Discover-then-Name: Task-Agnostic Concept Bottlenecks via Automated Concept Discovery

Sukrut Rao*[,1,2], Sweta Mahajan*[,1,2], Moritz Böhle[1], and Bernt Schiele[1]

[1] Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken
[2] RTG Neuroexplicit Models, Saarbrücken
{sukrut.rao,sweta.mahajan,mboehle,schiele}@mpi-inf.mpg.de

Abstract. Concept Bottleneck Models (CBMs) have recently been proposed to address the 'black-box' problem of deep neural networks, by first mapping images to a human-understandable concept space and then linearly combining concepts for classification. Such models typically require first coming up with a set of concepts relevant to the task and then aligning the representations of a feature extractor to map to these concepts. However, even with powerful foundational feature extractors like CLIP, there are no guarantees that the specified concepts are detectable. In this work, we leverage recent advances in mechanistic interpretability and propose a novel CBM approach — called Discover-then-Name-CBM (DN-CBM) — that inverts the typical paradigm: instead of pre-selecting concepts based on the downstream classification task, we use sparse autoencoders to first *discover* concepts learnt by the model, and then *name* them and train linear probes for classification. Our concept extraction strategy is *efficient*, since it is agnostic to the downstream task, and uses concepts *already known* to the model. We perform a comprehensive evaluation across multiple datasets and CLIP architectures and show that our method yields semantically meaningful concepts, assigns appropriate names to them that make them easy to interpret, and yields performant and interpretable CBMs. Code available at https://github.com/neuroexplicit-saar/discover-then-name.

Keywords: Inherent Interpretability · Concept Bottleneck Models

1 Introduction

Deep neural networks have been immensely successful for a variety of tasks, yet their 'black-box' nature poses a risk for their use in safety-critical applications. While attribution methods [5, 47, 52] have popularly been used to explain such models *post-hoc*, they have been shown to often provide explanations unfaithful to the model [2, 3, 46]. To address this, *inherently interpretable* models have been proposed [8, 12, 30] that constrain the model to yield more faithful and human-understandable explanations in the form of heatmaps, concepts, or prototypes.

* Equal contribution.

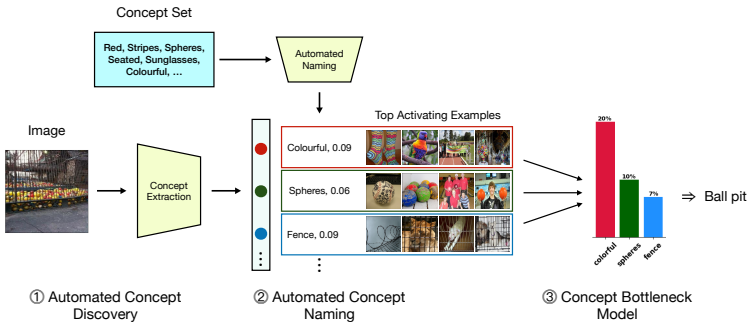

Fig. 1: Automated concept extraction and naming to construct task-agnostic concept bottlenecks. Our approach consists of three steps: **(1)** we use a sparse autoencoder to extract disentangled concepts from CLIP feature extractors, **(2)** automatically name extracted concepts by matching the dictionary vectors with the closest text embedding in CLIP space from a concept set of texts, and **(3)** use this named concept extractor layer as a concept bottleneck to create concept bottleneck models for classification on different datasets. In the example shown, the concepts 'colorful', 'spheres', and 'fence' are extracted from the image with high strengths, resulting in a prediction of 'ball pit'. For details, see Fig. 2 and Sec. 3.

Concept Bottleneck Models (CBMs) [30,36,59] are a class of inherently interpretable models that express their prediction as a linear combination of simpler but human-interpretable concepts detected from the input features. While typically constrained by the need of a labelled attribute dataset for training [30], recent CBMs leverage large-language models (LLMs) such as GPT-3 [10] to generate class-specific concepts and vision-language models (VLMs) such as CLIP [44] to learn the mapping from inputs to concepts in an attribute-label-free manner [34,36,41,58], and have been shown to be performant even on large datasets such as ImageNet [16]. However, such methods still require querying LLMs based on the classification task, and it is unclear if the concepts one *wants* the model to detect *can* be detected at all; in fact, recent works have suggested that while plausible, explanations from such CBMs may not be faithful [33, 49].

To address this, in this work, we invert the typical CBM paradigm, and aim to *discover* concepts the model *knows*, name them, and then perform classification (Fig. 1). We specifically use CLIP feature extractors to leverage vision-language alignment for automated naming of concepts. While raw features of a network are typically uninterpretable [18], sparse autoencoders (SAEs) have been shown to be a promising tool in the context of language models wherein they disentangle learned representations into a sparse set of human-understandable concepts. This is achieved by decomposing the representations into a sparse linear combination of a set of learned dictionary vectors [9,14]. We extend this to vision and find it to be similarly promising, and surprisingly, find that the dictionary vectors appear to align well with text embeddings of concepts they represent in CLIP space, thus making their corresponding concepts nameable (Fig. 3). Finally, we use this latent concept space as a concept bottleneck, and show that, once learnt, it can be frozen and used 'as is' to train classifiers to construct performant CBMs for a

variety of downstream classification tasks. Our approach is also computationally efficient since it learns concept bottlenecks in a *task-agnostic* manner, eliminating the need to make queries to external LLMs to find task-relevant concepts.

In summary, **our contributions are** • We propose DN-CBM, a novel CBM that leverages sparse autoencoders (SAEs) to *discover* concepts learnt by CLIP. We find that SAEs lend themselves well to our simple and intuitive approach for automated concept discovery. • We propose a novel approach to automatically *name* the discovered concepts, by mapping concepts to text with embeddings most similar to the corresponding dictionary vectors of the learned concept. We find that this often yields names semantically consistent to the images activating the concept (Fig. 3). • We show that, once discovered and named, the learnt concept mapping can be used to train concept bottleneck models (CBMs) out-of-the-box for a variety of downstream classification tasks. Specifically, we discover concepts using CC3M [53] in a task-agnostic fashion, and then construct CBMs for a variety of downstream datasets: ImageNet [16], Places365 [62], CIFAR10 [31], and CIFAR100 [31]. Importantly, this task-agnostic concept discovery approach yields both performant (Tab. 1) and interpretable (Figs. 7 and 8) classifiers.

2 Related Work

Concept-based Explanations (e.g. [1, 28, 30, 37]) aim to express a model's decision via human-understandable concepts. Unlike popularly used post-hoc attribution heatmaps (e.g. [5, 32, 47, 52, 54, 55]) that only inform *which regions* in the input is influential for the decision, such methods attempt to also answer *what* high-level concepts are important for the model [1]. In our work, we propose a pipeline for automatically extracting and naming such concepts from CLIP and using them to build interpretable models.

Concept Discovery [6, 9, 14, 19–22, 37–39, 60] methods have been proposed to better understand models by discovering and extracting semantically meaningful concepts learnt by them. They typically focus on explaining the function of neurons in a model [6,20,37,39], or on discovering features present in an input, and have been shown to be useful for diagnosing model failures [20]. However, these methods assign concepts to individual neurons, which may often be polysemantic and not decodable to human-understandable concepts [18]. Recently, [9] showed that sparse autoencoders (SAEs) can be effective to address the polysemanticity and superposition problem in deep networks [18] and extract mono-semantic concepts from language models (cf. [14]). We extend the setup of [9] to vision and use sparse autoencoders to automatically extract concepts learnt by CLIP.

Explanations using Language [13, 15, 24, 25, 35, 37, 39, 56, 61] have become popular to express a model's learnt representations [15,37,39] in an easily human-interpretable manner. To decode concepts to language, such methods typically use a large-language model (LLM) such as GPT-3 [10] and learn a mapping from vision features to the LLM input. More recently, [37,39] leverage CLIP, by aligning vision features of the model being explained to the representation space

of CLIP and finding the closest aligned texts from a large concept set. Similar to *Concept Discovery* methods, this line of work assigns concepts to individual neurons as well. In contrast, we use sparse autoencoders (SAEs) [9] to first disentangle the representation space into more human-interpretable concepts, and then name each concept. In particular, we encode the CLIP features to a high dimensional latent space, which we then pass through the SAE decoder to reconstruct back the CLIP features. Surprisingly, when using CLIP feature extractors, we find that the dictionary vectors of the SAE decoder can directly be decoded into text by finding most similar text embeddings in CLIP space, without needing to use LLMs as used by [15, 35, 56], and are effective in yielding semantically meaningful and human-interpretable concepts (Figs. 3 and 4).

Concept Bottleneck Models (CBMs) [4, 11, 26, 27, 29, 30, 34, 36, 40, 41, 51, 57–59, 63] are a recently popular class of inherently interpretable models (e.g. [8, 12]) that use a concept bottleneck layer (CBL) to extract named concepts and then learn a (typically sparse) linear classifier that predicts by combining such concepts, yielding highly interpretable explanations. While such methods typically require a labelled concept dataset [30] to learn the concept bottleneck, recent works leverage LLMs such as GPT-3 [10] and VLMs such as CLIP [44] to learn such bottlenecks without needing the concept labels [36, 40, 41, 58], making them scale to large datasets such as ImageNet [16] in a performant manner. Given a classification task, such methods first query an LLM for concepts relevant to the task, and use a VLM to learn a concept bottleneck where each neuron aligns to one of the desired concepts. However, it is unclear if the feature extractor can truly recognize all such concepts when specified a priori, since they may often be non-visual [49,58] and their faithfulness has also been called into question [33,49]. Further, the CBL needs to be trained separately for each classification dataset. In contrast, we flip the paradigm and first extract concepts that are detected by the model to train the concept bottleneck, using a dataset *independent* of the downstream classification task. We then *fix* the concept bottleneck and train linear classifiers for several datasets and show that this yields highly performant and interpretable models. Similar to us, [26] also first discover concepts before constructing a CBM; however, in contrast our method does not require any external text annotations for the images and the concept discovery can even be done using a dataset different from that of the downstream task.

Explaining CLIP. Several approaches have been proposed to specifically explain and understand CLIP [44] models [7, 41, 56]. Similar to [7], we disentangle CLIP features into human interpretable concepts. However, in contrast to [7], we do not optimize for a sparse concept representation per image using a predefined concept set, and instead first apply a general concept discovery framework [9] for extracting human understandable concepts and then name them post hoc in a task-agnostic manner to construct CBMs.

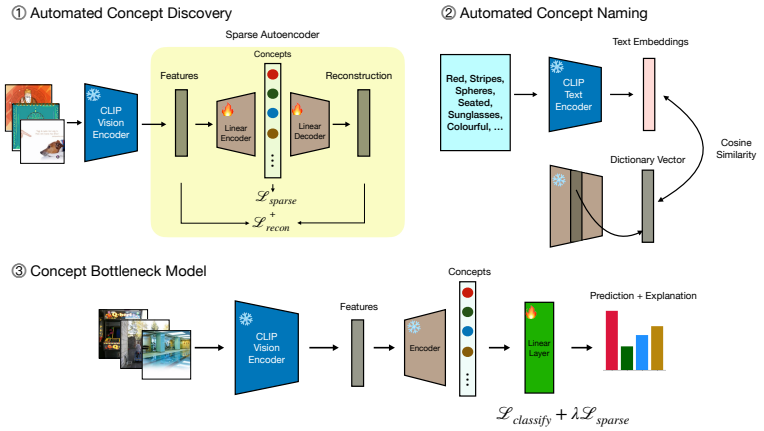

Fig. 2: Overview. Our approach consists of three steps. **(1)** We train a sparse autoencoder to extract disentangled concepts from a CLIP vision backbone. The autencoder is trained on a large dataset $\mathcal{D}_{extract}$ to reconstruct CLIP features using a linear combination of encoded concepts, which are optimized to be sparse using L_1 sparsity. The weights of the decoder can be interpreted as dictionary vectors whose linear sum with concept strengths reconstructs the original feature (Sec. 3.1). **(2)** We use a large concept set of texts $\mathcal{V}$ to name each extracted concept, by finding the text from the set whose embedding has the highest cosine similarity to concept's dictionary vector (Sec. 3.2). **(3)** We use the extracted and named concepts as a concept bottleneck layer, and train linear classifiers to construct inherently interpretable concept bottleneck models across downstream datasets $\mathcal{D}_{classify}$ using the same bottleneck layer (Sec. 3.3).

3 Constructing CBMs via Automated Concept Discovery

In this section, we describe our approach which consists of three stages: discovering the concepts the CLIP model has learnt via a sparse autoencoder (Sec. 3.1), naming those concepts in natural language by leveraging the CLIP text embeddings from a large vocabulary, (Sec. 3.2), and, lastly, training an interpretable concept bottleneck model (CBM) based on the discovered concepts (Sec. 3.3).

3.1 Extracting Concepts Learned by the Model

To discover the concepts learned by the model, we adapt the sparse autoencoder (SAE) approach as described by [9]. Specifically, we aim to discover concepts by representing the CLIP features in a high-dimensional, but very sparsely activating space. For language models, this has been shown to yield representations in which individual neurons (dimensions) are more easily interpretable [9].

The Sparse Autoencoders (SAEs) proposed by [9] consist of a linear encoder $f(\cdot)$ with weights $\boldsymbol{W}_E \in \mathbb{R}^{d \times h}$, a ReLU non-linearity ϕ, and a linear decoder $g(\cdot)$ with weights $\boldsymbol{W}_D \in \mathbb{R}^{h \times d}$. For a given input $\boldsymbol{a}$, the SAE computes:

$$\mathrm{SAE}(\boldsymbol{a}) = (g \circ \phi \circ f)(\boldsymbol{a}) = \boldsymbol{W}_D^T \, \phi\left(\boldsymbol{W}_E^T \boldsymbol{a}\right) \ . \tag{1}$$

Importantly, the hidden representation $f(\boldsymbol{a})$ is of significantly higher dimensionality than the CLIP embedding space (i.e. $h \gg d$), but optimised to activate only very sparsely. Specifically, the SAE is trained with an L_2 reconstruction loss, as well as an L_1 sparsity regularisation:

$$\mathcal{L}_{\text{SAE}}(\boldsymbol{a}) = \|\text{SAE}(\boldsymbol{a}) - \boldsymbol{a}\|_2^2 + \lambda_1 \|\phi(f(\boldsymbol{a}))\|_1 \tag{2}$$

with λ_1 a hyperparameter. To discover a diverse set of concepts for usage in downstream tasks, we train the SAE on a large dataset $\mathcal{D}_{extract}$; given the reconstruction objective, no labels for this dataset are required.

Note that sparsity does of course not *guarantee* that individual neurons in the hidden representation of the SAE align with human-interpretable concepts. However, similar to [9], in our experiments we find that this is often the case, and, as we discuss in the next section, can often even be automatically named.

Why SAEs? While SAEs are certainly not the only option for concept discovery in DNNs, recent work on language models suggests that they might be particularly well suited to discover interpretable concepts, see [9,14], and exhibit certain properties that lend themselves well for automatically naming visual concepts. Specifically, as we will see in the next section, by reconstructing the original feature space, we are able to leverage the dictionary vectors of the reconstruction matrix $\boldsymbol{W}_D$ for assigning names to individual concepts. Moreover, in contrast to dimensionality reduction techniques (e.g. PCA), SAEs are able to represent more features than there are neurons, which was shown to be advantageous to address the problem of polysemanticity [9].

3.2 Automated Concept Naming

Once we trained the SAE, we aim to automatically name the individual feature dimensions in the hidden representation of the SAE. For this, we propose using a large vocabulary of English words, say $\mathcal{V}=\{v_1, v_2, \dots\}$, which we embed via the CLIP text encoder $\mathcal{T}$ to obtain word embeddings $\mathcal{E}=\{\boldsymbol{e}_1, \boldsymbol{e}_2, \dots\}$.

To name the SAE's hidden features, we propose to leverage the fact that each of the SAE neurons c is assigned a specific dictionary vector $\boldsymbol{p}_c$, corresponding to a column of the decoder weight matrix:

$$\boldsymbol{p}_c = [\boldsymbol{W}_D]_c \in \mathbb{R}^d \tag{3}$$

If the SAE indeed succeeds to decompose image representations given by CLIP into individual concepts, we expect the $\boldsymbol{p}_c$ to resemble the embeddings of particular words that CLIP has learnt to expect in a corresponding image caption.

Hence, to name the 'concept' neuron c of the SAE, we propose to assign it the word s_c of the closest text embedding in $\mathcal{E}$:

$$s_c = \arg\min_{v \in \mathcal{V}} \left[\cos\left(\boldsymbol{p}_c, \mathcal{T}(v)\right)\right] \tag{4}$$

Note that this setting is equivalent to using the SAE to reconstruct a CLIP feature when only the concept to be named is present. As CLIP was trained to optimise cosine similarities between text and image embeddings, using the cosine similarity to assign names to concept nodes is a natural choice in this context.

3.3 Constructing Concept Bottleneck Models

Thus far, we trained an SAE to obtain sparse representations (Sec. 3.1), and named individual 'neurons' by leveraging the similarity between dictionary vectors $\boldsymbol{p}_c$ to word embeddings obtained via CLIP's text encoder $\mathcal{T}$ (Sec. 3.2).

Such a sparse decomposition into named 'concepts' constitutes the ideal starting point for constructing interpretable Concept Bottleneck Models (CBMs) [30, 36, 59]: for a given *labelled* dataset $\mathcal{D}_{\text{probe}}$, we can now train a linear transformation $h(\cdot)$ on the SAE's *sparse concept activations*, yielding our CBM $t(\cdot)$:

$$t(\boldsymbol{x}_i) = (\underbrace{h}_{\text{Probe}} \circ \underbrace{\phi \circ f}_{\text{SAE}} \circ \underbrace{\mathcal{I}}_{\text{CLIP}})(\boldsymbol{x}_i) \tag{5}$$

Here, $\boldsymbol{x}_i$ denotes an image from the probe dataset. The probe is trained using the cross-entropy loss, and to increase the interpretability of the resulting CBM classifier, we additionally apply a sparsity loss to the probe weights:

$$\mathcal{L}_{\text{probe}}(\boldsymbol{x}_i) = \text{CE}\left(t(\boldsymbol{x}_i), y_i\right) + \lambda_2 \|\omega\|_1 \tag{6}$$

where, λ_2 is a hyperparameter, y_i the ground truth label of $\boldsymbol{x}_i$ in the probe dataset, and ω denotes the parameters of the linear probe.

Importantly, note that the feature extractor, the dataset used for concept discovery, and the vocabulary used for naming can be freely chosen. As such, our approach is likely to benefit from advances in any of these directions.

4 Evaluation of Concept Discovery and Naming

In this section, we evaluate the effectiveness of using SAEs to discover and name concepts in CLIP vision encoders; see Sec. 5 for an evaluation of CBMs built on the SAEs. In Sec. 4.1, we first evaluate the accuracy and task agnosticity of the discovered concepts qualitatively and quantitatively, in Sec. 4.2, we discuss the impact of the vocabulary $\mathcal{V}$ towards the granularity of concept names, and in Sec. 4.3, we evaluate how well semantically similar concepts group together.

Setup. We use a CLIP [44] ResNet-50 [23] vision encoder for extracting features, and use the corresponding text encoder for labelling the extracted concepts. For additional results using CLIP ViT-B/16 and ViT-L/14 [17], see Appendices C and D. To extract concepts, we follow a setup similar to [9] and train SAEs using the CC3M dataset [53]. Following [37], we use the set of 20k most frequent English words as the vocabulary $\mathcal{V}$ (Eq. (4)). For details, see Appendix B.1.

4.1 Task-Agnosticity and Accuracy of Concepts

In this section, we qualitatively and quantitatively evaluate the extracted and named concepts for semantic consistency and accuracy.

Qualitative. To showcase the promise of our proposed approach, in Fig. 3 we visualize the top activating images across four datasets for various concepts that

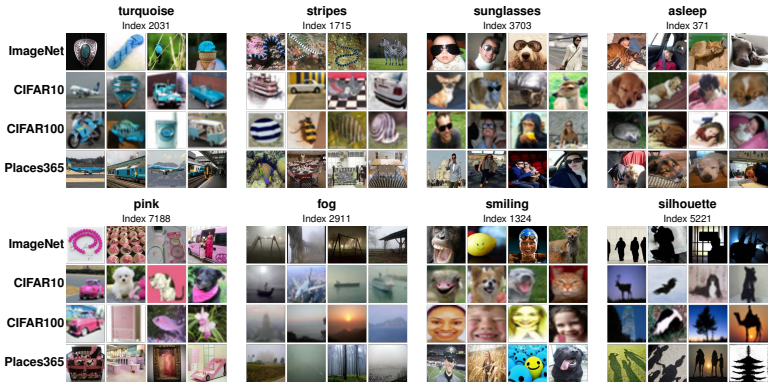

Fig. 3: Task-agnosticity of concept extraction. We show examples of named concepts (blocks) and top images activating them from four datasets (rows). We find that the images activating the concept are highly consistent with the concept name across datasets (e.g. the 'asleep' concept yields images across different species), despite not using these datasets for extraction and naming, showing the robustness of our approach.

were discovered and named as described in Secs. 3.1 and 3.2. For this, we select concepts c from the vocabulary with a high cosine similarity between $\boldsymbol{p}_c$ and $\mathcal{T}(s_c)$, see Eqs. (3) and (4). In particular, we show examples for various low-level concepts (turquoise, pink, striped), object and scene-specific concepts (sunglasses, fog, silhouette), as well as higher-level concepts (asleep, smiling), and find that the visualized concepts not only exhibit a high level of semantic consistency, but also that the automatically chosen names for the concepts accurately reflect the common feature in the images, despite coming from very different datasets. This highlights the promise of the SAE for disentangling representations into human interpretable concepts as well as of the proposed strategy for naming those concepts. Interestingly, as expected, we find that the accuracy of the ascribed names correlates with the cosine similarity between the text embedding $\mathcal{T}(s_c)$ and the dictionary vector $\boldsymbol{p}_c$ (cf. Eq. (3)), as we discuss next (see also Fig. 4). This indicates that our naming strategy could be significantly improved with a larger vocabulary, as we also discuss in Sec. 4.2.

Quantitative. To not only rely on the visual assessment of a few selected samples, we perform quantitative evaluations to assess the concept consistency and naming accuracy. This is generally challenging as only a few datasets include concept labels, and, even if they do, might describe different concepts in the image than those that were extracted by our task-agnostic approach. To address this, we perform a user study to evaluate concept accuracy. Specifically, we sort concepts based on how well their dictionary vector is aligned to the text embeddings of the name assigned to them (Sec. 3.2), and sample concepts with high, intermediate, and low alignments. We then extract the top activating images for each concept from three datasets (ImageNet, Places365, CC3M), and for each concept, we ask two questions: (1) how semantically consistent the concept is, i.e. if the top activating images map to some human interpretable concept, and (2) how accurate the assigned name is, if so. To evaluate if our SAE yields more

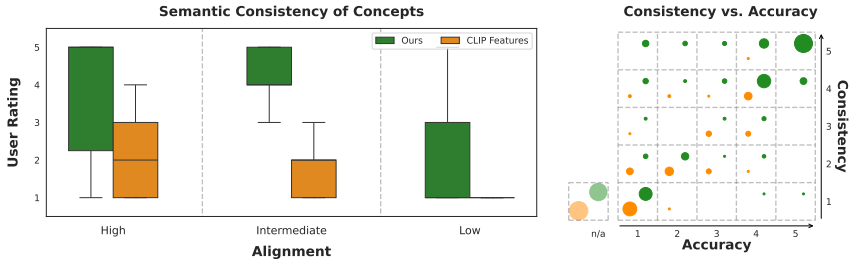

Fig. 4: User study on concept accuracy. *Left:* We evaluate the semantic consistency of concepts for nodes with high, intermediate, and low alignment with the text embeddings of the name assigned to them, both for nodes from our SAE (green) and the CLIP features (orange). We find that the concepts from the SAE are significantly more semantically consistent than CLIP features, and the consistency increases with alignment. The poor performance of the 'low alignment' group suggests that some nodes do not correspond to a consistent human interpretable concept. *Right:* We plot the scores for semantic consistency against name accuracy from human evaluators, both for nodes from our SAE (green) and the CLIP features (orange). We find that compared to the baseline, our SAE nodes are generally more consistent and accurately named.

disentangled concepts, we also compare with the neurons from the CLIP image features, named using CLIP-Dissect [37], as a baseline. For full details, see Appendix B.3. In Fig. 4 (left), we report the distribution of consistency scores both for our discovered concepts and the CLIP baseline each for the high, intermediate, and low aligned concepts, and find that our approach provides significantly more human interpretable concepts. Interestingly, for both sets of concepts, the consistency decreases as the alignment with the text embedding decreases, suggesting that some concepts are not human interpretable. In Fig. 4 (right), we evaluate the concept consistency against name accuracy, and find that our assigned names score highly in terms of accurately representing the concept (top right) as compared to the baseline. Note that some concepts, despite being consistent, are not named accurately, which could also be because of limitations in the vocabulary used; for more discussion, see Sec. 4.2.

In addition to the human evaluation, we also perform a small quantitative evaluation using the SUNAttributes dataset [42], following [41]. We use its labelled attributes as the vocabulary for naming the nodes in the SAE (Sec. 3.2) to match discovered concepts to ground truth labels. To account for concepts outside the labelled attribute set, we filter out nodes where the cosine similarity between the dictionary vector and the assigned text embedding is below a threshold, and merge concepts assigned to the same name. As a baseline, we compare against images obtained using CLIP retrieval from the ground truth attributes. We obtain a Jaccard index of 18.3, as compared to 22.0 for the CLIP retrieval baseline (for comparison, [41] report a Jaccard index of 15.7 under a similar setting) despite not optimizing the SAE to learn dataset-specific concepts.

Fig. 5: Impact of vocabulary. We show examples of pairs of concepts that, despite being assigned to the same coarse grained name (e.g. left: 'tree'), correspond to distinct fine-grained concepts. Better names that can distinguishing such concepts are assigned if added to the vocabulary (e.g. 'christmas tree' for the first concept, and 'tree in field' for the second). On the other hand, removing the assigned name from the vocabulary leads to worse names being assigned (e.g. 'ornaments' and 'branches'), which shows that the granularity of the vocabulary can impact name accuracy.

4.2 Impact of Vocabulary on Concept Name Granularity

As seen in Sec. 4.1 and Fig. 4, some of the SAE nodes may not map to human interpretable concepts (Fig. 4, left), or may not be named appropriately (Fig. 4, right). The latter could be a result of limitations in the vocabulary: it being finite and only consisting of single words, it is possible that even concepts that the SAE discovers cannot be named accurately.

To explore this, in Fig. 5 we visualize examples of concept pairs that are originally assigned the same name (e.g. right: 'bridges'), but visually correspond to distinct modalities of the concept. We find that a more fine-grained name is assigned to the concept when added to the vocabulary $\mathcal{V}$ (e.g. 'arch bridge', 'suspension bridge'). Conversely, removing the assigned name 'bridge' from the vocabulary leads to worse names being assigned (e.g. 'prague', 'lisbon'; interestingly, note that the cities contain a prominent arch and suspension bridge, respectively). This suggests that the granularity and size of the vocabulary can significantly affect the name accuracy, and can also serve as a tool for practitioners to control the granularity of assigned names depending on the use case.

4.3 Clustering Concept Vectors

To further measure semantic consistency, we also evaluate how well semantically related concepts cluster together in the latent concept space. To do this, we perform K-Means clustering on the concept representations across all images in the Places365 dataset, and visualize a random selection of clusters. For each cluster, we compute the cluster centroid and then visualize the strongest concepts. We find that semantically similar concepts and their associated images cluster together in concept space (e.g. farming related concepts and images in the right), showing that our concept-based (latent) representation does indeed result in semantically meaningful and nameable similarities.

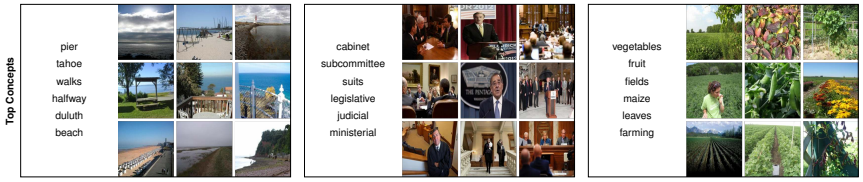

Fig. 6: Extracting meaningful clusters from concept strength vectors. We perform K-Means clustering over concept activation vectors on the Places365 dataset to evaluate the semantic consistency of these latent representations. We show a random subset of clusters: each block represents a cluster, and we show top concepts from the cluster centroid and randomly selected images assigned to the cluster. We find that highly semantically consistent clusters of concepts emerge (e.g. right: concepts and images from classes related to farming are grouped together).

5 Evaluation of DN-CBM

We now present results on the concept bottleneck models (DN-CBM) (Sec. 3.3) built on the discovered and named concepts (Secs. 3.1, 3.2), evaluating accuracy (Sec. 5.1), interpretability (Sec. 5.2), and effectiveness of interventions (Sec. 5.3).
Setup. Similar to prior work [36, 41], we train linear classifiers on top of the extracted concepts on four datasets—ImageNet [16], CIFAR10 [31], CIFAR100 [31], and Places365 [62]—and evaluate them for accuracy and interpretability. We train with various hyperparameters and pick the configurations based on performance on a heldout set. We compare our CBMs with recently proposed label-free approaches: LF-CBM [36], LaBo [58], DCLIP [34] and CDM [41], and also report the linear probe and zero-shot performance of the CLIP model we use as a backbone for reference. We use the respective concept sets of each baseline method, and for a fair comparison, the same feature extractor across methods.

5.1 Classification Performance

In Tab. 1, we show the classification performance of our DN-CBM on four datasets and two feature extractors and compare them with the baselines. We find that DN-CBM is highly performant across datasets and backbones. Despite being task-agnostic, DN-CBM almost always outperforms the baselines, which use concept sets optimized for the downstream task, showing the generality of our approach. The highest gains are with Places365 (i.e. 52.70→53.53 pp. on ResNet-50 and 52.58→55.11 pp. on ViT-B/16), which is a scene-classification dataset rich in a wide variety of objects, which correspond to coarser, higher level concepts than e.g. body parts of animals as in ImageNet or CIFAR10, and are likely more well-represented in our concept space trained on CC3M.

5.2 Interpretability of DN-CBM

Local Explanations (Image-Level). In Fig. 7, we show qualitative examples of *local* explanations from our DN-CBM, i.e., explanations of individual decisions. For each image, we show the most contributing concepts along with their

Table 1: Performance of our CBM in comparison to prior work. We report the classification accuracy (%) of our CBM and baselines using CLIP ResNet-50 and ViT-B/16 feature extractors (ViT-L/14 in Appendix C) on ImageNet, Places365, CIFAR10, and CIFAR100. We find that our CBM performs competitively and often outperforms prior work, despite using a common set of concepts across datasets. '*' indicates results reported for the respective baselines, and zero-shot performance is as reported by [44].

Model	Task Agnostic	CLIP ResNet-50				CLIP ViT-B/16			
		IMN	Places	Cif10	Cif100	IMN	Places	Cif10	Cif100
Linear Probe	-	73.3*	53.4	88.7*	70.3*	80.2*	55.1	96.2*	83.1*
Zero Shot	-	59.6*	38.7	75.6*	41.6*	68.6*	41.2	91.6*	68.7*
LF-CBM [36]	✗	67.5	49.0	86.4*	65.1*	75.4	50.6	94.6	77.4
LaBo [58]	✗	68.9	-	**87.9***	**69.1***	78.9	-	95.7	81.2
CDM [41]	✗	72.2*	52.7*	86.5*	67.6*	79.3*	52.6*	95.3*	80.5*
DCLIP [34]	✗	59.6	37.9	-	-	68.0*	40.3*	-	-
DN-CBM (Ours)	✓	**72.9**	**53.5**	87.6	67.5	**79.5**	**55.1**	**96.0**	**82.1**

contribution strengths. We find that the concepts used are intuitive and class-relevant, thus aiding interpretability. The concepts used are also diverse, and include objects in the scenes (e.g. 'rocks' for 'swimming hole', top-right), similar features (e.g. 'toaster' given the corroded surface for 'junkyard', bottom-left), and high-level concepts (e.g. 'abandoned' for 'junkyard', bottom-left). Interestingly, we also find concepts associated with the class (e.g. 'alps' or 'everest' for 'glacier', top-left), which shows that the model's decision is also based on what a scene *looks like*, akin to ProtoPNets [12]. Finally, we observe that the concepts for predicting the same class change based on the contents of the image, e.g. in the bottom row, we find that despite both images depicting a junkyard where the most influential concept is 'corrosion', the second highest concepts are 'car' and 'tractor' respectively, reflecting the image contents.

In Fig. 8, we also compare explanations from DN-CBM with baselines (LF-CBM, CDM) on the same images from Places365. Interestingly, we find that our approach yields similarly convincing explanations as prior state-of-the-art CBM models, despite the fact that it does not use a task-specific vocabulary and extracts the concepts on a separate dataset (CC3M).

Global Explanations (Class-level). In Fig. 9, we show qualitative examples of *global* explanations from our DN-CBM, i.e., explanations of which concepts contribute the most to a class *as a whole*. To do this, for each class, we compute the average contribution of all concepts for images from that class, and visualize the set of top concepts. Qualitatively, we find this set to be semantically consistent with what is contained in each class.

5.3 Effectiveness of Concept Interventions

In addition to understanding model decisions, explanations have also been used to debug models [30] and fix models' reasoning [43, 45, 48]. Specifically, concept bottleneck models allow human interventions on individual concepts to control

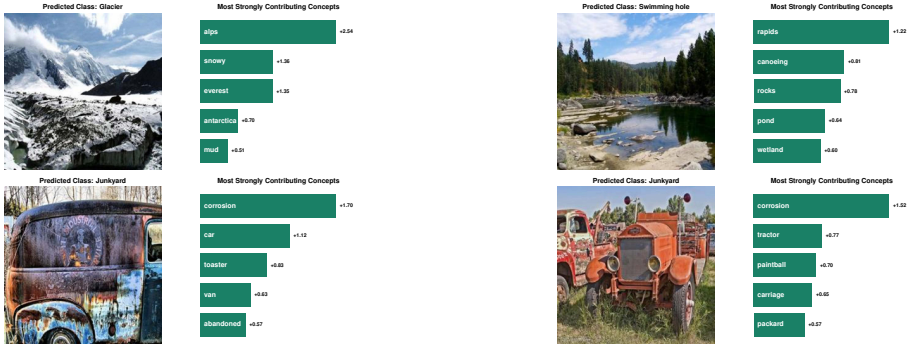

Fig. 7: Explaining decisions using our CBM. We show examples of images from the Places365 dataset along with the top concepts contributing to the decision. We find that our CBM classifies based on a diverse set of concepts present in the image, including objects, similar features, higher level concepts, and things associated with the class (e.g. similar locations), thus aiding interpretability.

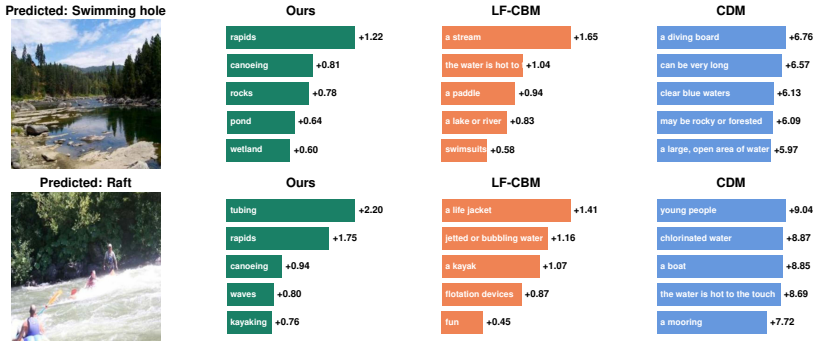

Fig. 8: Comparing interpretability across CBMs. We show an example from the Places365 dataset with explanations consisting of top contributing concepts using our CBM, LF-CBM [36], and CDM [41]. We find that our approach yields similar explanations despite not querying LLMs for concepts specific to the task and instead using a single task-agnostic concept bottleneck layer that is named post hoc.

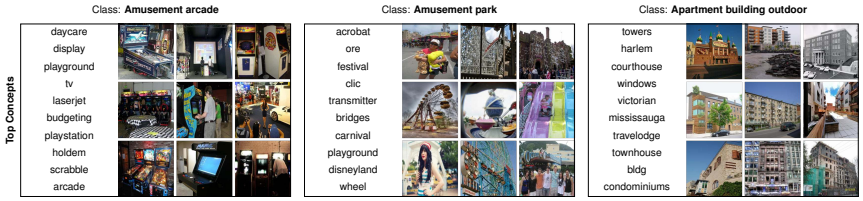

Fig. 9: Class-wise explanations of CBMs. We show examples of classes from the Places365 dataset with the top contributing concepts. For each class, we show random examples of images belonging to that class and select concepts with the highest average contribution across all images from the class in the validation set. We find that our approach yields classifiers that use concepts highly semantically related to each class.

the models' reliance on them. We assess the effectiveness of our DN-CBM with interventions by training on the Waterbirds-100 [43, 50] dataset. This contains

images of Landbirds and Waterbirds, with landbirds (waterbirds) on land (water) backgrounds during training, but without any such correlation in the test set. Following [43,46] we evaluate if intervening to (1) only keep bird related concepts, and (2) only remove such concepts increases (respectively, decreases) the performance on the worse group classification. To do this, we train a DN-CBM model that uses only five concepts for each class. For full details, see Appendix B.4.

In Tab. 2, we report the accuracy before and after the two interventions. We find that keeping only bird related concepts significantly improves the overall and worst group (Landbird on Water, Waterbird on Land) accuracies, with only a small drop in the other groups. Similarly, removing only such concepts leads to a large drop in accuracies, showing the effectiveness of interventions.

Table 2: Performance before and after intervening on the concept bottleneck model trained for the Waterbirds-100 dataset. We report the classification accuracy (%) on the full test set ('Overall') and each of the four groups (e.g. Landbird on Water, shown as 'L.Bird@W') before and after applying interventions. We find that intervening to only keep bird relevant concepts increases the overall and worst group [50] (Landbird on Water, Waterbird on Land) accuracy significantly, and conversely removing exactly these concepts leads to a large drop in accuracy, without adversely affecting performance on the groups in the training set ('Training Groups').

Model	Overall	Worst Groups		Training Groups	
		L.Bird@W	W.Bird@L	L.Bird@L	W.Bird@W
Before Intervention	82.8	71.3	57.5	**98.6**	**93.3**
Only Bird Concepts	**89.4** (+6.6)	**86.6** (+15.3)	**71.3** (+13.8)	96.8 (-1.8)	91.4 (-1.9)
No Bird Concepts	60.8 (-22.0)	28.5 (-42.8)	28.8 (-28.7)	95.0 (-3.6)	85.8 (-7.5)

6 Conclusion

In this work, we proposed Discover-then-Name CBM (DN-CBM), a novel CBM approach that uses sparse autoencoders to discover and automatically name concepts learnt by CLIP, and then use the learnt concept representations as a concept bottleneck and train linear layers for classification. We find that this simple approach is surprisingly effective at yielding semantically consistent concepts with appropriate names. Further, we find despite being task-agnostic, i.e. only extracting and naming concepts once, our approach can yield performant and interpretable CBMs across a variety of downstream datasets. Our results further corroborate the promise of sparse autoencoders for concept discovery. Training a more 'foundational' sparse autoencoder with a much larger dataset (e.g. at CLIP scale) and concept space dimensionality (with hundreds of thousands or millions of concepts) to obtain even more general-purpose CBMs, particularly for fine-grained classification, would be a fruitful area for future research.

Acknowledgements

Funded in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - GRK 2853/1 "Neuroexplicit Models of Language, Vision, and Action" - project number 471607914.

References

1. Achtibat, R., Dreyer, M., Eisenbraun, I., Bosse, S., Wiegand, T., Samek, W., Lapuschkin, S.: From Attribution Maps to Human-Understandable Explanations through Concept Relevance Propagation. Nature Machine Intelligence **5**(9), 1006–1019 (2023)
2. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity Checks for Saliency Maps. In: NeurIPS. vol. 31 (2018)
3. Adebayo, J., Muelly, M., Abelson, H., Kim, B.: Post Hoc Explanations may be Ineffective for Detecting Unknown Spurious Correlation. In: ICLR (2021)
4. Alukaev, D., Kiselev, S., Pershin, I., Ibragimov, B., Ivanov, V., Kornaev, A., Titov, I.: Cross-Modal Conceptualization in Bottleneck Models. In: EMNLP (2023)
5. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On Pixel-wise Explanations for Non-Linear Classifier Decisions by Layer-wise Relevance Propagation. PloS one **10**(7), e0130140 (2015)
6. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network Dissection: Quantifying Interpretability of Deep Visual Representations. In: CVPR. pp. 6541–6549 (2017)
7. Bhalla, U., Oesterling, A., Srinivas, S., Calmon, F.P., Lakkaraju, H.: Interpreting CLIP with Sparse Linear Concept Embeddings (SpLiCE). arXiv preprint arXiv:2402.10376 (2024)
8. Böhle, M., Fritz, M., Schiele, B.: B-cos Networks: Alignment is All We Need for Interpretability. In: CVPR. pp. 10329–10338 (2022)
9. Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J.E., Hume, T., Carter, S., Henighan, T., Olah, C.: Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. Transformer Circuits Thread (2023)
10. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language Models are Few-Shot Learners. In: NeurIPS. vol. 33, pp. 1877–1901 (2020)
11. Chauhan, K., Tiwari, R., Freyberg, J., Shenoy, P., Dvijotham, D.: Interactive Concept Bottleneck Models. In: AAAI (2023)
12. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This Looks Like That: Deep Learning for Interpretable Image Recognition. In: NeurIPS. vol. 32 (2019)
13. Chen, H., Yang, J., Vondrick, C., Mao, C.: Interpreting and Controlling Vision Foundation Models via Text Explanations. arXiv preprint arXiv:2310.10591 (2023)
14. Cunningham, H., Ewart, A., Riggs, L., Huben, R., Sharkey, L.: Sparse Autoencoders find Highly Interpretable Features in Language Models. arXiv preprint arXiv:2309.08600 (2023)
15. Dani, M., Rio-Torto, I., Alaniz, S., Akata, Z.: DeViL: Decoding Vision Features into Language. In: GCPR (2023)

16. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR. pp. 248–255 (2009)
17. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: ICLR (2021)
18. Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al.: Toy Models of Superposition. arXiv preprint arXiv:2209.10652 (2022)
19. Fel, T., Boutin, V., Béthune, L., Cadène, R., Moayeri, M., Andéol, L., Chalvidal, M., Serre, T.: A Holistic Approach to Unifying Automatic Concept Extraction and Concept Importance Estimation. In: NeurIPS. vol. 36 (2023)
20. Fel, T., Picard, A., Bethune, L., Boissin, T., Vigouroux, D., Colin, J., Cadène, R., Serre, T.: CRAFT: Concept Recursive Activation FacTorization for Explainability. In: CVPR. pp. 2711–2721 (2023)
21. Ghorbani, A., Wexler, J., Zou, J.Y., Kim, B.: Towards Automatic Concept-Based Explanations. In: NeurIPS. vol. 32 (2019)
22. Graziani, M., Nguyen, A.p., O'Mahony, L., Müller, H., Andrearczyk, V.: Concept Discovery and Dataset Exploration with Singular Value Decomposition. In: ICLRW (2023)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR. pp. 770–778 (2016)
24. Hendricks, L.A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., Darrell, T.: Generating Visual Explanations. In: ECCV. pp. 3–19. Springer (2016)
25. Hernandez, E., Schwettmann, S., Bau, D., Bagashvili, T., Torralba, A., Andreas, J.: Natural Language Descriptions of Deep Visual Features. In: ICLR (2022)
26. Jeyakumar, J.V., Dickens, L., Garcia, L., Cheng, Y.H., Echavarria, D.R., Noor, J., Russo, A., Kaplan, L., Blasch, E., Srivastava, M.: Automatic Concept Extraction for Concept Bottleneck-based Video Classification. arXiv preprint arXiv:2206.10129 (2022)
27. Kazmierczak, R., Berthier, E., Frehse, G., Franchi, G.: CLIP-QDA: An Explainable Concept Bottleneck Model. arXiv preprint arXiv:2312.00110 (2023)
28. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al.: Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In: ICML. pp. 2668–2677 (2018)
29. Kim, E., Jung, D., Park, S., Kim, S., Yoon, S.: Probabilistic Concept Bottleneck Models. In: ICML (2023)
30. Koh, P.W., Nguyen, T., Tang, Y.S., Mussmann, S., Pierson, E., Kim, B., Liang, P.: Concept Bottleneck Models. In: ICML. pp. 5338–5348 (2020)
31. Krizhevsky, A., Hinton, G., et al.: Learning Multiple Layers of Features from Tiny Images. Technical Report, Computer Science Department, University of Toronto (2009)
32. Lundberg, S.M., Lee, S.I.: A Unified Approach to Interpreting Model Predictions. NeurIPS **30** (2017)
33. Margeloiu, A., Ashman, M., Bhatt, U., Chen, Y., Jamnik, M., Weller, A.: Do Concept Bottleneck Models Learn as Intended? In: ICLRW (2021)
34. Menon, S., Vondrick, C.: Visual Classification via Description from Large Language Models. In: ICLR (2023)
35. Moayeri, M., Rezaei, K., Sanjabi, M., Feizi, S.: Text-to-Concept (and Back) via Cross-Model Alignment. In: ICML. pp. 25037–25060. PMLR (2023)

36. Oikarinen, T., Das, S., Nguyen, L.M., Weng, T.W.: Label-Free Concept Bottleneck Models. In: ICLR (2023)
37. Oikarinen, T., Weng, T.W.: CLIP-Dissect: Automatic Description of Neuron Representations in Deep Vision Networks. In: ICLR (2023)
38. O'Mahony, L., Andrearczyk, V., Müller, H., Graziani, M.: Disentangling Neuron Representations with Concept Vectors. In: CVPRW. pp. 3769–3774 (2023)
39. Panousis, K.P., Chatzis, S.: DISCOVER: Making Vision Networks Interpretable via Competition and Dissection. In: NeurIPS (2023)
40. Panousis, K.P., Ienco, D., Marcos, D.: Hierarchical Concept Discovery Models: A Concept Pyramid Scheme. arXiv preprint arXiv:2310.02116 (2023)
41. Panousis, K.P., Ienco, D., Marcos, D.: Sparse Linear Concept Discovery Models. In: ICCVW. pp. 2767–2771 (2023)
42. Patterson, G., Xu, C., Su, H., Hays, J.: The SUN Attribute Database: Beyond Categories for Deeper Scene Understanding. IJCV **108**(1-2), 59–81 (2014)
43. Petryk, S., Dunlap, L., Nasseri, K., Gonzalez, J., Darrell, T., Rohrbach, A.: On Guiding Visual Attention with Language Specification. In: CVPR. pp. 18092–18102 (2022)
44. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning Transferable Visual Models from Natural Language Supervision. In: ICML. pp. 8748–8763 (2021)
45. Rao, S., Böhle, M., Parchami-Araghi, A., Schiele, B.: Studying How to Efficiently and Effectively Guide Models with Explanations. In: ICCV. pp. 1922–1933 (2023)
46. Rao, S., Böhle, M., Schiele, B.: Towards Better Understanding Attribution Methods. In: CVPR. pp. 10213–10222 (2022)
47. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?" Explaining the Predictions of any Classifier. In: KDD. pp. 1135–1144 (2016)
48. Ross, A.S., Hughes, M.C., Doshi-Velez, F.: Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. In: IJCAI. pp. 2662–2670 (2017)
49. Roth, K., Kim, J.M., Koepke, A.S., Vinyals, O., Schmid, C., Akata, Z.: Waffling Around for Performance: Visual Classification with Random Words and Broad Concepts. In: ICCV. pp. 15746–15757 (October 2023)
50. Sagawa, S., Koh, P.W., Hashimoto, T.B., Liang, P.: Distributionally Robust Neural Networks. In: ICLR (2020)
51. Sawada, Y., Nakamura, K.: Concept Bottleneck Model with Additional Unsupervised Concepts. IEEE Access **10**, 41758–41765 (2022)
52. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In: ICCV. pp. 618–626 (2017)
53. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual Captions: A Cleaned, Hypernymed, Image Alt-Text Dataset for Automatic Image Captioning. In: ACL. pp. 2556–2565 (2018)
54. Shrikumar, A., Greenside, P., Kundaje, A.: Learning Important Features through Propagating Activation Differences. In: ICML. pp. 3145–3153. PMLR (2017)
55. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic Attribution for Deep Metworks. In: ICML. pp. 3319–3328. PMLR (2017)
56. Tewel, Y., Shalev, Y., Schwartz, I., Wolf, L.: ZeroCap: Zero-Shot Image-to-Text Generation for Visual-Semantic Arithmetic. In: CVPR. pp. 17918–17928 (2022)
57. Xu, X., Qin, Y., Mi, L., Wang, H., Li, X.: Energy-Based Concept Bottleneck Models: Unifying Prediction, Concept Intervention, and Conditional Interpretations. In: ICLR (2024)

58. Yang, Y., Panagopoulou, A., Zhou, S., Jin, D., Callison-Burch, C., Yatskar, M.: Language in a Bottle: Language Model Guided Concept Bottlenecks for Interpretable Image Classification. In: CVPR. pp. 19187–19197 (2023)
59. Yuksekgonul, M., Wang, M., Zou, J.: Post-hoc Concept Bottleneck Models. In: ICLR (2023)
60. Zhang, R., Madumal, P., Miller, T., Ehinger, K.A., Rubinstein, B.I.: Invertible Concept-based Explanations for CNN Models with Non-Negative Concept Activation Vectors. In: AAAI. pp. 11682–11690 (2021)
61. Zhao, C., Qian, W., Shi, Y., Huai, M., Liu, N.: Automated Natural Language Explanation of Deep Visual Neurons with Large Models. arXiv preprint arXiv:2310.10708 (2023)
62. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million Image Database for Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
63. Zhou, B., Sun, Y., Bau, D., Torralba, A.: Interpretable Basis Decomposition for Visual Explanation. In: ECCV. pp. 119–134 (2018)