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

Qihan Huang[1], Mengqi Xue[1], Wenqi Huang[2], Haofei Zhang[1],
Jie Song[1, 3, †], Yongcheng Jing[4], Mingli Song[1]
[1]Zhejiang University, [2]Digital Grid Research Institute, China Southern Power Grid,
[3]Zhejiang University - China Southern Power Grid Joint Research Centre on AI,
[4]The University of Sydney
{qh.huang,mqxue,haofeizhang,sjie,brooksong}@zju.edu.cn,
huangwqcsg@163.com, yjin9495@uni.sydney.edu.au

A. Experiment Details

A.1. Datasets

We conduct experiments on CUB-200-2011 dataset [43] and Stanford Cars dataset [4] following existing part-prototype networks. Besides, we also adopt a newly proposed dataset named *PartImageNet* [3]. *CUB-200-2011* covers 200 categories of 11,788 images, *Stanford Cars* covers 196 categories of 16,185 images and *PartImageNet* covers 158 categories of 24095 images. Furthermore, *CUB-200-2011* contains location annotations of object parts for each image, including 15 categories of object parts (back, beak, belly, breast, crown, forehead, left eye, left leg, left wing, nape, right eye, right leg, right wing, tail, throat). *PartImageNet* contains 41 categories of object parts. Our data augmentation settings strictly follow the first part-prototype network (ProtoPNet [1]). Precisely, each image of *CUB-200-2011* is cropped according to the bounding box annotations in the dataset. Besides, all these three datasets are augmented by offline data augmentation with random rotation, skew, shear, and left-right flip (each image is augmented 30 times).

A.2. Interpretability Benchmark

This section describes the calculation of the consistency score and stability score of ProtoTree [6] and ProtoPool [7] in detail. ProtoTree and ProtoPool both share the prototypes for different categories, and thereby we adopt a slightly different strategy to calculate their consistency score and stability score. For ProtoTree, we calculate the consistency score and stability score of each prototype on its Top-30 highest activated images, because each category in *CUB-200-2011* has around 30 images. For ProtoPool, we calculate the consistency score and stability score of each prototype in the category with the highest assignment score on it.

A.3. Details of Consistency Score

In the calculation of 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. Because some object parts are invisible in some images, we calculate each element of $a^{\mathbf{p}_j}$ 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 object part annotations in image x. Like $o^{\mathbf{p}_j}(x)$, $u(x)$ 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. And the final $a^{\mathbf{p}_j}$ is calculated as below ($\oslash$ denotes element-wise division):

$$a^{\mathbf{p}_j} = \Big(\sum_{x \in \mathcal{I}_{c(j)}} \mathrm{o}^{\mathbf{p}_j}(x) \Big) \oslash \Big(\sum_{x \in \mathcal{I}_{c(j)}} u(x) \Big).$$

A.4. Additional Components of ProtoPNet

ProtoPNet [1] adopts some additional components for model training: a cluster loss $\mathcal{L}_{\mathrm{clst}}$, a separation loss $\mathcal{L}_{\mathrm{sep}}$. This section specifies the details of these components:

Cluster Loss $\mathcal{L}_{\mathrm{clst}}$. The cluster loss is to encourage each training image to have some image patch that is close to at least one prototype from its category. Concretely, the cluster loss for image x with label y is calculated as below:

$$\mathcal{L}_{\mathrm{clst}} = \min_{j:\mathbf{p}_j \in \mathbf{P}_y} \min_{\tilde{z} \in f(x)} \|\tilde{z} - \mathbf{p}_j\|^2.$$

Separation Loss $\mathcal{L}_{\text{sep}}$. The separation loss is to keep every image patch of a training image away from the prototypes not from its category. Concretely, the separation loss for image x with label y is calculated as below:

$$\mathcal{L}_{\text{sep}} = -\min_{j:\mathbf{p}_j \notin \mathbf{P}_y} \min_{\tilde{z} \in f(x)} \|\tilde{z} - \mathbf{p}_j\|^2.$$

A.5. Hyper-parameters of the Revised ProtoPNet

We adopt some modifications to the vanilla ProtoPNet. Besides the activation function and orthogonality loss mentioned in Section 3.3.3 of the main paper, we also select some different hyper-parameters: (1) We select a single-layer add-on module instead of a double-layer one (*i.e.*, an add-on module consists of simple convolutional layers to align the dimension of the feature map to that of the prototypes). (2) We select a smaller dimension size for prototypes (*i.e.*, we set the dimension size to 64 instead of 128). (3) We set the decay rate of learning rates to 0.4 instead of 0.1.

A.6. SDFA Module

This section introduces the detailed implementation of the SDFA module. The "deep feature map" of the SDFA module is selected as the last feature map for different DNN backbones. In contrast, the feature map in the 0th layer, 8th layer and 4th layer of ResNet, VGG, and DenseNet are selected as the "shallow feature map", respectively. The common ground of these shallow feature maps is that they are all in the shape of 56 × 56.

B. Additional Experiments

B.1. Experiments on Stanford Cars Dataset

In this section, we demonstrate the performance of our method on the Stanford Cars dataset. As shown in Tab. 2, our method achieves the state-of-the-art performance superior to existing part-prototype networks over six backbones (ResNet34, ResNet152, VGG16, VGG19, DenseNet121, and DenseNet161). Besides, we can find that the performance rank of part-prototype networks on the Stanford Cars dataset is consistent with that on the CUB-200-2011 dataset.

B.2. Experiments on PartImageNet Dataset

In this section, we demonstrate the performance of our method on PartImageNet dataset [3]. PartImageNet is a newly proposed dataset with high-quality object part annotations, which is suitable for the interpretability evaluation of part-prototype networks. Therefore, we implement the experiments of ProtoPNet [1] (the first part-prototype network), TesNet [8] (the previous SOTA part-prototype network) and our model on PartImageNet. As shown in Tab. 3, our model outperforms ProtoPNet and TesNet by a large margin in both interpretability and accuracy.

Method	ProtoTree	ProtoPNet	ProtoPool	Deform.	TesNet	Ours
Original	21.6	53.8	57.6	57.0	65.4	**72.1**
IoU	18.2	35.1	38.5	41.4	56.7	**70.4**
PGD	16.4	26.2	39.9	36.5	53.1	**60.8**

Table 1. **Original:** The original stability score in the main paper. **IoU:** The stability score calculated with IoU. **PGD:** The stability score calculated using PGD attack for noise production. The results are from *CUB-200-2011* on ResNet34 backbone.

B.3. Other Variants of Stability Score

B.3.1 Stability Score with IoU

In the original version of our proposed stability score, we determine the stability of prototype $\mathbf{p}_j$ according to whether its corresponding object parts are the same in the original and perturbed images: $\mathbb{1}\{\mathrm{o}^{\mathbf{p}_j}(x) = \mathrm{o}^{\mathbf{p}_j}(x+\xi)\}$. Actually, other versions like measuring the IoU between corresponding regions of a prototype can also be used to determine stability. Intersection over Union (IoU) measures the matching degree between two regions, which calculates the ratio of intersection parts over union parts between two regions. Therefore, we slightly modify the stability score by replacing $\mathrm{o}^{\mathbf{p}_j}(x) = \mathrm{o}^{\mathbf{p}_j}(x+\xi)$ with $\mathrm{IoU}(r^{\mathbf{p}_j}(x), r^{\mathbf{p}_j}(x+\xi)) \geqslant \upsilon$:

$$S_{\text{sta}} = \frac{1}{M}\sum_{j=1}^{M} \frac{\sum_{x\in\mathcal{I}_{c(j)}} \mathbb{1}\{\mathrm{IoU}(r^{\mathbf{p}_j}(x), r^{\mathbf{p}_j}(x+\xi)) \geqslant \upsilon\}}{\|\mathcal{I}_{c(j)}\|}.$$

Here, $\mathrm{IoU}(\cdot,\cdot)$ denotes the IoU between two regions. We set υ to be 0.8 for all part-prototype networks, and as shown in Tab. 1, the evaluation results of IoU version are highly consistent with the original version.

B.3.2 Stability Score with PGD Attack

PGD attack [5] is a famous white-box attack method, which iteratively optimizes the adversarial image to attack the model through the gradients of a target loss function. Specifically, given an input image x, let x'_t denote the adversarial image generated at the t-th iteration, x'_{t+1} is generated as:

$$x'_{t+1} = \mathrm{Clip}_{x,\epsilon}(x'_t + \alpha \cdot \mathrm{sign}(\nabla_{x'_t}\mathcal{L}(x'_t, y))).$$

Here, α, $\mathcal{L}$ and y are the coefficient, the target loss function and the ground-truth, respectively. $\mathrm{sign}(\cdot)$ is a function that returns the sign of a real number. Besides, $\mathrm{Clip}_{x,\epsilon}(\cdot)$ is to clip the image values that exceed a pre-determined boundary (ϵ) of the original image x.

The vanilla PGD attack aims to attack the classification results of a network thereby it sets $\mathcal{L}$ to be the classification loss. In our case of attacking corresponding object part of prototypes, we set $\mathcal{L}$ to promote the highest activation values and suppress the smallest activation values in the activation map $v^{\mathbf{p}_j}(x) \in \mathbb{R}^{H \times W}$ of prototype $\mathbf{p}_j$ on x. In this way, the PGD attack method would attempt to reverse the activation values in the original $v^{\mathbf{p}_j}(x)$ and thus changes the corresponding region of prototype $\mathbf{p}_j$ on x. Specifically, given the prototype $\mathbf{p}_j$ and image x, the target loss function for prototype attack is defined as:

$$\mathcal{L} = -\frac{1}{\tilde{Z}}\Big(\mathrm{Top}_\omega\{v^{\mathbf{p}_j}(x)\} - \mathrm{Bottom}_\omega\{v^{\mathbf{p}_j}(x)\}\Big).$$

Here, $\tilde{Z}$ is the normalization term, $\mathrm{Top}_\omega\{v^{\mathbf{p}_j}(x)\}$ is to select the top ω elements in $v^{\mathbf{p}_j}(x)$ while $\mathrm{Bottom}_\omega\{v^{\mathbf{p}_j}(x)\}$ is to select the smallest ω ones. Finally, we utilize the generated adversarial images to calculate the stability score. For all part-prototype networks, we set ω to be 10 and conduct the PGD attack for 4 runs (with the standard hyper-parameters). As shown in Tab. 1, the evaluation results of the PGD-version stability score are overall consistent with the original version, and our model is still the most stable part-prototype network.

B.4. ViT Backbones

Existing part-prototype networks are mostly explored on the CNN backbones. In fact, they can be easily extended to ViT backbones by resizing the sequence of image tokens to be a feature map in the last layer of ViTs (*e.g.*, the sequence of image tokens with shape $\mathbb{R}^{196\times D}$ can be resized as a feature map with shape $\mathbb{R}^{14\times 14\times D}$). Therefore, we implement the Baseline model, ProtoPNet (the first part-prototype network), TesNet (the previous SOTA part-prototype network), and our model on *CUB-200-2011* with three ViT backbones (DeiT-Ti, DeiT-S, and DeiT-B). As shown in Tab. 4, the evaluation results on ViT backbones are similar to that on CNN backbones. Specifically, ProtoPNet still has poor interpretability and accuracy, and our model significantly outperforms other part-prototype networks in both interpretability and accuracy.

B.5. More Ablation Experiments

This section provides more ablation experiments of the coefficient λ_{align} and the thresh γ of $\mathcal{L}_{\text{align}}$ in the SDFA module on *CUB-200-2011*. As shown in Tab. 6, the increase of λ_{align} evidently improves the consistency score and stability score of our model over three backbones (especially when λ_{align} is small), but a too high λ_{align} may affect the learning of classification loss and thus hurts the accuracy. Furthermore, in Tab. 7, when γ increases, the consistency score and stability score of our model both reduce over three backbones. This is because $\mathcal{L}_{\text{align}}$ only restrains the pairs with dissimilar score larger than γ, and smaller γ signifies a stronger constraint of $\mathcal{L}_{\text{align}}$. But similar to λ_{align}, a too small γ would hurt the accuracy. Therefore, Tab. 6 and Tab. 7 collectively validate the availability of the SDFA module.

C. Additional Visualizations

We provide more visualization examples in this section. Fig. 1 demonstrates the visualization of object part annotations and the corresponding regions in three images. Fig. 2, Fig. 3 present the corresponding regions of two consistent prototypes from our model. To avoid the “cherry picks”, all the images in that category are visualized. Besides, Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, Fig. 11 are additional examples of reasoning process from the CUB-200-2011 dataset and the Stanford Cars dataset.

References

[1] 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, 2, 4, 6

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

[3] 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. 1, 2

[4] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV*, pages 554–561, 2013. 1

[5] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*. OpenReview.net, 2018. 2

[6] Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *CVPR*, pages 14933–14943, 2021. 1, 4, 6

[7] 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, 4, 6

[8] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *ICCV*, pages 875–884, 2021. 2, 4, 6

Method	ResNet34	ResNet152	VGG16	VGG19	Dense121	Dense161
ProtoTree [6]	86.6	N/A	N/A	N/A	N/A	N/A
ProtoPNet [1]	88.8	88.5	88.3	89.4	87.7	89.5
ProtoPool [7]	89.3	N/A	N/A	N/A	N/A	N/A
TesNet [8]	90.9	92.0	90.3	90.6	91.9	92.6
Ours + SA + SDFA	**92.0**	**92.8**	**90.8**	**91.0**	**92.4**	**92.9**

Table 2. The accuracy of part-prototype networks on *Stanford Cars*. The results are over six convolutional backbones pre-trained on ImageNet.

Method	ResNet34			ResNet152			VGG19			Dense121			Dense161		
	Con.	**Sta.**	**Acc.**	**Con.**	**Sta.**	**Acc.**	**Con.**	**Sta.**	**Acc.**	**Con.**	**Sta.**	**Acc.**	**Con.**	**Sta.**	**Acc.**
ProtoPNet [1]	15.6	70.3	79.3	27.3	72.7	82.4	15.7	69.7	77.3	11.6	69.1	80.6	10.6	70.1	83.4
TesNet [8]	42.8	82.1	83.5	36.7	77.0	87.1	23.9	71.3	81.2	47.3	77.8	83.9	48.1	80.8	86.3
Ours + SA + SDFA	**47.5**	**85.6**	**85.2**	**53.9**	**84.0**	**88.9**	**34.5**	**73.8**	**82.6**	**49.5**	**83.4**	**85.7**	**55.3**	**85.2**	**88.8**

Table 3. The comprehensive evaluation of interpretability and accuracy of part-prototype networks on *PartImageNet*. The results are over five convolutional backbones pre-trained on ImageNet. Con., Sta. and Acc. denote consistency score, stability score and accuracy, respectively. Bold font denotes the best result.

Method	DeiT-Ti			DeiT-S			DeiT-B		
	Con.	**Sta.**	**Acc.**	**Con.**	**Sta.**	**Acc.**	**Con.**	**Sta.**	**Acc.**
Baseline	N/A	N/A	81.2	N/A	N/A	82.2	N/A	N/A	82.5
ProtoPNet [1]	27.3	59.0	77.2	12.3	57.5	78.1	16.9	56.6	76.0
TesNet [8]	38.4	61.4	81.2	29.0	63.1	80.4	36.2	62.4	82.9
Ours + SA + SDFA	**48.8**	**65.7**	**84.3**	**58.3**	**67.9**	**84.6**	**51.7**	**70.4**	**85.0**

Table 4. Evaluation results of interpretability and accuracy of part-prototype networks on *CUB-200-2011*. The results are over three ViT backbones pre-trained on ImageNet. Con., Sta. and Acc. denote consistency score, stability score and accuracy, respectively. Bold font denotes the best result.

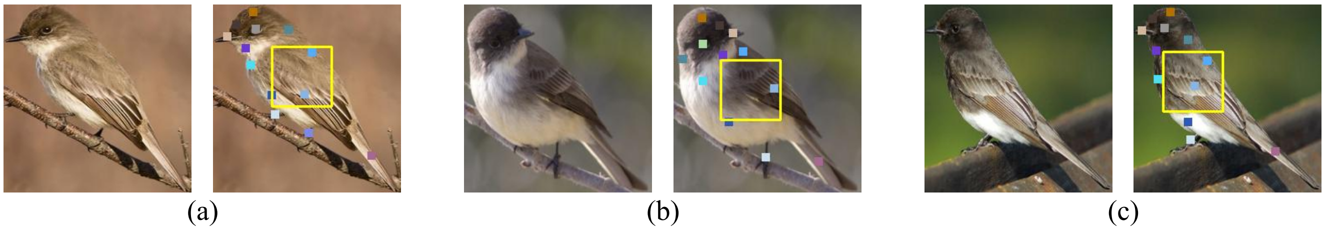


Figure 1. Visualization of object part annotations and the corresponding regions in three images. The colorful points are different object part annotations from the dataset, and the yellow bounding boxes are corresponding regions of a prototype. If an object part is inside the bounding box, we determine that this prototype corresponds to this object part in this image.

Symbol	Description	Symbol	Description
x	Input image	$c(j)$	The allocated category of prototype $\mathbf{p}_j$
f	Convolutional network of ProtoPNet	$a^{\mathbf{p}_j}$	Averaged corresponding object parts of $\mathbf{p}_j$, $a^{\mathbf{p}_j} \in \mathbb{R}^C$
$g_{\mathbf{P}}$	Prototype layer of ProtoPNet	μ	Threshold for determination of consistency of prototypes
h	Fully-connected layer of ProtoPNet	S_{con}	Consistency score of ProtoPNet
K	Number of categories in the dataset	ξ	Noise for determination of stability of prototypes
M	Number of prototypes in $g_{\mathbf{P}}$	S_{sta}	Stability score of ProtoPNet
N	Number of prototypes allocated to each category	$t(z)$	Spatial similarity structure of z, $t(z) \in \mathbb{R}^{HW \times HW}$
D	Dimension of the prototype	H_s, W_s, D_s	Height, width and dimension of a shallow feature map
$\mathbf{P}$	The set of M prototypes in $g_{\mathbf{P}}$	H_d, W_d, D_d	Height, width and dimension of a deep feature map
$\mathbf{p}_j$	The j-th prototype in $\mathbf{P}$, $\mathbf{p}_j \in \mathbb{R}^{1 \times 1 \times D}$	z_s	A shallow feature map, $z_s \in \mathbb{R}^{H_s W_s \times D_s}$
D	Dimension of the prototype	z_d	A deep feature map, $z_d \in \mathbb{R}^{H_d W_d \times D_d}$
H, W	Height and width of the feature map	$\mathcal{L}_{\text{align}}$	The shallow-deep feature alignment loss
z	The feature map extracted by f, $z \in \mathbb{R}^{H \times W \times D}$	logit_k	Classification score of category k
$\tilde{z}$	A unit of z, $\tilde{z} \in \mathbb{R}^{1 \times 1 \times D}$	γ	Threshold for $\mathcal{L}_{\text{align}}$
$v^{\mathbf{p}_j}(x)$	The activation map of $\mathbf{p}_j$ on z, $v^{\mathbf{p}_j}(x) \in \mathbb{R}^{H \times W}$	w^{SA}	Weight of SA module, $w^{\text{SA}} \in \mathbb{R}^M$
$g^{\mathbf{p}_j}(x)$	The activation value of $\mathbf{p}_j$ on z	$\tilde{w}^{\text{SA}}$	Normalization of w^{SA}
$\text{Sim}(\cdot, \cdot)$	Similarity score between two vectors	$\mathcal{L}_{\text{total}}$	The total loss
$r^{\mathbf{p}_j}(x)$	The corresponding region of $\mathbf{p}_j$ on x	$\mathbf{P}^k$	Concatenation of prototypes from category k, $\mathbf{P}^k \in \mathbb{R}^{N \times D}$
H_b, W_b	Height and width of the corresponding region	$\mathbb{I}_N$	The $N \times N$ identity matrix
C	Number of categories of object parts in the dataset	$\mathcal{L}_{\text{ortho}}$	The orthogonality loss
$o^{\mathbf{p}_j}(x)$	Corresponding object parts of $\mathbf{p}_j$ on x, $o^{\mathbf{p}_j}(x) \in \mathbb{R}^C$	$\mathcal{L}_{\text{clst}}$	The cluster loss
$u(x)$	Visible object parts in x, $u(x) \in \mathbb{R}^C$	λ_{align}	Coefficient of $\mathcal{L}_{\text{align}}$
$\mathcal{I}_k$	Test images belonging to category k	$\mathcal{L}_{\text{ce}}$	The cross entropy loss
$\oplus$	Concatenation of two vectors	$\mathcal{L}_{\text{sep}}$	The separation loss

Table 5. Summary of notations used in the main body of this paper.

Backbone	0.30			0.40			0.50[†]			0.60			0.70		
	Con.	Sta.	Acc.	Con.	Sta.	Acc.	Con.	Sta.	Acc.	Con.	Sta.	Acc.	Con.	Sta.	Acc.
ResNet34	69.5	70.6	83.7	70.4	71.2	83.7	70.6	72.1	84.0	71.5	72.4	83.8	71.8	72.4	83.6
VGG19	53.5	62.4	82.6	56.6	62.5	82.8	56.5	63.5	82.5	58.4	63.9	82.5	58.9	64.8	82.2
Dense121	67.1	65.4	85.4	67.9	66.5	85.2	68.1	67.6	85.4	69.7	68.0	85.2	70.5	68.6	84.8

Table 6. Ablation experiments of the coefficient λ_{align} of the alignment loss $\mathcal{L}_{\text{align}}$ on *CUB-200-2011*. † denotes the parameter we choose in the main experiments. The results indicate that the increase of λ_{align} improves the consistency score and stability score of our model, but a too high λ_{align} may affect the learning of classification loss and thus hurts the accuracy.

Backbone	**0.05**			**0.10†**			**0.15**			**0.20**			**0.25**		
	Con.	**Sta.**	**Acc.**	**Con.**	**Sta.**	**Acc.**	**Con.**	**Sta.**	**Acc.**	**Con.**	**Sta.**	**Acc.**	**Con.**	**Sta.**	**Acc.**
ResNet34	72.4	72.3	83.5	70.6	72.1	84.0	69.9	71.2	83.9	68.8	70.6	84.2	69.3	69.5	83.8
VGG19	58.6	64.2	82.1	56.5	63.5	82.5	56.6	62.7	82.5	55.9	62.8	82.8	55.5	62.0	82.7
Dense121	71.2	67.8	84.8	68.1	67.6	85.4	67.7	66.9	85.4	66.5	65.2	85.2	67.4	64.3	85.1

Table 7. Ablation experiments of the thresh γ of the alignment loss $\mathcal{L}_{\text{align}}$ on *CUB-200-2011*. † denotes the parameter we choose in the main experiments. The results indicate that the decrease of γ improves the consistency score and stability score of our model. But similar to λ_{align}, a too small γ hurts the accuracy.

Method	**ResNet34**	**ResNet152**	**VGG19**	**Dense121**	**Dense161**
Baseline	**21.39M**	58.55M	**20.17M**	7.16M	26.91M
ProtoTree [6]	21.55M	58.80M	20.29M	7.35M	27.17M
ProtoPNet [1]	22.02M	59.08M	20.76M	7.76M	27.43M
ProtoPool [7]	22.27M	59.53M	21.01M	8.07M	27.90M
Deformable [2]	26.84M	79.13M	25.58M	17.65M	49.07M
TesNet [8]	21.85M	58.80M	20.59M	7.55M	27.14M
Ours + SA + SDFA	21.45M	**58.40M**	20.19M	**7.15M**	**26.74M**

Table 8. The number of parameters in different part-prototype networks (Measured on *CUB-200-2011*). Bond font denotes the minimum number of parameters.

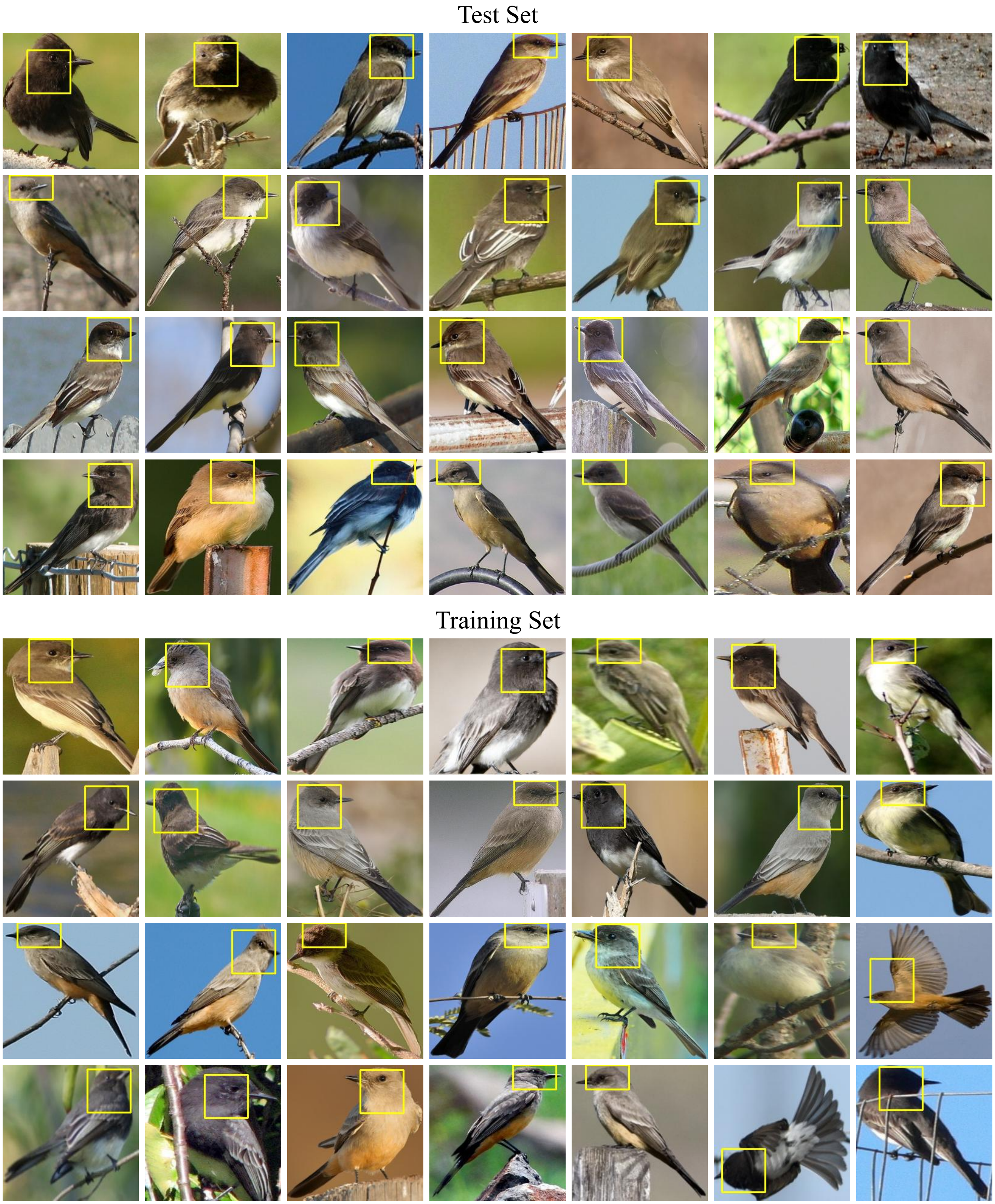
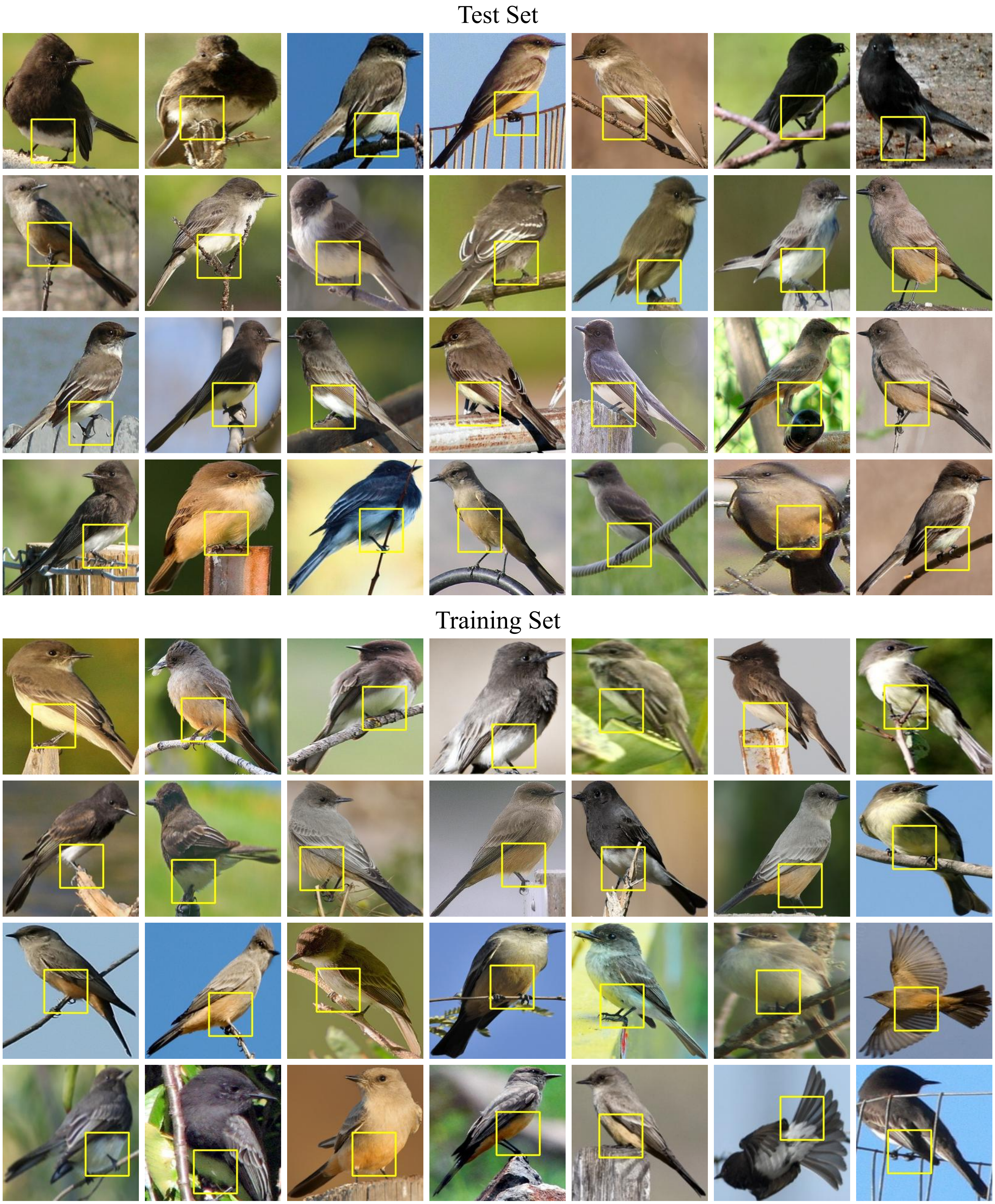


Figure 2. A consistent prototype $\mathbf{p}$ from our model, which corresponds to the head in all the images of category Sayornis ($\max(a^{\mathbf{p}}) = 1.00$).



Test Set

Training Set

Figure 3. A consistent prototype $\mathbf{p}$ from our model, which corresponds to the feet in all the images of category Sayornis ($\max(a^{\mathbf{p}}) = 0.93$).

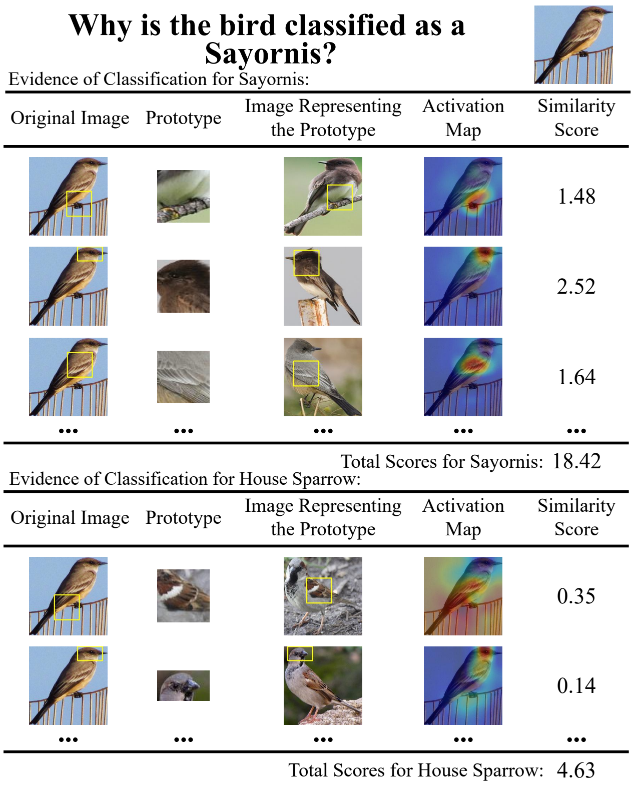

Why is the bird classified as a Sayornis?
Evidence of Classification for Sayornis:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
1.48
2.52
1.64
...
Total Scores for Sayornis: 18.42
Evidence of Classification for House Sparrow:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
0.35
0.14
...
Total Scores for House Sparrow: 4.63

Figure 4. Reasoning process of a Sayornis.

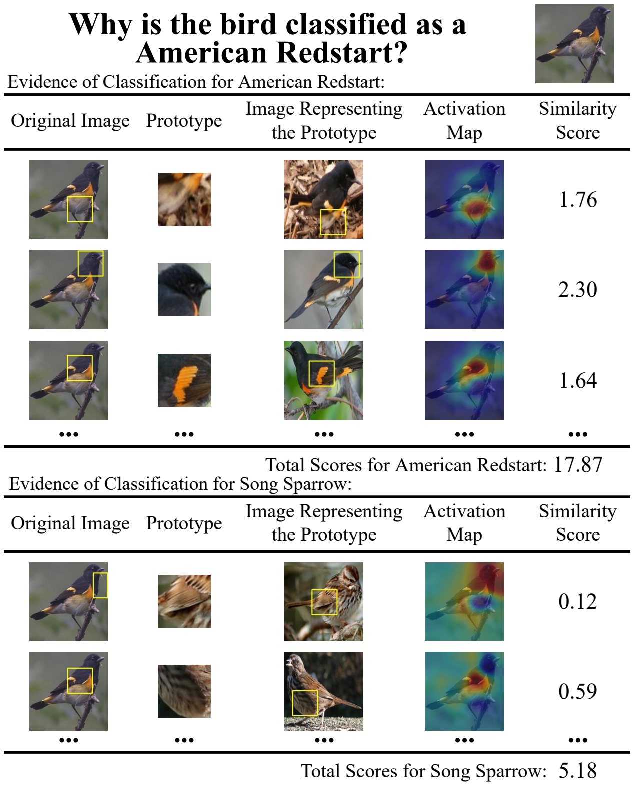

Why is the bird classified as a American Redstart?
Evidence of Classification for American Redstart:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
1.76
2.30
1.64
...
Total Scores for American Redstart: 17.87
Evidence of Classification for Song Sparrow:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
0.12
0.59
...
Total Scores for Song Sparrow: 5.18

Figure 5. Reasoning process of a American Redstart.

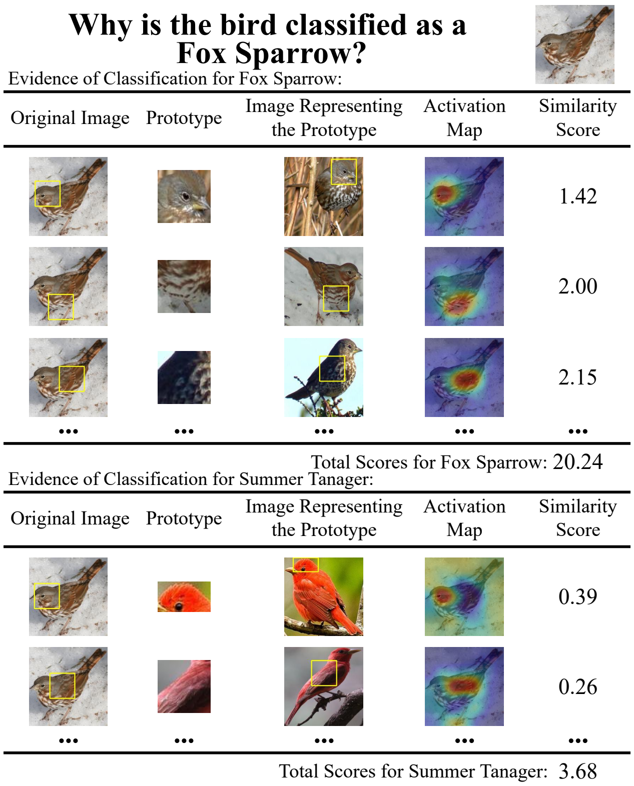

Why is the bird classified as a Fox Sparrow?
Evidence of Classification for Fox Sparrow:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
1.42
2.00
2.15
...
Total Scores for Fox Sparrow: 20.24
Evidence of Classification for Summer Tanager:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
0.39
0.26
...
Total Scores for Summer Tanager: 3.68

Figure 6. Reasoning process of a Fox Sparrow.

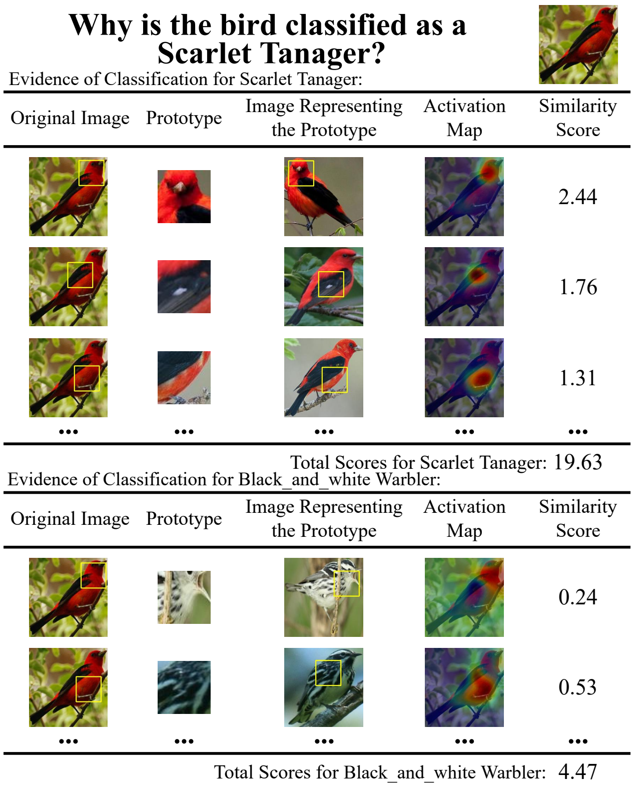

Why is the bird classified as a Scarlet Tanager?
Evidence of Classification for Scarlet Tanager:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
2.44
1.76
1.31
...
Total Scores for Scarlet Tanager: 19.63
Evidence of Classification for Black_and_white Warbler:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
0.24
0.53
...
Total Scores for Black_and_white Warbler: 4.47

Figure 7. Reasoning process of a Scarlet Tanager.

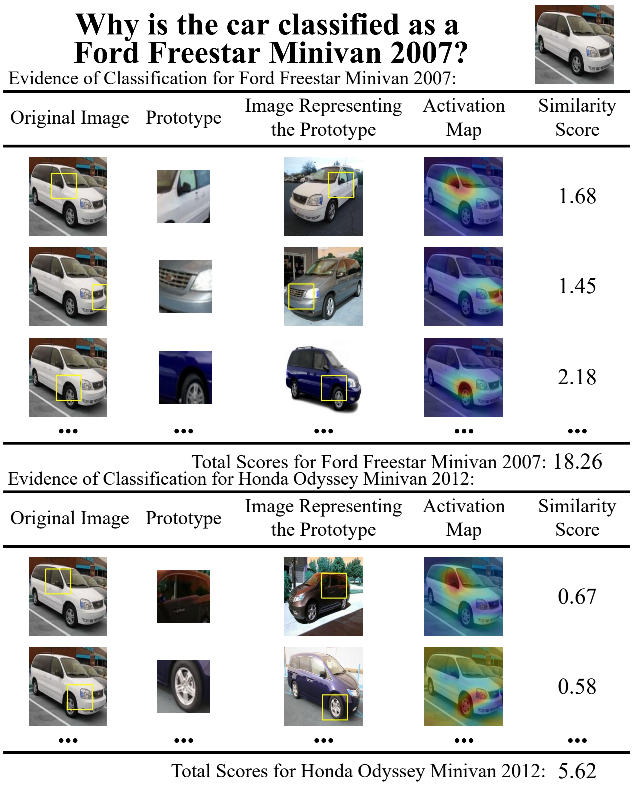

Why is the car classified as a
Ford Freestar Minivan 2007?
Evidence of Classification for Ford Freestar Minivan 2007:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
1.68
1.45
2.18
Total Scores for Ford Freestar Minivan 2007: 18.26
Evidence of Classification for Honda Odyssey Minivan 2012:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
0.67
0.58
Total Scores for Honda Odyssey Minivan 2012: 5.62

Figure 8. Reasoning process of a Ford Freestar Minivan 2007.

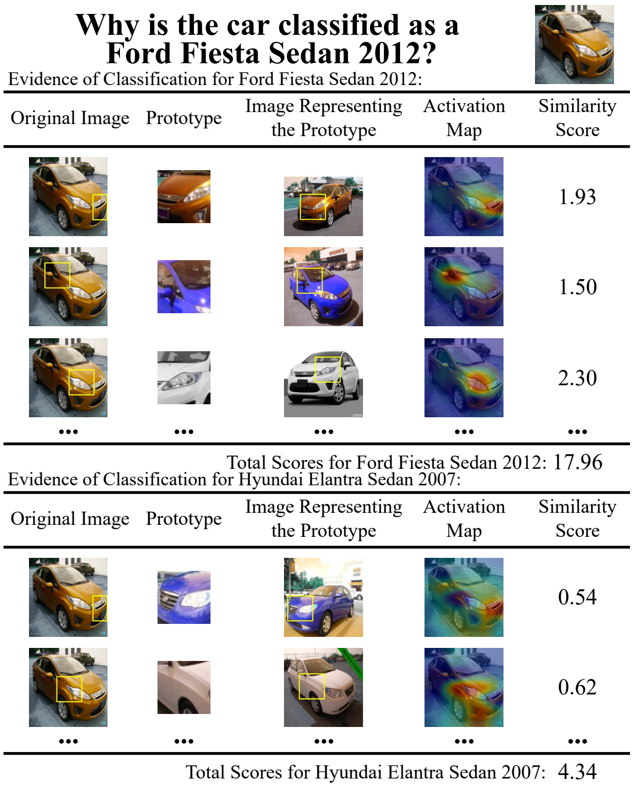

Why is the car classified as a
Ford Fiesta Sedan 2012?
Evidence of Classification for Ford Fiesta Sedan 2012:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
1.93
1.50
2.30
Total Scores for Ford Fiesta Sedan 2012: 17.96
Evidence of Classification for Hyundai Elantra Sedan 2007:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
0.54
0.62
Total Scores for Hyundai Elantra Sedan 2007: 4.34

Figure 9. Reasoning process of a Ford Fiesta Sedan 2012.

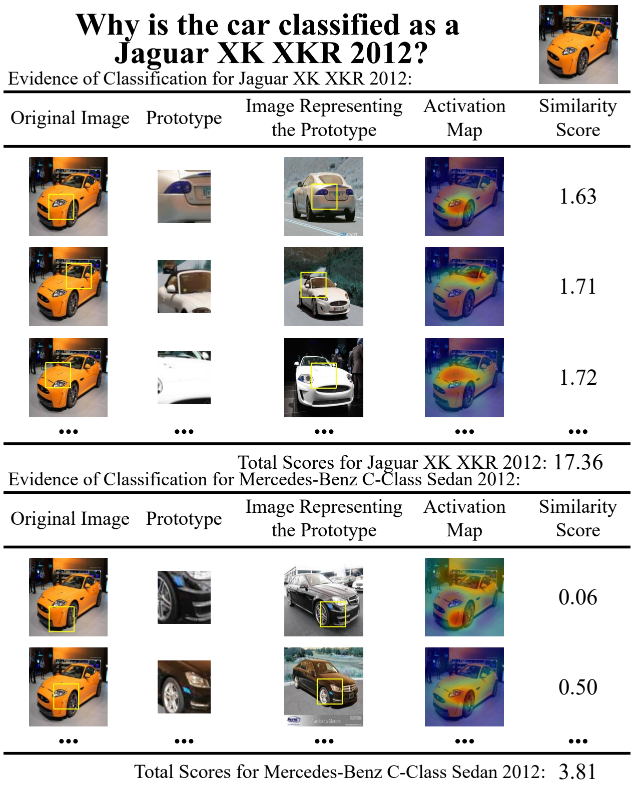

Why is the car classified as a
Jaguar XK XKR 2012?
Evidence of Classification for Jaguar XK XKR 2012:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
1.63
1.71
1.72
Total Scores for Jaguar XK XKR 2012: 17.36
Evidence of Classification for Mercedes-Benz C-Class Sedan 2012:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
0.06
0.50
Total Scores for Mercedes-Benz C-Class Sedan 2012: 3.81

Figure 10. Reasoning process of a Jaguar XK XKR 2012.

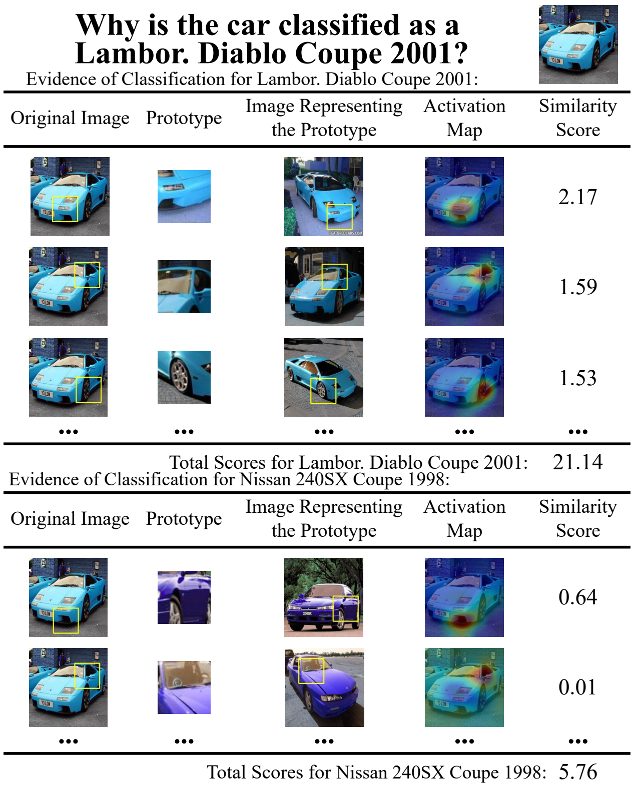

Why is the car classified as a
Lambor. Diablo Coupe 2001?
Evidence of Classification for Lambor. Diablo Coupe 2001:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
2.17
1.59
1.53
Total Scores for Lambor. Diablo Coupe 2001: 21.14
Evidence of Classification for Nissan 240SX Coupe 1998:
Original Image
Prototype
Image Representing the Prototype
Activation Map
Similarity Score
0.64
0.01
Total Scores for Nissan 240SX Coupe 1998: 5.76

Figure 11. Reasoning process of a Lambor. Diablo Coupe 2001.