Semi-supervised Concept Bottleneck Models

Lijie Hu[*,1,2], **Tianhao Huang**[*,1,2,3], **Huanyi Xie**[1,2], **Xilin Gong**[1,2,4],
Chenyang Ren[1,2], **Zhengyu Hu**[1,6], **Lu Yu**[5], **Ping Ma**[4], **Di Wang**[†,1,2],
[1]Provable Responsible AI and Data Analytics (PRADA) Lab,
[2]King Abdullah University of Science and Technology,
[3]University of Virginia, [4]University of Georgia, [5]Ant Group, [6]HKUST(GZ)
{lijie.hu, tianhao.huang, huanyi.xie, di.wang}@kaust.edu.sa

Abstract

Concept Bottleneck Models (CBMs) have garnered increasing attention due to their ability to provide concept-based explanations for black-box deep learning models while achieving high final prediction accuracy using human-like concepts. However, the training of current CBMs is heavily dependent on the precision and richness of the annotated concepts in the dataset. These concept labels are typically provided by experts, which can be costly and require significant resources and effort. Additionally, concept saliency maps frequently misalign with input saliency maps, causing concept predictions to correspond to irrelevant input features - an issue related to annotation alignment. To address these limitations, we propose a new framework called SS-CBM (Semi-supervised Concept Bottleneck Model). Our SSCBM is suitable for practical situations where annotated data is scarce. By leveraging joint training on both labeled and unlabeled data and aligning the unlabeled data at the concept level, we effectively solve these issues. We propose a strategy to generate pseudo labels and an alignment loss. Experiments demonstrate that our SSCBM is both effective and efficient. With only 10% labeled data, our model's concept and task accuracy on average across four datasets is only 2.44% and 3.93% lower, respectively, compared to the best baseline in the fully supervised learning setting.

1. Introduction

Deep learning models like ResNet [13] typically feature complex non-linear architectures, making it difficult for end-users to understand and trust their decisions. This lack of interpretability is a significant obstacle to their adoption, especially in critical fields such as healthcare [41] and finance [30], where transparency is crucial. Explain-

able artificial intelligence (XAI) models have been developed to meet the demand for transparency, providing insights into their behavior and internal mechanisms [16–18, 46]. Concept Bottleneck Models (CBMs) [25] are particularly notable among these XAI models for their ability to clarify the prediction process of end-to-end AI models. CBMs introduce a bottleneck layer that incorporates human-understandable concepts. During prediction, CBMs first predict concept labels from original input, then use these predicted concepts in bottleneck layer to determine the final label. This approach results in a self-explanatory decision-making process that users can comprehend.

A major issue in original CBMs is the need for expert labeling, which is costly in practice. Some researchers address this problem through unsupervised learning. For example, [34] proposes a Label-free CBM that transforms any neural network into an interpretable CBM without requiring labeled concept data while maintaining high accuracy. Similarly, Post-hoc Concept Bottleneck models [48] can be applied to various neural networks without compromising performance, preserving interpretability advantages. However, these methods have three issues. First, those unsupervised methods heavily rely on large language models like GPT-3, which have reliability issues [26]. Second, the concepts extracted by these models lack evaluation metrics, undermining their interpretability. Third, the assumption that no concept labels are available is too stringent in practice. In reality, obtaining a small portion of concept labels is feasible and cost-effective. Therefore, we can maximize the use of this small labeled concept dataset. This is the motivation for introducing our framework, which focuses on the semi-supervised setting in CBM.

In this paper, we introduce a framework called the **SS-CBM** (**S**emi-**s**upervised **C**oncept **B**ottleneck **M**odel). Compared to a supervised setting, semi-supervised CBMs have two main challenges. First, obtaining concept embeddings requires concept labels, so we need to generate pseudo labels to obtain these concept embeddings. To achieve this,

---

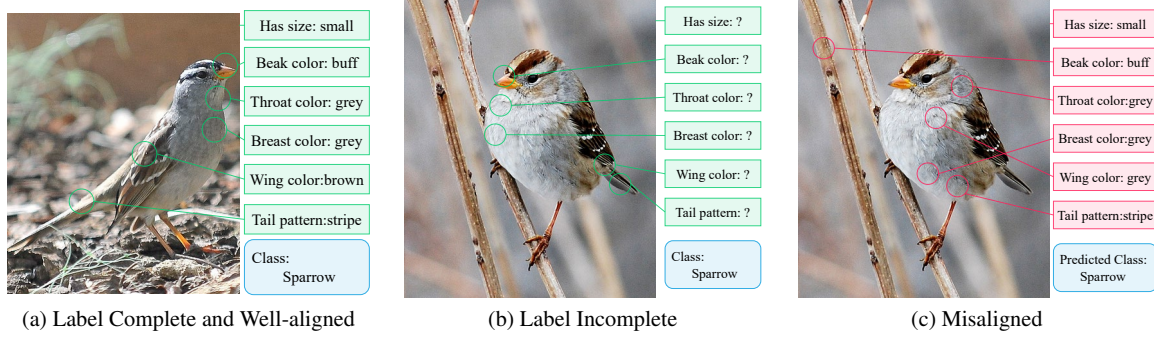*Equal contribution.
†Corresponding author.

Figure 1. (a) A sample of sparrow class with complete concept labels. (b) A sample of sparrow class with incomplete concept labels. (c) A sample of misalignment between input features and concepts resulting from existing CBM methods. Our framework simultaneously utilizes both (a) and (b) types of data and addresses the issue of (c) through an alignment loss.

SSCBM uses a KNN-based algorithm to assign pseudo-concept labels for unlabeled data. Second, while such a simple pseudo-labeling method is effective and has acceptable classification accuracy, we also find that the concept saliency map often misaligns with the input saliency map, meaning concept predictions frequently correspond to irrelevant input features. This misalignment often arises from inaccurate concept annotations or unclear relationships between input features and concepts, which is closely related to the broader issue of annotation alignment. In fact, in the supervised setting, there is a similar misalignment issue [11]. Existing research seeks to improve alignment by connecting textual and image information [15]. However, these methods only focus on supervised settings and cannot be directly applied to our settings, as our pseudo-labels contain noise. Our framework achieves excellent performance in both concept accuracy and concept saliency alignment by leveraging joint training on both labeled and unlabeled data and aligning the unlabeled data at the concept level. To achieve this, we leverage the relevance between the input image and the concept and get other pseudo-concept labels based on these similarity scores. Finally, we align these two types of pseudo-concept labels to give the concept encoder the ability to extract useful information from features while also inheriting the ability to align concept embeddings with the input. See Figure 1 for an illustration. Our contributions can be summarized as follows:

- We propose SSCBM, a framework designed to address both the semi-supervised annotation problem and the concept semantics alignment problem, which hold significant practical importance in real-world applications. SSCBM tackles them in an elegant and efficient manner. To the best of our knowledge, we are the first to tackle these two problems within a single framework, elucidating the behavior of CBMs through semi-supervised alignment.

- Comprehensive experiments demonstrate the superiority of our SSCBM in annotation and concept-saliency align-

ment, indicating its efficiency and effectiveness. Using only 10% labeled data, our model achieves concept and task accuracy on average across the four datasets that are merely 2.44% and 3.93% lower, respectively, than the best baseline under fully supervised learning settings.

2. Related Work

Concept Bottleneck Models. Concept Bottleneck Model (CBM) [25] is an innovative deep-learning approach for image classification and visual reasoning. By introducing a concept bottleneck layer into deep neural networks, CBMs enhance model generalization and interpretability by learning specific concepts. However, CBMs face two primary challenges: their performance often lags behind that of models without the bottleneck layer due to incomplete information extraction, and they rely heavily on laborious dataset annotation. Researchers have explored various solutions to these challenges. [5] extends CBMs into interactive prediction settings by introducing an interaction policy to determine which concepts to label, thereby improving final predictions. [34] addresses CBM limitations by proposing a Label-free CBM, which transforms any neural network into an interpretable CBM without requiring labeled concept data, maintaining high accuracy. Post-hoc Concept Bottleneck models [48] can be applied to various neural networks without compromising performance, preserving interpretability advantages. Related work in the image domain includes studies [12, 19, 23, 24, 29, 37, 38]. In the graph concept field, [31] provides a global interpretation of graph neural networks (GNNs) by mapping graphs into a concept space through clustering and offering a human-in-the-loop evaluation. [3, 44] extend this approach by incorporating both global and local explanations. For local explanations, they define a concept set, with each neuron represented as a vector with Boolean values indicating concept activation. However, existing works rarely consider semi-supervised settings, which are practical in real-world

applications. Our framework addresses these issues effectively.

Semi-supervised Learning. Semi-supervised learning (SSL) combines the two main tasks of machine learning: supervised learning and unsupervised learning [43]. It is typically applied in scenarios where labeled data is scarce. Examples include computer-aided diagnosis [6, 49], medical image analysis [8, 10, 20], and drug discovery [7]. In these cases, collecting detailed annotated data by experts requires considerable time and effort. However, under the assumption of data distribution, unlabeled data can also assist in building better classifiers [43]. SSL, also known as self-labeling or self-teaching in its earliest forms, involves the model iteratively labeling a portion of unlabeled data and adding it to the training set for the next round of training [35]. The expectation-maximization (EM) algorithm proposed by [32] uses both labeled and unlabeled data to produce maximum likelihood estimates of parameters. [27] and [40] focus on consistency regularization. Π-model [27] combines both supervised cross-entropy loss and unsupervised consistency loss while perturbing the model and data based on the consistency constraint assumption. A temporal ensembling model integrates predictions from models at various time points. Mean teacher [40] addresses the slow updating issue of the temporal ensembling model on large datasets by averaging model weights instead of predicting labels. MixMatch [4] unifies and refines the previous approaches of consistency regularization, entropy minimization, and traditional regularization into a single loss function, achieving excellent results. Pseudo labeling, as an effective tool for reducing the entropy of unlabeled data [28], has been increasingly attracting the attention of researchers in the field of semi-supervised learning. [2] proposes that directly using the model's predictions as pseudo-labels can achieve good results. FixMatch [39] predicts and retains the model, generating high-confidence pseudo-labels. [36] continuously adjusts the teacher based on feedback from the student, thereby generating better pseudo-labels. While there has been a plethora of work in the semi-supervised learning field, the focus on semi-supervised concept bottleneck models remains largely unexplored. Our work focuses on this new area.

3. Preliminaries

Concept Bottleneck Models [25]. We consider a classification task with a concept set denoted as $\mathcal{C} = \{p_1, \cdots, p_k\}$ with each p_i being a concept given by experts or LLMs, and a training dataset represented $\mathcal{D} = \{(x^{(i)}, y^{(i)}, c^{(i)})\}_{i=1}^N$. Here, for $i \in [N]$, $x^{(i)} \in \mathcal{X} \subseteq \mathbb{R}^d$ represents the feature vector (e.g., an image's pixels), $y^{(i)} \in \mathcal{Y} \subseteq \mathbb{R}^l$ denotes the label (l is the number of classes), $c^{(i)} = (c_i^1, \cdots, c_i^k) \in \mathbb{R}^k$ represents the concept vector (a binary vector of length k, where each value indicates whether the input $x^{(i)}$ contains

that concept). In CBMs, the goal is to learn two representations: one called a concept encoder that transforms the input space into the concept space, denoted as $g : \mathbb{R}^d \to \mathbb{R}^k$, and another called label predictor that maps the concept space to the downstream prediction space, denoted as $f : \mathbb{R}^k \to \mathbb{R}^l$. Usually, the map f is linear. For any input x, we aim to ensure that its predicted concept vector $\hat{c} = g(x)$ and prediction $\hat{y} = f(g(x))$ are close to their underlying counterparts, thus capturing the essence of the original CBMs.

Concept Embedding Models [9]. As the original CBM is based solely on concept features to determine the predictions of the model, compared to canonical deep neural networks, it will degrade prediction performance. To further improve the performance of CBMs, CEM is developed by [9]. It achieves this by learning interpretable high-dimensional concept representations (i.e., concept embeddings), thus maintaining high task accuracy while obtaining concept representations that contain meaningful semantic information. For CEMs, we use the same setting as that of [9, 21]. For each input x, the concept encoder learns k concept representations $\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_k$, each corresponding to one of the k ground truth concepts in the training dataset. In CEMs, each concept c_i is represented using two embeddings $\hat{c}_i^+, \hat{c}_i^- \in \mathbb{R}^m$, each with specific semantics, i.e., the concept is *True* (activate state) and concept is *False* (negative state), where hyper-parameter m is the embedding dimension. We use a DNN $\psi(x)$ to learn a latent representation $h \in \mathbb{R}^{n_h}$, which is used as input to the CEM embedding generator, where n_h is the dimension of the latent representation. The CEM embedding generator ϕ feeds h into two fully connected concept-specific layers to learn two concept embeddings in $\mathbb{R}^m$.

$$\hat{c}_i = \phi_i(h) = a(W_i h + b_i).$$

Then we use a differential scoring function $s : \mathbb{R}^{2m} \to [0, 1]$, to achieve the alignment of concept embeddings $\hat{c}_i^+, \hat{c}_i^-$ and ground-truth concepts c_i. It is trained to predict the probability $\hat{p}_i := s\left(\left[\hat{c}_i^+, \hat{c}_i^-\right]^\top\right) = \sigma\left(W_s\left[\left[\hat{c}_i^+, \hat{c}_i^-\right]^\top\right] + b_s\right)$ that the concept c_i is active in the embedding space. We get the final concept embedding $\hat{c}_i$, as follows:

$$\hat{c}_i := \hat{p}_i \hat{c}_i^+ + (1 - \hat{p}_i)\hat{c}_i^-.$$

At this point, we understand that we can obtain high-quality concept embeddings rich in semantics through CEMs. In the subsequent section 4, we will effectively utilize these representations of concepts and further optimize their interpretability through our proposed framework SSCBM.

Semi-supervised Setting. Now, we consider the setting of semi-supervised learning for concept bottleneck models. As mentioned earlier, a typical training dataset for CBMs

can be represented as $\mathcal{D} = \{(x^{(i)}, y^{(i)}, c^{(i)})\}_{i=1}^{N}$, where $x^{(i)} \in \mathcal{X}$ represents the input feature. However, in semi-supervised learning tasks, the set of feature vectors typically consists of two parts, $\mathcal{X} = \{\mathcal{X}_L, \mathcal{X}_U\}$, where $\mathcal{X}_L$ represents a small subset of labeled data and $\mathcal{X}_U$ represents the remaining unlabeled data. Here, the unlabeled data refers to data where the class label y is known, while the concept label c is unknown. Generally we assume $|\mathcal{X}_L| \ll |\mathcal{X}_U|$. We assume that $x^{(j)} \in \mathcal{X}_L$ is labeled with a concept vector $c^{(j)}$ and a label $y^{(j)}$, and for any $x^{(i)} \in \mathcal{X}$, there only exists a corresponding label $y^{(i)} \in \mathcal{Y}$.

Under these settings, given a training dataset $\mathcal{D} = \mathcal{D}_L \cup \mathcal{D}_U$ that includes both labeled and unlabeled data, the goal is to train a CBM using both the labeled data $\mathcal{D}_L$ and unlabeled data $\mathcal{D}_U$. This aims to get better mappings $g : \mathbb{R}^d \to \mathbb{R}^k$ and $f : \mathbb{R}^k \to \mathbb{R}^l$ than those trained by using only labeled data, ultimately achieving higher task accuracy and its corresponding concept-based explanation.

4. Semi-supervised Concept Bottleneck Models

In this section, we will elaborate on the details of the proposed SSCBM framework, which is shown in Figure 2. SS-CBM follows the main idea of CEM. Specifically, to learn a good concept encoder, we use different processing methods for labeled and unlabeled data. The labeled data first passes through a feature extractor ψ to be transformed into a latent representation $\boldsymbol{h}$, which then enters the concept embedding extractor to obtain the concept embeddings and the predicted concept vector $\hat{\boldsymbol{c}}$ for the labeled data, which is compared to the ground truth concept to compute the concept loss. Moreover, the label predictor predicts $\hat{\boldsymbol{y}}$ based on $\hat{\boldsymbol{c}}$, and calculates the task loss.

For unlabeled data, we first extract image features V from the input using an image encoder. Then, we use the KNN algorithm to assign a pseudo-label $\hat{c}_{img}$ to each unlabeled data, which has been experimentally proven to be simple and effective. In the second step, we compute a heatmap between concept embeddings and the input. After applying a threshold, we obtain the predicted alignment label $\hat{c}_{align}$. Finally, we compute the alignment loss between $\hat{c}_{img}$ and $\hat{c}_{align}$. During each training epoch, we simultaneously compute these losses and update the model parameters based on the gradients.

4.1. Label Anchor: Concept Embedding Encoder

Concept Embeddings. As described in Section 3, we obtain high-dimensional concept representations with meaningful semantics based on CEMs. Thus, our concept encoder should extract useful information from both labeled and unlabeled data.

For labeled training data $\mathcal{D}_L = \{(x^{(i)}, y^{(i)}, c^{(i)})\}_{i=1}^{|\mathcal{D}_L|}$, we follow the original CEM [9], i.e., using a backbone network (e.g., ResNet50) to extract features $\boldsymbol{h} =$

$\{\psi(x^{(i)})\}_{i=1}^{|\mathcal{D}_L|}$. Then, for each feature, it passes through an embedding generator to obtain concept embeddings $\hat{\boldsymbol{c}}_i \in \mathbb{R}^{m \times k}$ for $i \in [k]$. After passing through fully connected layers and activation layers, we obtain the predicted binary concept vector $\hat{\boldsymbol{c}} \in \mathbb{R}^k$ for the labeled data. The specific process can be represented by the following expression:

$$\boldsymbol{h^{(j)}} = \psi(x^{(j)}), \quad j = 1, \ldots, |\mathcal{D}_L|,$$

$$\hat{\boldsymbol{c}}_i^{(j)} = \sigma(\phi(\boldsymbol{h^{(j)}})), \quad i = 1, \ldots, k, \quad j = 1, \ldots, |\mathcal{D}_L|,$$

where ψ, ϕ and σ represent the backbone, embedding generator, and activation function, respectively.

To enhance the interpretability of concept embeddings, we calculate the concept loss utilizing binary cross-entropy to optimize the accuracy of concept predictions by computing $\mathcal{L}_c$ based on predicted binary concept vector $\hat{\boldsymbol{c}}$ and ground truth concept labels $\boldsymbol{c}$:

$$\mathcal{L}_c = BCE(\hat{\boldsymbol{c}}, \boldsymbol{c}), \tag{1}$$

where BCE is the binary cross entropy loss.

Task Loss. Since our ultimate task is classification, we also need to incorporate the task loss for the final prediction performance. After obtaining the predicted concept $\hat{\boldsymbol{c}}$, we use a label predictor to predict the final class $\hat{\boldsymbol{y}}$. Then we can define the task loss function using the categorical cross-entropy loss to train our classification model as follows:

$$\mathcal{L}_{task} = CE(\hat{\boldsymbol{y}}, \boldsymbol{y}), \tag{2}$$

Note that for the unlabeled data, we can also calculate the task loss since their class labels are known, to make full use of the data.

4.2. Unlabel Alignment: Image-Textual Semantics Alignment

Pseudo Labeling. Unlike supervised data, CEM cannot directly extract useful information from unlabeled data, as the concept encoder is a supervised training architecture. Thus, in practical situations lacking labeled data, one direct approach is to get high-quality pseudo concept labels to train the model. In the following, we will introduce the method we use to obtain pseudo-concept labels.

Firstly, we can naturally think of measuring the similarity between images by calculating the distance in the cosine space. Based on this idea, we can assign pseudo labels to unlabeled data by finding labeled data with similar image features to them. Specifically, for each unlabeled training data $x \in \mathcal{D}_U = \{(x^{(i)}, y^{(i)})\}_{i=1}^{|\mathcal{D}_U|}$, we calculate its cosine distance with all labeled data points $x^{(j)} \in \mathcal{D}_L = \{(x^{(j)}, y^{(j)}, c^{(j)})\}_{j=1}^{|\mathcal{D}_L|}$ and select the k samples with the smallest distances:

$$\text{dist}(x, x^{(j)}) = 1 - \frac{\Omega(x) \cdot \Omega(x^{(j)})}{\|\Omega(x)\|_2 \cdot \|\Omega(x^{(j)})\|_2}, \quad j = 1, \ldots, |\mathcal{D}_L|,$$
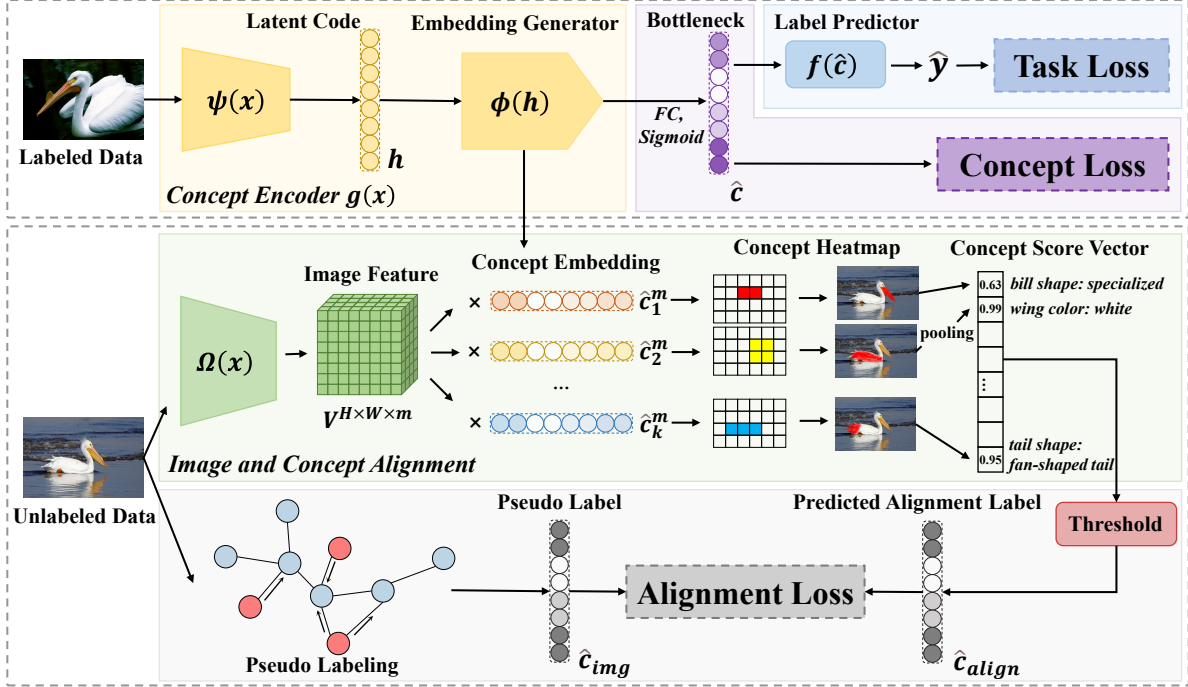
Figure 2. Overall framework of our proposed SSCBM.

where Ω is a visual encoder. Then, we normalize the reciprocal of the cosine distance between the nearest k data points and x as weights. We use these weights to compute a weighted average of the concept labels of these k data points, obtaining the pseudo concept label for x. In this way, we can obtain pseudo concept labels $\hat{c}_{img}$ for all $x \in \mathcal{D}_U$.

In our experiments, we find that directly feeding pseudo-concept labels generated by KNN to CEM has satisfactory performance. However, this simple labeling method can lead to alignment issues in the concept embeddings learned by the CEM's concept encoder. Specifically, the predicted concepts $\hat{c}$ could not have a relation to the corresponding features V in the image, which hinders the effectiveness of CEM as a reliable interpretability tool. Moreover, due to misalignment, this will also degrade the prediction performance of the concept encoder. In the following, we aim to address such a misalignment issue.

Generating Concept Heatmaps. Our above pseudo concept labels via KNN have already contained useful information for prediction. Thus, our goal is to provide these labels with more information about their relation to the corresponding features. To achieve this, we first get another pseudo-concept label via these relations, which are calculated by the similarity between the concept embedding and the input image, namely concept heatmaps. Specifically, given an image x, we first have its feature map $V \in \mathbb{R}^{H \times W \times m}$ for each concept extracted by $V = \Omega(x)$,

where H and W are the height and width of the feature map.

Given V and the i-th concept embedding c_i, we can obtain a heatmap $\mathcal{H}_i$, i.e., a similarity matrix that measures the similarity between the concept and the image can be obtained by computing their cosine distance:

$$\mathcal{H}_{p,q,i} = \frac{\hat{c}_i^m \cdot V_{p,q}}{||\hat{c}_i^m|| \cdot ||V_{p,q}||}, \quad p = 1, \ldots, H, \quad q = 1, \ldots, W$$

where p, q are the p-th and q-th positions in the heatmaps, and $\mathcal{H}_{p,q,i}$ represents a local similarity score between V and $\hat{c}_i^m$. Intuitively, $\mathcal{H}_i$ represents the relation of each part of the image with the i-the concept. Then, we derive heatmaps for all concepts, denoted as $\{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_k\}$.

Calculating Concept Scores and Concept Labels. As average pooling performs better in downstream classification tasks [45], we apply average pooling to the heatmaps to deduce the connection between the image and concepts: $s_i = \frac{1}{P \cdot Q} \sum_{p=1}^{P} \sum_{q=1}^{Q} \mathcal{H}_{p,q,i}$. Intuitively, s_i is the refined similarity score between the image and concept $\hat{c}_i^m$. Thus, a concept vector s can be obtained, representing the similarity between an image input x and the set of concepts: $s = (s_1, \ldots, s_k)^\top$. s can be considered as a soft concept label which is got by similarity. Next, we have to transform it into a hard concept label $\hat{c}_{align}$. we determine the presence of a concept attribute in an image based on a threshold value derived from an experiment. If the value s_i exceeds this threshold, we consider the image to possess that specific concept attribute and set the concept label to be *True*. We

can obtain predicted concept labels for all unlabeled data.

Alignment of Image. Based on our earlier discussions, on the one hand, the concept encoder should learn information from $\hat{c}_{img}$. On the other hand, it should also get concept embeddings which can get good similarity-based concept labels $\hat{c}_{align}$ for alignment with the input image. Thus, we need a further alignment loss to achieve these two goals. Specifically, we compute the alignment loss as follows:

$$\mathcal{L}_{align} = BCE(\hat{c}_{img}, \hat{c}_{align}). \quad (3)$$

In fact, during this alignment process, the CBM backbone is taught a reasonable latent space structure by the visual encoder, and thus we can regard it as an implicit *Student-Teacher* setup framework.

4.3. Final Objective

In this section, we will discuss how we derive the process of network optimization. First, we have the concept loss $\mathcal{L}_c$ in (1) to improve the interpretability of the concept embeddings. Also, since the concept embeddings are entered into the label predictor to output the final prediction, we also have a task loss between the predictions given by the concept bottleneck and ground truth, which is shown in (2). In the context of binary classification tasks, we employ binary cross-entropy (BCE) as our loss function. For multi-class classification tasks, we use cross-entropy as the measure. Finally, to align the images with the concept labels, we computed the loss of alignment in (3). Formally, the overall loss function of our approach can be formulated as:

$$\mathcal{L} = \mathcal{L}_{task} + \lambda_1 \cdot \mathcal{L}_c + \lambda_2 \cdot \mathcal{L}_{align}, \quad (4)$$

where λ_1, λ_2 are hyperparameters for the trade-off between interpretability and accuracy.

5. Experiments

In this section, we will conduct experimental studies on the performance of our framework. Specifically, we evaluate the utility of our method according to the concept and prediction of the class label. We also design interpretability evaluation and test-time intervention to verify the interpretability and alignment performance of our method. Finally, we give the ablation study to verify why we need the alignment loss. Details and additional results are in Appendix A.

5.1. Experimental Settings

Datasets. We evaluate our methods on four real-world image tasks: *CUB*[14], *AwA2*[33], *WBCatt*[42] and *7-point*[22]. See Appendix A.1 for a detailed introduction.

Baselines. We compare our SSCBM with Concept Bottleneck Model (CBM) and Concept Embedding Model (CEM)

Table 1. Concept and task accuracy results of SSCBM at a labeled ratio of 0.1, compared to baselines in the fully supervised setting.

Method	CUB		AwA2		WBCatt		7-point	
	Concept	Task	Concept	Task	Concept	Task	Concept	Task
CBM	93.99%	67.33%	96.48%	88.71%	94.18%	99.71%	74.34%	75.44%
CEM	96.39%	79.82%	95.91%	87.00%	95.33%	99.71%	77.15%	75.85%
SSCBM	90.88%	67.67%	96.48%	89.77%	93.98%	99.68%	73.67%	70.09%

mentioned in Section 3. Since those baselines do not inherently include settings for semi-supervised learning, to ensure fairness in the evaluation, we use the same pseudo-labeling approach as SSCBM. We utilize the KNN algorithm to annotate unlabeled data based on labeled data, which is mentioned in Section 4.2. For CBM [25], we adopt the same setting and architecture as in the original CBM. And for CEM, we follow the same setting as in [9]. We also compare label-free CBM [34], which is an unsupervised CBM method.

Evaluation Metrics. To evaluate the utility, we consider the accuracy for the prediction of class and concept labels. Specifically, *concept accuracy* measures the model's prediction accuracy for concepts: $\mathcal{C}_{acc} = \frac{1}{N} \cdot \frac{1}{k} \sum_{i=1}^{N} \sum_{j=1}^{k} \mathbb{I}(\hat{c}_j^{(i)} = c_j^{(i)})$. *Task accuracy* measures the model's performance in predicting downstream task classes: $\mathcal{A}_{acc} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{y}^{(i)} = y^{(i)})$.

To evaluate the interpretability, besides the concept accuracy, similar to previous work [9, 25], we show *visualization results*. Moreover, we evaluate the performance of the *test-time intervention*.

Implementation Details. All experiments are conducted on a Tesla V100S PCIe 32 GB GPU and an Intel Xeon Processor CPU. See Appendix A.2 for more details.

5.2. Evaluation Results on Utility

To comprehensively evaluate the model's performance in the semi-supervised learning setting, we simulate real-world application scenarios. We consider labeling K samples for each class in the dataset, with an extreme case where K=1. Clearly, K = total data size × labeled ratio ÷ number of classes. We also conduct experiments with labeled ratios of 0.05, 0.1, 0.15, and 0.2. We also compare the performance of SSCBM with the baseline under the fully supervised setting. The experimental results are shown in Table 1 and 2. Additionally, we also explored the impact of different backbones on model performance, which can be found in the Appendix A.3.

Table 1 shows the performance comparison of SSCBM at a labeled ratio of 0.1 with CEM and CBM under the fully supervised setting. It can be observed that, although CEM performs the best on CUB, WBCatt and 7-point, SSCBM is only slightly inferior to it. On the AwA2 dataset, its performance even surpasses that of CEM under the fully super-

Table 2. Results of concept and task accuracy for different datasets with different portions of labeled data. Since the label-free CBM [34] and LaBo [47] do not use the concept set included in the dataset but instead rely on GPT-3 to generate concepts based on class names, we do not report the concept accuracy here.

Dataset	Labeled Ratio	CBM+SSL		CEM+SSL		Label-free CBM		LaBo		SSCBM	
		Concept	Task	Concept	Task	Concept	Task	Concept	Task	Concept	Task
CUB	K=1	83.11%	5.51%	82.36%	59.35%	-	74.31%	-	66.38%	88.99%	66.72%
	0.05 (K=2)	84.51%	8.35%	83.72%	62.20%	-	74.31%	-	66.38%	90.04%	67.43%
	0.1 (K=3)	84.96%	9.84%	84.03%	63.12%	-	74.31%	-	66.38%	90.88%	67.67%
	0.15 (K=4)	85.47%	9.96%	84.30%	64.14%	-	74.31%	-	66.38%	91.47%	68.36%
	0.2 (K=5)	86.67%	16.43%	86.83%	67.64%	-	74.31%	-	66.38%	92.09 %	70.07%
AWA2	K=1	71.32%	90.87%	65.55%	84.44%	-	73.67%	-	93.73%	85.05%	89.02%
	0.05 (K=30)	77.23%	91.11%	68.31%	85.01%	-	73.67%	-	93.73%	95.42%	89.37%
	0.1 (K=60)	80.45%	91.03%	69.96%	85.48%	-	73.67%	-	93.73%	96.48%	89.77%
	0.15 (K=90)	83.46%	91.00%	72.14%	86.61%	-	73.67%	-	93.73%	96.81%	90.54%
	0.2 (K=120)	85.76%	91.12%	72.14%	86.61%	-	73.67%	-	93.73%	96.81%	90.44%
WBCatt	K=1	79.06%	99.39%	70.27%	98.64%	-	41.92%	-	92.26%	91.48%	99.13%
	0.05 (K=62)	81.08%	99.48%	73.82%	99.52%	-	41.92%	-	92.26%	93.53%	99.61%
	0.1 (K=124)	85.48%	99.32%	72.25%	99.29%	-	41.92%	-	92.26%	93.98%	99.68%
	0.15 (K=186)	85.39%	99.68%	72.68%	99.58%	-	41.92%	-	92.26%	94.42%	99.71%
	0.2 (K=247)	87.07%	99.74%	74.14%	99.52%	-	41.92%	-	92.26%	94.42%	99.71%
7-point	K=1	59.91%	55.95%	62.78%	66.09%	-	55.44%	-	64.97%	66.58%	66.84%
	0.05 (K=5)	65.36%	57.47%	67.85%	67.09%	-	55.44%	-	64.97%	70.98%	68.77%
	0.1 (K=9)	68.82%	55.70%	72.23%	66.33%	-	55.44%	-	64.97%	73.67%	70.09%
	0.15 (K=13)	66.14%	59.75%	66.54%	67.09%	-	55.44%	-	64.97%	73.94%	72.56%
	0.2 (K=17)	70.29%	60.44%	73.04%	66.84%	-	55.44%	-	64.97%	76.52%	74.56%

vised setting, which sufficiently demonstrates the superiority of our method.

In Table 2, it can be observed that as the proportion of labeled data gradually increases, the concept and class prediction accuracy of all models improve. SSCBM outperforms all baselines. On the CUB dataset, when only one labeled sample per class is available, SSCBM achieves 88.99% concept accuracy and 66.72% class accuracy, which are 6.63% and 7.37% higher, respectively, than the performance achieved by CEM in a semi-supervised learning setting. This demonstrates the effectiveness of the Image-Textual Semantics Alignment module. The 7-point dataset is the most challenging, and on this dataset, SSCBM still achieves the best performance across all settings. As the amount of labeled data increases, the concept prediction accuracy improves from 66.58% to 76.52%. We also note that on WBCatt and 7-point, the performance of Label-free CBM is poor. This is mainly because both datasets contain only five classes, which prevents Label-free CBM from fully leveraging the world knowledge of LLMs to generate an effective concept set, resulting in poor performance. This further demonstrates that semi-supervised learning scenarios are more realistic and practical for real-world datasets.

5.3. Interpretability Evaluation

Besides the concept accuracy, note that in SSCBM, we also have a pseudo label revived by the alignment between the concept embedding and the input saliency map. We use the alignment loss to inherit this alignment. Thus, we will evaluate the alignment performance here to show the faithfulness of the interpretability given by SSCBM. We measure our alignment performance by comparing the correctness of the concept saliency map with the concept semantics in Figure 3. See Appendix A.2 for detailed experimental procedures for generating saliency maps. Results show that our concept saliency map matches the concept semantics, indicating the effectiveness of our alignment loss. In Appendix C, we also provide our additional interpretability evaluation in Figure 6-18.

5.4. Test-time Intervention

Test-time intervention enables human users to interact with the model at inference time. We conduct this experiment by correcting the ratio 10% to 100% of the concept labels in the concept predictor. We expect that the model performance will steadily increase along with the ratio of concept intervention, indicating that the model learned such correct label information and automatically corrected other labels.

The results in Figure 4 demonstrate the robustness of our model and an increasing trend to learn the information about the concept, indicating our interpretability and model prediction performance. During the test phase, the SSCBM model generates probability scores for each concept (e.g., $has\_wing\_color :: brown$). Prior to intervention, we employ the COOP algorithm (following CEM) to identify the most critical concepts exhibiting both high predictive uncertainty and task importance. For these selected concepts, we replace the model's predicted values with their corresponding ground truth annotations. This rectified concept vector is subsequently fed into the classification head to produce the final task prediction. Finally, we successfully cause the model to predict the Great Crested Flycatcher instead of the

| (a) Original Image | (b) Tail Pattern: Multi-colored | (c) Underparts Color: Brown | (d) Wing Color: Brown |

Figure 3. The concept saliency map for the CUB dataset (savannah sparrow) demonstrates that our proposed SSCBM achieves meaningful alignment between the ground truth concepts and the input image features. The first image on the left is the original input image. The three images on the right show the aligned regions for different concepts using SSCBM.

Figure 4. **Left:** Performance with different ratios of intervened concepts on CUB dataset. **Right:** An example of successful intervention.

Swainson Warbler. More results are in Appendix B.

Table 3. Results of ablation study.

	Ratio	w/o **img**		w/o **align**		**full model**	
		Concept	**Task**	**Concept**	**Task**	**Concept**	**Task**
CUB	0.0001 (K=1)	87.77%	59.89%	80.87%	65.15%	88.99%	66.72%
	0.05 (K=2)	90.01%	65.98%	80.60%	65.67%	90.04%	67.43%
	0.1 (K=3)	91.02%	67.28%	80.60%	64.69%	90.88%	67.67%
	0.15 (K=4)	90.60%	67.97%	80.65%	65.43%	91.47%	68.36%
	0.2 (K=5)	91.94%	68.99%	80.73%	64.45%	92.09%	70.07%
WBCatt	0.0001 (K=1)	88.44%	99.68%	57.80%	99.51%	91.48%	99.13%
	0.05 (K=62)	92.17%	99.29%	57.93%	99.51%	93.53%	99.61%
	0.1 (K=124)	92.91%	99.52%	60.49%	99.64%	93.98%	99.68%
	0.15 (K=186)	93.55%	99.58%	59.13%	99.65%	94.42%	99.71%
	0.2 (K=247)	94.19%	99.61%	60.65%	99.70%	94.42%	99.71%
7-point	0.0001 (K=1)	62.31%	64.30%	55.07%	66.33%	66.58%	66.84%
	0.05 (K=5)	69.42%	65.06%	56.75%	68.10%	70.98%	68.77%
	0.1 (K=9)	70.06%	67.34%	56.07%	65.82%	73.67%	70.09%
	0.15 (K=13)	73.39%	68.10%	56.23%	65.82%	73.94%	72.56%
	0.2 (K=17)	74.44%	68.61%	56.11%	65.57%	76.52%	74.56%

5.5. Ablation Study

We then give a finer study on the two kinds of pseudo-labels to demonstrate that each one is indispensable in bolstering the efficacy of SSCBM. Specifically, based on our method in Section 4.2, we have three types of pseudo labels available: pseudo concept label $\hat{c}_{img}$, $\hat{c}_{align}$, and label $\hat{c}$ predicted by the concept embedding. We first remove $\hat{c}_{img}$ and calculate our alignment loss using $\hat{c}_{align}$ and $\hat{c}$; conversely, we remove $\hat{c}_{align}$ and calculate the alignment loss using $\hat{c}_{img}$ and $\hat{c}$.

From the results presented in Table 3, it is evident that removing the $\hat{c}_{img}$ component significantly degrades the performance of SSCBM at both the concept level and class level. This indicates that the pseudo-concept labels via KNN contain necessary information about the ground truth. The concept encoder needs to extract information from such labels to get better performance. This situation is similar when removing the $\hat{c}_{align}$ component, indicating that aligning the concept embedding and the input saliency map can further extract useful information from the input image and, thus, is beneficial to improve the performance. Our observations underscore the high degree of joint effectiveness of two kinds of pseudo-concept labels within our objective function, collectively contributing to the enhancement of model prediction and concept label prediction.

6. Conclusion

The training of current CBMs heavily relies on the accuracy and richness of annotated concepts in the dataset. These concept labels are typically provided by experts, which can be costly and require significant resources and effort. Additionally, concept saliency maps frequently misalign with input saliency maps, causing concept predictions to correspond to irrelevant input features - an issue related to annotation alignment. In this problem, we propose SSCBM, a strategy to generate pseudo labels and an alignment loss to solve these two problems. Results show our effectiveness.

Acknowledgements

References

[1] Andrea Acevedo, Santiago Alférez, Anna Merino, Laura Puigví, and José Rodellar. Recognition of peripheral blood cell images using convolutional neural networks. *Computer Methods and Programs in Biomedicine*, 180:105020, 2019. 11

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020. 3

[3] Pietro Barbiero, Gabriele Ciravegna, Francesco Giannini, Mateo Espinosa Zarlenga, Lucie Charlotte Magister, Alberto Tonda, Pietro Lio, Frédéric Precioso, Mateja Jamnik, and Giuseppe Marra. Interpretable neural-symbolic concept reasoning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, pages 1801–1825. PMLR, 2023. 2

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 3

[5] Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5948–5955, 2023. 2

[6] Asma Chebli, Akila Djebbar, and Hayet Farida Marouani. Semi-supervised learning for medical application: A survey. In *2018 International Conference on Applied Smart Systems (ICASS)*, pages 1–9, 2018. 3

[7] Xing Chen, Biao Ren, Ming Chen, Quanxin Wang, Lixin Zhang, and Guiying Yan. Nllss: predicting synergistic drug combinations based on semi-supervised learning. *PLoS computational biology*, 12(7):e1004975, 2016. 3

[8] Veronika Cheplygina, Marleen de Bruijne, and Josien P.W. Pluim. Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical Image Analysis*, 54:280–296, 2019. 3

[9] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, et al. Concept embedding models: Beyond the accuracy-explainability trade-off. *Advances in Neural Information Processing Systems*, 35:21400–21413, 2022. 3, 4, 6, 11

[10] Roman Filipovych and Christos Davatzikos. Semi-supervised pattern classification of medical images: Application to mild cognitive impairment (mci). *NeuroImage*, 55 (3):1109–1119, 2011. 3

[11] Jack Furby, Daniel Cunnington, Dave Braines, and Alun Preece. Can we constrain concept bottleneck models to learn semantically meaningful input features? *arXiv preprint arXiv:2402.00912*, 2024. 2

[12] Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. *Advances in Neural Information Processing Systems*, 35:23386–23397, 2022. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 11

[14] Xiangteng He and Yuxin Peng. Fine-grained visual-textual representation learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(2):520–531, 2019. 6, 11

[15] Junlin Hou, Jilan Xu, and Hao Chen. Concept-attention whitening for interpretable skin lesion diagnosis. *arXiv preprint arXiv:2404.05997*, 2024. 2

[16] Lijie Hu, Yixin Liu, Ninghao Liu, Mengdi Huai, Lichao Sun, and Di Wang. Improving faithfulness for vision transformers. *arXiv preprint arXiv:2311.17983*, 2023. 1

[17] Lijie Hu, Yixin Liu, Ninghao Liu, Mengdi Huai, Lichao Sun, and Di Wang. Seat: stable and explainable attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12907–12915, 2023.

[18] Lijie Hu, Liang Liu, Shu Yang, Xin Chen, Hongru Xiao, Mengdi Li, Pan Zhou, Muhammad Asif Ali, and Di Wang. A hopfieldian view-based interpretation for chain-of-thought reasoning. *arXiv preprint arXiv:2406.12255*, 2024. 1

[19] Lijie Hu, Chenyang Ren, Zhengyu Hu, Cheng-Long Wang, and Di Wang. Editable concept bottleneck models. *arXiv preprint arXiv:2405.15476*, 2024. 2

[20] Tri Huynh, Aiden Nibali, and Zhen He. Semi-supervised learning for medical image classification using imbalanced training data. *Computer Methods and Programs in Biomedicine*, 216:106628, 2022. 3

[21] Aya Abdelsalam Ismail, Julius Adebayo, Hector Corrada Bravo, Stephen Ra, and Kyunghyun Cho. Concept bottleneck generative models. In *The Twelfth International Conference on Learning Representations*, 2023. 3

[22] Jeremy Kawahara, Sara Daneshvar, Giuseppe Argenziano, and Ghassan Hamarneh. Seven-point checklist and skin lesion classification using multitask multimodal neural nets. *IEEE Journal of Biomedical and Health Informatics*, 23(2): 538–546, 2019. 6, 11

[23] Mert Keser, Gesina Schwalbe, Azarm Nowzad, and Alois Knoll. Interpretable model-agnostic plausibility verification for 2d object detectors using domain-invariant concept bottleneck models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3890–3899, 2023. 2

[24] Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. *arXiv preprint arXiv:2306.01574*, 2023. 2

[25] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 1, 2, 3, 6

[26] Songning Lai, Lijie Hu, Junxiao Wang, Laure Berti-Equille, and Di Wang. Faithful vision-language interpretation via concept bottleneck models. In *The Twelfth International Conference on Learning Representations*, 2023. 1

[27] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 3

[28] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, page 896. Atlanta, 2013. 3

[29] Jia Li, Lijie Hu, Zhixian He, Jingfeng Zhang, Tianhang Zheng, and Di Wang. Text guided image editing with automatic concept locating and forgetting. *arXiv preprint arXiv:2405.19708*, 2024. 2

[30] Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. Large language models in finance: A survey. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 374–382, 2023. 1

[31] Lucie Charlotte Magister, Dmitry Kazhdan, Vikash Singh, and Pietro Liò. Gcexplainer: Human-in-the-loop concept-based explanations for graph neural networks. *arXiv preprint arXiv:2107.11889*, 2021. 2

[32] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996. 3

[33] Li Niu, Qingtao Tang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Learning from noisy web data with category-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7689–7698, 2018. 6, 11

[34] Tuomas Oikarinen, Subhro Das, Lam Nguyen, and Lily Weng. Label-free concept bottleneck models. In *International Conference on Learning Representations*, 2023. 1, 2, 6, 7

[35] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020. 3

[36] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11557–11568, 2021. 3

[37] Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022. 2

[38] Ivaxi Sheth and Samira Ebrahimi Kahou. Auxiliary losses for learning generalizable concept-based models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[39] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 3

[40] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 3

[41] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023. 1

[42] Satoshi Tsutsui, Winnie Pang, and Bihan Wen. Wbcatt: A white blood cell dataset annotated with detailed morphological attributes. In *Advances in Neural Information Processing Systems (NeurIPS).*, 2023. 6, 11

[43] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020. 3

[44] Han Xuanyuan, Pietro Barbiero, Dobrik Georgiev, Lucie Charlotte Magister, and Pietro Liò. Global concept-based interpretability for graph neural networks via neuron analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10675–10683, 2023. 2

[45] An Yan, Yu Wang, Yiwu Zhong, Zexue He, Petros Karypis, Zihan Wang, Chengyu Dong, Amilcare Gentili, Chun-Nan Hsu, Jingbo Shang, et al. Robust and interpretable medical image classifiers via concept bottleneck models. *arXiv preprint arXiv:2310.03182*, 2023. 5

[46] Shu Yang, Lijie Hu, Lu Yu, Muhammad Asif Ali, and Di Wang. Human-ai interactions in the communication era: Autophagy makes large models achieving local optima. *arXiv preprint arXiv:2402.11271*, 2024. 1

[47] 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, 2023. 7

[48] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2

[49] Gang Zhang, Shan-Xing Ou, Yong-Hui Huang, and Chun-Ru Wang. Semi-supervised learning methods for large scale healthcare data analysis. *International Journal of Computers in Healthcare*, 2(2):98–110, 2015. 3