Interpretable Object Recognition by Semantic Prototype Analysis

Qiyang Wan[1,2], Ruiping Wang[1,2], Xilin Chen[1,2]
[1]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
qiyang.wan@vipl.ict.ac.cn, {wangruiping, xlchen}@ict.ac.cn

Abstract

People can usually give reasons for recognizing a particular object as a specific category, using various means such as body language (by pointing out) and natural language (by telling). This inspires us to develop a recognition model with such principles to explain the recognition process to enhance human trust. We propose ***S****emantic* ***P****rototype* ***A****nalysis* ***Net****work (SPANet), an interpretable object recognition approach that enables models to explicate the decision process more lucidly and comprehensibly to humans by "pointing out where to focus" and "telling about why it is" simultaneously. With the proposed method, some* ***part prototypes*** *with* ***semantic concepts*** *will be provided to elaborate on the classification together with a group of* ***visualized samples*** *to achieve both part-wise and semantic interpretability. The results of extensive experiments demonstrate that SPANet is able to recognize objects almost as well as the non-interpretable models, at the same time generating intelligible explanations for its decision process.*

1. Introduction

With the rapid increase in performance of neural networks, recognition models have been widely applied in various fields. In recent years, an increasing number of researchers have concentrated their efforts on explainable AI (XAI), in order to deploy AI models towards real application scenarios with high-reliability requirements, such as autonomous driving, healthcare, transportation, security and other fields. Besides, previous work [11] states that when using recognition models, humans prefer a model with explanations over the non-interpretable one, because the provided explanations indeed help humans determine when to trust the AI systems and solve problems better.

Various methods have been proposed to significantly improve the interpretability of recognition models by generating diverse forms of explanation. Previous user study [11] shows that among the existing XAI approaches (explanations presented in heatmaps, examples, concepts and prototypes), **concepts** and **prototypes** are preferred by human users than other explanation forms. Concept-based explanations use class-agnostic, text-based concepts to deconstruct the categories. Prototype-based explanations explain the model with visualized samples, and usually focus on the local area of images. Concepts and prototypes are the most intuitive views that humans use to explain their reasoning and recognition [11].

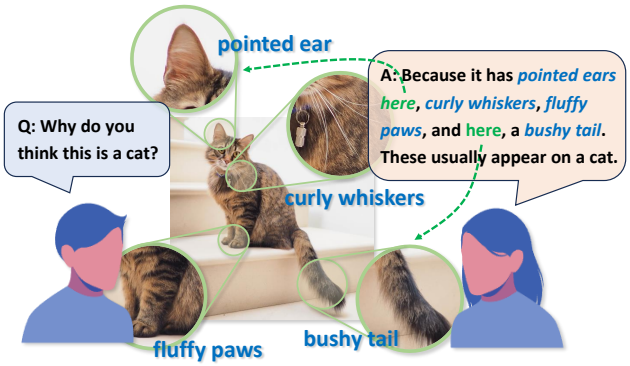

Figure 1. Example of the explanation provided by humans to recognize an object, in which "pointing out" (in green) and "telling" (in blue) are often used. Best in color viewing.

Consider what humans usually do when asked to explain a recognition process. As shown in Fig.1, when asked "*Why do you think this is a cat?*", we usually explain by **pointing out** where to focus and **telling** about why it is, such as "*Because this part is a pointed ear, and that part is a pair of fluffy paws. These features usually appear on a cat.*" The explanation not only involves highlighting regions of typical patterns, which are called **part prototypes**, but also provides some text descriptions, which are called **semantic concepts**. These two forms of explanation are related to XAI's part-wise and semantic interpretability respectively, and previous works always concentrate on one of them. The natural combination of these two forms of explanation in human communication inspires us to design an interpretable

model that takes both into account.

In this paper, an interpretable object recognition method, **S**emantic **P**rototype **A**nalysis **Net**work (SPANet), is proposed to recognize by **semantic prototypes**, which combine the part prototypes and semantic concepts to generate comprehensive explanations. As Fig.2 shows, with SPANet, local features of the input image are extracted and compared with the learned part prototypes to achieve recognition based on similarity. The learned part prototypes are labeled with semantic tags by a fine-tuned vision-language model for semantic reasoning similar to that in the human recognition, and are visualized by a retrieval-based reconstruction method to enhance the intrepretability.

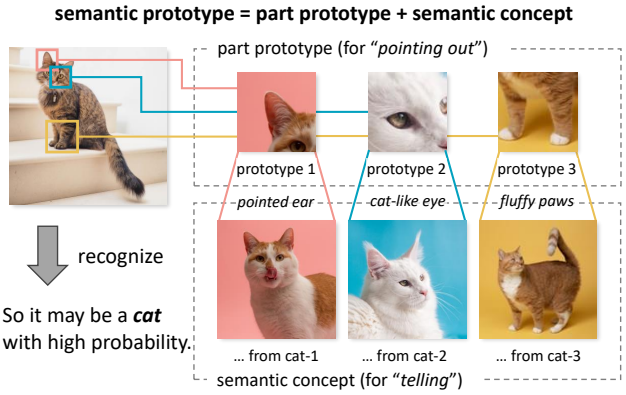

Figure 2. SPANet uses semantic prototypes to recognize an object to achieve both (prototype-based) part-wise and (concept-based) semantic interpretability.

Due to the significant differences when constructing two forms of explanations, there have been very few prior works that have successfully integrated both ideas and techniques. As far as we know, almost all of the current interpretable recognition methods only focus on one of these two aspects, either on part-wise interpretability or semantic interpretability, and SPANet represents an early endeavor to achieve these two forms at the same time. The case studies on bird species identification and car model recognition are conducted, and the quantitative results demonstrate that SPANet outperforms the compared interpretable recognition methods, and achieve comparetable results with the non-interpretable models. What's more, qualitative analyses verify that SPANet can effectively model the semantic concepts, and align them to the learned part prototypes. Our method shows protential influences on generating comprehensive explanations in the interpretable methods to build more reliable and practical recognition models.

2. Related Work

Post-hoc and self-explainable interpretability. According to the purpose of interpretable methods, they can be grouped into two types typically [16, 27]. The purpose of *post-hoc* interpretability [16, 27] is to analyze an existing model, which is usually able to be utilized on any DNN models and shows how they work. These works can be further divided into several categories, such as methods by attribution (gradient or relevance propagation) [2, 28, 34, 41], by perturbation (occlusion or counterfactual case generation) [7, 23], by example inversion or generation [14, 29, 39], and so on. In contrast, *self-explainable* models [27], or transparent models [16], are designed for intrinsic interpretability, which means that they can output the task results and the corresponding explanations simultaneously, including some prototype-based [3, 17, 18, 25, 26, 35] and concept-based [12, 19, 40, 42] methods. Almost all of the current self-explainable methods only focus on one of the explanation forms, while SPANet achieves part-wise interpretability based on prototypes and semantic interpretability based on concepts at the same time.

Prototype-based object recognition. In the recognition process of prototype-based methods, the distances between the image features and the learned prototypes of the categories (or concepts) are used. Prototype-based methods are traditionally used in the field of few-shot learning [30, 32]. The summarization ability of prototypes for categories makes them particularly suitable for scenarios that require efficient use of support sets. Recently, there are also many works using prototypes for interpretable object recognition, in which a prototype usually represents a region (or a patch) in an image. ProtoPNet [3] is proposed to find prototypical parts for each class and classify objects by combining evidence from prototypes. ProtoTree [18] conducts prototype learning with decision trees, and explain the recognition process by tracing the path through the tree. TesNet [35] constructs a transparent embedding space on Grassmann Manifold to replace the $L2$-distance in ProtoPNet. The proposed SPANet relies on semantic prototypical parts to recognize objects. In order to attach semantic to each learned part prototype, the local features are extracted from a vision-language model with multi-modal alignment, instead of a vanilla CNN used in previous works.

Concept learning. Concept learning is a fundamental process in cognitive science, which refers to learning attributes to distinguish exemplars from others among various categories [1]. In XAI, concept learning is generally used in both post-hoc interpretability and self-explainable models to generate meaningful, easy-to-understand explanations. For post-hoc interpretability, TCAV [10] learns meaningful concepts from the probe datasets to analyze an existing recognition model. Yeh et al. [38] investigate the *completeness* of a concept set and propose a new concept discovery method, as well as a metric (ConceptSHAP) to evaluate the importance of learned concepts. To design self-explainable models, Concept Bottleneck Models [12] learns human interpretable concepts by a bottleneck layer before

the last fully connected layer. CSG [15] trains a Class-Specific Gate layer to sparse the connections between concepts and categories. Zarlenga et al. [5] propose Concept Embedding Models to improve the accuracy of concept bottleneck models. In these works, concepts are always learned on the entire image, while SPANet tries to align the concepts with local regions so as to attach semantic labels on the learned part prototypes.

3. Method

In this section, we will introduce the proposed interpretable object recognition model. In Sec.3.1, the problem definition and some notations will be introduced first, and then the framework of the method will be presented. After that the specific designing details of each module will be shown in Sec.3.2, 3.3, and 3.4 respectively.

3.1. SPANet's framework

In a vanilla single-label object recognition task, a class label $\hat{y}_l$ is required to predict accroding to the input image x_l. Suppose that $\mathbf{X}$ represents all the image samples in the dataset, and $n = |\mathbf{X}|$ is the size of the dateset. $\mathbf{X}^{(c)} \subset \mathbf{X}$ refers to all the image samples of the certain category c. $x_l \in \mathbf{X}^{(c)}$ is one of the samples in the category c, whose class label y_l equals to c. Suppose that $C = \{c\}$ represents all the categories to be recognized.

In SPANet, there are three main functional modules: *Semantic Attachment Module*, *Prototype Recognition Module*, and *Reconstruction Module*. *Semantic Attachment Module* is able to encode images and texts to a common embedding space (semantic space), which has been implemented by many previous works such as CLIP [21]. *Prototype Recognition Module* is designed to assign one class label by comparing the image feature encoded in the *Semantic Attachment Module* with the stored part prototypes, and the semantic label of the nearest part prototype is provided simultaneously, which is used to generate the final explanation of the classification process. *Reconstruction Module* is not a part of the classification workflow, but is very important to the model interpretability, which can restore the semantic prototypes into the visual space to enable human-friendly explanations.

As Fig.3 shows, in the training stage a training pair (x, y) with corresponding semantic descriptions (such as class-level attributes or captions) is provided. Then the semantic descriptions are preprocessed to the input text r, which will be encoded to the text feature $z_T = f_T(r)$, with the input image x encoded to the global feature $z_I = f_I(x)$ and the local feature $z_L = f_L(x)$ by *Semantic Attachment Module*. The image features z_L can be used to train the following *Prototype Recognition Module* to generate the class label, which will be introduced in Sec.3.2, and all of them are used to learn the relation between the part-wise features and the specific semantic description, which will be introduced in Sec.3.3. Finally, a reconstruction method is proposed to reconstruct the part-wise feature to the corresponding image patch, in order to generate a visualized explanation in Sec.3.4.

3.2. Recognition by part prototype

In this section, the structure of SPANet's main branch, the recogniton module, will be elaborated. We use the most common framework of case-based and part-based interpretable object recognition method [3] as our baseline.

Given an input image x, the feature map $z_L \in \mathbb{R}^{h \times w \times d}$ is extracted by a local feature encoder f_L, such as a CNN or a Vision Transformer, in which h and w refer to the height and the width of the feature map, and d refers to the number of output channels. Each feature vector $\mathfrak{z}^{(i,j)} \in \mathbb{R}^d$ from the feature map z_L's position (i, j) can be traced back to a region (or a patch) on the input image, which can be regarded as the local feature of the region:

$$z_L = \left\{\mathfrak{z}^{(i,j)}\right\}_{i=1,j=1}^{(h,w)} = f_L(x) \in \mathbb{R}^{h \times w \times d} \tag{1}$$

For the semantic tags on the part prototypes in SPANet, in the semantic attachment step (see Sec.3.3), a vision-language pretrained model is used in the backbone instead of a visual feature extractor. The choice of vision-language models can be diverse, and in the following sections we will take CLIP [21] as an example. In CLIP, a global feature vector is obtained by its image encoder; however, a feature map is required in the part-based recognition. It is crucial to get local features with spatial information embedded while not losing the alignment to the text embeddings. We design some schemes for each backbone to modify the top layers of CLIP to obtain the required local features. More details are provided in the supplementary material.

SPANet learns m prototypes $\mathbf{P} = \{p_k\}_{k=1}^m$, and a prototype layer g is used to calculate the similarity between every feature vector $\mathfrak{z}^{(i,j)}$ on the feature map z_L and each prototype p_k. The prototype layer g gets a full size similarity map firstly, and then applys the max pooling along the spatial dimension (to get the "most similar score" to each prototype):

$$g(z_L; \mathbf{P}) = \max_{loc} sim(z_L, \mathbf{P}) = \max_{(i,j)} sim\left(\mathfrak{z}^{(i,j)}, \mathbf{P}\right) \tag{2}$$

For implementation, $g(z_L)$ calculates the $L2$ distance, and uses a monotonically decreasing transformation to transfer the distance to the similarity score $g(z_L; \mathbf{P}) \in \mathbb{R}^m$:

$$g(z_L; \mathbf{P})_k = \max_{(i,j)} \log \frac{\|\mathfrak{z}^{(i,j)} - p_k\|_2^2 + 1}{\|\mathfrak{z}^{(i,j)} - p_k\|_2^2 + \epsilon} \tag{3}$$

SPANet allocates m_c prototypes for each category c. We mark the allocation as $\mathbf{P}^{(c)} \subset \mathbf{P}$, so that $|\mathbf{P}^{(c)}| = m_c$.

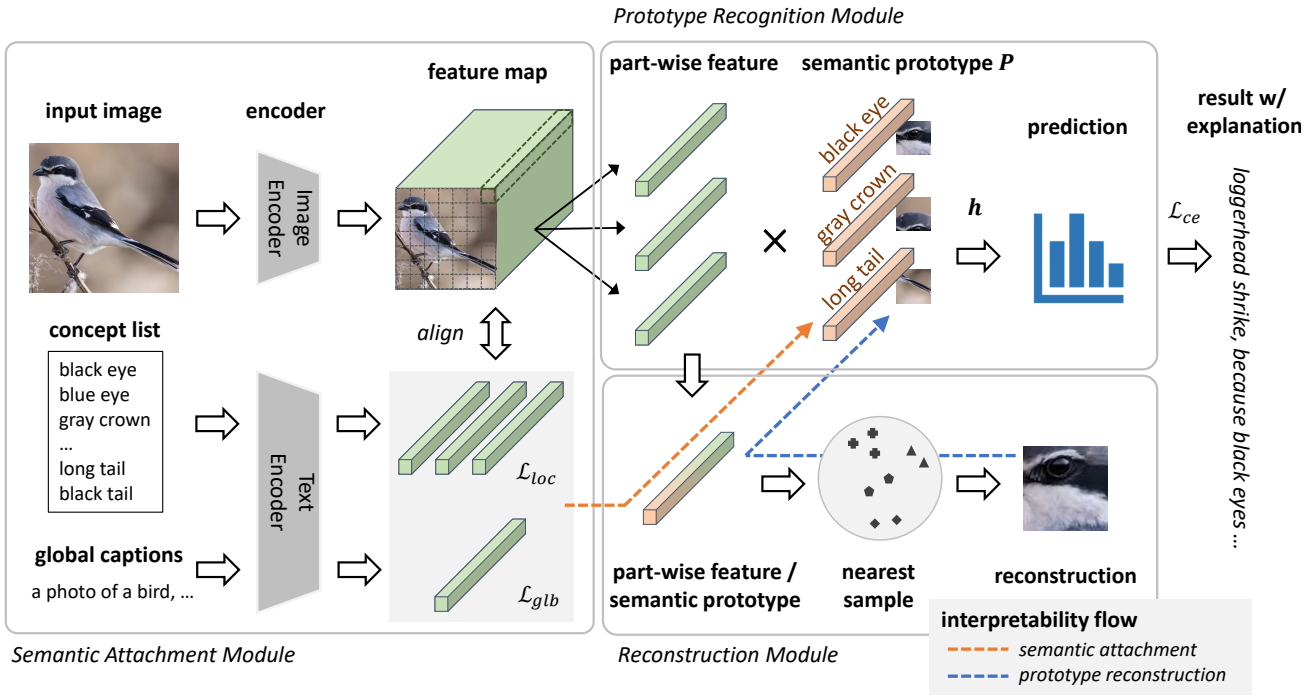

Figure 3. The framework of the proposed SPANet. The input image is encoded to a feature map by an image encoder, and the similarities between part-wise features in the feature map and the learned semantic prototypes are calculated to make final predictions (*Prototype Recognition Module*). The semantic prototypes are interpreted by both semantic tags and patch visualization, in which semantic tags are labeled by the fine-tuned vision-language model [21] (*Semantic Attachment Module*, the red dashed line), and the patch visualization of the prototypes comes from a retrieval-based, parameter-free reconstruction method (*Reconstruction Module*, the blue dashed line).

Then the input samples can be finally classified into a certain category by identifying which prototype its $g(z_L)$ has the highest similarity score with, and which category this most similar prototype belongs to. This function is implemented by an FC layer h (without a bias), and the output of this layer (after a *softmax* layer) can be regarded as the probability distribution over the categories to be classified:

$$h(g(z_L); \mathbf{W}_h) = \mathbf{W}_h \cdot g(z_L) \in \mathbb{R}^{|C|} \tag{4}$$

To optimize g and h, we follow the most loss functions in ProtoPNet [3], including a cross entropy loss ($\mathcal{L}_{ce}$) to constrain the final classification results, a cluster loss ($\mathcal{L}_{clst}$) to minimize the distance between the nearest (local feature $\mathfrak{z}^{(i,j)}$, part prototype p_k from the **positive** class) pair, and a separation loss ($\mathcal{L}_{sep}$) to maximize the distance between the nearest (local feature $\mathfrak{z}^{(i,j)}$, part prototype p_k from the **negative** classes) pair. Across the training set $\mathcal{X} = \{(x_l, y_l)\}_{l=1}^{n}$, the losses can be calculated as:

$$\mathcal{L}_{ce} = \frac{1}{n}\sum_{l=1}^{n} CE\left(h \circ g \circ f_L(x_l), y_l\right) \tag{5}$$

$$\mathcal{L}_{clst} = \frac{1}{n}\sum_{l=1}^{n} \min_{p_k \in \mathbf{P}_{y_i}} \min_{(i,j)} \|\mathfrak{z}_l^{(i,j)} - p_k\|_2^2 \tag{6}$$

$$\mathcal{L}_{sep} = -\frac{1}{n}\sum_{l=1}^{n} \min_{p_k \notin \mathbf{P}_{y_i}} \min_{(i,j)} \|\mathfrak{z}_l^{(i,j)} - p_k\|_2^2 \tag{7}$$

Besides, we find that there should be an $L2$ penalization on the part prototypes $\mathbf{P}$ to prevent overfitting:

$$\mathcal{L}_{L2} = \sum_{p_k \in \mathbf{P}} \|p_k\|_2^2 \tag{8}$$

Finally, several losses are weighted and summed to obtain our final optimization objective of the prototype recognition module:

$$\mathcal{L}_{pt} = \mathcal{L}_{ce} + \lambda_{clst}\mathcal{L}_{clst} + \lambda_{sep}\mathcal{L}_{sep} + \lambda_{L2}\mathcal{L}_{L2} \tag{9}$$

3.3. Semantic attachment

It is important to attach semantic meanings on the learned part prototypes, or these part prototypes are presented in numerical vector form and extremely unintelligible to humans. We use the *Semantic Attachment Module* to "label" the part prototypes with semantic labels, which we refer to as the process of *semantic attachment*. Just like in the previous section, we will take a vanilla CLIP [21] as an example of a vision-language model, and then fine-tune it to adapt to our data domain. Due to the modifications made to CLIP's visual backbone (referring to Sec.3.2), it can directly obtain the local features of images without any extra parameters, which means it does not require any training or fine-tuning process if the parameters in the backbone are frozen. These obtained local features can be directly matched with the semantic text features.

However, as the algorithm may be applied to highly specialized domains, a pretrained vision-language model may be ill-suited for real, fine-grained application scenario. Hence, if additional semantic data are available, preliminary adjustments can be implemented to adapt the model to the target domain, which we term as *semantic fine-tuning*. Various strategies of fine-tuning can be performed depending on the type of additional data provided, such as *global fine-tuning* and *local fine-tuning*.

Global fine-tuning for image-text pairs. If the image-text pairs are provided (in the most cases when the vision-language pretraining), the *global fine-tuning* can be conducted. The goal of *global fine-tuning* is to enable CLIP to achieve correspondence between images and descriptive text in the specific data domain. The fine-tuning process is identical to that of vanilla CLIP, but different data are used. Across the training set with image-text pairs $\mathcal{X} = \{(x_l, r_l)\}_{l=1}^{n}$, in a training batch $\mathcal{B}$, the global fine-tuning loss $\mathcal{L}_{glb}$ can be represented as a single-label contrastive loss in Equation 10:

$$\mathcal{L}_{glb} = -\frac{1}{n}\sum_{l=1}^{n}\log\frac{\exp\left[s_c\left(z_{I,l}, z_{T,l}\right)/\tau\right]}{\sum_{(x_j,r_j)\in\mathcal{B}}\exp\left[s_c\left(z_{I,l}, z_{T,j}\right)/\tau\right]}, \tag{10}$$

where

$$s_c\left(z_I, z_T\right) = \frac{z_I \cdot z_T}{\|z_I\|_2 \cdot \|z_T\|_2}.$$

Note that here we use the original **global** feature $z_I = f_I(x)$ in the pretrained model, instead of the **local** feature $z_L = f_L(x)$ generally used in Sec.3.2. Besides, the text embedding $z_T = f_T(r)$ is used to align the visual feature and text. s_c is the cosine similarity function, which will also be used in the following sections.

Local fine-tuning for image-concepts pairs. If the image-concepts pairs are provided, the *local fine-tuning* can be presented. Sometimes *attributes* are confused with *concepts*, but *attributes* are actually just one type of *concepts*. The definition of *concept* should be broader, including any abstract class-agnostic semantic "building blocks" able to deconstruct visual categories. Here the form of *concepts* labeled on the image can be a list of text tags. The purpose of *local fine-tuning* is to enable CLIP to align specific *concepts* with local features extracted from the input image.

We use $\mathbf{A} = \{a_k\}$ to represent the set of all the concepts defined in the datasets, and use $\mathbf{A}^{(l)} \subset \mathbf{A}$ to represent the set of concepts labeled on the sample x_l, which means that a training set consisting of image-concepts pairs can be represented as $\mathcal{X} = \left\{\left(x_l, \mathbf{A}^{(l)}\right)\right\}_{l=1}^{n}$. All the concepts are labeled on the entire image, and no extra local keypoints or bboxes are used. Just like what we do in Sec.3.2, the similarities between every feature vector $\mathfrak{z}_l^{(i,j)}$ and each concept embedding $f_T(a_k)$ are calculated, and then max-pooled across the spatial locations:

$$\begin{aligned} g'\left(z_L; \mathbf{A}\right) &= \max_{(i,j)} s_c\left(z_L, f_T\left(\mathbf{A}\right)\right) \\ &= \max_{(i,j)} s_c\left(\mathfrak{z}^{(i,j)}, f_T\left(a_k\right)\right) \in \mathbb{R}^{|\mathbf{A}|} \end{aligned} \tag{11}$$

where $g'\left(z_L; \mathbf{A}\right)$ indicates the similarity between the highest activated vector in the feature map z_L and the text embedding $f_T\left(a_k\right)$ of each concept a_k. Then when the positive concept set $\mathbf{A}^{(l)}$ for the image x_l is given, we can regard the concept prediction as a multi-label classification task, that is fitting $\mathbf{A}$ with the output $g'\left(z_L; \mathbf{A}\right)$.

Considering that CLIP is pretrained by contrastive learning, we also use a multi-label contrastive loss [8] $\mathcal{L}_{loc}$ to optimize the objective of multi-label concept prediction. $\mathcal{L}_{loc}$ are defined in Equation 12, in which the most-activated feature vector is pulling closer to the embedding of positive concept $a_j \in \mathbf{A}^{(l)}$, and pushing away from the embedding of negative concept $a_k \in \mathbf{A}\backslash\mathbf{A}^{(l)}$:

$$\mathcal{L}_{loc} = -\frac{1}{n}\sum_{l=1}^{n}\frac{1}{|\mathbf{A}^{(l)}|} \sum_{a_j\in\mathbf{A}^{(l)}}\log\left[\frac{\exp\left[g'\left(z_{L,l}; \mathbf{A}\right)_j/\tau\right]}{\sum_{a_k\in\mathbf{A}\backslash\mathbf{A}^{(l)}}\exp\left[g'\left(z_{L,l}; \mathbf{A}\right)_k/\tau\right]}\right] \tag{12}$$

where $g'\left(z_{L,l}; \mathbf{A}\right)_j$ means the j-th value in the $g'\left(z_{L,l}; \mathbf{A}\right)$, namely the similarity between the nearest local feature and the embedding of the concept a_j. As Fig.4 shows, the multi-label contrastive loss can also be visually interpreted as that, the concept embeddings are "solid anchors" in the feature space, and multi-label input samples are pulled closely or pushed away to the anchors depending on their ground truth label.

3.4. Semantic prototype reconstruction

To achieve visual part-wise interpretation, restoring semantic prototypes from the "invisible" feature space to the "visible" image space is necessary. This process is referred to as *semantic prototype reconstruction*. Many different reconstruction or generative technologies can be applied, such as reconstruction methods used in segmentation tasks (U-Net [24]), variational vutoencoders (VQ-VAE [31]), or generative adversarial networks [6]. These reconstruction algorithms play similar roles in SPANet. However, during the exploration, we find that both generation-based methods and reconstruction-based methods each have their own set of issues: due to the extreme locality of semantics in the part prototype, generation-based methods either are with faithfulness but in poor rendering quality, or in good rendering quality but fabricate non-existent details; reconstruction-based methods are affected by input images and cannot

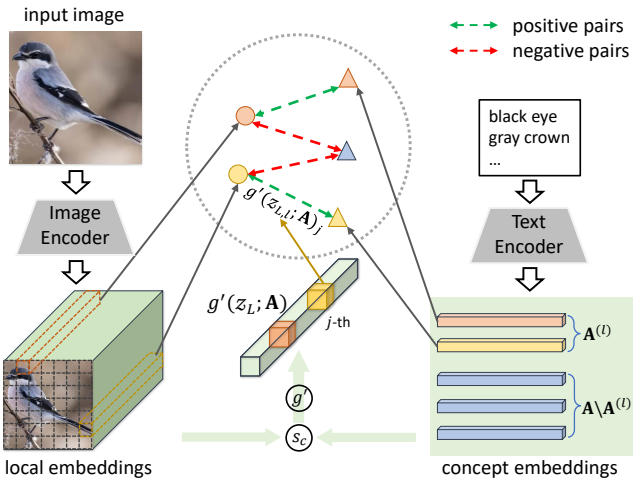

Figure 4. Illustration of the multi-label contrastive loss used in the *local fine-tuning*. With the multi-label contrastive loss, the image patch embeddings from feature map can be aligned to the corresponding text (concept) embeddings.

completely restore the visual images from the semantic prototypes only by their own representations. We believe that these phenomena are due to current technological limitations or the imperfect implementation; however, in order to ensure the faithfulness of semantic prototype restoration, we choose to fallback and adopt a retrieval method as our reconstruction module, that guarantees semantic faithfulness at the expense of slight loss of restoration accuracy.

Assuming that semantic prototypes $\mathbf{P}$ are fully trained, for each semantic prototype $p_k \in \mathbf{P}$, a reconstruction image patch I_k will be retrieved from the training set $\mathbf{X} = \{x_i\}$:

$$\begin{aligned} I_k &= \text{image\_patch}\left(x_l, (i,j)\right) \\ (x_l, i, j) &= \text{argmin}_{(x_l,i,j)} \left\| \mathfrak{z}_l^{(i,j)} - p_k \right\|_2^2 \end{aligned} \tag{13}$$

In Equation 13, $\text{image\_patch}\left(x, (i,j)\right)$ is to obtain the patch at position i of row and j of column on the image x.

4. Case study

In this section, we firstly introduce how to prepare the dataset in the case studies (Sec.4.1) and some implementation details (Sec.4.2). Then qualitive and quantitative results on both recognition accuracy and interpretability are shown with the corresponding analysis (Sec.4.3, 4.4). In the end, ablation study is conducted to analyze different modules in SPANet to demonstrate their effectiveness (Sec.4.5).

4.1. Data preparation

For bird species identification, Caltech-UCSD Birds-200-2011 (CUB) [33] is what we used to evaluate SPANet, which contains 5,994 training images and 5,794 test images from 200 bird species. For car model identification, experiments are conducted on Stanford Cars [13], which includes 8,144 training samples and 8,041 test samples of 196 car models. As in previous works, we augment the training samples by a factor of 40 using rotation, skew, shear, random distortion, and random erasing.

As stated previously, since CUB and Stanford Cars are designed for fine-grained recognition, fine-tuning the vision-language model is essential. We conduct both global fine-tuning and local fine-tuning on them, in which image-text pairs and image-concepts pairs are required. The image-text pairs on CUB can be obtained in labeled captions [22, 36]. As concepts, following the settings in CBM [12], 112 binary attributes are selected by majority voting, that if more than 50% samples in a category share one attribute, then the attribute is labeled positive to all samples in this category. For Stanford Cars, both global captions and category concepts are obtained from GPT-4 [20]. Similar to the attributes in CompCars [37], 20 concepts related to body shape, headlights, doors, etc., are automatically generated for local fine-tuning. In the supplementary material, more details, data examples, and extracted concepts are provided.

4.2. Implementation details

We conduct empirical experiments on different backbone models, including ResNet-50 and ResNet-101 [9] for CNN backbones, and ViT-B/32 and ViT-B/16 [4] for transformer backbones. The pretrained weights are all from CLIP [21]. A two-stage training strategy is used: In the warm-up epoches, the parameters in the backbone are frozen, and only the semantic prototypes are optimized. Then in the joint-training epoches, all the parameters in SPANet are free to be optimized, but the backbone and the semantic prototypes have different learning rates. The global fine-tuning loss $\mathcal{L}_{glb}$ and local fine-tuning loss $\mathcal{L}_{loc}$ are added into the final optimazation loss with the respective weight $\lambda_{glb} = 0.1$ and $\lambda_{loc} = 0.1$. More hyperparameters are listed in the supplementary material.[1]

4.3. Recognition accuracy

The recognition results of SPANet with different CNN and transformer architectures are shown in Table 1. We compare SPANet with several representative methods from different groups based on their interpretability, including *non-interpretable* baselines (vanilla CNNs and transformers), *part-wise interpretable* methods (ProtoPNet [3], etc.), *semantic interpretable* methods (CBM [12], etc.).

When providing part-wise explanation and semantic explanation simultaneously, SPANet limits the decline of performance into an acceptable level compared with non-interpretable baselines. Especially, due to the excellent compatibility, SPANet can take into account different backbone architectures. It can take advantage of the performance benefits of transformers to outperform that with

[1]The code is available at https://github.com/WanQiyang/SPANet.

Table 1. Recognition accuracy on CUB and Stanford Cars. Column "Interpret." refers to the type of interpretability. "RN" is short for ResNet, and "Inc.-v3" is short for Inception-v3.

Method	Interpret.	Backbone	CUB	Cars
Vanilla CNN/ViT	None	RN50	84.5	86.3*
		RN101	83.5	91.2
		ViT-B/32	82.3*	89.5*
		ViT-B/16	89.4	93.7
ProtoPNet [3]	Part-wise	RN34	79.2	86.1
		RN50	79.4*	87.9*
		RN101	77.3*	87.6*
ProtoPShare [26]	Part-wise	RN34	74.7	86.4
TesNet [35]	Part-wise	RN34	82.8	90.9
ProtoPool [25]	Part-wise	RN34	80.3	89.3
ProtoPool [25]	Part-wise	RN50	85.5	88.9
ProtoTree [18]	Part-wise	RN50	82.2	86.6
PIP-Net [17]	Part-wise	RN50	82.0	86.5
CBM [12]	Semantic	Inc.-v3	80.1	-
		RN18	62.9	-
PCBM [40]	Semantic	RN18	58.8	-
LFCBM [19]	Semantic	RN18	74.3	-
SPANet (Ours)	Both	RN50	81.7	88.9
		RN101	77.7	85.6
		ViT-B/32	83.0	90.3
		ViT-B/16	87.2	93.7

* indicates results reproduced based on official codes.

CNN backbones. We believe that the form of image patch used in Vision Transformer [4] is more suitable as part prototypes than traditional CNN, which is the key to achieving better performance.

For qualitative analysis, as shown in Fig.5, SPANet is able to generate the comprehensible explanations in part prototypes with their semantic labels. Some visualization of the semantic prototypes is shown in Fig.6.

4.4. Evaluation of interpretability

To evaluate the interpretability of the methods, we conduct a user study among ProtoPNet [3], ProtoTree [18], and SPANet. We invite 15 participants, ranging from ordinary users to experts, to complete the user study. In the experiments, about 17% test samples are randomly selected from CUB to be recognized, and corresponding explanations are generated by these methods. The participants are asked to click the explanation that they think most convincing (including a "None" option to represent "none of them are satisfying"). The screenshot of the user interface is shown in the supplementary material. The answers are collected and summarized as shown in Table 2.

The user study is conducted in two settings, in which different explanations are provided by SPANet. "w/o sem." means that no semantic labels are attached on part prototypes, and "w/ sem." means that sematic labels (such as "color: black") are provided. The results in Table 2 demonstrate that interpretability of SPANet with semantic labels is the most satisfying among these methods. Surprisingly, ProtoTree [18] based on decision trees, which we believe to be elegant and sophisticated, does not seem to be favored by most of participants. Based on the results and the conversations with participants, we draw the following conclusions:

1. Semantic labels are **helpful**. A part prototype with **correct** semantic labels is more convincing.
2. When the attached semantic labels are incorrect, **even partially incorrect**, it will greatly reduce the trust of the participants in the explanations.
3. Participants are more inclined to trust explanations based on **positive** factors. The frequent use of "absent" prototypes by ProtoTree [18] without any additional semantic information may confuse the participants.

The second point is the main reason why SPANet only marginally outperforms ProtoPNet, because prediction errors sometimes occur when attaching the semantic labels. For SPANet, simple concepts used in previous study such as "**color: black**" are easier to predict, but the attachment of complex concepts including part names (such as "**tail** color: black") may be inaccurate due to weak supervision on part location. Thus, in our extra study, if the semantic labels including these complex concepts with some wrong part names are provided, users' satisfaction of SPANet will decrease by about 0.1. Nevertheless, all participants believe that semantic labels are beneficial, and more accurate semantic attachment will further enhance the interpretability.

Table 2. Interpretability evaluation on a subset of CUB. A higher value indicates higher satisfaction, with each row's total sum being 1. "sem." is short for semantic interpretability, which means the semantic labels generated by SPANet for part prototypes.

	ProtoPNet	ProtoTree	SPANet	None
w/o sem.	0.36	0.17	0.33	0.14
w/ sem.	0.37	0.13	**0.43**	0.07

Table 3. The results of ablation study on CUB. SPANet with both global and local fine-tuning is able to bridge the gap between global captions and concept texts, making more efficient use of semantic information and achieving better performance.

Settings (ResNet-50)	Accuracy
SPANet (frozen backbone)	53.9
SPANet (base w/o semantic intp.)	78.5
SPANet (w/ global fine-tuning)	79.7
SPANet (w/ local fine-tuning)	79.6
SPANet (w/ global & local fine-tuning)	**81.7**

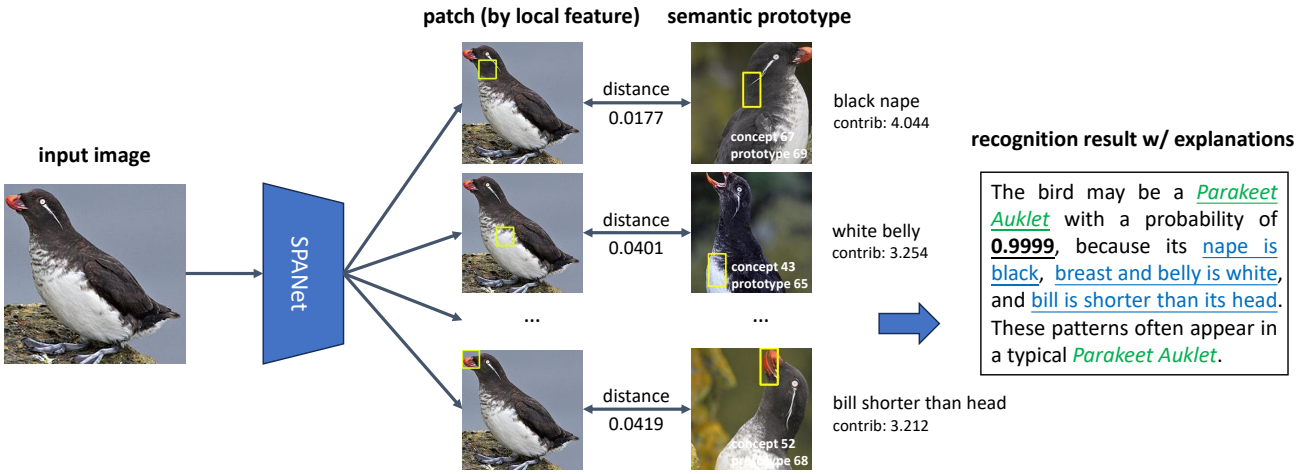

Figure 5. Example of the recognition process of SPANet. The distances between local features of input image and learned semantic prototypes are calculated, and the (local feature, semantic prototype) pair contributes positively to the category that the protoype belongs to and negatively to the other categories. The recognition result with explanations in natural language is provided as a result.

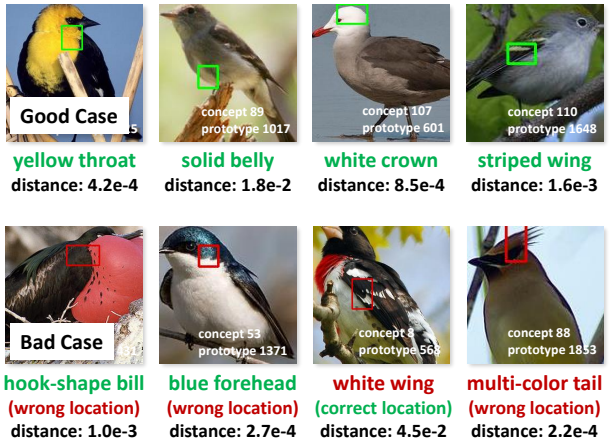

Figure 6. Visualization of the semantic prototypes from the trained SPANet. Correctly predicted concepts and the corresponding locations are indicated in green, while incorrect ones are in red.

4.5. Ablation study

The results of ablation study on CUB are listed in Table 3. The module of prototype-based recognition and its losses have been thoroughly validated in previous works. Therefore, we focus on the validation of the proposed module of semantic fine-tuning and attachment.

As shown in Table 3 and mentioned in Sec.3.3, the perfromance on the fine-grained datasets (such as CUB) of using the pretrained vision-language model directly, which is pretrained on large amounts of image and text data in various generic scenarios, is not satisfactory (see "SPANet w/ frozen backbone"). SPANet can also be implemented without any semantic interpretability, which is similar to ProtoPNet [3]'s structure, and the results are also very similar to ProtoPNet [3] with the same backbone. Additionally, global fine-tuning and local fine-tuning not only enable the model with semantic interpretability, but also improve the recognition accuracy to some extent by introducing cross-modal knowledge from language. However, relying solely on image-text pairs is difficult to obtain the ability of local semantic alignment, while relying solely on image-concepts pairs makes the corpus scope extremely limited (limited to some attribute vocabulary). Combining the both can integrate their advantages comprehensively. Therefore, the complete SPANet can better integrate knowledge from natural language and achieve optimal recognition performance.

5. Conclusion

In this paper, we focus on interpretable object recognition and propose SPANet to generate both part-wise explanations and semantic labels through the learned semantic prototypes. Consequently, SPANet can provide intelligible and comprehensible explanations for its decision process, and its effectiveness has been demonstrated in empirical experiments. However, some limitations still exist and more improvements could be considered in the future work: enhancing the effectiveness of semantic attachment can further improve interpretability, and the opacity of the complicated backbone also requires more attention. Nevertheless, the user study shows that the explanations in the form of semantic prototypes are promising and worth further development. We call for more attention to the user-friendly explanations to help human make precise decision in the scenarios with high-reliability requirements.

Acknowledgements

This work is partially supported by National Key R&D Program of China No. 2021ZD0111901, and Natural Science Foundation of China under contracts Nos. U21B2025, U19B2036. We are grateful to Bin Fu for many helpful discussions.

References

[1] Jerome Seymour Bruner and George Allen Austin. *A study of thinking*. Transaction publishers, 1986. 2

[2] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018. 2

[3] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 2, 3, 4, 6, 7, 8

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 6, 7

[5] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lió, and Mateja Jamnik. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 21400–21413, 2022. 3

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 5

[7] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning (ICML)*, pages 2376–2384. PMLR, 2019. 2

[8] Rohit Gupta, Anirban Roy, Claire Christensen, Sujeong Kim, Sarah Gerard, Madeline Cincebeaux, Ajay Divakaran, Todd Grindal, and Mubarak Shah. Class prototypes based contrastive learning for classifying multi-label and fine-grained educational videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19923–19933, 2023. 5

[9] 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 (CVPR)*, pages 770–778, 2016. 6

[10] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 2668–2677. PMLR, 2018. 2

[11] Sunnie SY Kim, Elizabeth Anne Watkins, Olga Russakovsky, Ruth Fong, and Andrés Monroy-Hernández. "help me help the ai": Understanding how explainability can support human-ai interaction. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2023. 1

[12] 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 (ICML)*, pages 5338–5348. PMLR, 2020. 2, 6, 7

[13] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV 2013 Workshop on 3D Representation and Recognition (3dRR-13)*, 2013. 6

[14] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T. Freeman, Phillip Isola, Amir Globerson, Michal Irani, and Inbar Mosseri. Explaining in style: Training a gan to explain a classifier in stylespace. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 693–702, 2021. 2

[15] Haoyu Liang, Zhihao Ouyang, Yuyuan Zeng, Hang Su, Zihao He, Shu-Tao Xia, Jun Zhu, and Bo Zhang. Training interpretable convolutional neural networks by differentiating class-specific filters. In *European Conference on Computer Vision (ECCV)*, pages 622–638. Springer, 2020. 3

[16] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018. 2

[17] Meike Nauta, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. Pip-net: Patch-based intuitive prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2744–2753, 2023. 2, 7

[18] Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14933–14943, 2021. 2, 7

[19] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 2, 7

[20] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 3, 4, 6

[22] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning (ICML)*, pages 1060–1069. PMLR, 2016. 6

[23] Jie Ren, Zhanpeng Zhou, Qirui Chen, and Quanshi Zhang. Can we faithfully represent absence states to compute shap-

ley values on a dnn? In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 2

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, pages 234–241. Springer, 2015. 5

[25] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. In *European Conference on Computer Vision (ECCV)*, pages 351–368. Springer, 2022. 2, 7

[26] Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, 2021. 2, 7

[27] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021. 2

[28] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. 2

[29] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9374–9384, 2021. 2

[30] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. 2

[31] Aaron Van Den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. 5

[32] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016. 2

[33] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. Caltech-ucsd birds-200-2011 (cub-200-2011). Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 6

[34] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *CVPR 2020 Workshop on Fair, Data-Efficient and Trusted Computer Vision*, 2020. 2

[35] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 895–904, 2021. 2, 7

[36] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1316–1324, 2018. 6

[37] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 6

[38] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 20554–20565, 2020. 2

[39] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8715–8724, 2020. 2

[40] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR 2022 Workshop on PAIR^2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022. 2, 7

[41] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016. 2

[42] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. 2