huang, Chenming Shang, Aolin Xiong, Yuxian Pang, and Zhi Jin. Seeing health with eyes: Feature combination for image-based human bmi estimation. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021. 1

[22] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. *arXiv preprint arXiv:2202.00622*, 2022. 2

[23] Dmitry Kazhdan, Botty Dimanov, Mateja Jamnik, Pietro Liò, and Adrian Weller. Now you see me (cme): concept-based model extraction. *arXiv preprint arXiv:2010.13233*, 2020. 2

[24] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 2, 4

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[26] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 1, 2, 3, 5

[27] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 4

[28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[29] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022. 3

[30] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1

[32] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 2021. 1

[33] Max Losch, Mario Fritz, and Bernt Schiele. Interpretability beyond classification output: Semantic bottleneck networks. *arXiv preprint arXiv:1907.10882*, 2019. 2

[34] Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom. Automatic speech recognition: a survey. *Multimedia Tools and Applications*, 80: 9411–9457, 2021. 1

[35] Emanuele Marconato, Andrea Passerini, and Stefano Teso. Glancenets: Interpretable, leak-proof concept-based models. *Advances in Neural Information Processing Systems*, 35:21212–21227, 2022. 2

[36] Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended? *arXiv preprint arXiv:2105.04289*, 2021. 2, 5

[37] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 5

[38] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. *arXiv preprint arXiv:2304.06129*, 2023. 3, 5, 6

[39] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications: A survey. *Applied Soft Computing*, 93:106384, 2020. 1

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 3, 5

[41] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023. 1

[42] Vikram V Ramaswamy, Sunnie SY Kim, Ruth Fong, and Olga Russakovsky. Overlooked factors in concept-based explanations: Dataset choice, concept salience, and human capability. *arXiv preprint arXiv:2207.09615*, 2022. 2

[43] Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 464–483. IEEE, 2023. 2

[44] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps: Automation of Decision Making*, pages 323–350, 2018. 1

[45] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3): 247–278, 2021. 1

[46] Gesina Schwalbe. Concept embedding analysis: A review. *arXiv preprint arXiv:2203.13909*, 2022. 1

[47] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2

[48] Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A closer look at the intervention procedure of concept bottleneck models. *arXiv preprint arXiv:2302.14260*, 2023. 1

[49] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017. 1

[50] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, 2017. 2, 4, 5

[51] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5

[52] Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Learning bottleneck concepts in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10962–10971, 2023. 2

[53] Dan Wang, Xinrui Cui, and Z Jane Wang. Chain: Concept-harmonized hierarchical inference interpretation of deep convolutional neural networks. *arXiv preprint arXiv:2002.01660*, 2020. 2

[54] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33:15173–15184, 2020. 2

[55] Shirley Wu, Mert Yuksekgonul, Linjun Zhang, and James Zou. Discover and cure: Concept-aware mitigation of spurious correlation. *arXiv preprint arXiv:2305.00650*, 2023. 4

[56] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197, 2023. 3, 6, 8

[57] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks.

Advances in neural information processing systems, 33: 20554–20565, 2020. 2

[58] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022. 1, 2, 3, 4, 5, 6

[59] Hengyuan Zhang, Dawei Li, Yanran Li, Chenming Shang, Chufan Shi, and Yong Jiang. Assisting language learners: Automated trans-lingual definition generation via contrastive prompt learning. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 260–274, 2023. 1

[60] Hengyuan Zhang, Zitao Liu, Chenming Shang, Dawei Li, and Yong Jiang. A question-centric multi-experts contrastive learning framework for improving the accuracy and interpretability of deep sequential knowledge tracing models, 2024. 1

[61] 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, 5