Flexible Concept Bottleneck Model

Xingbo Du[1*†], Qiantong Dou[2*], Lei Fan[2], Rui Zhang[3†]

[1]Mohamed bin Zayed University of Artificial Intelligence
[2]School of Computer Science and Engineering, University of New South Wales
[3]Gaoling School of Artificial Intelligence, Renmin University of China
Xingbo.Du@mbzuai.ac.ae, q.dou@student.unsw.edu.au, lei.fan1@unsw.edu.au, rayzhang@ruc.edu.cn

Abstract

Concept bottleneck models (CBMs) improve neural network interpretability by introducing an intermediate layer that maps human-understandable concepts to predictions. Recent work has explored the use of vision-language models (VLMs) to automate concept selection and annotation. However, existing VLM-based CBMs typically require full model retraining when new concepts are involved, which limits their adaptability and flexibility in real-world scenarios, especially considering the rapid evolution of vision-language foundation models. To address these issues, we propose Flexible Concept Bottleneck Model (FCBM), which supports dynamic concept adaptation, including complete replacement of the original concept set. Specifically, we design a hypernetwork that generates prediction weights based on concept embeddings, allowing seamless integration of new concepts without retraining the entire model. In addition, we introduce a modified sparsemax module with a learnable temperature parameter that dynamically selects the most relevant concepts, enabling the model to focus on the most informative features. Extensive experiments on five public benchmarks demonstrate that our method achieves accuracy comparable to state-of-the-art baselines with a similar number of effective concepts. Moreover, the model generalizes well to unseen concepts with just a single epoch of fine-tuning, demonstrating its strong adaptability and flexibility.

Code — https://github.com/deepopo/FCBM

1 Introduction

Understanding black-box models is a core challenge in developing trustworthy and interpretable systems for fairness (Mehrabi et al. 2021), value alignment (Gabriel 2020), and human–machine interaction (Zhang and Zhang 2022). To better understand how deep models work internally, researchers (Dwivedi et al. 2023) have proposed a range of interpretability methods, such as case-based reasoning (Li et al. 2018; Zhu et al. 2025) and disentangled neural networks (Kim et al. 2018). Among them, concept bottleneck models (CBMs) (Koh et al. 2020) enhance model transparency by explicitly introducing an intermediate concept layer, where the model first predicts human-interpretable concepts before making the final task prediction. Given training data annotated by experts, a standard CBM architecture consists of two components: a concept predictor, which maps the input into a bottleneck layer where each unit represents a human-understandable concept; and a linear classifier, which uses these predicted concepts to produce the final output. By disentangling high-level concepts from raw inputs, CBMs allow users to understand and directly intervene in the model's decision process (Chauhan et al. 2023).

CBMs require large-scale expert-labeled concepts to cover a sufficient range of data features, resulting in substantial costs and limited scalability. To address this issue, recent studies (Yuksekgonul, Wang, and Zou 2022; Oikarinen et al. 2023; Yang et al. 2023; Tang et al. 2025) use pre-trained vision–language models (VLMs) to eliminate the need for manual annotation. Specifically, a VLM-based CBM first uses a large language model (LLM) (Brown et al. 2020) to generate a pool of candidate concepts, and then employs foundation models (*e.g.*, CLIP (Radford et al. 2021)) to compute similarity scores between inputs and each concept in the pool as concept values. This approach enables training without expert annotations and allows any black-box classifier to be converted into a transparent white-box model.

VLM-based CBMs use the powerful alignment capabilities of VLM to automatically construct concept pools from images, enhancing the scalability of concept acquisition. However, these concept sets are typically fixed during both training and inference, which limits the model's flexibility and adaptability to new contexts. In many real-world scenarios, such as medicine, newly discovered biomarkers need to be incorporated into the concept space. Likewise, the rapid iteration of vision-language foundation models often shifts the underlying semantic representations, necessitating updates to the concept embeddings. Therefore, the fixed concept pool must be redefined and re-aligned, requiring costly end-to-end retraining. This rigidity severely constrains the practical deployment of existing CBMs, particularly in real-world applications where domain knowledge evolves continuously and foundation models are frequently updated (Zhang et al. 2025b).

To address these challenges, we propose Flexible Concept Bottleneck Model (FCBM), which extends existing VLM-based CBMs by allowing the concept pool to be modi-

[*]These authors contributed equally.
[†]Corresponding authors.

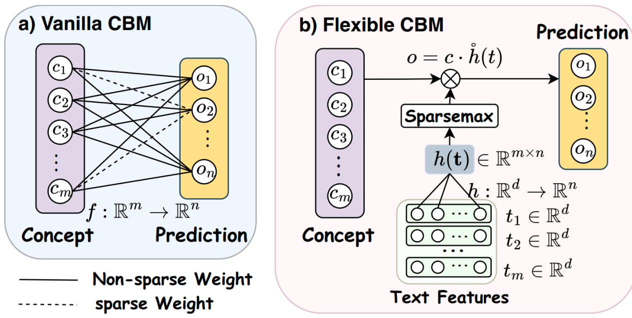


Figure 1: Difference between the vanilla CBM and FCBM.

fied dynamically, as illustrated in Fig. 1. Unlike previous CBM variants that rely on a fixed concept-to-label mapping, FCBM employs a hypernetwork (Ha, Dai, and Le 2016) to disentangle concept representations from their contribution weights. Equipped with the hypernetwork, FCBM incorporates new concepts and generates adapted weights dynamically, thereby supporting the integration of evolving knowledge. To maintain interpretability and mitigate concept dilution, we introduce a sparsity strategy (Martins and Astudillo 2016) to recalibrate the balance between prediction accuracy and concept activation. This encourages sparse explanations, ensuring that only a concise set of key concepts is activated for each prediction. **The main contributions are as follows**:
1) **Flexible and scalable framework.** We propose a novel framework named **F**lexible **C**oncept **B**ottleneck **M**odel (FCBM). Unlike existing approaches using a fixed concept pool, FCBM adapts to evolving knowledge when involving new concepts, which is more scalable and adaptable in scenarios where different concept subsets are preferable or when users want to leverage more advanced LLMs.
2) **Dynamic concept modeling with sparse and interpretable selection.** To enable flexible concept usage, FCBM introduces a hypernetwork to generate weights dynamically for new or updated concepts. In addition, we design a sparsemax module with a learnable temperature, which promotes sparsity in concept selection and ensures that only the most informative concepts are activated, preserving interpretability while avoiding concept redundancy.
3) **Comparable accuracy to state-of-the-art baselines.** We execute extensive experiments to show that FCBM is comparable to state-of-the-art baselines with two backbones and outperforms all baselines on more than half of the benchmarks. Moreover, the model generalizes well to unseen concepts with just a single epoch of fine-tuning, demonstrating its strong adaptability and flexibility.

2 Related Work

Concept Bottleneck Models (CBMs). As deep neural networks are increasingly deployed, understanding their internal decision processes has become a key challenge (Doshi-Velez and Kim 2017). A common approach is to align internal representations with human-interpretable concepts (Kim et al. 2018), promoting transparency and trust. While post-hoc methods offer several insights, they often fail to reflect the model's true reasoning (Rudin 2019). This motivated a shift towards methods that embed interpretability directly into the model by design, among which one prominent design is CBM (Koh et al. 2020). CBMs (Koh et al. 2020; Losch, Fritz, and Schiele 2021) and their variants (Kazhdan et al. 2020; Kim et al. 2023) follow a two-stage architecture: they first map input images to a set of expert-defined concepts, and then perform classification based on these concepts via a linear layer. This structure allows predictions to be explained through human-understandable concepts and provides strong interactivity. Beyond interpretable image classification, CBMs have been applied to diverse areas including human–AI interaction (Chauhan et al. 2023), decision-making processes (Das, Chernova, and Kim 2023), and causal inference (Dominici et al. 2025). However, training CBMs requires large expert-labeled concepts, which are costly to obtain. Theoretically, under limited expert supervision, CBMs face a trade-off between interpretability and predictive accuracy (Zhang et al. 2025a). Although several methods have been proposed to mitigate this trade-off, such as CBM-AUC (Sawada and Nakamura 2022), DCBM (Zhang et al. 2025a), and CEM (Espinosa Zarlenga et al. 2022), they still rely on concept-level supervision and are not applicable in scenarios where expert-labeled concepts are unavailable or extremely limited.

VLM-based CBMs. Recent studies have also explored constructing CBMs by leveraging the rich knowledge reserve of LLMs (Berrios et al. 2023) and the semantic understanding capabilities of VLMs (Wang et al. 2024; Zang et al. 2025), to reduce the reliance on concept annotations. The Post-hoc CBM (Yuksekgonul, Wang, and Zou 2022) leverages external knowledge such as ConceptNet (Speer, Chin, and Havasi 2017) to identify meaningful concepts and align them with input features through CLIP (Radford et al. 2021). Label-free CBM (Oikarinen et al. 2023) first generates the concept set by prompting GPT-3 (Brown et al. 2020), and the mapping from image embeddings to concept values in the CLIP space is optimized using a cosine cubed loss. In contrast, Labo (Yang et al. 2023) uses submodular optimization (Bach 2010) to filter candidate concepts, ensuring the discriminative representations and semantic diversity. Recently, Panousis, Ienco, and Marcos proposed a hierarchical reasoning strategy to improve the predictive performance and interoperability. VLG-CBM (Srivastava, Yan, and Weng 2024) leverages an open-domain grounded object detection model (Liu et al. 2024b) to generate localized and visually interpretable concept annotations. By filtering out non-visual concepts using visual signals from open-vocabulary object detectors, it enhances both transparency and explainability while improving prediction performance. In addition, OpenCBM (Tan, Zhou, and Chen 2024) aims to support flexible, open-vocabulary concepts specified by users at test time. While offering a degree of flexibility, its capabilities are limited to adding or removing concepts, and it cannot accommodate complete updates to the concept pool, a requirement in scenarios where foundation models are updated.

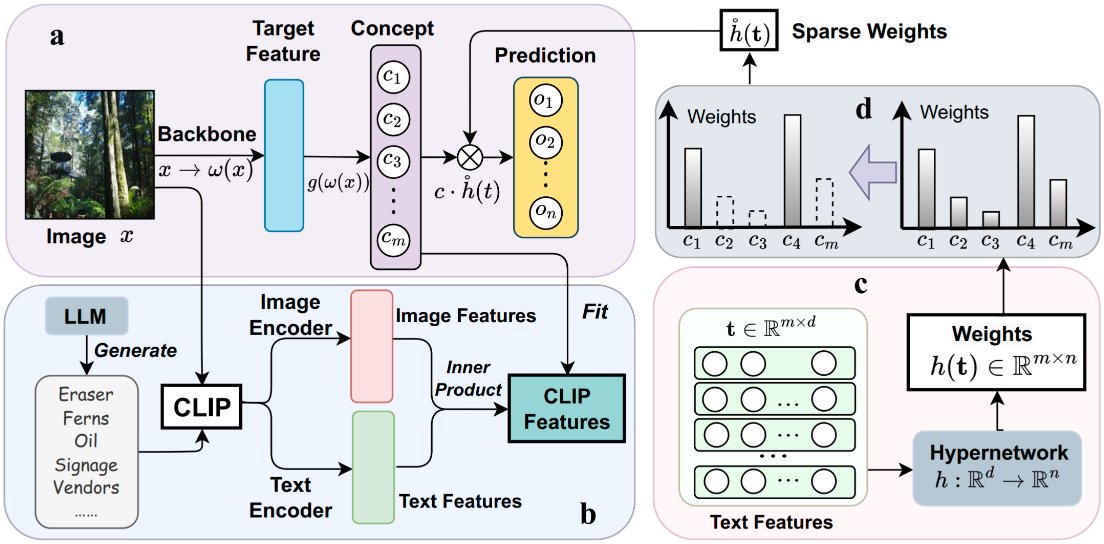


Figure 2: The pipeline of FCBM consists of four key components: a) A two-stage learning framework; b) Concept sets generated by LLMs, which are used to form CLIP-derived features; c) A hypernetwork that generates weights based on text features; and d) A tailored sparsemax module that enforces sparsity in the weights during both training and inference.

3 Methodology

Our proposed framework FCBM is illustrated in Fig. 2, which consists of four main components: 1) a two-stage learning framework (Sec. 3.1), 2) concept set generation via LLMs and CLIP feature formation (Sec. 3.2), 3) a hypernetwork for dynamic concept-to-label mapping (Sec. 3.3), and 4) a tailored sparsemax module for interpretable concept selection (Sec. 3.4).

3.1 Preliminaries and Framework of FCBM

In Concept Bottleneck Models (CBMs), three core components are involved: input images, concepts, and labels, which are represented as vectors $\boldsymbol{x_i}$, $\boldsymbol{c_i}$, and $\boldsymbol{y_i}$ $(i = 1, 2, \cdots, N)$, respectively. CBMs typically follow a two-stage learning framework: a concept predictor $g : \mathcal{X} \to \mathcal{C}$ that maps images $\boldsymbol{x}$ to concepts $\boldsymbol{c}$, followed by a label predictor $f : \mathcal{C} \to \mathcal{Y}$ that maps concepts $\boldsymbol{c}$ to final predictions $\boldsymbol{y}$.

In vanilla CBMs (Koh et al. 2020), ground-truth concepts $\boldsymbol{c}$ are assumed to be annotated by human experts, which is both time-consuming and challenging. To address this issue, VLM-based CBMs (Oikarinen et al. 2023) automatically generate concepts by combining capabilities of VLMs and LLMs (see Sec. 3.2). By eliminating the dependence on expert-annotated concepts, CBMs can be trained on a much broader range of datasets, such as Places365 (Zhou et al. 2017) and ImageNet (Deng et al. 2009).

FCBM also follows a two-stage learning framework but introduces several design differences. Specifically, given images $\boldsymbol{v}$, a backbone model ω is used to extract target features $\omega(\boldsymbol{x})$. Let $\boldsymbol{q} \triangleq g \circ \omega(\boldsymbol{x})$, then the first stage optimizes the concept predictor by minimizing the following objective:

$$g^* = \arg\min_g \sum_{j=1}^{m} \left[-\mathrm{sim}(\boldsymbol{c}_{:,j}, \boldsymbol{q}_{:,j}) \right], \tag{1}$$

where m is the number of concepts, $\mathrm{sim}(\cdot, \cdot)$ is the *cosine-cubed* similarity introduced in LF-CBM (Oikarinen et al. 2023), and $\boldsymbol{c}$ represents the CLIP-derived features (Sec. 3.2).

In most VLM-based CBMs (Oikarinen et al. 2023; Srivastava, Yan, and Weng 2024), the second stage simply aims to minimize the Cross Entropy (CE) between the predicted outcome $f \circ g^* \circ \omega(\boldsymbol{x})$ and labels $\boldsymbol{y}$, where $f : \mathcal{C} \to \mathcal{Y}$ is a linear-like projection. However, this formulation restricts f to a fixed number of concepts in $\mathcal{C}$, which is not well-suited for dynamic concepts. In contrast, FCBM introduces a hypernetwork h that generates weights $h(\boldsymbol{t})$ from text features $\boldsymbol{t}$ and uses a tailored sparsemax operation $\mathcal{S}^{\tau}_{\max}$ with temperature τ to obtain the sparse weights $\mathring{h}(\boldsymbol{t}) \triangleq \mathcal{S}^{\tau}_{\max}(h(\boldsymbol{t}))$.

Thus, the second learning stage can be written as:

$$h^* = \arg\min_h \mathcal{L}, \quad \mathcal{L} = \sum_{i=1}^{N} \mathrm{CE}(g^* \circ \omega(\boldsymbol{x_i}) \cdot \mathring{h}(\boldsymbol{t}), \boldsymbol{y_i}), \tag{2}$$

where the $\mathring{h}^*(\boldsymbol{t})$ is used for the classification tasks.

3.2 Concept Generation and CLIP Features

Concepts are generated through large language models (LLMs), utilizing the same prompt templates introduced in LF-CBM (Oikarinen et al. 2023), where GPT-3 (Brown et al. 2020) serves as the base model. To ensure comparability, we adopt the same generated concepts for the main experiments. Additionally, we include concepts generated by more advanced LLMs, such as DeepSeek-V3 (Liu et al. 2024a) and GPT-4o (OpenAI 2024), to demonstrate the zero-shot generalization capability of FCBM with respect to concepts.

A vision-language model (*e.g.*, CLIP (Radford et al. 2021)) is then employed as both an image encoder and a text encoder to extract image features $\boldsymbol{z} \in \mathbb{R}^{N \times d}$ from images and text features $\boldsymbol{t} \in \mathbb{R}^{m \times d}$ from concept texts, where

Algorithm 1: Sparsemax with temperature τ ($\mathcal{S}^{\tau}_{\max}$)

Input: Input vector $\boldsymbol{s}$, temperature τ.
Output: $\tilde{s}$, where $\tilde{s}_i = [\boldsymbol{s}_i - \xi(\boldsymbol{s})]_+$.

1: Sort $\boldsymbol{s}$: $\boldsymbol{s}_{(1)} \geq \boldsymbol{s}_{(2)} \cdots \geq \boldsymbol{s}_{(m)}$.
2: Find $k(\boldsymbol{s}) \triangleq \max \left\{ k \in [m] | \tau + k\boldsymbol{s}_{(k)} > \sum_{j \leq k} \boldsymbol{s}_{(j)} \right\}$.
3: Let $\xi(\boldsymbol{s}) = \frac{(\sum_{j \leq k(\boldsymbol{s})} \boldsymbol{s}_{(j)}) - \tau}{k(\boldsymbol{s})}$.

N, m, and d represent the number of images, the number of concepts, and the feature dimensions, respectively. Both modalities are encoded into a shared feature space of dimension d, as they are pretrained through a contrastive learning mechanism (Desai and Johnson 2021). As a result, when a concept text accurately describes an image's characteristics, their corresponding features exhibit high similarity in the joint embedding space.

The CLIP-derived features are computed by taking the inner product between $\boldsymbol{z}$ and $\boldsymbol{t}$, forming $\boldsymbol{c} = \boldsymbol{z} \cdot \boldsymbol{t}^\top \in \mathbb{R}^{N \times m}$. Here, each element $\boldsymbol{c_i} \in \mathbb{R}^m$ represents the concept vector for the image $\boldsymbol{x_i}$.

3.3 Hypernetwork

To better extract informative details from concepts and support a dynamic number of concepts, we introduce a hypernetwork $h : \mathbb{R}^d \to \mathbb{R}^n$, which maps the text feature dimension to the class dimension, where n represents the number of classes. Notably, the scale of h is independent of the number of concepts m, making it scalable to varying concept set sizes. The output of hypernetwork $h(\boldsymbol{t}) \in \mathbb{R}^{m \times n}$ shares the same shape as the weight matrix used in the linear projection $f : \mathcal{C} \to \mathcal{Y}$ in vanilla VLM-based CBMs (Oikarinen et al. 2023). Intuitively, $h(\boldsymbol{t})$ also acts as the weights, determining the contribution of each concept to the final predictions.

To preserve the zero-shot generalization capability with respect to concepts, we align the feature distributions during training and inference. Specifically, given the text features $\boldsymbol{t} \in \mathbb{R}^{m \times d}$ during training, we compute their mean $\bar{\boldsymbol{t}} \in \mathbb{R}^d$ and standard deviation $\sigma_{\boldsymbol{t}} \in \mathbb{R}^d$, as well as the mean $\bar{h}(\boldsymbol{t}) \in \mathbb{R}^n$ and standard deviation $\sigma_{h(\boldsymbol{t})} \in \mathbb{R}^n$ of the hypernetwork's output $h(\boldsymbol{t}) \in \mathbb{R}^{m \times n}$. During inference, given the new text features $\boldsymbol{t}'$ generated from a new set of concepts, the weights $\tilde{h}(\boldsymbol{t}')$ are obtained as follows:

$$\tilde{\boldsymbol{t}}' \triangleq \frac{\sigma_{\boldsymbol{t}}}{\sigma_{\boldsymbol{t}'}}(\boldsymbol{t}' - \overline{\boldsymbol{t}'}) + \bar{\boldsymbol{t}}, \quad \tilde{h}(\boldsymbol{t}') \triangleq \frac{\sigma_{h(\boldsymbol{t})}}{\sigma_{h(\tilde{\boldsymbol{t}}')}}(h(\tilde{\boldsymbol{t}}') - \bar{h}(\tilde{\boldsymbol{t}}')) + \bar{h}(\boldsymbol{t}), \tag{3}$$

where $\overline{\boldsymbol{t}'}$ and $\sigma_{\boldsymbol{t}'}$ are the mean and standard deviation of $\boldsymbol{t}'$, and $\bar{h}(\tilde{\boldsymbol{t}}')$ and $\sigma_{h(\tilde{\boldsymbol{t}}')}$ are the mean and standard deviation of $h(\tilde{\boldsymbol{t}}')$. Intuitively, Eq. 3 ensures that the distribution of the final weights $\tilde{h}(\boldsymbol{t}')$ is aligned with the distribution of $h(\boldsymbol{t})$ in both the text feature dimension and the weight dimension.

3.4 Sparsemax with a Learnable Temperature

The weights $h(\boldsymbol{t})$ generated by the hypernetwork are generally non-sparse, which may hinder interpretability. To address this issue, we introduce a modified sparsemax module based on (Martins and Astudillo 2016), further incorporating a learnable temperature parameter τ to dynamically control the level of sparsity. This module can enforce sparsity in the weight selection by controlling the number of active concepts used for each prediction. Intuitively, sparsemax acts similarly to softmax but generates sparse outputs, enabling the model to focus on the most relevant concepts. The temperature parameter τ adjusts the level of sparsity: higher temperatures lead to fewer active concepts, while lower temperatures increase the number of concepts considered in the prediction. By making τ learnable in the training stage, the model can automatically balance prediction accuracy and interpretability. The overall algorithm of the modified sparsemax is shown in Alg. 1.

To optimize the temperature parameter jointly with other model parameters, we need to compute the gradient of the sparsemax operation with respect to both $\boldsymbol{s}$ and τ. The original sparsemax optimization was proposed in (Martins and Astudillo 2016), where the Jacobian of the transformation given a vector $\boldsymbol{v}$ is expressed as:

$$J_{\text{sparsemax}}(\boldsymbol{s}) \cdot \boldsymbol{v} = \boldsymbol{e} \odot (\boldsymbol{v} - \hat{v}\mathbf{1}), \quad \text{with} \quad \hat{v} := \frac{\sum_{j \in P(\boldsymbol{s})} v_j}{|P(\boldsymbol{s})|}, \tag{4}$$

where $P(\boldsymbol{s})$ represents the support set (*i.e.*, the indices where $\mathcal{S}^{\tau}_{\max}(\boldsymbol{s})_i > 0$), $\odot$ denotes the Hadamard (element-wise) product, and $\mathbf{1}$ is an all-ones vector. $\boldsymbol{e}$ is an indicator vector with $\boldsymbol{s_i} = 1$ if $i \in P(\boldsymbol{s})$ and 0 otherwise. This computation is efficient, requiring only $O(|P(\boldsymbol{s})|)$ operations, especially when the support set is small. The derivation of τ is:

$$\frac{\partial \mathcal{L}}{\partial \tau} = \sum_{i \in P(\boldsymbol{s})} \frac{1}{|P(\boldsymbol{s})|} \cdot \frac{\partial \mathcal{L}}{\partial \tilde{\boldsymbol{s}}_i}. \tag{5}$$

Detailed derivations of Eq. 4 and 5 are in the Appendix.

4 Experiments

The experimental protocols are outlined in Sec. 4.1. Sec. 4.2 presents the main results, comparing results under a similar average number of effective concepts (NEC). To show the ability of FCBM's concept generalization, Sec. 4.3 shows the zero-shot prediction accuracy using newly-generated concepts. Sec. 5 provides ablation studies.

4.1 Experimental Protocols

Datasets. The experiments use five public datasets: CIFAR10 (Krizhevsky, Nair, and Hinton 2009), CIFAR100 (Krizhevsky, Nair, and Hinton 2009), CUB (Welinder et al. 2010), Places365 (Zhou et al. 2017), and ImageNet (Deng et al. 2009), consistent with previous works (Oikarinen et al. 2023; Zhang et al. 2025a). CIFAR10 and CIFAR100 are common image classification datasets containing 50,000 images with 10 and 100 classes, respectively. CUB is a fine-grained bird-species classification dataset with 4,900 training images and 200 classes. Places365 and ImageNet are large datasets with 1-2 million training images and 365 and 1,000 classes, respectively. The concepts used by FCBM are generated by LLMs using three

Backbone	Method	Average NEC	Dataset				
			CIFAR10	CIFAR100	CUB	Places365	ImageNet
ResNet50	Standard	-	88.55 ± 0.02	70.19 ± 0.06	71.00 ± 0.16	53.28 ± 0.04	73.14 ± 0.04
	Standard (sparse)	29.84	82.11 ± 0.00	57.54 ± 0.01	53.34 ± 0.01	44.00 ± 0.00	57.03 ± 0.01
	PCBM	32.37	76.43 ± 0.02	56.24 ± 0.06	58.39 ± 0.06	42.26 ± 0.35	61.38 ± 0.22
	LF-CBM	27.37	**86.16 ± 0.05**	$\underline{64.62 \pm 0.06}$	56.91 ± 0.14	$\underline{48.88 \pm 0.05}$	$\underline{66.03 \pm 0.05}$
	CF-CBM	28.46	85.42 ± 0.07	64.31 ± 0.25	**64.23 ± 0.29**	46.39 ± 0.34	65.95 ± 0.27
	FCBM (ours)	28.85	$\underline{85.59 \pm 0.29}$	**64.77 ± 0.41**	$\underline{63.46 \pm 0.41}$	**49.13 ± 0.67**	**66.34 ± 0.34**
ViT-L/14	Standard	-	98.02 ± 0.03	86.99 ± 0.09	85.22 ± 0.05	55.66 ± 0.06	84.11 ± 0.05
	Standard (sparse)	28.92	96.61 ± 0.00	80.28 ± 0.00	72.07 ± 0.00	44.90 ± 0.01	72.96 ± 0.13
	PCBM	32.56	92.13 ± 0.05	74.29 ± 0.13	68.25 ± 0.32	43.18 ± 0.36	65.79 ± 0.53
	LF-CBM	27.85	$\underline{97.18 \pm 0.01}$	81.98 ± 0.02	75.44 ± 0.04	$\underline{50.51 \pm 0.02}$	$\underline{79.70 \pm 0.03}$
	CF-CBM	28.56	96.35 ± 0.05	$\underline{82.33 \pm 0.08}$	$\underline{79.56 \pm 0.37}$	48.55 ± 0.28	79.16 ± 0.44
	FCBM (ours)	28.08	**97.21 ± 0.06**	**83.63 ± 0.20**	**80.52 ± 0.27**	**51.39 ± 0.11**	**80.62 ± 0.32**

Table 1: We compare prediction accuracy across five datasets using two backbones: ResNet50 and ViT-L/14. The average NEC is controlled to approximately 30, except for the *Standard* model, to ensure a fair comparison. The best and the second best results, excluding the *Standard* model, for each dataset and backbone are shown in **bold** and $\underline{\text{underline}}$, respectively.

types of prompts introduced in (Oikarinen et al. 2023): including the key features of a class, features commonly associated with a class, and the superclasses of a class.

Implementation Details. Experiments were run using an AMD EPYC 7402 24-Core Processor, an NVIDIA GeForce RTX 4090, and 512GB RAM. The batch size was set to 50,000 for each dataset, with the Adam (Kingma and Ba 2015) optimizer and a learning rate of 0.001 for both objectives in Eq.1 and Eq.2. The maximum number of epochs was set to 5,000 for the three small-scale datasets (CIFAR10, CIFAR100, and CUB), and 500 for the two large-scale datasets (ImageNet and Places365). This setting was empirically determined through a grid search.

Metrics. The prediction accuracy is used as the main metric. To make a fairer comparison, we incorporate the average number of effective concepts (NEC), first introduced in (Srivastava, Yan, and Weng 2024) to keep similar sparsity of concept usages:

$$\mathrm{NEC}(W) = \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbb{I}\{W_{ij} \neq 0\}, \tag{6}$$

where $\mathbb{I}[\cdot]$ is an indicator returning $1/0$ with correct/wrong inputs. W is the weights of the linear function in vanilla VLM-based CBMs (Oikarinen et al. 2023), and is $\mathring{h}(\boldsymbol{t})$ in our approach. In the main result, we set NEC near 30, and we also conduct ablation studies in terms of NEC in Sec. 5. Experiments are repeated three times with different seeds, reporting the average performance with standard deviation.

Baselines. Six approaches are utilized as baselines, including the sparse and non-sparse standard model (linear projection from the concept layer to the prediction layer), PCBM (Yuksekgonul, Wang, and Zou 2022), LF-CBM (Oikarinen et al. 2023), and CF-CBM (Panousis, Ienco, and Marcos 2024). Note that all baselines are with similar NEC, except the non-sparse standard model.

Other Settings. Other details are shown in the Appendix.

4.2 Main Results

In this section, we compare the prediction accuracy of different approaches using two backbone models, i.e., ResNet50 (He et al. 2016) and CLIP-ViT-L/14 (Radford et al. 2021). The NEC for all approaches on each dataset is controlled to be approximately 30.

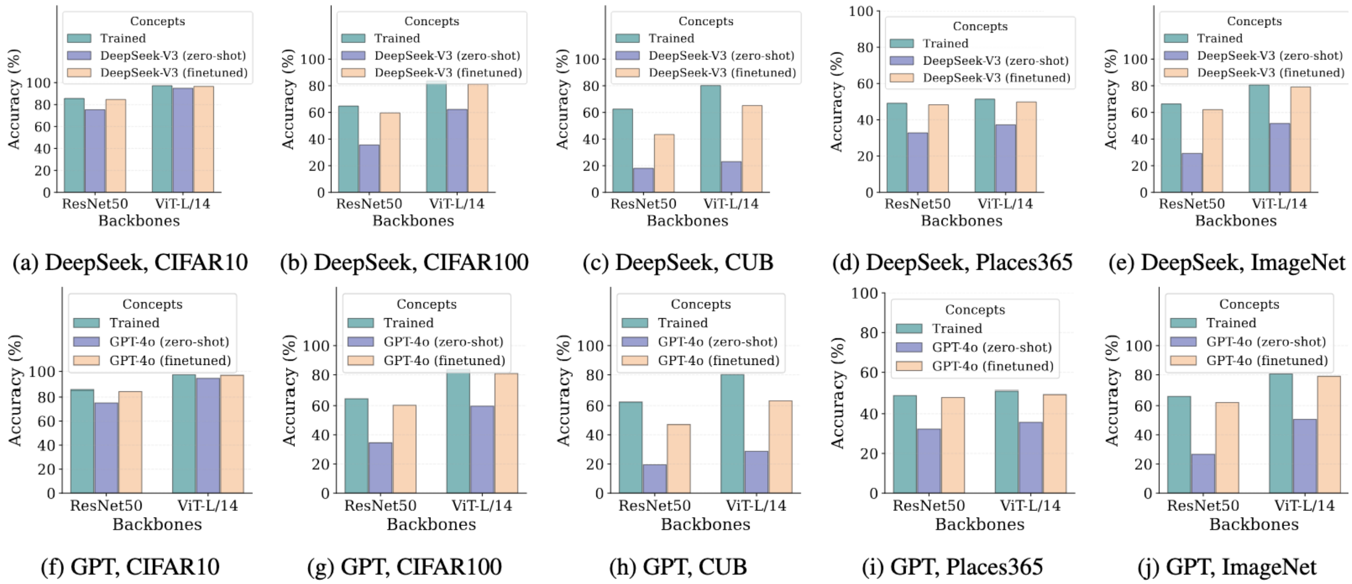
From a general perspective, the standard model achieves the highest accuracy across all datasets. This is expected, as it incorporates the most significant number of informative concepts for making predictions. However, its sparse version performs poorly, emphasizing the challenge of maintaining high accuracy while ensuring sparsity.

Among the other four baselines, the accuracy of PCBM (Yuksekgonul, Wang, and Zou 2022) is not particularly remarkable and is roughly comparable to that of the standard sparse model. Moreover, note that PCBM was not originally designed for larger datasets. In contrast, LF-CBM (Oikarinen et al. 2023) and CF-CBM (Panousis, Ienco, and Marcos 2024) exhibit similarly strong accuracy with different design strategies, illustrating possible empirical upper bounds for accuracy when using sparse concepts.

Compared to these baselines, FCBM achieves the highest accuracy in most cases and ranks second in the remaining two, showing performance that matches or exceeds these state-of-the-art methods. This result suggests that replacing the linear function with a hypernetwork-sparsemax module does not degrade the final prediction quality. Beyond prediction accuracy, the next section will discuss zero-shot generalization and reveal the benefits of explicit concept learning.

4.3 Adaptability Across Unseen Concept Pools

To demonstrate FCBM's zero-shot generalization capability in terms of concepts, we follow the generation configuration proposed in (Oikarinen et al. 2023) and regenerate the concepts using two widely used LLMs: DeepSeek-V3 (Liu et al.



(a) DeepSeek, CIFAR10 (b) DeepSeek, CIFAR100 (c) DeepSeek, CUB (d) DeepSeek, Places365 (e) DeepSeek, ImageNet

(f) GPT, CIFAR10 (g) GPT, CIFAR100 (h) GPT, CUB (i) GPT, Places365 (j) GPT, ImageNet

Figure 3: Adaptability of different concept pools across five datasets using ResNet50 and ViT-L/14 backbones. For each subfigure, we test the accuracy of FCBM using three types of concepts: the trained concepts, the LLM-generated concepts without training (zero-shot), and the LLM-generated concepts with only one epoch of fine-tuning (finetuned). DeepSeek-V3 (a-e) and GPT-4o (f-j) are employed as the LLM backbones, respectively.

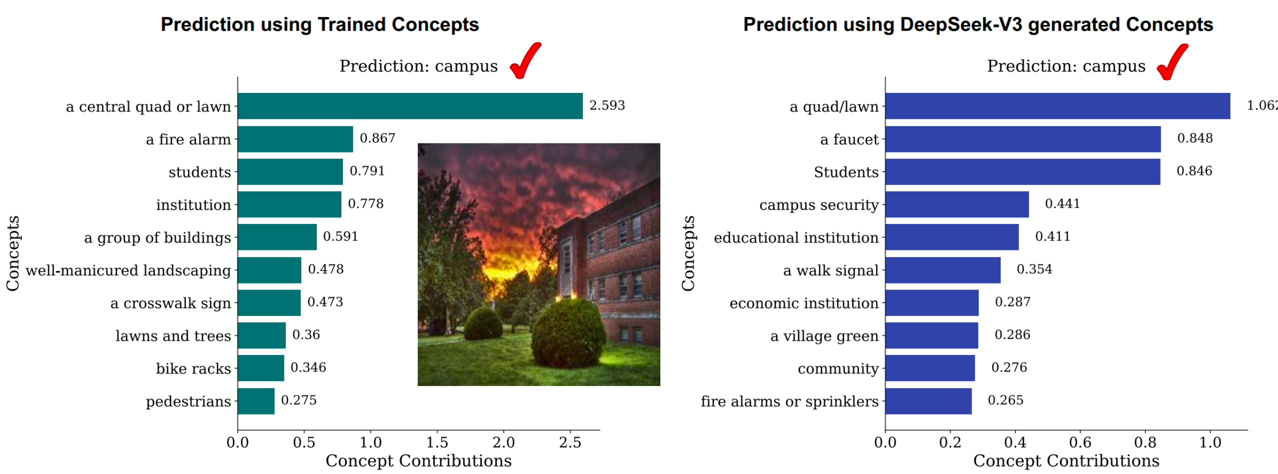
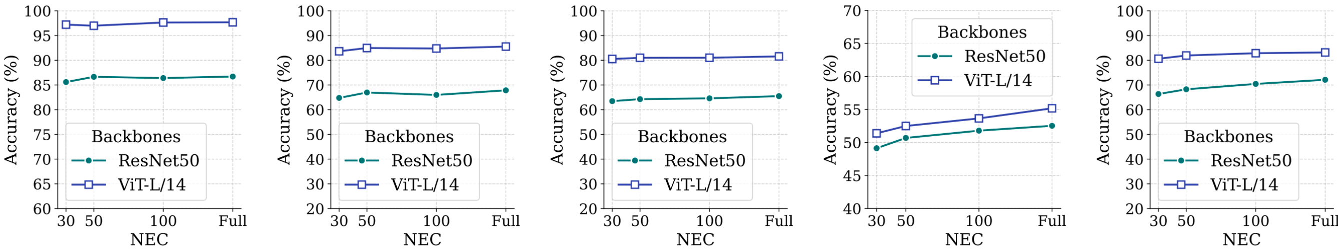


Figure 4: The adaptability of FCBM across different concept pools on the *Places365* dataset. The left histograms illustrates the prediction made using the trained concepts, while the right histograms show the prediction based on the DeepSeek-V3-generated concepts. The left image belong to the ground-truth class, i.e., 'campus'.

2024a) and GPT-4o (OpenAI 2024).

FCBM demonstrates varying degrees of zero-shot generalization and fine-tuning adaptability across different concept pools (Fig. 3). Our experimental setting is particularly challenging: we replace the entire concept pool with a new set of concepts, reflecting a realistic scenario in the era of rapidly evolving foundation models. Despite this difficulty, FCBM possesses a degree of zero-shot generalization, successfully transferring to new concepts without any fine-tuning, suggesting the learned structure within FCBM captures generalizable patterns in the semantic space. Remarkably, FCBM can adapt to the entirely new concept set with only a single epoch of fine-tuning, highlighting its ability to quickly incorporate evolving domain knowledge.

Fig. 4 shows a example of concept contributions using the trained concepts (left) and the concepts generated by DeepSeek-V3 (right), where the image used for prediction belongs to the 'campus' class in the *Places365* dataset. Overall, the highly contributing concepts are similar, such as 'a central quad or lawn' versus 'a quad/lawn', 'students' versus 'Students', and 'institution' versus 'educational institution'. However, in the left prediction, the concept 'a central



(a) CIFAR10 (b) CIFAR100 (c) CUB (d) Places365 (e) ImageNet

Figure 5: Sparsity analysis. We test the accuracy of FCBM with NEC = 30, 50, 100, and full concepts across five datasets using ResNet50 and ViT-L/14 backbones.

Backbone	Method	CIFAR10			CIFAR100			CUB			Places365			ImageNet		
		Tr.	DS	GPT	Tr.	DS	GPT	Tr.	DS	GPT	Tr.	DS	GPT	Tr.	DS	GPT
ResNet50	Hard#	69.17	54.00	47.56	63.01	17.71	16.13	50.95	6.82	9.02	**49.85**	9.77	8.86	61.50	17.79	16.92
	FCBM$\setminus\tau$#	81.27	62.62	59.37	49.80	29.13	31.89	29.45	10.97	12.38	33.95	22.98	23.61	39.60	17.95	16.60
	FCBM	**85.59**	**75.32**	**75.09**	**64.77**	**35.57**	**34.54**	**63.92**	**18.07**	**19.43**	49.13	**32.79**	**32.27**	**66.34**	**29.23**	**26.69**
ViT-L/14	Hard#	**97.27**	78.78	73.10	65.15	23.61	24.68	68.15	9.14	12.87	50.37	11.80	11.60	75.22	15.07	14.36
	FCBM$\setminus\tau$#	89.05	75.58	78.13	62.42	38.54	37.30	40.00	18.00	14.40	34.97	25.58	24.41	49.13	23.65	22.55
	FCBM	97.21	**94.89**	**94.19**	**83.63**	**62.27**	**59.67**	**80.52**	**23.06**	**28.69**	**51.39**	**37.24**	**35.70**	**80.62**	**51.70**	**50.56**

* Tr. = Using trained concepts, DS = Using DeepSeek-V3-generated concepts, GPT = Using GPT-4o-generated concepts.
# Hard: forcibly selecting a group of most effective concepts. FCBM$\setminus\tau$: FCBM using the default unlearnable temperature.

Table 2: Accuracy of FCBM with different modules across five datasets using three groups of concepts. The best results for each backbone, dataset, and concept group are shown in **bold**.

quad or lawn' contributes the most, while in the right prediction, based on the DeepSeek-V3-generated concepts, the contributions are more dispersed. This dispersal may suggest that the concepts used in the right prediction do not fully align with those in the training set, which could potentially lead to classification errors. An additional case on the *CUB* dataset can be seen in the Appendix.

To summarize, FCBM demonstrates the potential of zero-shot generalization in terms of concepts in VLM-based CBMs. However, we suggest that users exercise caution when using LLM-generated concepts for fine-grained categories or highly specialized datasets.

5 Ablation Studies

Sparsity. We evaluate the accuracy of FCBM with different numbers of effective concepts (NEC), including 30, 50, 100, and full concepts (all available concepts), across five datasets. As shown in Fig. 5, the accuracy increases slightly as the number of NECs increases, but overall it remains stable. The most noticeable improvement is observed on the *Places365* dataset, indicating the higher prediction difficulty for this dataset.

Sparsemax and Temperature. In Table 2, we evaluate the accuracy of FCBM with three different modules: FCBM without sparsemax (denoted as 'Hard', using hard truncation instead), FCBM without a learnable temperature (denoted as 'FCBM$\setminus\tau$'), and the full version of FCBM. Two backbones (ResNet50 and ViT-L/14) and three groups of concepts (trained concepts, DeepSeek-V3 generated concepts, and GPT-4o generated concepts) are also included in this analysis. The hard truncation refers to the process of forcibly selecting the 30 most effective concepts (NEC).

From a general perspective, the full version of FCBM outperforms other methods in most cases, particularly in terms of zero-shot generalization performance. Although 'Hard' occasionally achieves good results using trained concepts, its zero-shot generalization performance is the weakest, indicating that hard truncation does not effectively preserve concept understanding. Furthermore, 'FCBM\temp.' lacks the ability to control sparsity due to the absence of a learnable temperature. As a result, the sparsity is typically very low, making it difficult to select effective concepts.

6 Conclusion and Outlook

We proposed the Flexible Concept Bottleneck Model (FCBM), which addresses key challenges in CBMs by allowing dynamic adaptation of the concept pool. Through experiments on five datasets, FCBM demonstrated comparable performance to state-of-the-art methods, achieving competitive accuracy with a similar sparsity. Moreover, FCBM showed promising zero-shot generalization, successfully handling unseen concepts. Future work includes evaluating FCBM in more complex, real-world scenarios. It is interesting to investigate whether novel biomarkers can be seamlessly integrated into FCBM without sacrificing interpretability or predictive performance.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (No. 62506367). R.Z. would like to acknowledge the supported by the China Postdoctoral Science Foundation under Grant Number 2025M771582 and the Postdoctoral Fellowship Program of CPSF under Grant Number GZB20250408. Additional experimental details and full appendices are available in the arXiv version of this paper at https://arxiv.org/abs/2511.06678.

References

Bach, F. 2010. Convex analysis and optimization with submodular functions: a tutorial. *arXiv preprint arXiv:1010.4207*.

Berrios, W.; Mittal, G.; Thrush, T.; Kiela, D.; and Singh, A. 2023. Towards language models that can see: Computer vision through the lens of natural language. *arXiv preprint arXiv:2306.16410*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901.

Chauhan, K.; Tiwari, R.; Freyberg, J.; Shenoy, P.; and Dvijotham, K. 2023. Interactive concept bottleneck models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 5948–5955.

Das, D.; Chernova, S.; and Kim, B. 2023. State2explanation: Concept-based explanations to benefit agent learning and user understanding. In *Advances in Neural Information Processing Systems*, volume 36, 67156–67182.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. Ieee.

Desai, K.; and Johnson, J. 2021. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11162–11173.

Dominici, G.; Barbiero, P.; Zarlenga, M. E.; Termine, A.; Gjoreski, M.; Marra, G.; and Langheinrich, M. 2025. Causal Concept Graph Models: Beyond causal opacity in deep learning. In *International Conference on Learning Representations*.

Doshi-Velez, F.; and Kim, B. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Dwivedi, R.; Dave, D.; Naik, H.; Singhal, S.; Omer, R.; Patel, P.; Qian, B.; Wen, Z.; Shah, T.; Morgan, G.; et al. 2023. Explainable AI (XAI): Core ideas, techniques, and solutions. *ACM Computing Surveys*, 55(9): 1–33.

Espinosa Zarlenga, M.; Barbiero, P.; Ciravegna, G.; Marra, G.; Giannini, F.; Diligenti, M.; Shams, Z.; Precioso, F.; Melacci, S.; Weller, A.; et al. 2022. Concept embedding models: Beyond the accuracy-explainability trade-off. *Advances in Neural Information Processing Systems*, 35: 21400–21413.

Gabriel, I. 2020. Artificial intelligence, values, and alignment. *Minds and Machines*, 30(3): 411–437.

Ha, D.; Dai, A.; and Le, Q. V. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Kazhdan, D.; Dimanov, B.; Jamnik, M.; Liò, P.; and Weller, A. 2020. Now You See Me (CME): Concept-based model extraction. In *Conference on Information and Knowledge Management (Workshops)*.

Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; Viegas, F.; et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, 2668–2677. PMLR.

Kim, E.; Jung, D.; Park, S.; Kim, S.; and Yoon, S. 2023. Probabilistic concept bottleneck models. In *International Conference on Machine Learning*, 16521–16540. PMLR.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Koh, P. W.; Nguyen, T.; Tang, Y. S.; Mussmann, S.; Pierson, E.; Kim, B.; and Liang, P. 2020. Concept bottleneck models. In *International Conference on Machine Learning*, 5338–5348. PMLR.

Krizhevsky, A.; Nair, V.; and Hinton, G. 2009. Cifar-10 and cifar-100 datasets. *URl: https://www.cs.toronto.edu/kriz/cifar.html*, 6(1): 1.

Li, O.; Liu, H.; Chen, C.; and Rudin, C. 2018. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. 2024b. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, 38–55. Springer.

Losch, M.; Fritz, M.; and Schiele, B. 2021. Semantic Bottlenecks: Quantifying and improving inspectability of deep representations. *International Journal of Computer Vision*, 129(11): 3136–3153.

Martins, A.; and Astudillo, R. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, 1614–1623. PMLR.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6): 1–35.

Oikarinen, T.; Das, S.; Nguyen, L. M.; and Weng, T.-W. 2023. Label-free concept bottleneck models. In *International Conference on Learning Representations*.

OpenAI. 2024. Hello GPT-4o. Accessed: 2024-05-13.

Panousis, K.; Ienco, D.; and Marcos, D. 2024. Coarse-to-fine concept bottleneck models. *Advances in Neural Information Processing Systems*, 37: 105171–105199.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763. PMLR.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5): 206–215.

Sawada, Y.; and Nakamura, K. 2022. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10: 41758–41765.

Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 31.

Srivastava, D.; Yan, G.; and Weng, L. 2024. Vlg-cbm: Training concept bottleneck models with vision-language guidance. *Advances in Neural Information Processing Systems*, 37: 79057–79094.

Tan, A.; Zhou, F.; and Chen, H. 2024. Explain via any concept: Concept bottleneck model with open vocabulary concepts. In *European Conference on Computer Vision*, 123–138. Springer.

Tang, Q.; Fan, L.; Pagnucco, M.; and Song, Y. 2025. Prototype-based image prompting for weakly supervised histopathological image segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 30271–30280.

Wang, H.; Vasu, P. K. A.; Faghri, F.; Vemulapalli, R.; Farajtabar, M.; Mehta, S.; Rastegari, M.; Tuzel, O.; and Pouransari, H. 2024. Sam-clip: Merging vision foundation models towards semantic and spatial understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3635–3647.

Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; and Perona, P. 2010. The Caltech-UCSD Birds 200. *California Institute of Technology*.

Yang, Y.; Panagopoulou, A.; Zhou, S.; Jin, D.; Callison-Burch, C.; and Yatskar, M. 2023. 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*, 19187–19197.

Yuksekgonul, M.; Wang, M.; and Zou, J. 2022. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*.

Zang, G.; Li, X.; Di, D.; Nie, L.; Zhan, D.; Song, Y.; and Fan, L. 2025. SAGE: A Visual Language Model for Anomaly Detection via Fact Enhancement and Entropy-aware Alignment. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 5030–5039.

Zhang, R.; Du, X.; Yan, J.; and Zhang, S. 2025a. The Decoupling Concept Bottleneck Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(2): 1250–1265.

Zhang, R.; and Zhang, S. 2022. Rethinking influence functions of neural networks in the over-parameterized regime. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 9082–9090.

Zhang, Y.; Chen, L.; Li, S.; Cao, N.; Shi, Y.; Ding, J.; Qu, Z.; Zhou, P.; and Bai, Y. 2025b. Way to specialist: Closing loop between specialized llm and evolving domain knowledge graph. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 1996–2007.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6): 1452–1464.

Zhu, Z.; Fan, L.; Pagnucco, M.; and Song, Y. 2025. Interpretable Image Classification via Non-parametric Part Prototype Learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 9762–9771.