

# Hybrid Concept Bottleneck Models

Yang Liu, Tianwei Zhang, Shi Gu

University of Electronic Science and Technology of China  
 Shenzhen Institute for Advanced Study, UESTC

liu.yang.mine@std.uestc.edu.cn, gsbyztw@gmail.com, gus@uestc.edu.cn

## Abstract

Concept Bottleneck Models (CBMs) provide an interpretable framework for neural networks by mapping visual features to predefined, human-understandable concepts. However, the application of CBMs is often constrained by insufficient concept annotations. Recently, multi-modal pre-trained models have shown promise in reducing annotation costs by aligning visual representations with textual concept embeddings. Nevertheless, the quality and completeness of the predefined concepts significantly affect the performance of CBMs. In this work, we propose **Hybrid Concept Bottleneck Model (HybridCBM)**, a novel CBM framework to address the challenge of incomplete predefined concepts. Our method consists of two main components: a Static Concept Bank and a Dynamic Concept Bank. The Static Concept Bank directly leverages large language models (LLMs) for concept construction, while the Dynamic Concept Bank employs learnable vectors to capture complementary and valuable concepts continuously during training. After training, a pre-trained translator converts these vectors into human-understandable concepts, further enhancing model interpretability. HybridCBM is highly flexible and can be easily integrated with existing CBMs to improve both interpretability and performance. Experimental results<sup>1</sup> on multiple datasets demonstrate that HybridCBM outperforms current state-of-the-art CBMs and achieves comparable results to black-box models. Additionally, we propose novel metrics to assess the quality of learned concepts, showing that they perform comparably to predefined concepts.

## 1. Introduction

Deep Neural Networks (DNNs) have become increasingly dominant in various fields, including computer vision, natural language processing, and speech recognition. However their complex and deep structures present a significant challenge to interpretability, often earning them the label

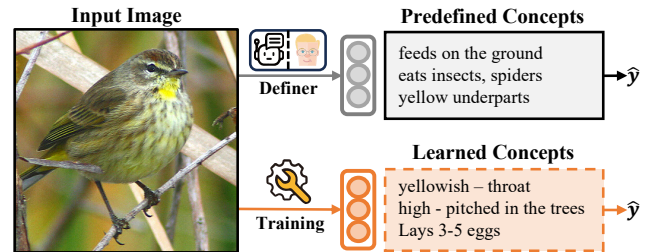


Figure 1. Our proposed concept-trainable CBM reduces the need for meticulously designed concepts by incorporating learned concepts (e.g., ‘yellowish throat’), thereby enhancing interpretability and performance while minimizing manual design effort.

of ‘blackbox’ [23]. Addressing this challenge is crucial not only for ensuring accurate predictions, but also for providing scientists and engineers with more accessible tools for designing, developing, and debugging models. Interpretability is particularly critical in domains such as healthcare [10], medicine [33], education [7], and finance [28], where high-stakes decision-making requires transparency and trustworthiness to ensure reliability and accountability. Therefore, there has been a growing focus on research aimed at developing interpretable models. While post-hoc explanation methods [34, 37, 55] have gained popularity, they often fall short by providing unfaithful representations of the model’s underlying computations [35].

A natural approach to improving interpretability in DNNs is to design inherently interpretable models that leverage high-level concepts to describe features extracted by black-box models [36], imitating the way humans encode and understand the world [20]. One notable approach following such methodology is the Concept Bottleneck Models (CBM) [17]. CBMs operate by first predicting concepts from image representations mid-way through the decision-making process. These models pair each image with its predefined concepts and are trained in an end-to-end manner—initially predicting concepts and subsequently using these predictions to make final decisions. While this method of concept prediction enhances model interpretability, it often faces two fundamental challenges: (1) it is perceived to underperform compared to black-box models [11],

<sup>1</sup>Code available at <https://github.com/ly1998117/HybridCBM>

and (2) the requirement for precise annotations for each concept significantly limits their broader application.

To mitigate these shortcomings, recent research has explored multimodal pretrained model-based CBMs [15, 52, 54], leveraging CLIP [32] to establish correspondences between visual and textual representations, thereby reducing the reliance on costly concept annotations. Specifically, CLIP encodes images and human-understandable concepts into a shared feature space, enabling the creation of a concept bottleneck by directly mapping visual representations to concept embeddings instead of through manual annotations. Regarding the trade-off between interpretability and performance, some studies [15, 38, 52] have emphasized the construction of high-quality concept banks through efficient concept selection. However, establishing a comprehensive concept bank remains a significant challenge [53], often constrained by the inherent limitations in language expressiveness and precision. In response, other research efforts have explored the integration of a residual linear layer to address the issue of incomplete concept extraction [39, 54], albeit at the cost of interpretability. For example, ResCBM [39] introduces optimizable vectors to capture missing concepts, utilizing a candidate concept bank in the residual component to regularizing these vectors toward specific candidates during training. However, the model is limited by the scope of the candidate concept bank, restricting its ability to learn concepts outside the predefined set. Additionally, the effectiveness of the residual structure remains an open question.

In this work, we propose the **Hybrid Concept Bottleneck Model** (HybridCBM), specifically designed to address the challenges of incomplete concept representation, predefined concept bank dependence, and scalability issue by dynamically discovering new concepts directly from visual representations. Different from prior methods that rely on residual structure or external concept banks, our approach incorporates a hybrid concept bank comprising both a static and a dynamic concept bank. Initially, we establish the static concept bank by leveraging a large language model (LLM) like GPT-3.5 [27, 31], which is celebrated for its broad world knowledge [12, 30, 45]. For instance, in Figure 1, when prompted about a palm warbler, GPT-3.5 provides information such as ‘feeds on the ground’. This textual descriptions is then encoded into embeddings by the text encoder of CLIP. Next, we initialize a set of optimizable vectors that form the dynamic concept bank. These vectors are refined to identify concepts present in the input image but absent from the static concept bank. To further enhance the concept discovery, we pre-train a concept translator, such as GPT-2 [31], to translate these newly discovered unknown concepts into textual descriptions, a process we term as ‘concept labeling’. To evaluate the understandability, visual relevance and factual accuracy of the dynamic concepts, we

further design a set of evaluation metrics, some of which are based on Vision-Language Model (VLM) GPT-4o<sup>2</sup>. Our main contributions can be summarized as follows:

1. **Innovative Hybrid Concept Bank:** Our model introduces a hybrid concept bank that combines static and dynamic concepts, allowing it to adapt and refine its interpretative capabilities dynamically. This structure leverages both predefined expert knowledge while enabling the discovery of new relevant concepts.
2. **Seamless Integration of LLM Technologies:** By employing advanced language models (LLM for defining static concepts, translating new concepts and VLM for evaluation) alongside the visual-textual bridging capabilities of CLIP, our model bridges the gap between visual representation and textual interpretation.
3. **Enhanced Interpretability and Performance:** Our approach maintains high interpretability through transparent concept labeling, while improving performance by discovering new concepts from visual information.

## 2. Related Work

**Interpretable Neural Networks.** One way to build an interpretable neural networks is the use of a concept-based explanation [3, 13, 15, 21, 29, 38, 43, 49]. Concept Bottleneck Models (CBMs) [17] are among the most popular approaches for making predictions based on human-interpretable concepts but require labor-intensive concept annotations for each image. Similarly, Concept Activation Vectors (CAVs) [14] represent concepts as normal vectors to the decision boundaries that separate positive and negative samples of a concept, though they also need extra datasets to train SVMs for each concept. Owing to the simple structure based on human-defined concepts, there have been studies aiming to alleviate these drawbacks, e.g., costly concept annotation and trade-off between interpretability and performance. Post-hoc Concept Bottleneck (PCBM) [54] utilizes ConceptNet [42] to obtain concepts and Leveraging CLIP to generate bottlenecks by projecting visual representations into the concept subspace. However, PCBM-h [54] incorporates a residual linear predictor that compensates for missing concepts by directly adding logits to PCBM’s predictor. While this modification helps to recover accuracy, it does so at the expense of interpretability. The label-free CBM [26] uses CLIP for concept annotation, allowing arbitrary visual backbone to be transformed into an CBM without the need for labeled concept data. Labo [52] concentrates on concepts selection after generating candidate concepts through LLM. ResCBM [39] builds on PCBM-h’s residual structure by using a candidate concept bank to guide and refine optimizable vectors. However, the model’s expressive potential is limited by the extent of the candidate concept bank.

<sup>2</sup><https://openai.com/index/hello-gpt-4o/>

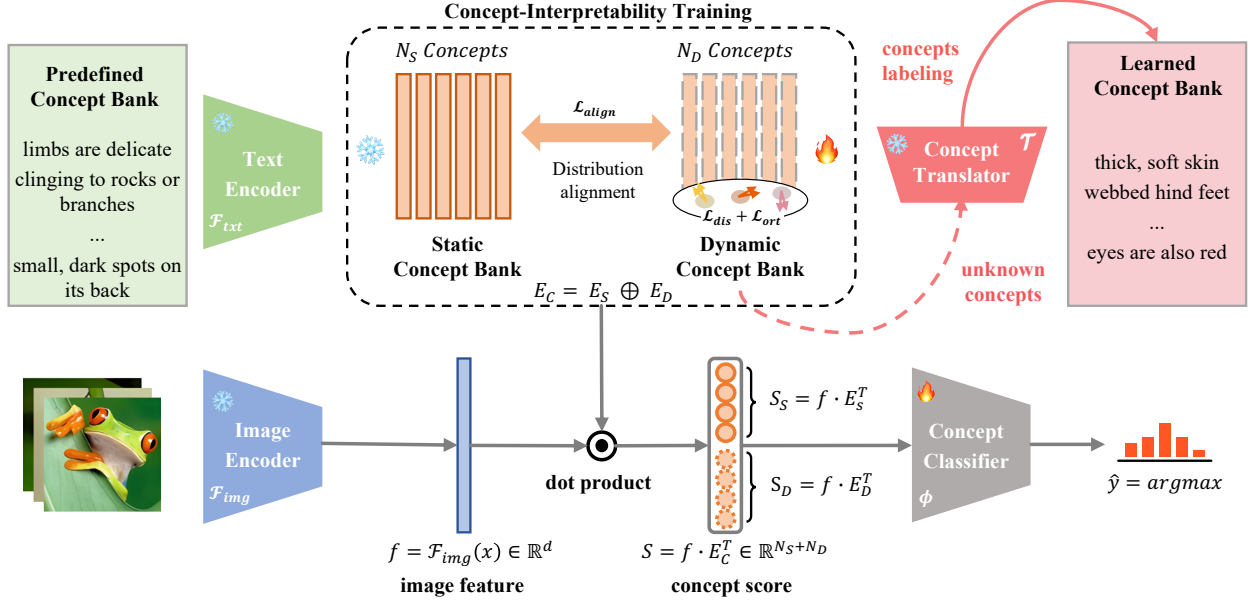


Figure 2. Overview of the **Hybrid Concept Bottleneck Model (HybridCBM)**, which is designed for interpretable concept learning and labeling. First, a Concept Translator is pre-trained to do concept-labeling. Next, a pretrained alignment model (CLIP) is used to obtain embeddings of predefined concepts, forming the Static Concept Bank. The Dynamic Concept Bank learns missing concepts and aligns its distribution with the Static Concept Bank to create the final concept bottleneck. Finally, an interpretable concept-based predictor classifies targets based on the concept scores.

**CLIP in Image Captioning.** Multimodal pretrained models like CLIP [32], trained on over 400 million image-text pairs using unsupervised contrastive loss, exhibit impressive capabilities in image captioning tasks [1, 19, 46]. Previous studies, such as [24] and [40], have primarily used CLIP as a visual encoder for captioning tasks, while [19] projects visual embeddings into CLIP’s text embedding space to preserve visual information. However, these approaches often underutilize CLIP’s aligned multi-modal latent space. Similar to [19, 24], we collect image-caption pairs and a vast corpus to train a GPT-2 from scratch as our concept translator, fully leveraging CLIP’s latent space and bridging the gap between image and concept embeddings and their corresponding textual descriptions.

### 3. Method

We provide an overview of our methodology in Figure 2. Our model incorporates a hybrid concept bank and a pre-trained concept translator (Section 3.2). The concept bank consists of a set of optimizable vectors, which are refined to discover new concepts through a *concept-interpretability training scheme* (Section 3.3). As a result, the learned concepts are not only discriminative and orthogonal but also aligned with human-understandable semantics. The pre-trained concept translator performs concept labeling by mapping these learned, initially unknown vectors to meaningful human concepts. Finally, we apply a sparse linear layer over the similarity scores between hybrid concepts

and images, enabling us to identify the specific concepts the model relies on for its decisions.

#### 3.1. Problem Formulation

Consider a dataset  $\mathcal{D} = (x, y)$ , where each image  $x$  is paired with a label  $y \in \mathcal{Y}$ . We use the image encoder  $\mathcal{F}_{img}$  and text encoder  $\mathcal{F}_{txt}$  of CLIP [32] to map images and text into a shared  $d$ -dimensional feature space, respectively. A pre-trained concept translator  $\mathcal{T}$  then maps these shared  $d$ -dimensional features back to textual concepts. In order to store the interpretable information of the model, we construct a hybrid concept bank  $E_C \in \mathbb{R}^{N_C \times d}$ , comprising:

- **Static Concept Bank**  $E_S \in \mathbb{R}^{N_S \times d}$ . Contains text features  $\mathcal{F}_{txt}(c) \in \mathbb{R}^d$  extracted by the text encoder  $\mathcal{F}_{txt}$ . Each row corresponds to a predefined concept  $C_S = \{c_1, c_2, \dots, c_{N_S}\}$  generated from a large language model.
- **Dynamic Concept Bank**  $E_D \in \mathbb{R}^{N_D \times d}$ . Consists of a set of randomly initialized learnable vectors  $e_i \in \mathbb{R}^d$  that are optimized to capture missing visual-specific features. After training, These vectors could be mapped to the textual concept space  $C_D = \{c_1, c_2, \dots, c_{N_D}\}$  by the translator  $\mathcal{T}$ , where each concept  $c = \mathcal{T}(e)$ .

The overall hybrid concept bank is defined as  $E_C = E_S \oplus E_D$ , where  $\oplus$  denotes concatenation. The total number of concepts in the bank is indicated by  $N_C = N_S + N_D$ . To compute the concept scores, we utilize the CLIP image encoder to extract the image feature  $f = \mathcal{F}_{img}(x) \in \mathbb{R}^d$ . Therein, the similarity between  $f$  and  $E_C$  is computed using projection length, resulting in the concept score  $S \in \mathbb{R}^{N_C}$ ,

which reflects the presence of particular concepts in the image  $x$ :  $S_x = \|f\|_2 \cos(f, E_C) = f \cdot \hat{E}_C^T$ , where  $\hat{\cdot}$  denotes  $L_2$  normalization and  $\cos$  is the cosine similarity. Since  $S$  provides a unique representation for each image, it serves as a concept bottleneck. These per-image representations can be naturally used to support an interpretable classification. Following the existing CLIP-based CBMs [26, 52, 54], we train a linear layer as the concept classifier  $\phi : \hat{y} = \phi(S), \mathbb{R}^{N_C} \rightarrow \mathcal{Y}$  to make final predictions in the label space based on the concept scores. In our perspective, we intend to improve the performance and learn abundant concepts from the visual information at the same time. This can be solved by the following optimization problem:

$$\min_{\phi, E_D} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ \mathcal{L}_{cls} \left( \phi(f \cdot \hat{E}_S^T \oplus f \cdot \hat{E}_D^T), y \right) \right] + \lambda \Omega(\phi) + \Gamma_{E_D} \quad (1)$$

where  $\mathcal{L}_{cls}(\hat{y}, y)$  is the cross-entropy loss function,  $\Omega$  is a complexity measure used to promote sparsity in the linear layer,  $\lambda$  is the regularization strength and  $\Gamma$  represents the optimization objective for the Dynamic Concept Bank  $E_D$ , introduced in Section 3.3.

### 3.2. Hybrid Concept Bank Establishment

Previous methods, such as Labo [52], generate a set of relevant concepts and then form a concept bank through the submodular selection method. However, the predefined concepts often suffer from inherent flaws due to their limited scope, and the selection process may inadvertently omit important information. To address these issues, we propose a Hybrid Concept Bank that dynamically discovers concepts directly from the visual representation, combining both a Static Concept Bank and a Dynamic Concept Bank.

**Static Concept Bank.** Following Labo, we utilize GPT-3.5 to generate a set of candidate concepts for each class name  $y \in \mathcal{Y}$  with prompts such as “describe what the <CLASS NAME> looks like”. After initially generating 500 concepts per class, we perform concept selection to ensure a balanced representation. Specifically, we select  $N$  concepts for each class using both submodular and random selection methods for comparison. Assume  $K = |\mathcal{Y}|$ , this process yields a total of  $N_S = N \times K$  concepts, which are then encoded by the text encoder  $\mathcal{F}_{txt}$  of CLIP and fixed during training.

**Dynamic Concept Bank.** Unlike the static counterpart, the Dynamic Concept Bank is initialized with a set of concept vectors  $e_d \in \mathbb{R}^d$ , which are optimized through the proposed *Concept-Interpretability Training Scheme* (Section 3.3). Thus, it can adaptively capture the unique and rich visual-specific features during the training process.

**Concept Labeling.** To enhance the interpretability of learned concepts, we pre-train a concept translator using GPT-2 [31] architecture to invert the CLIP text encoder.

Following recent work [19, 24, 50], we use prefix language modeling for training. Details of the pre-training process are provided in Appendix B.2. The translator maps each concept embedding  $e$  to its textual description  $c$  through  $c = \mathcal{T}(e)$ , assigning semantic meaning to the learned concepts and improving interpretability.

### 3.3. Concept-Interpretability Training Scheme

The previous CLIP-based CBMs rely solely on predefined concepts, often overlooking the potential of learning concepts directly from visual representation. Since CLIP maps both images and texts into a shared feature space, we could then leverage this capability to optimize the dynamic concept bank  $E_D$  based on the given image embedding  $f$ . To achieve this, we introduce a concept-interpretability training scheme that regularizes  $E_D$  through multiple loss functions. The goal is to obtain a set of learnable concepts that are discriminative, diverse, and semantically aligned with human-understandable meanings, thereby enhancing both model interpretability and performance.

**Discriminability Loss.** The discriminability loss ensures that learned concepts are highly activated for images of their own class and less activated for images of other classes. We first divide the dynamic concept vectors into several subsets corresponding to the classes, such that  $E_D = \{E_D^1, E_D^2, \dots, E_D^K\}$ , where  $K$  is the number of classes. Each subset  $E_D^k$  contains  $N_D/K$  dynamic concept vectors associated with class  $k$ . To encourage vectors  $\mathbf{c}_k \in E_D^k$  to learn specific information of images within class  $k$  and incorporate semantic information from the class name, we define the intra-class loss:

$$\mathcal{L}_{intra}^i = y_k \left\{ -\hat{f} \cdot \hat{\mathbf{c}}_k + \alpha \left| m - \hat{\mathcal{F}}_{txt}(k) \cdot \hat{\mathbf{c}}_k \right| \right\} \quad (2)$$

where  $f = \mathcal{F}_{img}(x)$  is the image feature,  $\mathcal{F}_{txt}(k)$  is the class embedding,  $\mathbf{c}_k$  is the sampled concept vector for class  $k$  from the distribution of vectors  $E_D^k$  via reparameterization<sup>3</sup>,  $\alpha$  is a hyperparameter and  $m$  is a margin used to prevent excessive similarity to class  $k$ , with a value of 0.85 as defined in Label-Free [26]. To ensure that concepts are distinct between different classes, we further define the inter-class loss:

$$\mathcal{L}_{inter}^i = (y_k - 1) \left\{ -\hat{f} \cdot \hat{\mathbf{c}}_k + \alpha \left| m - \hat{\mathcal{F}}_{txt}(k) \cdot \hat{\mathbf{c}}_k \right| \right\} \quad (3)$$

The total discriminability loss is defined as  $\mathcal{L}_{dis} = \mathcal{L}_{intra} + \beta \mathcal{L}_{inter}$ , where  $\beta$  is a balancing hyperparameter.

**Orthogonality Loss.** To enhance the diversity of dynamic concept embeddings and reduce redundancy, we introduce an orthogonality loss, which comprises two components:

<sup>3</sup>  $\mathbf{c}_k = \mu_k + \sigma_k \cdot \epsilon$ , with  $\mu_k$  and  $\sigma_k$  being the mean and standard deviation of  $E_D^k$ , respectively, and  $\epsilon \sim \mathcal{N}(0, I)$  is sampled from the standard normal distribution.

intra-class diversity and inter-bank diversity. First, we promote the dynamic concept embeddings within the same class to be orthogonal to each other. Specifically, we define the *intra-class diversity loss* as follows:

$$\mathcal{L}_{ort-intra} = \frac{1}{K} \sum_{k=1}^K \frac{1}{(N_D/K)^2} \sum_{i \neq j} |\hat{\mathbf{e}}_{D,i}^k \cdot \hat{\mathbf{e}}_{D,j}^k| \quad (4)$$

where  $\mathbf{e}_{D,i}^k$  and  $\mathbf{e}_{D,j}^k$  are dynamic concept embeddings associated with class  $k$ ,  $K$  is the number of classes, and  $N_D$  is the total number of dynamic concepts. Second, to minimize redundancy between static and dynamic concepts, we define the *inter-bank diversity loss* as follows:

$$\mathcal{L}_{ort-inter} = \frac{1}{N_S N_D} \sum_{i=1}^{N_S} \sum_{j=1}^{N_D} |\mathbf{e}_{S,i} \cdot \hat{\mathbf{e}}_{D,j}| \quad (5)$$

where  $\mathbf{e}_{S,i}$  and  $\mathbf{e}_{D,j}$  are static and dynamic concept embeddings, respectively, and  $N_S$  is the number of static concepts. The total orthogonality loss is then defined as  $\mathcal{L}_{ort} = \mathcal{L}_{ort-intra} + \mathcal{L}_{ort-inter}$ .

**Distribution Alignment Loss.** To enhance consistency between the distributions of dynamic and static concepts, we employ the Sinkhorn divergence [8], an entropy-regularized version of the Sinkhorn distance. We define the *distribution alignment loss* as:  $\mathcal{L}_{align} = \mathfrak{S}_{div,\epsilon}(E_D, E_S)$ , where  $\mathfrak{S}_{div,\epsilon}$  denotes the Sinkhorn divergence,  $E_D \in \mathbb{R}^{N_D \times d}$  and  $E_S \in \mathbb{R}^{N_S \times d}$  are the dynamic and static concept embeddings, respectively. This loss encourages the distribution of dynamic concepts to align with that of static concepts, thus promoting semantic consistency and interpretability.

**Overall Loss Function.** The total loss function combines the classification loss with the proposed regularization terms  $\mathcal{L} = \mathcal{L}_{cls} + \lambda_{dis} \mathcal{L}_{dis} + \lambda_{ort} \mathcal{L}_{ort} + \lambda_{align} \mathcal{L}_{align}$ , where  $\lambda_{dis}$ ,  $\lambda_{ort}$ , and  $\lambda_{align}$  are hyperparameters to balance the losses.

## 4. Experiments

### 4.1. Experiments Setup

**Dataset.** We collect image-caption pairs from MSCOCO [4] and then compile a large corpus from ConceptNet [42] and MSCOCO, which also includes concepts generated by GPT-3.5. After filtering out concepts that contain fewer than 15 words, the dataset includes 1,738,985 concepts and 566,747 image-concept pairs, used to pre-train our concept translator for translating both image and text embeddings. We also perform comprehensive experiments on 11 classification datasets, covering a broad spectrum of domains, including: ImageNet [9], CIFAR-10 and CIFAR-100 [18], Food-101 [2], FGVC-Aircraft [22], Flower-102 [25], CUB-200-2011 [48], UCF-101 [41], DTD [6], HAM10000 [47] and RESISC45 [5]. Each dataset is split into training, validation, and test sets, as per standard practices.

**Baselines.** To evaluate our model, we compare it with a linear probe and several interpretable methods.

- **Linear Probe.** Following the implementation of CLIP [32], we train a logistic regression model using `cuML`'s L-BFGS solver with an L2 penalty term, directly from the visual representations encoded by CLIP.
- **PCBM [54].** PCBM conceptualizes each class as a node in ConceptNet [42] and aggregates neighboring nodes as concepts. PCBM-h [54] introduces an additional classifier that maps image embeddings into the label space, serving as residual shortcuts to the original classifier.
- **Labo [52].** Labo employs a submodular function to select concepts for each class from candidate concepts generated by LLM. Unlike other models, it uses dot-product, without any normalization, to determine the presence of particular concepts in an image.
- **ResCBM[39].** Building on PCBM-h, ResCBM utilizes ConceptNet [42] to construct a candidate concept bank. It optimizes a set of vectors to align with some concepts from this bank, thereby enhancing the interpretability.

**Implementation Details.** The hybrid concept bank is constructed with a default ratio of 0.5 between the dynamic and static concept bank, ensuring an equal number of concepts for each class. Our model is trained using the Adam [16] optimizer within the `PyTorch Lightning` framework. We save checkpoints that achieve the highest validation accuracy and perform evaluations on the test set. Further details can be found in the supplementary material.

### 4.2. Classification Performance

**Few-shot Comparison.** To assess the performance gap between HybridCBM and 'black-box' models, we compare our method with the linear probe (LP) in a few-shot setting. We follow the few-shot evaluation protocol introduced by CLIP [32]. Specifically, we randomly sample 1, 2, 4, 8, and 16 images per class from the training set and use all available images in the fully-supervised setting. For all experiments, we use CLIP-ViT-L/14 as the backbone and compare the performance with submodular [52] and random static concept selection methods. As shown in Table 1, the mean test accuracy across all datasets indicates that HybridCBM performs similarly to LP in the fully-supervised setting and significantly surpasses LP in the few-shot scenario. Figure 3 presents detailed comparisons across eight datasets, with the last three provided in the appendix.

**CBMs Comparison.** We further compare HybridCBM with other CBMs and employ CLIP-RN50 as the backbone to ensure fairness. Additionally, since our dynamic concepts are derived from visual representation, we create a variant called CaptionCBM, where  $N_D/K$  images for each class are sampled and translated into captions to replace the dynamic concepts. As summarized in Table 2, despite strong baselines like LaBo, and ResCBM, Hybrid-

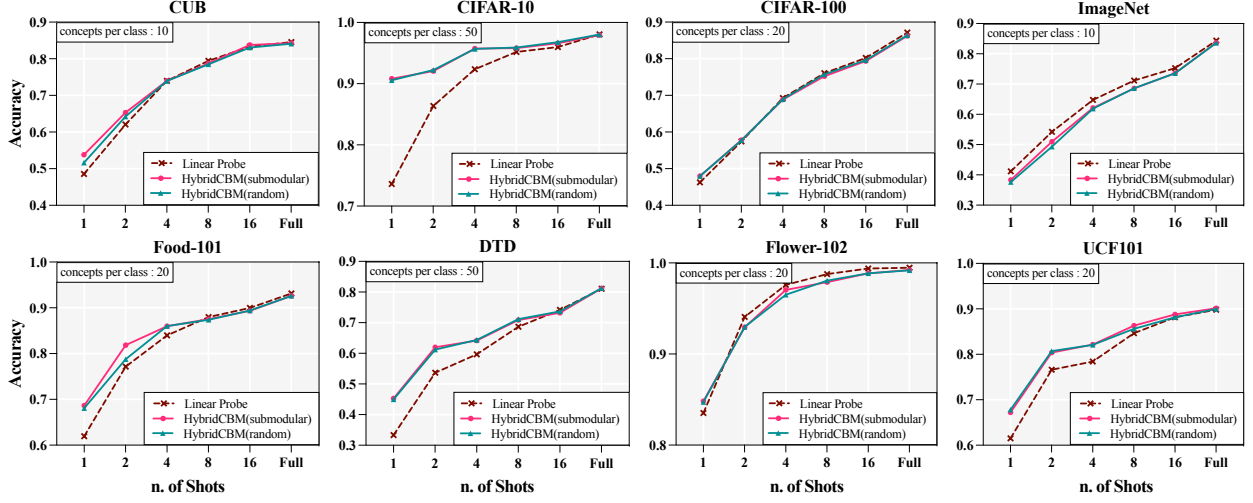


Figure 3. Comparison of test accuracy between HybridCBM with submodular and random selection methods, and Linear Probe across 8 datasets. The x-axis denotes the number of labeled images.

Method	1	2	4	8	16	Full	Avg
Linear Probe	52.30	64.60	71.94	78.33	81.55	<b>86.84</b>	72.59
Labo	N/A	N/A	N/A	N/A	N/A	85.72	N/A
HybridCBM (submodular)	<b>58.64</b>	<b>69.52</b>	<b>75.08</b>	<b>78.71</b>	<b>81.93</b>	<u>86.83</u>	<b>75.12</b>
HybridCBM (random)	<b>57.58</b>	<b>66.69</b>	<b>74.62</b>	78.02	80.98	<u>86.82</u>	<b>74.16</b>

Table 1. Mean accuracy across all datasets, at different shots.

Method	Interpretability	CIFAR-10	CIFAR-100
Linear Probe	✗	88.8	70.1
PCBM-h [54]	✗	87.6	69.9
PCBM [54]	✓	84.5	56.0
Label-Free [26] <sup>†</sup>	✓	86.3	65.2
Labo [52]	✓	87.6	65.2
ResCBM [39] <sup>†</sup>	✓	88.0	67.9
CaptionCBM	✓	81.8	60.7
HybridCBM	✓	<b>88.03</b>	<b>68.38</b>

Table 2. Test accuracy comparison between HybridCBM and on CIFAR-10 and CIFAR-100. <sup>†</sup> indicates the reported performance.

CBM achieves state-of-the-art performance on CIFAR-10 and CIFAR-100, and delivers results comparable to LP.

### 4.3. Interpretability of the dynamic concepts

In addition to evaluating classification performance, we assess the interpretability of the learned concepts in this section using the following metrics:

- **Concept Purity:** This metric measures how well each concept aligns with its respective class, reflecting the ability of the concepts to capture class-specific features. For each class  $k$ , we compute the average cosine similarity between the normalized embedding of the concept  $\hat{e}_k$  and the class name  $\hat{\mathcal{F}}_{txt}(k)$ .

$$Purity = \frac{1}{K} \sum_{k=1}^K (\bar{e}_k \cdot \hat{\mathcal{F}}_{txt}(k)) \quad (6)$$

where  $K$  is the total number of classes, and  $\bar{e}_k$  denotes the mean embedding of concepts for class  $k$ . Higher purity indicates stronger alignment with class-specific features.

- **Concept Separation:** The Separation metric quantifies how distinct the learned concepts are across classes, capturing their semantic independence. We calculate the average cosine similarity between the mean concept embeddings of all distinct class pairs, excluding self-similarity.

$$Separation = 1 - \frac{2}{K(K-1)} \sum_{i=1}^K \sum_{j=i+1}^K (\bar{e}_i \cdot \bar{e}_j) \quad (7)$$

where  $\bar{e}_i$  denotes the mean embedding of concepts for class  $i$ . Higher separation indicates greater independence between concepts of different classes in semantic space.

- **Semantic Alignment:** To ensure semantic alignment between translated concepts and intended dataset classes, we use GPT-3.5 to evaluate each translated concept’s accuracy in representing a specific class with a binary “yes” or “no” response. The validation metric, “Semantics”, is calculated as the proportion of concepts that GPT-3.5 correctly associates with the intended classes, reflecting the semantic interpretability. The detailed information is provided in Appendix B.3.

$$Semantics = \frac{1}{N_D} \sum_{i=1}^{N_D} \delta(LLM(c_i, classes), 1) \quad (8)$$

where  $LLM(c_i, classes)$  represents the response to whether concept  $c_i$  belongs to one of the classes, and  $\delta$  is a binary function that returns 1 for “yes” and 0 otherwise. To mitigate potential limitations of the LLM, we emphasize comparing the relative size of static and dynamic concept groups over absolute values.

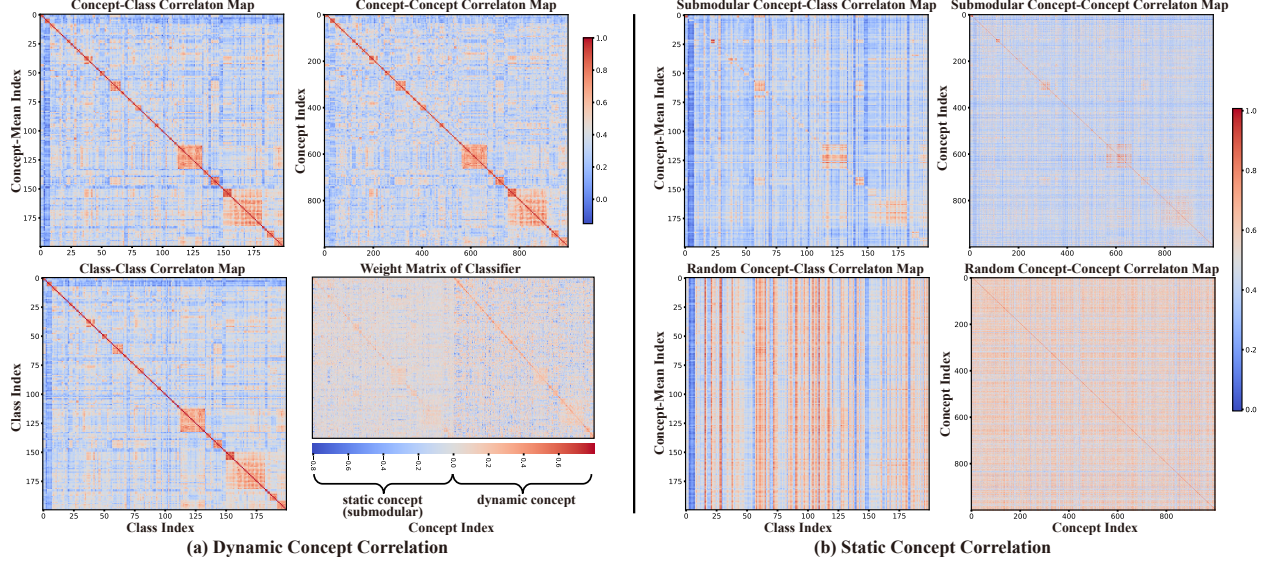


Figure 4. Comparison of dynamic and static concept correlation structures on the CUB dataset. (a) Dynamic concepts: Correlation maps and classifier weights illustrate how dynamic concepts adjust their relationships with classes and among themselves. (b) Static concepts: Selected and random concepts show their contributions to class relationships and inter-concept dependencies.

Loss	Concept Bank	Purity (%)	Separation (%)	Semantics (%)	Precision@t (%)
$\mathcal{L}_{cls}$	Static (submodular)	19.3	86.2	<b>38.3</b>	48.5
	Static (random)	19.2	82.4	30.9	47.3
	Dynamic	0.1	100.0	0.01	0.05
$\mathcal{L}_{cls+dis}$	Dynamic	71.9	76.9	30.0	34.8
$\mathcal{L}_{cls+dis+ort}$	Dynamic	72.0	77.1	28.9	34.1
$\mathcal{L}_{cls+dis+ort+align}$	Dynamic	39.8	80.0	<b>32.8</b>	<b>46.2</b>

Table 3. Evaluation of HybridCBM’s static and dynamic concepts across various metrics.

- **Concept-Image Relevance:** As shown on the right in Figure 5, to evaluate the accuracy of the concepts aligning with the actual content of each image, we use the ability of GPT-4o in image understanding to calculate the retrieval *Precision@t*. For a sampled image of each class, we measure the average proportion of  $t$  concepts that GPT-4o confirms as relevant.




$$Precision@t = \frac{1}{K} \sum_{i=1}^K \frac{\delta(VLM(c_{i,t}, img), 1)}{t} \quad (9)$$

where  $VLM(c_{i,t}, img)$  is the response of GPT-4o and  $K$  is the total number of classes.  $c_{i,t}$  is  $t$  concepts of class  $i$ .

**Concept Analysis.** We perform concept analysis on the CUB dataset. Figure 4 compares the correlation structures of dynamic and static concepts. In Figure 4a, the Concept-Class Correlation Map shows that our dynamic concepts align closely with their corresponding classes while effectively capturing diversity across classes. The Concept-Concept Correlation Map further highlights the diversity among concepts. Notably, regions with higher coefficients


reflect natural semantic relationships between certain class names, such as “red-eyed vireo” and “white-eyed vireo”. The pattern also seen in the Class-Class Correlation Map. The weight matrix of the classifier in Figure 4a reveals that the dynamic concept part has a more pronounced diagonal compared to the static concept, indicating that dynamic concepts are more discriminative and align better with their corresponding classes. In Figure 4b, we present similar relation maps of concepts selected by a submodular function and randomly selected concepts from each class. The selected concepts are more cohesive with their classes, whereas the randomly selected concepts, despite belonging to each class, do not exhibit clear patterns. Table 3 details the evaluation metrics proposed. For static concepts, submodular selection achieves a higher semantics score than random selection (38.3% vs 30.9%). For dynamic concepts, after introducing the loss functions  $\mathcal{L}_{dis}$ ,  $\mathcal{L}_{ort}$ , and  $\mathcal{L}_{align}$ , the semantic score of dynamic concepts improved to 32.8%, approaching static concepts. This indicates that our proposed loss functions effectively enhance the alignment and discriminative power of dynamic concepts.

**Interpretability.** Interpretability can be divided into pattern and case interpretability. Pattern interpretability focuses on general patterns the model has learned, providing insight into its global behavior and how features influence classes. For example, a classifier’s weight matrix reflecting bias toward each class. Case interpretability explains the model’s decisions for individual instances, offering a local understanding of why a specific prediction was made for an image—for instance, concept scores indicating how strongly certain concepts are present. Following MMCBM [51], We


	Class Name	Sample	Case interpretability	
			static concepts	dynamic concepts
CUB	Laysan Albatross		1.lays 4-6 white eggs 2.black and white striped body	1.black and white albatross 2.largest albatross species
Food-101	Baklava		1.flaky, phyllo dough texture 2.filled with nuts and sweetened with syrup	1.light brown center 2.a food is arranged on a large platter
Flower	Clematis		1.popular choice for making garlands and wreaths 2.borne on a climbing vine	1.large , a seed plant 2.violet

**Concept Retrieval**

You are an expert binary concept classifier, capable of determining whether a given concept has any form of relationship with the provided image. If it has any relationship with the image, respond with "yes". Otherwise, respond with "no".



- sometimes called the "black albatross"
- known as the black-browed albatross
- most abundant albatross species



(no, yes, yes)

Figure 5. On the left, the top-2 static and dynamic concepts for randomly selected classes across three datasets are presented, focusing on case interpretability. The concept retrieval process with vision-language model GPT-4o is illustrated on the right.

Method	concepts per class	Ratio of Dynamic Concepts					
		0	0.2	0.4	0.6	0.8	1
LP	N/A	<b>87.09</b>	N/A	N/A	N/A	N/A	N/A
Hybrid (submodular)	5	83.21	84.46	85.45	86.08	86.34	86.78
	10	84.27	85.47	86.28	86.51	86.87	87.13
	15	84.75	85.84	86.45	86.78	87.08	87.25
	20	84.93	86.09	86.44	86.84	86.88	87.24
Hybrid (random)	5	80.95	84.04	85.28	85.96	86.07	86.53
	10	82.88	85.28	86.30	86.62	86.82	87.13
	15	83.65	85.81	86.54	86.80	86.89	87.25
	20	83.85	86.09	86.66	86.86	87.06	87.24

Table 4. Ablation average results on 10 datasets for varying the ratio of dynamic concepts under different methods.

Method	Metric	$\mathcal{L}_{cls}$	$\mathcal{L}_{cls+dis}$	$\mathcal{L}_{cls+dis+ort}$	$\mathcal{L}_{cls+dis+ort+align}$
Hybrid (submodular)	accuracy	84.60	84.36	84.29	84.54
	precision@t	0.047	34.8	34.1	46.2
Hybrid (random)	accuracy	84.50	84.16	83.91	84.17
	precision@t	0.044	34.8	35.5	47.2

Table 5. Ablation results for varying training loss with a dynamic concept ratio of 0.5 and 10 concepts per class.

use the attention matrix, computed as  $S_i \times W$ , where  $W$  is the classifier’s weight, to show that the prediction is based on specific concepts within the image. As illustrated in Figure 5, we show the case interpretability by selecting the top 2 concepts for each image. we evaluate case interpretability by computing retrieval precision@t, measuring how accurately the top 5 identified concepts align with the image content. The results are shown in Table 3, where the static concepts serve as a baseline for comparison with the dynamic concepts; we focus on relative differences due to the potential limitations of GPT-4o. The dynamic concepts, across all loss functions, achieve 46.2%, closely approaching the static concepts’ performance of 47.3%. This demonstrates that dynamic concepts offer high interpretability and can effectively serve as an alternative to static concepts.

#### 4.4. Ablation Study

**Dynamic Concept Bank Ratio.** To examine the impact of dynamic concept ratios on performance, we test various ratios on 10 datasets except ImageNet, as shown in Table 4.

As the dynamic concept ratio increases, performance will also improve. Notably, using random concepts per class results in an performance drop compared to the submodular selection method. Furthermore, employing learned concepts, even at a ratio of 0.2, significantly improves performance.

**Concept-Interpretability Loss.** Table 5 reports the accuracy and precision@t (t=5) of learned concepts across different combinations of loss functions, comparing the Hybrid (submodular) and Hybrid (random) methods. A slight decrease in accuracy is observed as additional terms are added to the  $\mathcal{L}_{cls}$  but significantly improves precision@t. This suggests a trade-off, where additional constraints, while slightly reducing accuracy (e.g., accuracy drops from 84.60 to 84.29 for the Hybrid (submodular) method), enhance concept interpretability.

## 5. Conclusion

In this paper, we introduced HybridCBM, a novel hybrid concept bottleneck model designed to enhance interpretability and performance by combining both static and dynamic concept banks. Unlike traditional CBMs, our method allows the model to dynamically discover new concepts during training, thus expanding its ability to capture unique and relevant features. Our approach also incorporates a concept translator to provide human-readable interpretations for the learned dynamic concepts, making the model’s decision process more accessible and understandable. Through extensive experiments on various datasets, our method achieves comparable accuracy to ‘black-box’ models while retaining transparency, bridging the gap between interpretability and performance. We believe that HybridCBM represents a significant step forward in the development of interpretable machine learning models. Future work could further explore methods for optimizing the accuracy of concept labeling and exploring more sophisticated strategies for dynamic concept discovery.

## 6. Acknowledgements.

This project is supported by NSFC Key Program 62236009, Shenzhen Fundamental Research Program (General Program) JCYJ 20210324140807019, NSFC General Program 61876032, and Key Laboratory of Data Intelligence and Cognitive Computing, Longhua District, Shenzhen.

## References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 3
- [2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 5
- [3] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 2
- [4] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server, 2015. 5, 1, 2
- [5] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 5
- [6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 5
- [7] Cristina Conati, Kaska Porayska-Pomsta, and Manolis Mavrikis. Ai in education needs interpretable machine learning: Lessons from open learner modelling, 2018. 1
- [8] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2013. 5
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5
- [10] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019. 1
- [11] David Gunning and David Aha. Darpa’s explainable artificial intelligence (xai) program. *AI magazine*, 40(2):44–58, 2019. 1
- [12] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8: 423–438, 2020. 2
- [13] Dmitry Kazhdan, Boty Dimanov, Mateja Jamnik, Pietro Liò, and Adrian Weller. Now you see me (cme): Concept-based model extraction, 2020. 2
- [14] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, pages 2668–2677. PMLR, 2018. 2
- [15] Injae Kim, Jongha Kim, Joonmyung Choi, and Hyunwoo J. Kim. Concept bottleneck with visual concept filtering for explainable medical image classification, 2023. 2
- [16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5, 4
- [17] 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
- [18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5
- [19] Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. Decap: Decoding clip latents for zero-shot captioning via text-only training. In *The Eleventh International Conference on Learning Representations*, 2023. 3, 4, 2
- [20] Yuanning Li, Huzheng Yang, and Shi Gu. Enhancing neural encoding models for naturalistic perception with a multi-level integration of deep neural networks and cortical networks. *Science Bulletin*, 2024. 1
- [21] Max Losch, Mario Fritz, and Bernt Schiele. Interpretability beyond classification output: Semantic bottleneck networks, 2019. 2
- [22] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5
- [23] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. 1
- [24] Ron Mokady, Amir Hertz, and Amit H Bermano. Clip-cap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 3, 4, 2
- [25] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 5
- [26] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 4, 6
- [27] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information*

- Processing Systems*, pages 27730–27744. Curran Associates, Inc., 2022. 2
- [28] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications: A survey. *Applied Soft Computing*, 93:106384, 2020. 1
- [29] Konstantinos Panagiotis Panousis, Dino Ienco, and Diego Marcos. Sparse linear concept discovery models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2767–2771, 2023. 2
- [30] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, 2019. Association for Computational Linguistics. 2
- [31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2, 4
- [32] 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. 2, 3, 5
- [33] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. *Deep Learning for Medical Image Processing: Overview, Challenges and the Future*, page 323–350. Springer International Publishing, 2017. 1
- [34] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ” why should i trust you?” explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 1
- [35] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. 1
- [36] Gesina Schwalbe. Concept embedding analysis: A review. *arXiv preprint arXiv:2203.13909*, 2022. 1
- [37] 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. 1
- [38] Chenming Shang, Hengyuan Zhang, Hao Wen, and Yujiu Yang. Understanding multimodal deep neural networks: A concept selection view, 2024. 2
- [39] Chenming Shang, Shiji Zhou, Hengyuan Zhang, Xinzhe Ni, Yujiu Yang, and Yuwang Wang. Incremental residual concept bottleneck models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11030–11040, 2024. 2, 5, 6
- [40] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can CLIP benefit vision-and-language tasks? In *International Conference on Learning Representations*, 2022. 3
- [41] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5
- [42] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 2, 5, 1
- [43] Divyansh Srivastava, Ge Yan, and Tsui-Wei Weng. Vlg-cbm: Training concept bottleneck models with vision-language guidance, 2024. 2
- [44] Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*, 2018. 2
- [45] Mujeen Sung, Jinhyuk Lee, Sean Yi, Minji Jeon, Sungdong Kim, and Jaewoo Kang. Can language models be biomedical knowledge bases? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4723–4734, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. 2
- [46] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zero-shot image-to-text generation for visual-semantic arithmetic. *arXiv preprint arXiv:2111.14447*, 2021. 3
- [47] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018. 5
- [48] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5
- [49] Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Learning bottleneck concepts in image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2
- [50] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations*, 2022. 4
- [51] Yifan Wu, Yang Liu, Yue Yang, Michael S. Yao, Wenli Yang, Xuehui Shi, Lihong Yang, Dongjun Li, Yueming Liu, James C. Gee, Xuan Yang, Wenbin Wei, and Shi Gu. A concept-based interpretable model for the diagnosis of choroid neoplasias using multimodal data, 2024. 7
- [52] 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. 2, 4, 5, 6
- [53] 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. In *Advances in Neural Information Processing Systems*, pages 20554–20565. Curran Associates, Inc., 2020. 2

- [54] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR 2022 Workshop on PAIR<sup>2</sup> Struct*, 2022. [2](#), [4](#), [5](#), [6](#)
- [55] Yu Zhang, Peter Tiño, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021. [1](#)
- [56] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. [1](#)