

Evaluation and Improvement of Interpretability for Self-Explainable Part-Prototype Networks

Qihan Huang¹, Mengqi Xue¹, Wenqi Huang², Haofei Zhang¹,
Jie Song^{1,3,†}, Yongcheng Jing⁴, Mingli Song¹

¹Zhejiang University, ²Digital Grid Research Institute, China Southern Power Grid,

³Zhejiang University - China Southern Power Grid Joint Research Centre on AI,

⁴The University of Sydney

{qh.huang,mqxue,haofeizhang,sjie,brooksong}@zju.edu.cn,

huangwqcsq@163.com, yjin9495@uni.sydney.edu.au

Abstract

Part-prototype networks (e.g., ProtoPNet, ProtoTree, and ProtoPool) have attracted broad research interest for their intrinsic interpretability and comparable accuracy to non-interpretability counterparts. However, recent works find that the interpretability from prototypes is fragile, due to the semantic gap between the similarities in the feature space and that in the input space. In this work, we strive to address this challenge by making the first attempt to quantitatively and objectively evaluate the interpretability of the part-prototype networks. Specifically, we propose two evaluation metrics, termed as “consistency score” and “stability score”, to evaluate the explanation consistency across images and the explanation robustness against perturbations, respectively, both of which are essential for explanations taken into practice. Furthermore, we propose an elaborated part-prototype network with a shallow-deep feature alignment (S DFA) module and a score aggregation (SA) module to improve the interpretability of prototypes. We conduct systematical evaluation experiments and provide substantial discussions to uncover the interpretability of existing part-prototype networks. Experiments on three benchmarks across nine architectures demonstrate that our model achieves significantly superior performance to the state of the art, in both the accuracy and interpretability. Our code is available at <https://github.com/hqhQAQ/EvalProtoPNet>.

1. Introduction

Part-prototype networks are recently emerged deep self-explainable models for image classification, which achieve

[†] Corresponding author.

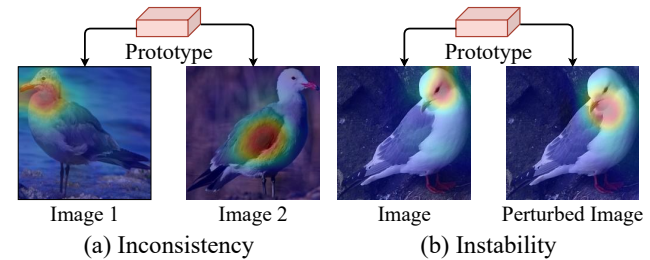


Figure 1. (a) **Inconsistency**. A prototype may mistakenly correspond to different object parts in different images. (b) **Instability**. A prototype may mistakenly correspond to different object parts in the original image and the slightly perturbed image. The samples are from ProtoPNet trained on ResNet34 backbone [12].

excellent performance in an interpretable decision-making manner. In particular, ProtoPNet [7] is the first part-prototype network, with the follow-up part-prototype networks (e.g., ProtoTree [31], ProtoPool [33], TesNet [44], and ProtoP-Share [34]) built upon its framework. At its core, part-prototype networks define multiple trainable prototypes that represent specific object parts and emulate human perception by comparing object parts across images to make predictions. Currently, part-prototype networks have been extended to various domains (e.g., graph neural network [54], deep reinforcement learning [19], and image segmentation [35]).

However, the current part-prototype networks only demonstrate their interpretability with only a few visualization examples, and ignore the problem that the learned prototypes of part-prototype networks do not have adequately credible interpretability [13, 20, 36]. The reasons for such unreliable interpretability are twofold: (1) **Inconsistency**. The basic design principle of part-prototype networks [7] is that *each prototype is associated with a specific object part*, but it is not guaranteed that the corresponding object part of a

prototype is consistent across images, as shown in Fig. 1 (a); **(2) Instability.** Previous interpretability methods [1, 50, 55] claim that the explanation results should be stable, but the prototype in part-prototype networks is easily mapped to a vastly different object part in a perturbed image [13], as shown in Fig. 1 (b). Recently, Kim *et al.* [20] have proposed a human-centered method named HIVE to evaluate the interpretability of part-prototype networks. Nevertheless, HIVE requires redundant human interactions and the evaluation results are subjective. Therefore, for the further research on the part-prototype networks, there is an urgent need for more formal and rigorous evaluation metrics that can quantitatively and objectively evaluate their interpretability.

In this work, we strive to take one further step towards the interpretability of part-prototype networks, by making the first attempt to quantitatively and objectively evaluate the interpretability of part-prototype networks, rather than the qualitative evaluations by several visualization examples or subjective evaluations from humans. To this end, we propose two evaluation metrics named “consistency score” and “stability score”, corresponding to the above inconsistency and instability issues. Specifically, the consistency score evaluates whether and to what extent a learned prototype is mapped to the same object part across different images. Meanwhile, the stability score measures the robustness of the learned prototypes being mapped to the same object part if the input images are slightly perturbed. In addition, our evaluation metrics generate objective and reproducible evaluation results using object part annotations in the dataset. With the proposed metrics, we make the first systematic quantitative evaluations of existing part-prototype networks in Sec. 4.2. Experiments demonstrate that current part-prototype networks are, in fact, not sufficiently interpretable.

To strengthen the interpretability of prototypes, we propose an elaborated part-prototype network built upon a revised ProtoPNet with two proposed modules: a shallow-deep feature alignment (SDFA) module and a score aggregation (SA) module. These two modules aim to accurately match the prototypes with their corresponding object parts across images, benefiting both consistency and stability scores. Part-prototype networks match prototypes with object parts in two steps: (1) feature extraction of object parts; (2) matching between prototypes and features of object parts. SDFA module improves the first step by promoting deep feature maps to spatially align with the input images. Specifically, SDFA module aligns the spatial similarity structure of shallow and deep feature maps with the observation that shallow feature maps retain spatial information that deep feature maps lack (Fig. 2 (a)). Meanwhile, SA module improves the second step based on the observation that the matching of each prototype with its corresponding object part is disturbed by other categories (Fig. 2 (b)). To mitigate this problem, SA module aggregates the activation values of prototypes only

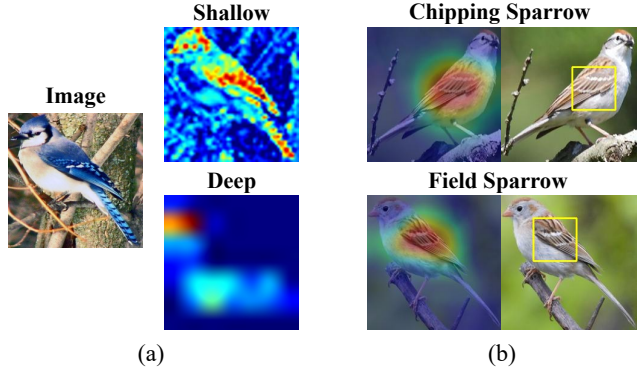


Figure 2. (a) Shallow feature maps of an image retain spatial information that deep feature maps lack (the features are from one channel of the feature map). (b) A prototype from Chipping Sparrow tends to match the wing, but meanwhile it has to paradoxically ignore almost the same wing of Field Sparrow. The samples are from ProtoPNet trained on ResNet34 backbone.

into their allocated categories to concentrate the matching between prototypes and their corresponding object parts.

We perform extensive experiments to validate the performance of our proposed model. Experiment results demonstrate that without using any object part annotations in training, our model achieves the state-of-the-art performance in both interpretability and accuracy on CUB-200-211 dataset [43], Stanford Cars dataset [23] and PartImageNet dataset [11], over six CNN backbones and three ViT backbones. Furthermore, experiment results show that the proposed consistency and stability scores are strongly positively correlated with accuracy in part-prototype networks, nicely reconciling the conflict between interpretability and accuracy in most prior interpretability methods.

To sum up, the key contributions of this work can be summarized as follows:

- We establish a benchmark to quantitatively evaluate the interpretability of prototypes of part-prototype networks with the proposed evaluation metrics (consistency score and stability score), uncovering pros and cons of various part-prototype networks.
- We propose an elaborated part-prototype network built upon ProtoPNet with a shallow-deep feature alignment (SDFA) module and a score aggregation (SA) module to enhance its interpretability.
- Experiment results verify that our proposed model significantly outperforms existing part-prototype networks by a large margin, in both accuracy and interpretability. Besides, the consistency and stability scores are positively correlated with accuracy, nicely reconciling the conflict between interpretability and accuracy.

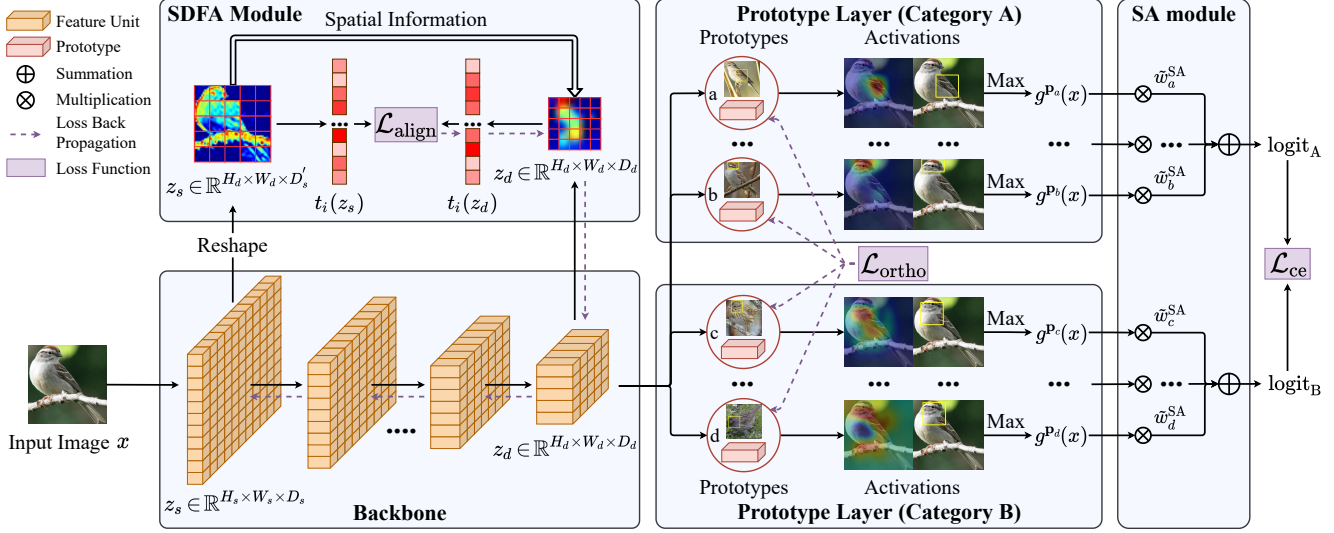


Figure 3. Overview of our proposed model (only two categories are presented for brevity). Backbone is a deep convolutional network to extract the features of input image x . SDFa module incorporates spatial information from shallow layers into deep layers by aligning the spatial similarity structure in deep layers with that in shallow layers using an alignment loss \mathcal{L}_{align} . The last feature map z_d is fed into prototype layer for different categories. In the prototype layer, each prototype p_i generates an activation map on z_d and selects the maximum value as the activation value $g^{p_i}(x)$. Finally, SA module aggregates the activation values of prototypes into their allocated categories for classification. Note that $D'_s = \frac{H_s}{H_d} \cdot \frac{W_s}{W_d} \cdot D_s$, \mathcal{L}_{clst} , \mathcal{L}_{sep} and the loss back propagation of \mathcal{L}_{ce} are omitted for brevity in the figure.

2. Related Work

2.1. Part-Prototype Networks

ProtoPNet [7] is the first work of part-prototype networks which define interpretable prototypes to represent specific object parts for image classification. ProtoPNet has explicit explanations of DNNs and comparable performance with its analogous non-interpretable counterpart, which inspires many variants of ProtoPNet. ProtoTree [31] aggregates prototype learning into a decision tree, which generates local explanations of prototypes through a specific route of the decision tree. TesNet [44] organizes the prototypes on the Grassmann manifold with several regularization loss functions as constraints. Deformable ProtoPNet [8] proposes deformable prototypes, which consist of multiple prototypical parts with changeable relative positions to capture pose variations. PW-Net [19] extends part-prototype networks into deep reinforcement learning and ProtoPDebug [5] proposes to utilize part-prototypes to correct the mistakes of network. However, these methods rely on the assumption that deep features of networks retain spatial information which is not guaranteed, and they lack a quantitative metric to evaluate their explanation results.

2.2. Evaluation of Interpretability Methods

With the development of many tasks in computer vision (e.g., image classification [12, 38–40], detection [45, 46, 52, 53], and generation [15, 16, 48], 3D object process-

ing [9, 17, 18], dataset condensation [25, 26, 51], and model reassembly [47, 49]), numerous XAI (i.e., explainable AI) methods and the corresponding evaluation methods are also proposed. Zhou *et al.* [55] propose that current interpretability evaluation methods can be categorized into human-centered methods [4, 6, 24] and functionality-grounded methods [2, 10, 14, 29, 30, 32, 37, 41, 50]. Human-centered methods require end-users to evaluate the explanations of XAI methods, which typically demand high labor costs and cannot guarantee reproducibility. Conversely, functionality-grounded methods utilize the formal definition of XAI methods as a policy to evaluate them. Model size, runtime operation counts assess the quality of explanations according to their explicitness, e.g., a shallow decision tree tends to have better interpretability. Many functionality-grounded methods are also proposed to evaluate the eminent attribution-based XAI methods. However, these functionality-grounded methods are not directed against part-prototype networks, and our work aims to propose quantitative metrics to evaluate the interpretability of part-prototype networks.

3. Method

3.1. Preliminaries

Existing part-prototype networks are built upon the framework of ProtoPNet [7], and this section specifies this framework. ProtoPNet mainly consists of a regular convolutional network f , a prototype layer g_p and a fully-connected layer

h (w^h denotes the parameters of h). The prototype layer $g_{\mathbf{p}}$ contains M learnable prototypes $\mathbf{P} = \{\mathbf{p}_j \in \mathbb{R}^{1 \times 1 \times D}\}_{j=1}^M$ for total K categories (D is the dimension of prototypes).

Given an input image x , ProtoPNet uses the convolutional network f to extract the feature map $z = f(x)$ of x ($z \in \mathbb{R}^{H \times W \times D}$). The prototype layer $g_{\mathbf{p}}$ generates an activation map $v^{\mathbf{p}_j}(x) \in \mathbb{R}^{H \times W}$ of each prototype \mathbf{p}_j on the feature map z by calculating and concatenating the similarity score between \mathbf{p}_j and all units $\tilde{z} \in \mathbb{R}^{1 \times 1 \times D}$ of z (z consists of $H \times W$ units). Then the activation value $g^{\mathbf{p}_j}(x)$ of \mathbf{p}_j on x is computed as the maximum value in $v^{\mathbf{p}_j}(x)$:

$$\begin{aligned} g^{\mathbf{p}_j}(x) &= \max v^{\mathbf{p}_j}(x) \\ &= \max_{\tilde{z} \in \text{units}(z)} \text{Sim}(\tilde{z}, \mathbf{p}_j). \end{aligned} \quad (1)$$

Here, $\text{Sim}(\cdot, \cdot)$ denotes the similarity score between two vectors (named activation function). ProtoPNet allocates N pre-determined prototypes to each category k (note that $M = N \cdot K$), and $\mathbf{P}_k \subseteq \mathbf{P}$ denotes the prototypes from category k . Finally, the total M activation values $\{g^{\mathbf{p}_j}(x)\}_{j=1}^M$ are concatenated and multiplied with the weight matrix $w^h \in \mathbb{R}^{K \times M}$ in the fully-connected layer h to generate the classification logits of x . Specifically, ProtoPNet sets $w_{k,j}^h$ to be positive for all j with $\mathbf{p}_j \in \mathbf{P}_k$, and $w_{k,j}^h$ to be negative for all j with $\mathbf{p}_j \notin \mathbf{P}_k$. This design ensures that the high activation value of a prototype increases the probability that the image belongs to its allocated category and decreases the probability that the image belongs to other categories.

After training, the value of $g^{\mathbf{p}_j}(x)$ reflects whether the object part represented by prototype \mathbf{p}_j exists in image x . Besides, ProtoPNet visualizes the corresponding region of prototype \mathbf{p}_j on x by resizing the activation map $v^{\mathbf{p}_j}(x)$ to be a visualization heatmap with the same shape as image x .

3.2. Interpretability Benchmark of Part-Prototype Networks

The interpretability benchmark of part-prototype networks includes two evaluation metrics: consistency score (or named *part-consistency score*) and stability score (or named *stability score under perturbations*). To calculate these two metrics, we first calculate the corresponding object part of each prototype on the images. Next, we determine the consistency of each prototype according to whether the corresponding object parts of it are consistent across different images, and determine the stability of each prototype according to whether the corresponding object parts of it are the same on the original and perturbed images.

3.2.1 Corresponding Object Part of Prototype

The corresponding object part of each prototype on the image is calculated using object part annotations in the dataset,

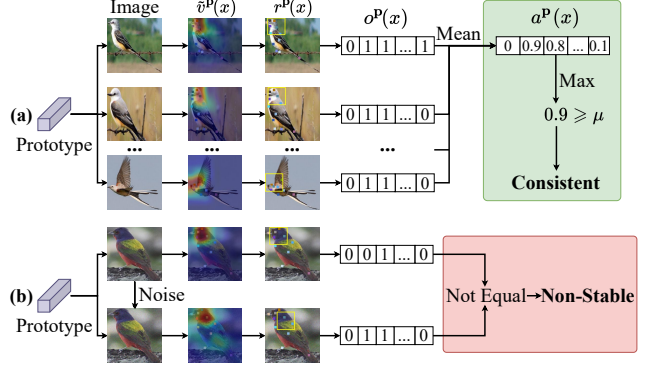


Figure 4. (a) Example for determination of **consistency** of a prototype. (b) Example for determination of **stability** of a prototype. The colorful points represent the object part annotations.

which guarantees objective and reproducible evaluation results. First, given a prototype \mathbf{p}_j and an input image x , we follow ProtoPNet to resize the activation map $v^{\mathbf{p}_j}(x) \in \mathbb{R}^{H \times W}$ to be $\tilde{v}^{\mathbf{p}_j}(x)$ with the same shape as x , then calculate the corresponding region $r^{\mathbf{p}_j}(x)$ of \mathbf{p}_j on x as a fix-sized bounding box (with a pre-determined shape $H_b \times W_b$) whose center is the maximum unit in $\tilde{v}^{\mathbf{p}_j}(x)$. Next, let C denote the number of categories of object parts in the dataset, and the corresponding object part $o^{\mathbf{p}_j}(x) \in \mathbb{R}^C$ of prototype \mathbf{p}_j on x is calculated from $r^{\mathbf{p}_j}(x)$ and the object part annotations in x . Specifically, $o_i^{\mathbf{p}_j}(x) = 1$ if the i -th object part is inside $r^{\mathbf{p}_j}(x)$ and $o_i^{\mathbf{p}_j}(x) = 0$ otherwise, as shown in Fig. 4. Note that “ i -th” is the index of this object part in the dataset.

3.2.2 Consistency Score

We generate the averaged corresponding object part $a^{\mathbf{p}_j} \in \mathbb{R}^C$ of each prototype \mathbf{p}_j over the test images from the allocated category of \mathbf{p}_j , and determine the consistency of \mathbf{p}_j according to whether the maximum element in $a^{\mathbf{p}_j}$ exceeds a threshold. Specifically, let \mathcal{I}_k denote the test images belonging to category k , and for each prototype \mathbf{p}_j , $c(j)$ denotes the allocated category of \mathbf{p}_j , and the averaged corresponding object part $a^{\mathbf{p}_j}$ of \mathbf{p}_j is calculated as below ($\|\cdot\|$ denotes cardinality of a set):

$$a^{\mathbf{p}_j} = \frac{\sum_{x \in \mathcal{I}_{c(j)}} o^{\mathbf{p}_j}(x)}{\|\mathcal{I}_{c(j)}\|}. \quad (2)$$

For $\forall i \in \{1, 2, \dots, C\}$, $a_i^{\mathbf{p}_j} \in [0, 1]$ since each $o_i^{\mathbf{p}_j}$ is either 0 or 1. If there exists an element in $a^{\mathbf{p}_j}$ not less than a pre-defined threshold μ , the prototype \mathbf{p}_j is determined to be consistent. Finally, the consistency score S_{con} of a part-prototype network is defined concisely as the ratio of

consistent prototypes over all M prototypes ($\mathbb{1}\{\cdot\}$ is the indicator function):

$$S_{\text{con}} = \frac{1}{M} \sum_{j=1}^M \mathbb{1}\{\max(a^{\mathbf{P}_j}) \geq \mu\}. \quad (3)$$

3.2.3 Stability Score

Stability score estimates whether prototypes retain the same corresponding object parts in the images perturbed by noise ξ . As consistency score, the stability of a prototype is estimated using all the test images from its allocated category. Finally, the stability score S_{sta} of a part-prototype network is calculated averagely over all prototypes:

$$S_{\text{sta}} = \frac{1}{M} \sum_{j=1}^M \frac{\sum_{x \in \mathcal{I}_{c(j)}} \mathbb{1}\{\mathbf{o}^{\mathbf{P}_j}(x) = \mathbf{o}^{\mathbf{P}_j}(x + \xi)\}}{\|\mathcal{I}_{c(j)}\|}. \quad (4)$$

We implement two types of noise ξ : (1) **Random noise**. In this way, the noise ξ is randomly sampled from the same Gaussian Distribution for all models: $\xi \sim \mathcal{N}(0, \sigma^2)$. (2) **Adversarial noise**. We utilize the famous PGD-attack method [28] to generate adversarial noise, which attempts to perturb the activation map of prototypes.

3.3. Towards a Stronger Part-Prototype Network

The interpretability benchmark for part-prototype networks can be established using the proposed consistency and stability scores. Next, we propose an elaborated part-prototype network built upon a revised ProtoPNet with two proposed modules: shallow-deep feature alignment (SDFA) module and score aggregation (SA) module. Part-prototype networks match prototypes with object parts in two steps: (1) feature extraction of object parts; (2) matching between prototypes and features of object parts. SDFA and SA modules respectively optimize these two steps to concentrate the matching between prototypes and their corresponding object parts, improving both consistency and stability scores.

3.3.1 Shallow-Deep Feature Alignment Module

The shallow-deep feature alignment (SDFA) module is proposed to improve the feature extraction of object parts. Part-prototype networks extract the features of object part as the feature unit \tilde{z} with the corresponding spatial position in z , requiring that deep feature maps preserve spatial information and spatially align with the input images. However, this requirement is not guaranteed and leads to inaccurate feature extraction of object parts. According to previous work [3, 27] and our pre-experiments, units of shallow feature maps have small effective receptive fields and thereby retain spatial information. Therefore, SDFA module preserves the spatial

information of deep feature maps by incorporating spatial information from shallow layers into deep layers.

To this end, SDFA module utilizes the spatial similarity structure to represent the spatial information of a feature map and constrain the feature map of deep layers to have the identical spatial similarity structure with that of shallow layers, which is inspired by Kornblith *et al.* [22] that similarity structures within representations can be used to compare two representations. Specifically, the spatial similarity structure $t(z) \in \mathbb{R}^{HW \times HW}$ of a feature map $z \in \mathbb{R}^{HW \times D}$ (z is resized from $H \times W \times D$ to $HW \times D$ for convenience) is defined as a matrix whose element represents the similarity between two units in z :

$$t_{i,j}(z) = \text{Sim}(z_i, z_j). \quad (5)$$

We adopt cosine similarity for $\text{Sim}(\cdot, \cdot)$ here, because cosine similarity is invariant to the norm of units which differ a lot in different layers. $z_s \in \mathbb{R}^{H_s \times W_s \times D_s}$, $z_d \in \mathbb{R}^{H_d \times W_d \times D_d}$ are used to denote the feature map of a shallow layer and a deep layer, respectively. To keep consistent with z_d , z_s is first resized to be $H_d \times W_d \times (\frac{H_s}{H_d} \cdot \frac{W_s}{W_d} \cdot D_s)$, which implies that each unit of z_d corresponds to an ‘‘image patch’’ in z_s . Besides, SDFA module adopts a ReLU function to restrain only the extremely dissimilar pairs, which is to stabilize model training. Finally, the shallow-deep feature alignment loss $\mathcal{L}_{\text{align}}$ is calculated as below ($\tilde{Z} = H_d W_d$):

$$\mathcal{L}_{\text{align}} = \frac{1}{\tilde{Z}^2} \sum_{i=0}^{\tilde{Z}-1} \sum_{j=0}^{\tilde{Z}-1} \max(|t_{i,j}(z_d) - t_{i,j}(z_s)| - \gamma, 0). \quad (6)$$

Here, γ denotes the threshold for ReLU function. Besides, $t(z_s)$ is set to be detached so that it never requires gradient.

3.3.2 Score Aggregation Module

The score aggregation (SA) module is proposed to address the problem that the matching of each prototype with its corresponding object part is disturbed by other categories. This problem stems from the fully-connected layer h that the classification score of a category is dependent on prototypes of other categories. The vanilla ProtoPNet has proposed a convex optimization step to mitigate this problem, which optimizes the weight w^h of last layer by minimizing this loss: $\mathcal{L}_{\text{ce}} + \sum_{k=1}^K \sum_{j: \mathbf{p}_j \notin \mathbf{P}_k} |w_{k,j}^h|$ (\mathcal{L}_{ce} is the classification loss). However, we find that this optimization step will make some $w_{k,j}^h$ with $\mathbf{p}_j \in \mathbf{P}_k$ be negative, which causes high activation values of these prototypes to paradoxically contribute negatively to their allocated categories and hurts the interpretability of the model.

Instead, our work directly addresses this problem by replacing the fully-connected layer with the SA module. SA module aggregates the activation values of prototypes only

into their allocated categories, followed by a learnable layer with weights $w^{\text{SA}} \in \mathbb{R}^M$ to adjust the importance of all M prototypes. Specifically, let $\tilde{w}_j^{\text{SA}} = e^{w_j^{\text{SA}}} / (\sum_{\mathbf{p}_i \in \mathbf{P}_k} e^{w_i^{\text{SA}}})$, the classification score logit_k of category k is calculated in SA module as below:

$$\text{logit}_k = \sum_{\mathbf{p}_j \in \mathbf{P}_k} \tilde{w}_j^{\text{SA}} \cdot g_{\mathbf{p}_j}(x). \quad (7)$$

3.3.3 Revised Baseline & Loss Function

The vanilla ProtoPNet adopts some components for model training: a cluster loss $\mathcal{L}_{\text{clst}}$, a separation loss \mathcal{L}_{sep} . The details of them are presented in Section A.4 of the appendix. Besides, we additionally adopt several simple but effective modifications on the vanilla ProtoPNet: (1) **Activation function**. We select inner product as the activation function following TesNet [44] (another part-prototype network), which enlarges the gap between high activations and low activations, and thus better discriminates regions with different activations. (2) **Orthogonality loss**. We use the orthogonality loss following Deformable ProtoPNet [8] to diversify prototypes within the same category, as shown in Eq. (8) ($\mathbf{P}^k \in \mathbb{R}^{N \times D}$ denotes the concatenation of prototypes from category k and \mathbb{I}_N is an $N \times N$ identity matrix). (3) **Hyper-parameters**. We select some different hyper-parameters, as shown in Section A.5 of the appendix.

$$\mathcal{L}_{\text{ortho}} = \sum_{k=1}^K \|\mathbf{P}^k (\mathbf{P}^k)^\top - \mathbb{I}_N\|^2. \quad (8)$$

This revised ProtoPNet significantly improves the performance of the vanilla ProtoPNet, and we add S DFA and SA modules into it for further improvement (Fig. 3). The total loss $\mathcal{L}_{\text{total}}$ of our final model is as below (\mathcal{L}_{ce} denotes the cross entropy loss, λ_{align} denotes the coefficient of $\mathcal{L}_{\text{align}}$):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \underbrace{\mathcal{L}_{\text{clst}} + \mathcal{L}_{\text{sep}} + \mathcal{L}_{\text{ortho}}}_{\text{Previous Methods}} + \underbrace{\lambda_{\text{align}} \mathcal{L}_{\text{align}}}_{\text{Ours}}. \quad (9)$$

4. Experiments

4.1. Experimental Settings

Datasets. We follow existing part-prototype networks to conduct experiments on CUB-200-2011 [43] and Stanford Cars [23]. CUB-200-2011 contains location annotations of object parts for each image, including 15 categories of object parts (back, breast, eye, leg, ...) that cover the bird’s whole body. Therefore, our interpretability benchmark is mainly established based on this dataset. Besides, we also validate the performance of our model on PartImageNet [11].

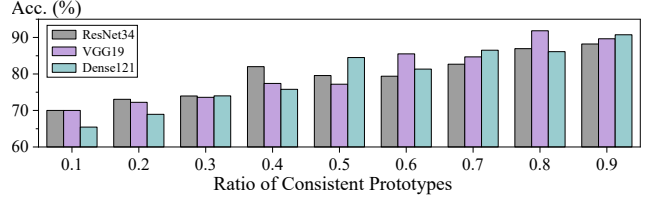


Figure 5. The test accuracy on images from each class correlates positively with its ratio of consistent prototypes in ProtoPNet (over three backbones).

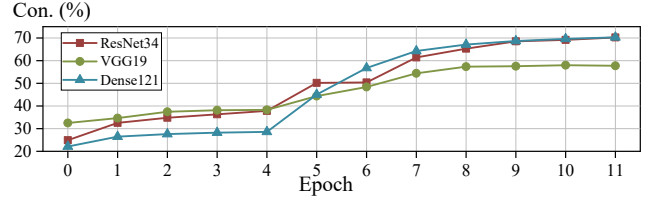


Figure 6. The consistency score increases along with the training process of ProtoPNet (over three backbones).

Benchmark Setup. Our benchmark evaluates five current part-prototype networks: ProtoPNet [7], ProtoTree [31], TesNet [44], Deformable ProtoPNet [8] and ProtoPool [33]. The consistency score and stability score of these methods are re-implemented faithfully following their released codes.

Parameters. We set H_b , W_b , μ and σ to be 72, 72, 0.8 and 0.2 for the interpretability evaluation of all part-prototype networks. Our models are trained for 12 epochs with Adam optimizer [21] (including 5 epochs for warm-up). We set the learning rates of the backbone, the add-on module and the prototypes to be $1e^{-4}$, $3e^{-3}$ and $3e^{-3}$ for our method. 0.5 and 0.1 are chosen for λ_{align} and γ . The number of prototypes per category and the dimension of prototypes are 10 and 64 for our method. More details of our experiment setup are presented in Section A of the appendix.

4.2. Benchmark of Part-Prototype Networks

With the proposed consistency and stability scores, the benchmark of part-prototype networks on CUB-200-2011 can be established (over five convolutional backbones pre-trained on ImageNet), as shown in Tab. 1. The stability score in this table is calculated with random noise, we provide the evaluation results of other variants of stability score (including adversarial noise) in Section B.3 of the appendix and find that they are consistent with the random-noise version. In the table, “Baseline” is the simplest non-interpretable model with a fully-connected layer on the last feature map for classification. Existing part-prototype networks are listed in ascending order of their accuracy in the table, and some important conclusions can be drawn from it.

The vanilla ProtoPNet has poor interpretability. The consistency score of the vanilla ProtoPNet ranges from 15.1 to

Method	ResNet34			ResNet152			VGG19			Dense121			Dense161		
	Con.	Sta.	Acc.	Con.	Sta.	Acc.	Con.	Sta.	Acc.	Con.	Sta.	Acc.	Con.	Sta.	Acc.
Baseline	N/A	N/A	82.3	N/A	N/A	81.5	N/A	N/A	75.1	N/A	N/A	80.5	N/A	N/A	82.2
ProtoTree [31]	10.0	21.6	70.1	16.4	23.2	71.2	17.6	19.8	68.7	21.5	24.4	73.2	18.8	28.9	72.4
ProtoPNet [7]	15.1	53.8	79.2	28.3	56.7	78.0	31.6	60.4	78.0	24.9	58.9	80.2	21.2	58.2	80.1
ProtoPool [33]	32.4	57.6	80.3	35.7	58.4	81.5	36.2	62.7	78.4	48.5	55.3	81.5	40.6	61.2	82.0
Deformable [8]	39.9	57.0	81.1	44.2	53.5	82.0	40.6	61.5	77.9	61.4	64.7	82.6	46.7	63.9	83.3
TesNet [44]	53.3	65.4	82.8	48.6	60.0	82.7	46.8	58.2	81.4	63.1	66.1	84.8	62.2	67.5	84.6
Ours	52.9	66.3	82.3	50.7	65.7	83.8	44.6	56.9	80.6	52.9	59.1	83.4	56.3	61.5	84.7
Ours + SA	67.5	69.9	83.6	59.2	68.4	84.3	50.4	60.5	82.1	65.3	61.5	84.8	70.0	66.6	85.8
Ours + SA + SDFa	70.6	72.1	84.0	62.1	70.8	85.1	56.5	63.5	82.5	68.1	67.6	85.4	72.0	71.8	86.5

Table 1. The comprehensive evaluation of interpretability and accuracy of part-prototype networks on CUB-200-2011 dataset. The results are over five convolutional backbones pre-trained on ImageNet. Con., Sta. and Acc. denote consistency score, stability score and accuracy, respectively. Our results are averaged over 4 runs with different seeds. Bold font denotes the best result.

Method	Backbone	Con.	Sta.	Acc.
ProtoTree	ResNet50 (iN)	16.4	18.4	82.2
ProtoPool	ResNet50 (iN)	34.6	45.8	85.5
Deformable	ResNet50 (iN)	39.7	48.5	85.6
Ours	ResNet50 (iN)	56.9	67.8	87.1
ProtoPNet	VGG + ResNet + Dense	23.9	57.7	84.8
TesNet	VGG + ResNet + Dense	54.4	63.2	86.2
ProtoTree	ResNet50 (iN) × 3	17.2	19.8	86.6
ProtoPool	ResNet50 (iN) × 3	34.3	46.2	87.5
Ours	ResNet50 (iN) × 3	56.7	67.4	88.3
ProtoTree	ResNet50 (iN) × 5	16.8	18.5	87.2
ProtoPool	ResNet50 (iN) × 5	34.4	45.7	87.6
Ours	ResNet50 (iN) × 5	57.0	67.5	88.5

Table 2. Results of the benchmark on CUB-200-2011 dataset over ResNet50 backbone pre-trained on iNaturalist2017 dataset and combination of multiple backbones. “× 3” denotes combining the classification logits of 3 models trained with different seeds. Bold font denotes the best result.

31.6 on five backbones, meaning that most of its prototypes are not interpretable because they cannot represent the same object parts in different images. This points out that qualitative analysis (cherry picks) of explanation results is not reliable, and quantitative analysis is more meaningful and essential. However, many current methods directly transfer the paradigm of the vanilla ProtoPNet to other domains (*e.g.*, image segmentation, person re-identification, deep reinforcement learning) with only qualitative analysis.

The accuracy of part-prototype networks correlates positively with their consistency and stability scores overall. For each backbone in Tab. 1, the part-prototype network with higher accuracy generally has higher consistency and stability scores. This phenomenon accords with the definition of part-prototype networks that a prototype represents a specific object part, and part-prototype networks make pre-

dictions by comparing the object parts which are activated by the same prototypes in the test image and training images. In this definition, the mismatch of object parts in the test image and training images will severely drop the accuracy of the model. For example, a non-consistent and non-stable part-prototype network will make wrong predictions by mistakenly comparing the head part in the test image with the stomach part in the training images.

Besides, we conduct two experiments to analyze the relation between interpretability and accuracy of ProtoPNet. First, we calculate the accuracy of each category and get the average accuracy of categories with the same ratio of consistent prototypes in Fig. 5, showing that the accuracy on different categories positively correlates with the ratio of consistent prototypes. Second, we calculate the consistency score after each training epoch in Fig. 6, showing that the consistency score increases along with the model training and thus has a positive correlation with the model accuracy.

4.3. Comparisons with State-of-the-Art Methods

As shown in Tab. 1, we integrated the SDFa and SA modules into our revised ProtoPNet (“Ours” denotes this revised ProtoPNet), and the consistency score, stability score and accuracy of final model are significantly superior to current part-prototype networks on CUB-200-2011 (the main dataset adopted by previous methods) over five backbones. Tab. 2 demonstrates the experiment results on ResNet50 backbone pre-trained on iNaturalist2017 dataset [42] and combination of multiple backbones, which also verifies the significant performance of our model. Besides, our model achieves the best performance on Stanford Cars and PartImageNet, shown in Section B.1 and B.2 of the appendix.

4.4. Ablation Study

Tab. 1 demonstrates that the SDFa and SA modules both effectively improve consistency score, stability score and accuracy of the model. We provide ablation experiments of

Method	ResNet34	ResNet152	VGG19	Dense121
w/o S DFA	12.4	22.8	9.8	15.9
w/ S DFA	70.8(+58.4)	74.3(+51.5)	30.5(+20.7)	47.1(+31.2)

Table 3. Similarity (%) between spatial similarity structures of shallow layers and deep layers without/with S DFA module on the whole test set of CUB-200-2011.






Test Image	Shallow Layer	Deep Layer	Similarity Score
			12.18 (w/o S DFA)
			73.94 (w/ S DFA)

Figure 7. Visualization of feature maps from shallow layers and deep layers without/with S DFA module.

Method	ResNet34	ResNet152	VGG19	Dense121
w/o SA	3.9	4.7	7.3	6.2
w/ SA	8.5(+4.6)	9.8(+5.1)	10.8(+3.5)	11.8(+5.6)

Table 4. Average number of similar prototypes from other categories of each prototype on CUB-200-2011 dataset.

the hyper-parameters used in our method in Section B.5 of the appendix. Besides, we conduct two ablation experiments to analyze the effect of S DFA and SA modules.

S DFA Module. We calculate the similarity between spatial similarity structures ($t(z_s)$ and $t(z_d)$ with shape $\mathbb{R}^{H_d W_d \times H_d W_d}$) of shallow and deep layers with/without S DFA module, and generate the average results over all test images. Specifically, the similarity $\text{Sim}(t(z_s), t(z_d))$ between $t(z_s)$ and $t(z_d)$ is calculated in Eq. (10) ($\bar{Z} = H_d W_d$). Tab. 3 shows that S DFA module improves the similarity between spatial similarity structure of shallow layers and deep layers by a large margin. Besides, Fig. 7 shows that the shallow feature maps both explicitly contain spatial information, and the deep feature map can highlight the object instead of background with S DFA module.

$$\text{Sim}(t(z_s), t(z_d)) = \frac{1}{\bar{Z}} \sum_{i=0}^{\bar{Z}-1} e^{-\|t_i(z_s) - t_i(z_d)\|^2}. \quad (10)$$

SA Module. We calculate the number of similar prototypes from other categories for each prototype and present the averaged results without/with SA module (two prototypes with cosine similarity over 0.6 are considered to be similar here). As shown in Tab. 4, each prototype has fewer similar prototypes from other categories without SA module, indicating that prototypes are suppressed to represent similar

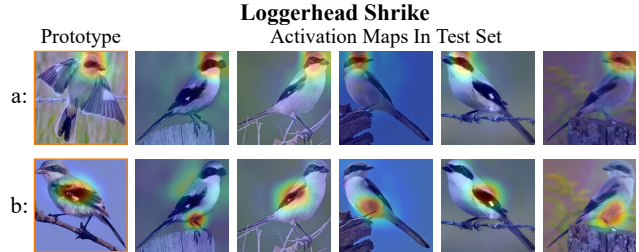


Figure 8. A consistent prototype (a) and a non-consistent prototype (b). Note that $\max(a^{P_a}) = 0.97$ and $\max(a^{P_b}) = 0.33$. The first image in each row is the training image representing this prototype.

object parts among categories without SA module, due to the original paradoxical learning paradigm.

4.5. Visualization Results

Consistency Score. To analyze the effect of our proposed consistency score, we provide visualization of activation maps of a consistent prototype p_a and a non-consistent prototype p_b from Yellow Billed Cuckoo category ($\max(a^{P_a}) = 0.97$ and $\max(a^{P_b}) = 0.13$). As shown in Fig. 8, prototype p_a consistently activates the head part in test images and training images, while prototype p_b desultorily activates the wing part, belly part and feet part.

Additionally, we demonstrate more comprehensive visualization analysis on the corresponding regions of consistent prototypes from our model in Section C of the appendix.

5. Conclusion

This work establishes an interpretability benchmark to quantitatively evaluate the interpretability of prototypes for part-prototype networks, based on two evaluation metrics (consistency score and stability score). Furthermore, we propose a S DFA module to incorporate the spatial information from shallow layers into deep layers and a SA module to concentrate the learning of prototypes. We add these two modules into a simply revised ProtoPNet, and it significantly surpasses the performance of existing part-prototype networks on three datasets, in both accuracy and interpretability. Our work has great potential to facilitate more quantitative metrics to evaluate the explanation results of interpretability methods, instead of using limited visualization samples which can be easily misled by cherry picks. In the future, we will extend this work to other concept embedding methods towards a unified benchmark for visual concepts.

Acknowledgements. This work is funded by National Natural Science Foundation of China (61976186, U20B2066, 62106220), Ningbo Natural Science Foundation (2021J189), and the Fundamental Research Funds for the Central Universities (2021FZZX001-23, 226-2023-00048).

References

- [1] David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *NeurIPS*, pages 7786–7795, 2018. 2
- [2] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *ICLR*, 2018. 3
- [3] André Araujo, Wade Norris, and Jack Sim. Computing receptive fields of convolutional neural networks. *Distill*, 2019. <https://distill.pub/2019/computing-receptive-fields>. 5
- [4] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S. Lasecki, Daniel S. Weld, and Eric Horvitz. Beyond accuracy: The role of mental models in human-ai team performance. In *AAAI*, pages 2–11, 2019. 3
- [5] Andrea Bontempelli, Stefano Teso, Fausto Giunchiglia, and Andrea Passerini. Concept-level debugging of part-prototype networks. In *ICLR*, 2023. 3
- [6] Jonathan D. Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *NeurIPS*, pages 288–296, 2009. 3
- [7] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan Su. This looks like that: Deep learning for interpretable image recognition. In *NeurIPS*, pages 8928–8939, 2019. 1, 3, 6, 7
- [8] Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable protopnet: An interpretable image classifier using deformable prototypes. In *CVPR*, pages 10255–10265, 2022. 3, 6, 7
- [9] Mingtao Feng, Haoran Hou, Liang Zhang, Yulan Guo, Hongshan Yu, Yaonan Wang, and Ajmal Mian. Exploring hierarchical spatial layout cues for 3d point cloud based scene graph prediction. *IEEE Transactions on Multimedia*, 2023. 3
- [10] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, pages 93:1–93:42, 2019. 3
- [11] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jieneng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan L. Yuille. Partimagenet: A large, high-quality dataset of parts. In *ECCV*, pages 128–145. Springer, 2022. 2, 6
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 3
- [13] Adrian Hoffmann, Claudio Fanconi, Rahul Rade, and Jonas Kohler. This looks like that... does it? shortcomings of latent space prototype interpretability in deep networks. *arXiv preprint arXiv:2105.02968*, 2021. 1, 2
- [14] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *NeurIPS*, pages 9734–9745, 2019. 3
- [15] Yongcheng Jing, Xiao Liu, Yukang Ding, Xinchao Wang, Errui Ding, Mingli Song, and Shilei Wen. Dynamic instance normalization for arbitrary style transfer. In *AAAI*, volume 34, pages 4369–4376, 2020. 3
- [16] Yongcheng Jing, Yining Mao, Yiding Yang, Yibing Zhan, Mingli Song, Xinchao Wang, and Dacheng Tao. Learning graph neural networks for image style transfer. In *ECCV*, pages 111–128. Springer, 2022. 3
- [17] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Amalgamating knowledge from heterogeneous graph neural networks. In *CVPR*, pages 15709–15718, 2021. 3
- [18] Yongcheng Jing, Chongbin Yuan, Li Ju, Yiding Yang, Xinchao Wang, and Dacheng Tao. Deep graph reprogramming. In *CVPR*, pages 24345–24354, 2023. 3
- [19] Eoin M Kenny, Mycal Tucker, and Julie Shah. Towards interpretable deep reinforcement learning with human-friendly prototypes. In *ICLR*, 2023. 1, 3
- [20] Sunnie SY Kim, Nicole Meister, Vikram V Ramaswamy, Ruth Fong, and Olga Russakovsky. Hive: evaluating the human interpretability of visual explanations. In *ECCV*, pages 280–298, 2022. 1, 2
- [21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6
- [22] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. In *ICML*, pages 3519–3529, 2019. 5
- [23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV*, pages 554–561, 2013. 2, 6
- [24] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Samuel J. Gershman, and Finale Doshi-Velez. Human evaluation of models built for interpretability. In *HCOMP*, pages 59–67, 2019. 3
- [25] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. *NeurIPS*, 35:1100–1113, 2022. 3
- [26] Songhua Liu, Jingwen Ye, Rungpeng Yu, and Xinchao Wang. Slimmable dataset condensation. In *CVPR*, pages 3759–3768, 2023. 3
- [27] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *NeurIPS*, pages 4898–4906, 2016. 5
- [28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*. OpenReview.net, 2018. 5
- [29] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning - A brief history, state-of-the-art and challenges. In *ECML*, pages 417–431, 2020. 3
- [30] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.*, pages 1–15, 2018. 3
- [31] Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *CVPR*, pages 14933–14943, 2021. 1, 3, 6, 7
- [32] An-phi Nguyen and María Rodríguez Martínez. On quantitative aspects of model interpretability. *arXiv preprint arXiv:2007.07584*, 2020. 3

- [33] Dawid Rymarczyk, Lukasz Struski, Michal Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zielinski. Interpretable image classification with differentiable prototypes assignment. In *ECCV*, pages 351–368, 2022. 1, 6, 7
- [34] Dawid Rymarczyk, Lukasz Struski, Jacek Tabor, and Bartosz Zielinski. Protoshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *SIGKDD*, pages 1420–1430, 2021. 1
- [35] Mikolaj Sacha, Dawid Rymarczyk, Lukasz Struski, Jacek Tabor, and Bartosz Zielinski. Protoseg: Interpretable semantic segmentation with prototypical parts. In *WACV*, pages 1481–1492. IEEE, 2023. 1
- [36] Zohaib Salahuddin, Henry C. Woodruff, Avishek Chatterjee, and Philippe Lambin. Transparency of deep neural networks for medical image analysis: A review of interpretability methods. *Comput. Biol. Medicine*, page 105111, 2022. 1
- [37] Dylan Slack, Sorelle A Friedler, Carlos Scheidegger, and Chitradeepta Roy. Assessing the local interpretability of machine learning models. *arXiv preprint arXiv:1902.03501*, 2019. 3
- [38] Xiu Su, Tao Huang, Yanxi Li, Shan You, Fei Wang, Chen Qian, Changshui Zhang, and Chang Xu. Prioritized architecture sampling with monte-carlo tree search. In *CVPR*, pages 10968–10977, 2021. 3
- [39] Xiu Su, Shan You, Fei Wang, Chen Qian, Changshui Zhang, and Chang Xu. Bcnet: Searching for network width with bilaterally coupled network. In *CVPR*, pages 2175–2184, 2021. 3
- [40] Xiu Su, Shan You, Jiyang Xie, Mingkai Zheng, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Vitas: Vision transformer architecture search. In *ECCV*, pages 139–157. Springer, 2022. 3
- [41] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, pages 3319–3328, 2017. 3
- [42] Grant Van Horn, Oisín Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, pages 8769–8778, 2018. 7
- [43] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2, 6
- [44] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *ICCV*, pages 875–884, 2021. 1, 3, 6, 7
- [45] Haoning Xi. Data-driven optimization technologies for maas. In *Big Data and Mobility as a Service*, pages 143–176. Elsevier, 2022. 3
- [46] Haoning Xi, Yi Zhang, and Yi Zhang. Detection of safety features of drivers based on image processing. In *18th COTA International Conference of Transportation Professionals*, pages 2098–2109. American Society of Civil Engineers Reston, VA, 2018. 3
- [47] Xingyi Yang, Jingwen Ye, and Xinchao Wang. Factorizing knowledge in neural networks. In *ECCV*, pages 73–91, 2022. 3
- [48] Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *CVPR*, pages 22552–22562, 2023. 3
- [49] Xingyi Yang, Daquan Zhou, Songhua Liu, Jingwen Ye, and Xinchao Wang. Deep model reassembly. *NeurIPS*, 35:25739–25753, 2022. 3
- [50] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. On the (in)fidelity and sensitivity of explanations. In *NeurIPS*, pages 10965–10976, 2019. 2, 3
- [51] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *arXiv preprint arXiv:2301.07014*, 2023. 3
- [52] Wei Zhai, Yang Cao, Jing Zhang, and Zheng-Jun Zha. Exploring figure-ground assignment mechanism in perceptual organization. *NeurIPS*, 35:17030–17042, 2022. 3
- [53] Wei Zhai, Hongchen Luo, Jing Zhang, Yang Cao, and Dacheng Tao. One-shot object affordance detection in the wild. *IJCV*, 130(10):2472–2500, 2022. 3
- [54] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. Protgnn: Towards self-explaining graph neural networks. In *AAAI*, pages 9127–9135, 2022. 1
- [55] Jianlong Zhou, Amir H Gandomi, Fang Chen, and Andreas Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, page 593, 2021. 2, 3