Is ProtoPNet Really Explainable? Evaluating and Improving the Interpretability of Prototypes

Qihan Huang, Mengqi Xue, Haofei Zhang, Jie Song, Mingli Song†
Zhejiang University
{qh.huang,mqxue,haofeizhang,sjie,brooksong}@zju.edu.cn

Abstract

ProtoPNet and its follow-up variants (ProtoPNets) have attracted broad research interest for their intrinsic interpretability from prototypes and comparable accuracy to non-interpretable counterparts. However, it has been recently found that the interpretability of prototypes can be corrupted due to the semantic gap between similarity in latent space and that in input space. In this work, we make the first attempt to quantitatively evaluate the interpretability of prototype-based explanations, rather than solely qualitative evaluations by some visualization examples, which can be easily misled by cherry picks. To this end, we propose two evaluation metrics, termed consistency score and stability score, to evaluate the explanation consistency cross images and the explanation robustness against perturbations, both of which are essential for explanations taken into practice. Furthermore, we propose a shallow-deep feature alignment (SDFA) module and a score aggregation (SA) module to improve the interpretability of prototypes. We conduct systematical evaluation experiments and substantial discussions to uncover the interpretability of existing ProtoPNets. Experiments demonstrate that our method achieves significantly superior performance to the state-of-the-arts, under both the conventional qualitative evaluations and the proposed quantitative evaluations, in both accuracy and interpretability. Codes are available at https://github.com/hqhQAQ/EvalProtoPNet.

1. Introduction

Explainable artificial intelligence (XAI) [8, 11] is artificial intelligence (AI) in which humans can understand the decisions or predictions made by the AI. It has attracted wide attention, and fruitful approaches have been proposed in recent years due to its crucial role in security-critical scenarios. Among these works, concept embedding [26, 30] interprets a black-box model, e.g., deep neural networks (DNNs) [12, 14, 31, 34], by finding a collection of human understandable concepts from internal representations. Prototypical part networks (ProtoPNets) [6, 7, 15, 24, 27, 28, 38, 39] are one representative type of concept embedding to transform a DNN into a self-explanatory model for image classification. Specifically, ProtoPNets define and train a number of learnable prototypes to represent various object parts and make model decisions through a linear combination of prototypes’ similarity scores with deep features, which endows the DNN with the self-interpretable characteristic.

Although ProtoPNets have achieved comparable or even superior classification performance to their non-interpretable baseline models (i.e., using the identical DNN backbones), some researchers [16, 29, 35] argue that the learned prototypes do not have credible interpretability owing to the semantic gap between similarity in the latent space and that in the input space. The reasons for the unreliable interpretability are twofold: (1) The corresponding object parts of the same prototype may not be consistent across images, as shown in Fig. 1 (a); (2) Prototypes are easily attacked, which means that the same prototype can be mapped to a vastly different
object part if the input image is modified with imperceptible perturbations, as shown in Fig. 1 (b). The fragile explanations of prototypes make existing qualitative evaluations (e.g., example visualization) easily misled by cherry picks. Recently, Kim et al. [16] proposed HIVE, a human-centered method to evaluate the interpretability of ProtoPNets. However, HIVE requires redundant human interactions and fails to guarantee the reproducibility of evaluation results due to subjective judgments from human users. In view of foregoing reasons, more formal and rigorous evaluation metrics are urgently needed to quantitatively and objectively benchmark existing prototype-based explanations.

In this work, we make to our best knowledge the first attempt to quantitatively evaluate the interpretability of prototype-based explanations, rather than solely qualitative evaluations by some visualization examples. Our evaluation metrics consider two key ingredients in interpretability: consistency and stability. Following the basic design principle of concept embedding [30] that each prototype is associated with a semantic concept which represents one specific object part in the image, we propose consistency score to evaluate whether and to what extent a learned prototype is mapped to the same object parts across different images. On the other hand, following viewpoint [1, 40, 41] that the explanations provided by XAI methods should satisfy stability property, we further propose stability score to measure the robustness of the learned prototypes being mapped to the same object parts if the input images are slightly perturbed. With the proposed metrics, we make the first systematic quantitative evaluations of existing prototype-based explanations, uncovering pros and cons of various ProtoPNets variants.

We further propose two modules to enhance the interpretability of prototypes: the shallow-deep feature alignment (SDFA) module and the score aggregation (SA) module. SDFA module stems from the implicit assumption of ProtoPNets that the features in the deep layers should retain spatial information as they directly interact with the prototypes. However, as shown in Fig. 2 (a), this assumption is not guaranteed [3, 21], incurring the inconsistency and instability issues of prototype explanations. Inspired by Kornblith et al. [18] that similarity structures can be used to compare two representations, SDFA module adopts an alignment loss to constrain the spatial similarity structures to be consistent between shallow features and deep features. SA module is motivated by the observation that ProtoPNets can make predictions based on absence of prototypes from other categories (named negative reasoning [27]), which causes each prototype to learn information of other categories and thereby disturbs it from accurately matching object parts of its category. As shown in Fig. 2 (b), a prototype tends to concentrate on the wing of Chipping Sparrow, but meanwhile it has to paradoxically ignore almost the same wing of Field Sparrow. To alleviate this problem, SA module aggregates the similarity scores of prototypes only into their allocated categories to concentrate the learning of prototypes.

We make extensive experiments to validate the two proposed modules and evaluation metrics. Without any object part annotations during the training phase, the two proposed modules significantly improve the performance of ProtoPNets, achieving the state-of-the-art performance in both interpretability and accuracy on CUB-200-211 [37] and Stanford Cars [19]. Furthermore, experimental results demonstrate that the proposed consistency and stability scores are strongly positively correlated with accuracy, nicely reconciling the conflict between interpretability and accuracy in most prior explaining methods.

In a nutshell, the main contributions of this work are listed as follows:

- We propose two evaluation metrics, consistency score and stability score, to quantitatively evaluate the interpretability of ProtoPNets variants. Furthermore, we establish a benchmark to quantitatively evaluate the interpretability of prototypes of ProtoPNets with the proposed evaluation metrics, uncovering pros and cons of various ProtoPNets variants.
- Based on our observations of shallow-deep feature misalignment and negative reasoning in prior ProtoPNets, we propose the shallow-deep feature alignment (SDFA) module and the score aggregation (SA) module to enhance their interpretability and accuracy.
- Experimental results verify that the two proposed modules significantly improve the performance, surpassing existing ProtoPNets by a large margin. Furthermore, the proposed consistency and stability scores are strongly positively correlated with accuracy, nicely reconciling the conflict between interpretability and accuracy.
2. Related Work

2.1. Prototypical Part Networks

ProtoPNet [6] is a self-explainable model that defines interpretable prototypes to represent specific object parts for image classification. The vanilla ProtoPNet has explicit explanations of DNNs and comparable performance with its analogous non-interpretable counterpart, which inspires many variants of ProtoPNet. ProtoTree [24] aggregates prototype learning into a decision tree, which generates local explanations of prototypes through a specific route of the decision tree. TesNet [38] organizes the prototypes on the Grassmann manifold with several regularization loss functions as constraints. ProtoPShare [28] shares the prototypes between categories of a trained ProtoPNet to reduce the number of prototypes while maintaining its performance. Deformable ProtoPNet [7] proposes deformable prototypes, which consist of multiple prototypical parts with changeable relative positions to capture pose variations. 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 emergence of numerous XAI methods, many quantitative evaluation methods of XAI methods are also proposed. Zhou et al. [41] propose that current interpretability evaluation methods can be categorized into human-centered methods [4,5,20] and functionality-grounded methods [2,10,13,22,23,25,32,33,40]. 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 ProtoPNet, and our work aims to propose quantitative metrics to evaluate the interpretability of ProtoPNet.

3. Method

3.1. Preliminaries

The vanilla prototypical part network (ProtoPNet) consists of a regular convolutional network \(f\), a small add-on module, a prototype layer \(g_p\), and a fully-connected layer \(h\) (\(w_h\) denotes the parameters of \(h\)). Detailedly, the add-on module is to adjust the dimension of the feature map, and the prototype layer \(g_p\) contains \(M\) learnable prototypes \(P = \{p_j \in \mathbb{R}^{1 \times 1 \times D} \}_{j=1}^M\) for total \(K\) categories.

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

$$g_p(x) = \max v_p(x) = \max_{\tilde{z} \in \text{units}(z)} \text{Sim}(\tilde{z}, p_j).$$ (1)

Here, $\text{Sim}(\cdot, \cdot)$ denotes the similarity score between two vectors with the same shape (e.g., $\text{Sim}(\cdot, \cdot)$ can be the sum of Hadamard product of two vectors). ProtoPNets allocate $N$ pre-determined prototypes to each category $k$ ($M = N \cdot K$), and $P_j \subseteq \mathbb{P}$ denotes the prototypes from category $k$. Finally, the $M$ activation values $\{g_p(x)\}_{j=1}^M$ are 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 $x$. Particularly, ProtoPNets sets $w_h^{k,j}$ to 1 for all $j$ with $p_j \in P_k$, and $w_h^{k,j}$ to $-0.5$ for all $j$ with $p_j \notin P_k$. The such 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, ProtoPNets visualizes the corresponding region of prototype $p_j$ by mapping the high activated units of its activation map $v_p(x)$ to the original input image $x$.

### 3.2. Interpretability Benchmark of Prototypical Part Networks

The interpretability benchmark of ProtoPNets is based on two evaluation metrics: consistency score and stability score. Unlike human-centered evaluation methods, these two metrics guarantee objective and reproducible evaluation results using object part annotations in the dataset. To ensure that the two evaluation metrics are fair for all ProtoPNets, this work primarily unifies the calculation of corresponding region $r_p(x)$ of prototype $p_j$ on image $x$. For all ProtoPNets, we resize the activation map $v_p(x) \in \mathbb{R}^{H \times W}$ to be $	ilde{v}_p(x)$ with the same shape as $x$, then calculate the corresponding region $r_p(x)$ of $p_j$ on $x$ as a fix-sized bounding box with a pre-defined shape $H_b \times W_b$ whose center is the maximum unit in $\tilde{v}_p(x)$. Next, we use $C$ and $o_p(x) \in \mathbb{R}^C$ to denote the number of categories of object parts in the dataset and the corresponding object parts of prototype $p_j$, respectively. Specifically, $o_p(x)$ is a binary vector calculated from $r_p(x)$ and the object parts annotations in $x$, with $o_p^i(x) = 1$ if the $i$-th object part is inside $r_p(x)$ and $o_p^i(x) = 0$ if the $i$-th object part is not inside $r_p(x)$. Note that “$i$-th” is the index of this object part in the dataset.

#### 3.2.1 Consistency Score

In the definition of the vanilla ProtoPNet, each prototype is associated with an object part of its allocated category. Therefore, this work determines the consistency of a prototype according to the averaged corresponding object parts on all the test images of its allocated category. In other word, if there exists an absolutely high value in the averaged corresponding object parts, the prototype is determined to be associated with this object part. Note that some object parts are invisible in the image, we calculate each element of the averaged corresponding object parts only in the images that contain this object part. Specifically, we use $u(x) \in \mathbb{R}^C$ to denote the visible object parts in image $x$ according to annotations. Like $o_p(x), u(x)$ is a binary vector with $u^i(x) = 1$ if the $i$-th object part is visible in $x$ and $u^i(x) = 0$ otherwise. Let $\mathcal{I}_k$ denote the test images belonging to category $k$, and for each prototype $p_j$, $c(j)$ denotes the allocated category of $p_j$, the averaged corresponding object parts $a_p \in \mathbb{R}^C$ of $p_j$ is calculated as below ($\odot$ denotes element-wise division):

$$a_p(x) = \left( \sum_{x \in \mathcal{I}_{c(j)}} o_p(x) \right) \odot \left( \sum_{x \in \mathcal{I}_{c(j)}} u(x) \right).$$ (2)

For $\forall i \in \{1, 2, ..., C\}$, $a_p^i \in [0, 1]$ because $\alpha_p^i(x) = 1$ only if $u^i(x) = 1$. The value of $\alpha_p^i$ reflects the correspondence degree between prototype $p_j$ and the $i$-th object part. Therefore, if there exists an element in $\alpha_p$ not less than a pre-defined threshold $\mu$, the prototype $p_j$ is determined to be consistent. Finally, the consistency score $S_{\text{con}}$ of a ProtoPNet is defined concisely as the ratio of consistent prototypes over all $M$ prototypes ($\{\cdot\}$ is the indicator function):

$$S_{\text{con}} = \frac{1}{M} \sum_{j=1}^M \{\text{max}(a_p) \geq \mu\}.$$ (3)

#### 3.2.2 Stability Score

Stability score estimates whether prototypes retain the same corresponding object parts in the slightly-perturbed images. This metric needs not consider the invisible object parts since each image contains the same object parts as its perturbed counterpart. To ensure the fairness of this metric, the noise $\xi$ for perturbation is randomly sampled from the same Gaussian Distribution for all models: $\xi \sim \mathcal{N}(0, \sigma^2)$. As consistency score, the stability of a prototype is estimated using all the test images of its allocated category. Finally, the stability score $S_{\text{sta}}$ of a ProtoPNet is calculated averagely over all prototypes ($\| \cdot \|$ denotes cardinality of a set):

$$S_{\text{sta}} = \frac{1}{M} \sum_{j=1}^M \sum_{x \in \mathcal{I}_{c(j)}} \mathbb{I}\{\alpha_p^i(x) = \alpha_p^i(x + \xi)\}.$$ (4)
### 3.3. Shallow-Deep Feature Alignment Module

In ProtoPNets, the corresponding region of prototype \( p_j \) on image \( x \) is calculated from the last feature map of the DNN backbone. Current ProtoPNets never constrain that the last feature map of DNNs retains spatial information, i.e., each unit of the feature map represents the image patch with the same spatial position in the input image. Without this constraint, the corresponding region of prototype \( p_j \) on the input image \( x \) is not guaranteed to be identical to that in the activation map \( v_{p_j}(x) \). According to previous work [3, 21] and our pre-experiments, units of shallow feature maps have small effective receptive fields and thereby retaining spatial information. Therefore, shallow-deep feature alignment (SDFA) module is proposed to retain the spatial information of deep layers of DNNs by incorporating spatial information from shallow layers into deep layers.

Inspired by Kornblith et al. [18] that similarity structures within representations can be used to compare two representations, we utilize the spatial similarity structure to represent the spatial information of a feature map and constrains the feature map of deep layers to have the identical spatial similarity structure with that of shallow layers. 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 \( z \):

\[
t_{i,j}(z) = \text{Sim}(z_i, z_j).
\]  

Here, \( \text{Sim}(.,. \) is the cosine similarity. \( z_d \in \mathbb{R}^{H_d \times W_d \times D_d} \) and \( z_s \in \mathbb{R}^{H_s \times W_s \times D_s} \) 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 \( H_d \times W_d \times (\frac{H_s}{H_d}, \frac{W_s}{W_d} \cdot D_s) \), which implies that each unit of \( z_d \) corresponds to an “image patch” in \( z_s \). To stabilize model training, this work constrains only the similarities between the highest activated unit and other units to be identical between shallow and deep layers, together with a RELU function to restrain only the extremely dissimilar pairs. Therefore, the shallow-deep feature alignment loss \( L_{\text{align}} \) is calculated as below (\( Z = H_dW_d \)):

\[
L_{\text{align}} = \frac{1}{Z} \sum_{i=0}^{Z-1} \max(|t_{e,i}(z_d) - t_{e,i}(z_s)| - \gamma, 0).
\]  

Here, \( e \) denotes the index of the highest activated unit in \( z_d \), and \( \gamma \) denotes the thresh for RELU function. Moreover, \( t(z_s) \) is set to be detached so that it never requires gradient.

### 3.4. Score Aggregation Module

The negative reasoning problem stems from the fully-connected layer \( h \) that the classification score of a category relates to prototypes of other categories. Wang et al. [38] and Rymarczyk et al. [27] have proposed a constraint loss function and a shared prototype pool to mitigate this problem, respectively. However, their solutions are indirect and may bring an additional burden for training. Instead, our work addresses this problem directly by replacing the fully-connected layer with the score aggregation (SA) module. SA module aggregates the activation values of prototypes only into their allocated categories, followed by a learnable layer with weights \( \omega_{\text{SA}} \in \mathbb{R}^M \) to adjust the importance of each prototype. Specifically, let \( \tilde{w}_{\text{SA}} = e^{w_{\text{SA}}}/(\sum_{p \in \mathbb{P}_k} e^{w_{\text{SA}}}) \), the classification score \( \logit_k \) of category \( k \) is calculated in SA module as below:

\[
\logit_k = \sum_{p \in \mathbb{P}_k} \tilde{w}_{\text{SA}}^j \cdot g_{p_j}(x).
\]  

### 3.5. Training Strategy

This work is built upon the vanilla ProtoPNets but removes its cluster loss, separation loss, the prototype projection strategy and the convex optimization strategy. Additionally, this work uses an orthogonality loss \( L_{\text{ortho}} \) to diversify prototypes within the same category, which is also adopted by TesNet [38] and Deformable ProtoPNets [7]:

\[
L_{\text{ortho}} = \sum_{k=1}^{K} \| \mathbf{P}_k^k (\mathbf{P}_k^k)^\top - I_N \|^2.
\]  

Here, \( \mathbf{P}_k^k \in \mathbb{R}^{N \times D} \) denotes the concatenation of prototypes from category \( k \) and \( I_N \) is an \( N \times N \) identity matrix. Finally, the total loss \( L_{\text{total}} \) of our method is as below:

\[
L_{\text{total}} = L_{\text{ce}} + \lambda_{\text{ortho}} L_{\text{ortho}} + \lambda_{\text{align}} L_{\text{align}}.
\]  

Here, \( L_{\text{ce}} \) denotes the cross entropy loss, \( \lambda_{\text{ortho}} \) and \( \lambda_{\text{align}} \) denote the coefficient of \( L_{\text{ortho}} \) and \( L_{\text{align}} \), respectively.
4. Experiments

4.1. Experimental Settings

Dataset. We conduct experiments on CUB-200-2011 dataset [37] and Stanford Cars dataset [19] as current ProtoPNets. CUB-200-2011 dataset 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.

Benchmark Setup. Our benchmark evaluates five current ProtoPNets: ProtoPNet [6], ProtoTree [24], TesNet [38], Deformable ProtoPNet [7] and ProtoPool [27]. The consistency score and stability score of these methods are re-implemented faithfully following their released codes.

Parameters. We set $H_b$, $W_b$ and $\mu$ to be 72, 72 and 0.8 for the interpretability evaluation of all ProtoPNets. Our models are trained for 12 epochs with Adam optimizer [17] (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. These learning rates are decayed by 0.4 every 2 epochs. $1e^{-4}$, 0.3 and 0.1 are chosen for $\lambda_{\text{ortho}}$, $\lambda_{\text{align}}$ and $\gamma$. The number of prototypes per category and the dimension of prototypes are 10 and 64 for our method. More details of our experimental setup are presented in supplementary materials.

4.2. Benchmark of Prototypical Part Networks

Fig. 5 demonstrates the benchmark on CUB-200-2011 dataset over five convolutional backbones pre-trained on ImageNet. Baseline is the simplest non-interpretable model with a fully-connected layer on the last feature map for classification. In the table, we sort the ProtoPNets in ascending order of their performance. From this perspective, we can find a clear positive correlation among consistency score, stability score and accuracy within each backbone. This phenomenon accords with the definition of ProtoPNet that a prototype represents specific object parts, and ProtoPNet makes predictions by comparing the object parts from the test image and training images which are activated by the same prototypes. In such a definition, the mismatch of object parts in the test image and training images will severely drop the performance of the model, e.g., a non-consistent and non-stable model may make wrong predictions by mistakenly comparing the head part in the test image with the stomach part in training images.

The vanilla ProtoPNet [6] has inferior consistency score, stability score and accuracy owing to its simplicity. ProtoTree shares prototypes among different categories by allocating prototypes to nodes of a decision tree in one-to-one correspondence. The such design increases the interpretability of the reasoning process, but it causes a more profound negative reasoning problem [27] and leads inconsistency to prototypes. From this benchmark, ProtoPool, Deformable ProtoPNet and TesNet improve the accuracy of ProtoPNet by implicitly increasing its consistency score and stability score. Specifically, ProtoPool proposes focal similarity that can better recognize object parts by concentrating the prototype on salient visual features in different images. Deformable ProtoPNet proposes deformable prototypes that can better distinguish object parts by capturing their pose transformations. And TesNet organizes prototypes of different categories into different subspaces on the Grassmann manifold with several loss functions for constraint, which indirectly mitigates the negative reasoning problem.

We conduct more experiments to analyze the properties of consistency score based on our revised ProtoPNet introduced in Sec. 3.5. First, 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. 5, prototype $p_a$ consistently activates the head part in test images and training images, while the activation maps of prototype $p_b$ are scattered over the images desultorily.

To analyze the relation between consistency score and accuracy, we calculate the accuracy of different categories
Table 1. The comprehensive evaluation of interpretability and accuracy of ProtoPNets 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.

![Figure 7](image)

Figure 7. The consistency score increases along with the training of ProtoPNet (over three backbones).

along with their ratio of consistent prototypes. As shown in Fig. 6, the accuracy on different categories positively correlates with the ratio of consistent prototypes. Besides, Fig. 7 demonstrates the consistency score of our revised ProtoPNet at each training epoch, indicating that consistency score increases along with the model training and has a positive correlation with the model accuracy.

4.3. Analysis of Our Method

As shown in Tab. 1, with SDFA module and SA module, our method’s consistency score, stability score and accuracy are significantly superior to current ProtoPNets on CUB-200-2011 dataset over five backbones. Tab. 2 demonstrates the experimental results on 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.

![Table 2](image)

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.

4.4. Ablation Experiments

We conduct ablation experiments of our proposed SDFA module and SD module mainly on CUB-200-2011 dataset.
over five backbones. As shown in Tab. 1, SDFA module and SA module both effectively improve consistency score, stability score and accuracy of the model.

Next, we conduct experiments on the similarity between spatial similarity structures of shallow and deep layers with/without SDFA module. Given spatial similarity structure of shallow layer \( t(z_s) \) and that of deep layer \( t(z_d) \) with the same shape \( \mathbb{R}^{H_dW_d \times H_dW_d} \), the similarity \( \text{Sim}(t(z_s), t(z_d)) \) between them is calculated as below (\( Z = H_dW_d \)):

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

For accurate estimation, we generate the averaged results of Eq. (10) over all the images in the test set. As shown in Tab. 3, SDFA module improves the similarity between spatial similarity structure of shallow layers and deep layers by a large margin, which meets our expectations. Fig. 9 shows that the shallow feature maps both explicitly contain spatial information, and the deep feature map can roughly capture the object profile with SDFA module.

Besides, to verify that SA module mitigates negative reasoning, we calculate the number of similar prototypes from other categories for each prototype and present the averaged results (two prototypes with cosine similarity over 0.4 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 object parts among categories without SA module, due to negative reasoning problem.

### 5. Conclusion

This work establishes an interpretability benchmark to quantitatively evaluate the interpretability of prototypes of ProtoPNets, based on two evaluation metrics (consistency score and stability score). Furthermore, we propose an SDFA module to incorporate the spatial similarity structure from shallow layers of DNNs into deep layers and an SA module to concentrate the learning of prototypes. With these two modules, our method significantly surpasses the performance of existing ProtoPNets on three evaluation metrics (accuracy, consistency score, and stability score). Still there are a few limitations of this work: First, we require object part annotations in the dataset for interpretability evaluation, which is only suitable for some datasets. Second, we do not evaluate all the interpretability properties of ProtoPNets, e.g., whether the correspondence between prototypes and object parts is really faithful. Nevertheless, our work has great potential to facilitate more quantitative metrics to evaluate the explanation results of interpretability methods, instead of demonstrating the interpretability using limited visualization samples which can be easily misled by cherry picks. In the future, we will extend this work to more DNNs like ViT [9] and other concept embedding methods towards a unified interpretability benchmark for visual concepts.
References

[1] David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. In NIPS, 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. 2, 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] Jonathan D. Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading tea leaves: How humans interpret topic models. In NIPS, pages 288–296, 2009. 3

[6] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan Su. This looks like that: Deep learning for interpretable image recognition. In NIPS, pages 8928–8939, 2019. 1, 3, 6, 7

[7] Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable protopnet: An interpretable image classifier using deformable prototypes. In CVPR, pages 10255–10265, 2022. 1, 3, 5, 6, 7

[8] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO), pages 0210–0215, 2018. 1

[9] 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 ICLR, 2021. 8

[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] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. Xai—explainable artificial intelligence. Science robotics, page eaay7120, 2019. 1

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016. 1

[13] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In NIPS, pages 9734–9745, 2019. 3

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In CVPR, pages 4700–4708, 2017. 1

[15] Sangwon Kim, Jae-Yeal Nam, and ByoungChul Ko. Vit-net: Interpretable vision transformers with neural tree decoder. In ICML, pages 11162–11172, 2022. 1

[16] 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

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015. 6

[18] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. In ICML, pages 3519–3529, 2019. 2, 5

[19] 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

[20] 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

[21] Wenjie Luo, Yujia Li, Raquell Urtasun, and Richard S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In NIPS, pages 4898–4906, 2016. 2, 5

[22] 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

[23] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. Digit. Signal Process., pages 1–15, 2018. 3

[24] 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

[25] An-phi Nguyen and María Rodríguez Martínez. On quantitative aspects of model interpretability. arXiv preprint arXiv:2007.07584, 2020. 3

[26] Giang Nguyen, Mohammad Reza Taesiri, and Anh Nguyen. Visual correspondence-based explanations improve ai robustness and human-ai team accuracy. arXiv preprint arXiv:2208.00780, 2022. 1

[27] Dawid Rymarczyk, Łukasz Struski, Michal Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zielinski. Interpretable image classification with differentiable prototypes assignment. In ECCV, pages 351–368, 2022. 1, 2, 5, 6, 7

[28] Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zielinski. Protoshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In SIGKDD, pages 1420–1430, 2021. 1, 3

[29] 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

[30] Gesina Schwalbe. Concept embedding analysis: A review. arXiv preprint arXiv:2203.13909, 2022. 1, 2

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015. 1
[32] Dylan Slack, Sorelle A Friedler, Carlos Scheidegger, and Chitradeep Dutta Roy. Assessing the local interpretability of machine learning models. arXiv preprint arXiv:1902.03501, 2019. 3

[33] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In ICML, pages 3319–3328, 2017. 3

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In CVPR, pages 1–9, 2015. 1

[35] Stefano Teso, Özmur Alkan, Wolfgang Stammer, and Elizabeth Daly. Leveraging explanations in interactive machine learning: An overview. arXiv preprint arXiv:2207.14526, 2022. 1

[36] Grant Van Horn, Oisin 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

[37] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2, 6

[38] 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, 5, 6, 7

[39] Mengqi Xue, Qihan Huang, Haofei Zhang, Lechao Cheng, Jie Song, Minghui Wu, and Mingli Song. Protopformer: Concentrating on prototypical parts in vision transformers for interpretable image recognition. arXiv preprint arXiv:2208.10431, 2022. 1

[40] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. On the (in)fidelity and sensitivity of explanations. In NIPS, pages 10965–10976, 2019. 2, 3

[41] 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