Supplementary Material of Paper: Keep the Faith: Faithful Explanations in Convolutional Neural Networks for Case-Based Reasoning

A Appendix

A.1 Derivation of ReLU1 Moments

We analytically derive expectation E and variance V of a layer input mean μ and variance σ^2 for ReLU1 for $E(\mu,\sigma) = \int g(x)\frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx$ and variance $V(\mu,\sigma) = \int (g(x)-E(\mu,\sigma))^2\frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx$, with ϕ the standard Gaussian probability density function $\phi = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$, the corresponding cumulative distribution function $\Phi(x) = \int_{-\infty}^{x}\phi(t)\,dt$ (Frey and Hinton 1999), and $g(x) = \text{ReLU1}(x)$ defined as:

$$\text{ReLU1}(x) = \begin{cases} 0, & \text{if x} < 0, \\ x, & \text{if } 0 \leq \text{x} \leq 1, \\ 1, & \text{if x} > 1. \end{cases}$$

Note that this derivation can serve as a blueprint to derive mean and variance for any ReLU with an upper bound by adapting the integration limits. We use color coding to emphasize where replaced equations come from.

$$\begin{aligned} E(\mu,\sigma) &= \int_{-\infty}^{\infty} \text{ReLU1}(x)\frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx \\ &= \int_{-\infty}^{0} \text{ReLU1}(x)\frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx + \int_{0}^{1} \text{ReLU1}(x)\frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx + \int_{1}^{\infty} \text{ReLU1}(x)\frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx \\ &= \int_{0}^{1} \frac{x}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx + \int_{1}^{\infty} \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx. \end{aligned}$$

$$\begin{aligned} \int_{0}^{1} \frac{x}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx &= \frac{1}{\sigma}\int_{0}^{1} x\phi(a+bx)\,dx, \text{ with } a = -\frac{\mu}{\sigma},\ b = \frac{1}{\sigma} \\ &\overset{\text{Eq.101 (Owen 1980)}}{=} \frac{1}{\sigma}[-\frac{1}{b^2}(\phi(a+bx) + a\Phi(a+bx))]\Big|_0^1 \\ &= \frac{1}{\sigma}[-\sigma^2(\phi(\frac{x-\mu}{\sigma}) - \frac{\mu}{\sigma}\Phi(\frac{x-\mu}{\sigma}))]\Big|_0^1 \\ &= -\sigma\phi(\frac{x-\mu}{\sigma}) + \mu\Phi(\frac{x-\mu}{\sigma})\Big|_0^1 \\ &= -\sigma\phi(\frac{1-\mu}{\sigma}) + \mu\Phi(\frac{1-\mu}{\sigma}) + \sigma\phi(-\frac{\mu}{\sigma}) - \mu\Phi(-\frac{\mu}{\sigma}) \\ &= \sigma(\phi(-\frac{\mu}{\sigma}) - \phi(\frac{1-\mu}{\sigma})) + \mu(\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})). \end{aligned}$$

$$\int_1^\infty \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx = \Phi(\frac{x-\mu}{\sigma})\Big|_1^\infty$$
$$= 1 - \Phi(\frac{x-\mu}{\sigma})\Big|_{-\infty}^1$$
$$= 1 - \Phi(\frac{1-\mu}{\sigma}).$$

$$E(\mu,\sigma) = \sigma(\phi(-\frac{\mu}{\sigma}) - \phi(\frac{1-\mu}{\sigma})) + \mu(\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})) + 1 - \Phi(\frac{1-\mu}{\sigma}).$$

$$V(\mu,\sigma) = \int_\infty^\infty (\text{ReLU1}(x) - E(\mu,\sigma))^2 \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx$$

Let $E(\mu,\sigma) = \bar{\mu}$.

$$V(\mu,\sigma) = \int_\infty^\infty (\text{ReLU1}(x) - \bar{\mu})^2 \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx$$
$$= \int_{-\infty}^0 \bar{\mu}^2 \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx + \int_0^1 (x-\bar{\mu})^2 \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx + \int_1^\infty (1-\bar{\mu})^2 \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx.$$

$$\int_{-\infty}^0 \bar{\mu}^2 \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx = \bar{\mu}^2 \int_{-\infty}^0 \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx$$
$$= \bar{\mu}^2 \Phi(-\frac{\mu}{\sigma}).$$

$$\int_0^1 (x-\bar{\mu})^2 \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx$$
$$= \int_0^1 (x^2 - 2\bar{\mu}x + \bar{\mu}^2) \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx$$
$$= \int_0^1 \frac{x^2}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx + \int_0^1 -\frac{2\bar{\mu}x}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx + \int_0^1 \frac{\bar{\mu}^2}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx.$$

$$\int_0^1 \frac{x^2}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx = \frac{1}{\sigma}\int_0^1 x^2 \phi(\frac{x-\mu}{\sigma})\,dx$$
$$= \frac{1}{\sigma}\int_0^1 x^2 \phi(a+bx)\,dx, \text{ with } a = -\frac{\mu}{\sigma},\ b = \frac{1}{\sigma}$$
$$\overset{\text{Eq.102 (Owen 1980)}}{=} \frac{1}{\sigma}[\frac{1}{b^3}((a^2+1)\Phi(a+bx) + (a-bx)\phi(a+bx))]\Big|_0^1$$
$$= \frac{1}{\sigma}\,[\sigma^3((\frac{\mu^2}{\sigma^2}+1)\Phi(\frac{x-\mu}{\sigma}) - \frac{x+\mu}{\sigma}\phi(\frac{x-\mu}{\sigma}))]\Big|_0^1$$
$$= \sigma^2((\frac{\mu^2}{\sigma^2}+1)\Phi(\frac{x-\mu}{\sigma}) - \frac{x+\mu}{\sigma}\phi(\frac{x-\mu}{\sigma}))\Big|_0^1$$
$$= (\mu^2+\sigma^2)\Phi(\frac{x-\mu}{\sigma}) - \sigma(x+\mu)\phi(\frac{x-\mu}{\sigma})\Big|_0^1$$
$$= (\mu^2+\sigma^2)\Phi(\frac{1-\mu}{\sigma}) - (\sigma+\mu\sigma)\phi(\frac{1-\mu}{\sigma}) - (\mu^2+\sigma^2)\Phi(-\frac{\mu}{\sigma}) + \sigma\mu\phi(-\frac{\mu}{\sigma})$$
$$= (\mu^2+\sigma^2)(\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})) - (\sigma+\mu\sigma)\phi(\frac{1-\mu}{\sigma}) + \sigma\mu\phi(-\frac{\mu}{\sigma}).$$

$$\int_0^1 -\frac{2\bar{\mu}x}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx = -2\bar{\mu}(\frac{1}{\sigma}\int_0^1 x\phi(\frac{x-\mu}{\sigma})\,dx)$$
$$= -2\bar{\mu}(\sigma(\phi(-\frac{\mu}{\sigma}) - \phi(\frac{1-\mu}{\sigma})) + \mu(\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma}))).$$

$$\int_0^1 \frac{\bar{\mu}^2}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx = \bar{\mu}^2\int_0^1 \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx$$
$$= \bar{\mu}^2(\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})).$$

$$\int_0^1 (x-\bar{\mu})^2\frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx$$
$$= (\mu^2+\sigma^2)(\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})) - (\sigma+\mu\sigma)\phi(\frac{1-\mu}{\sigma}) + \sigma\mu\phi(-\frac{\mu}{\sigma})$$
$$- 2\bar{\mu}(\sigma(\phi(-\frac{\mu}{\sigma}) - \phi(\frac{1-\mu}{\sigma})) + \mu(\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})))$$
$$+ \bar{\mu}^2(\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})).$$

$$\int_1^\infty (1-\bar{\mu})^2\frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx$$
$$= (1-\bar{\mu})^2\int_1^\infty \frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})\,dx$$
$$= (1-\bar{\mu})^2(1 - \Phi(\frac{1-\mu}{\sigma})).$$

$$V(\mu,\sigma) = \bar{\mu}^2\Phi(-\frac{\mu}{\sigma})$$
$$+ (\mu^2+\sigma^2)(\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})) - (\sigma+\mu\sigma)\phi(\frac{1-\mu}{\sigma}) + \sigma\mu\phi(-\frac{\mu}{\sigma})$$
$$- 2\bar{\mu}(\sigma(\phi(-\frac{\mu}{\sigma}) - \phi(\frac{1-\mu}{\sigma})) + \mu(\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})))$$
$$+ \bar{\mu}^2(\Phi(\frac{1-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})) + (1-\bar{\mu})^2(1 - \Phi(\frac{1-\mu}{\sigma}))$$
$$= (\mu^2 - 2\mu\bar{\mu} + \sigma^2 + 2\bar{\mu} - 1)\Phi(\frac{1-\mu}{\sigma})$$
$$- (\mu^2 - 2\mu\bar{\mu} + \sigma^2)\Phi(-\frac{\mu}{\sigma})$$
$$- (\mu\sigma - 2\bar{\mu}\sigma + \sigma)\phi(\frac{1-\mu}{\sigma})$$
$$+ (\mu\sigma - 2\bar{\mu}\sigma)\phi(-\frac{\mu}{\sigma})$$
$$+ \bar{\mu}^2 - 2\bar{\mu} + 1.$$

A.2 Training, Dataset and Implementation Details

Model Architecture. For the backbones, we use the pre-trained feature extractors of ResNet50, ResNeXt50, and Wide-ResNet50 provided by torchvision[1]. The resulting number of trainable parameters for ProtoPFaith

[1] torchvision is available at: https://github.com/pytorch/vision

are 24,314,816 (ResNet50), 23,786,688 (ResNeXt50), and 67,641,024 (Wide-ResNet50) for the models trained on CUB-200-2011. ResNet follows the ResNet18 backbone with just one input channel. In the original ResNet18, the number of channels is increased every two ResBlocks. We increase the capacity by adding one additional ResBlock before every increase in the number of channels (the resulting number of trainable parameters for ProtoPFaith is 17,825,528). The ConvNet is the same architecture as our ResNet but without skip connections (the resulting number of trainable parameters for ProtoPFaith is 17,319,928).

Data. For CUB-200-2011 and Stanford Dogs, we use the official test set as a hold-out test set and randomly split the training set into five folds stratified by the class label. For RSNA, we use the official training set only. We split 20% for a hold-out test set, stratified by age, sex, and class labels. Additionally, we split the remaining 80% into five folds. We resize all images to 224x224 resolution (bilinear interpolation).

Training Details and Evaluation. For ResNet50, ResNeXt50, and Wide-ResNet50, we initialize weights of the encoder V with pre-trained weights of ImageNet-1k[2] provided by torchvision. We optimize with AdamW (Loshchilov and Hutter 2017). Following the implementation of Chen et al. (2019), we first do a warm-up of parameters of Z and Q for ten epochs, starting at $\frac{1}{10}$-th of the initial learning and linearly increasing every iteration to $\frac{1}{5}$-th of the initial learning rate. Then, we cycle between optimizing all parameters related to feature extraction (V, Z, Q), and classification (F) for five and ten epochs, respectively. During each of these cycles, we perform two learning rate scheduling cycles with the Cyclic Learning Rate Scheduler provided by torchvsision (base learning rate of $\frac{1}{5}$-th of the initial lr, max learning rate is the initial learning rate, linear increase and decrease updated every iteration). We carry out a grid search for hyper-parameters on one split of CUB-200-2011 with ResNet50 based on the validation balanced accuracy of the validation set of this split. We follow the training routine proposed in Chen et al. (2019) and set the coefficients before cluster $\mathcal{L}_{\text{clst}}$ and separation cost $\mathcal{L}_{\text{sep}}$ to $\lambda_1 = \lambda_2 = 0.5$:

$$\begin{aligned}
\mathcal{L}(y, \mathcal{I}) &= \mathcal{L}_{\text{CE}}(y, (F \circ Q \circ Z \circ V)(\mathcal{I})) \\
&\quad + \lambda_1 \mathcal{L}_{\text{clst}}(y, (Z \circ V)(\mathcal{I})) \\
&\quad + \lambda_2 \mathcal{L}_{\text{sep}}(y, (Z \circ V)(\mathcal{I})), \\
\mathcal{L}_{\text{clst}}(y, (Z \circ V)(\mathcal{I})) &= \min_{c=y,k,i,j} \|p_k^c - z_{i,j}\|_2^2, \\
\mathcal{L}_{\text{sep}}(y, (Z \circ V)(\mathcal{I})) &= \min_{c\neq y,k,i,j} \|p_k^c - z_{i,j}\|_2^2,
\end{aligned}$$

with $\mathcal{L}_{\text{CE}}$ the cross-entropy loss and y the target label of the input image $\mathcal{I}$. Then, we re-train the model with the best set of hyper-parameters five times (once on each training set of each split) with early stopping for 1000 epochs. We use this hyper-parameter set to train for all models on CUB-200-2011 and Stanford Dogs. The hyper-parameter search space is (fat numbers denote the best configuration):

- Learning Rate: [0.022, 0.01, 0.0022, 0.001, **0.00022**, 0.0001]
- Weight Decay: [0.0, **0.001**]
- (K) Prototypes per Class: [3, **5**, 7]
- (L) Channels for Prototypes: [**128**, 256, 512].

We apply random affine transformations (-25..25 degrees, shear of 15, probability of 1.0), followed by random horizontal flip (probability of 0.5), resizing to 224x224 (bilinear interpolation), and normalization (mean=[0.485, 0.456, 0.406], standard deviation=[0.229, 0.224, 0.225]).

For RSNA, we pre-train ConvNet and ResNet backbones five times (once on each training set of each split) to yield five sets of pre-trained weights (one for each split), which prevents bias towards the validation set. We use a learning rate of 0.001, weight decay of 0.0, and train for 100 epochs. Then, we carry out a hyper-parameter grid search for ProtoPNet on one split (same search space as for CUB-200-2011) and re-train with the best set of hyper-parameters five times (once on each training set of each split, for 100 epochs with early stopping).

[2]ImageNet-1k is available at https://www.image-net.org/download.php

The best configuration for ConvNet is: Learning Rate of 0.0001, Weight Decay of 0.0, (K) Prototypes per Class of 7, (L) Channels for Prototypes of 256. The best configuration for ResNet is: Learning Rate of 0.00022, Weight Decay of 0.0, (K) Prototypes per Class of 7, (L) Channels for Prototypes of 512. We apply random affine transformations (-45..45 degrees, translation of -0.15..0.15% of the image size in each direction, the scale of 85..115%, probability of 1.0), and rescale intensities to 0..1.

As we have five models for each architecture and dataset, we report the mean and standard deviation of the balanced accuracy on the five validation sets and their performance on the test set. For each model trained on CUB-200-2011 or Stanford Dogs, we randomly select 100 prototypes (total prototypes are 1000 and 600, respectively) and all 14 prototypes for models trained on RSNA. We create our proposed explanations using 32 network evaluations per input feature for DASP (Ancona, Oztireli, and Gross 2019) (approximately 90 minutes per image on an NVIDIA A10040GB). We calculate the AOPC score by iterative removal of the most important features, i.e., in every iteration, one more feature (the next, most important one) is replaced with 0. Therefore, the evaluation of Eq. 10 is expected to decrease faster for higher-quality explanations. Note that the squared L2-distance has its minimum at 0. As we evaluate the change in the input image of a prototype, $s_{p_k^c}(\mathcal{I}^{(0)}_{\xi(p_k^c)})$ is zero and the whole Eq. 10 becomes negative for all steps.

A.3 Additional Results

We report the Balanced Accuracy (mean and standard deviation in %) for all models and the datasets on which they were evaluated in Tab. A1. For all models, there is only marginal overfitting towards the validation set (drop-off in performance is $\leq 1\%$ for all models). Note that ResNeXt50 outperforms Wide-ResNet50 on CUB-200-2011, while it lacks behind 0.6% on Stanford Dogs. ConvNet outperforms ResNet on RSNA. Additionally, the drop-off in performance from the validation set to the test set is 1.4% for the ConvNet and 1.8% for the ResNet. Compared to ResNet, the standard deviation of ConvNet is +5.6% on the test set, which indicates training instabilities w.r.t. the dataset split. While it can be argued that ResNet is more stable in this regard, we can observe from Fig. A1 that ResNet achieved a high test performance by just learning a single prototype for one class. Therefore, this model should not be applied in clinical practice, which is an important finding only enabled by the architecture of case-based reasoning.

Dataset	Model Backbone	Validation BAcc	Test BAcc
CUB-200-2011	ResNeXt50	72.8 ± 0.5	72.0 ± 0.6
CUB-200-2011	ResNet50	70.8 ± 1.1	69.8 ± 0.8
CUB-200-2011	Wide-ResNet50	71.0 ± 0.9	70.2 ± 0.7
Stanford Dogs	ResNeXt50	84.1 ± 0.3	83.4 ± 0.6
Stanford Dogs	ResNet50	81.0 ± 0.6	80.4 ± 0.5
Stanford Dogs	Wide-ResNet50	84.1 ± 1.0	84.0 ± 0.6
RSNA	ConvNet	74.0 ± 10.1	72.6 ± 10.8
RSNA	ResNet	73.7 ± 4.1	71.9 ± 5.2

Table A1: Balanced Accuracy (BAcc) scores for the models evaluated in this work.

We show the AOPC for all evaluated models in Tab. A2 (lower numbers are better). In all cases, the explanations given by ProtoPFaith outperform the explanations given by ProtoPNet by a factor $>10^3$. Note that the training of ResNet on RSNA collapsed: The model learned only one prototype for one class and just two prototypes for the other class (see Fig. A1). However, the model still achieved a Balanced Accuracy of 79.8%. The even bigger gap between explanations for this model shows a multiplicative effect with respect to duplicate prototypes.

Dataset	Model	Explanation	AOPC($Q \circ Z \circ V$)
CUB-200-2011	ResNeXt50	ProtoPNet	-0.000446
CUB-200-2011	ResNeXt50	ProtoPFaith	-3.721184
CUB-200-2011	ResNet50	ProtoPNet	-0.000223
CUB-200-2011	ResNet50	ProtoPFaith	-2.848353
CUB-200-2011	Wide-ResNet50	ProtoPNet	-0.000834
CUB-200-2011	Wide-ResNet50	ProtoPFaith	-3.075180
Stanford Dogs	ResNeXt50	ProtoPNet	-0.001767
Stanford Dogs	ResNeXt50	ProtoPFaith	-1.850909
Stanford Dogs	ResNet50	ProtoPNet	-0.000516
Stanford Dogs	ResNet50	ProtoPFaith	-3.012445
Stanford Dogs	Wide-ResNet50	ProtoPNet	-0.000717
Stanford Dogs	Wide-ResNet50	ProtoPFaith	-3.006323
RSNA	ConvNet	ProtoPNet	-0.001360
RSNA	ConvNet	ProtoPFaith	-8.816450
RSNA	ResNet	ProtoPNet	-0.000012
RSNA	ResNet	ProtoPFaith	-31.912595

Table A2: Area Over the Perturbation Curve (AOPC) scores for the models evaluated in this work.

A.4 Additional Visualization of Explanations

We perform forward passes for all architectures evaluated in this work and visualize the explanations. This is a ResNet in Fig. A1 and a ConvNet in Fig. A2 on the RSNA dataset. On the Standford Dogs dataset, this is a Wide-ResNet50 in Fig. A3, a ResNeXt50 in Fig. A4, and a ResNet50 in Fig. A5. On the CUB-200-2011 dataset, this is a Wide-ResNet50 in Fig. A6, a ResNeXt50 in Fig. A7, and a ResNet50 in Fig. A8. All figures follow the structure presented in the top left of Fig. 3.

We can see in Fig. A1 that the prototypes collapsed during training as there are only three prototypes learned in total. The explanations given by ProtoPNet show high activation for a lot of background of the test sample. In contrast, the ProtoPFaith extracts explanations that lie within the torso. Additionally, they distinguish between the lung and other parts of the image. The same accounts for ConvNet trained on RSNA (see Fig. A2), where some explanations given by ProtoPFaith discard the rips. In contrast to ResNet, the explanations show that the model learned local features, as most of the input image has little effect on the Shapley value. However, the explanations given by ProtoPNet are mostly in the background. For models trained on Stanford Dogs (see Fig. A3-A5), the explanations given by ProtoPNet again suggest locality. ProtoPFaith extracts explanations across the whole input domain, but the contours or body parts of the dogs are clearly visible for most explanations. On CUB-200-2011 (see Fig. A6-A8), explanations given by ProtoPFaith show a bias towards the background, which is also the case for some explanations of ProtoPNet. Like the explanations for the models trained on Stanford Dogs, the explanations of ProtoPFaith highlight the contours or body parts of birds, like the head, tail, or beak.

In summary, ProtoPNet extracts misleading local explanations, and ProtoPFaith extracts granular, faithful explanations. This prohibits the extraction of crops of the input image for visualization of prototypes, but the explanations of ProtoPFaith faithfully show *how* the model transformed the input and *what* lead to the model prediction. Additionally, granularity showed benefits when applied to medical images.

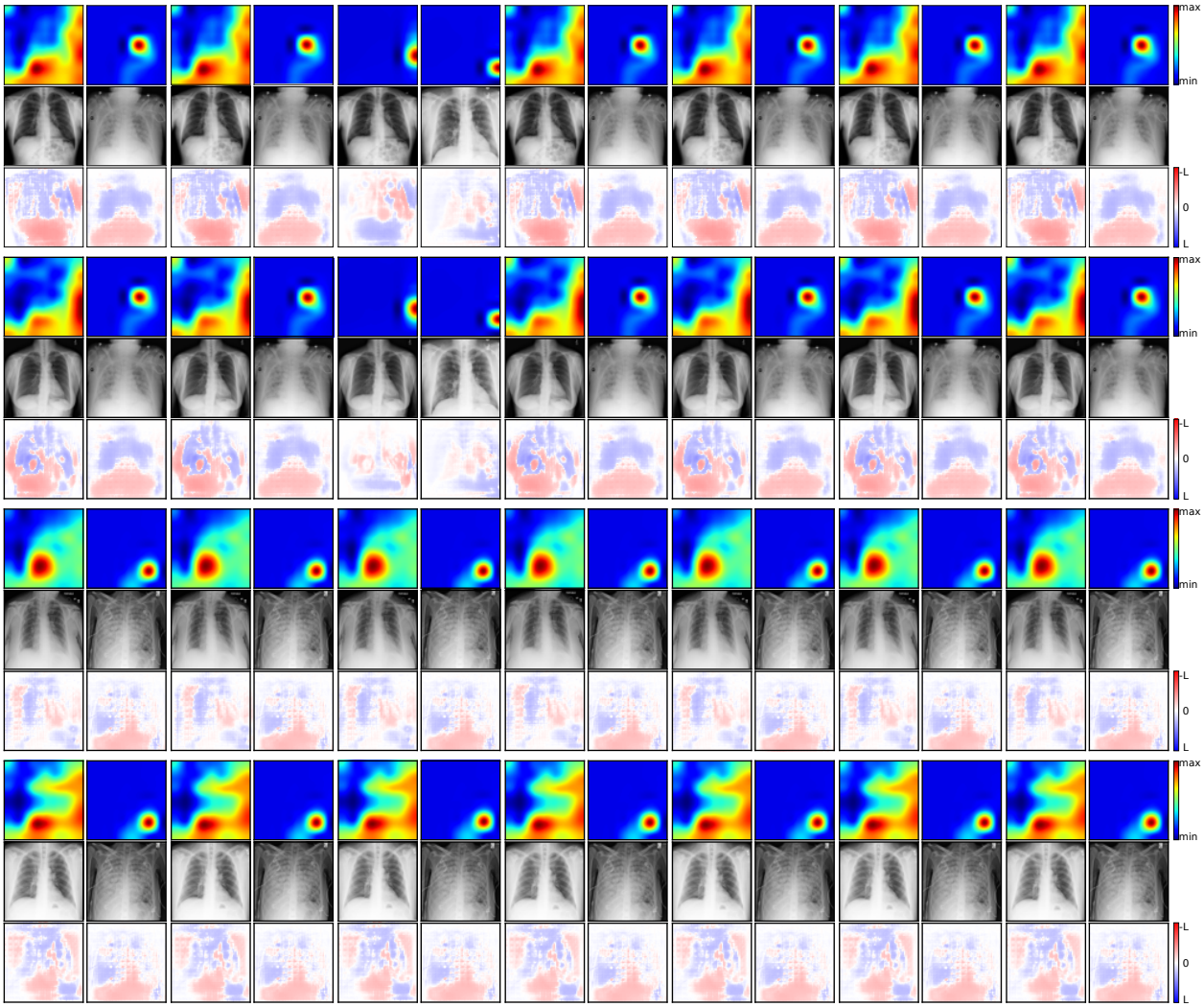

Figure A1: Explanations of the forward pass of a sample with **ResNet** trained for pneumonia detection on **RSNA**. We explain the layout of each explanation in the top left with formal notation (left to right, top to bottom): (1) explanation of ProtoPNet for the occurrence of a prototype within the test image; (2) explanation of ProtoPNet for the activation of an image, from which a prototype was extracted; (3) test image; (4) training image, from which the prototype was extracted; (5) explanation of ProtoPFaith for the occurrence of a prototype within the test image; (6) explanation of ProtoPFaith about the image, from which the prototype was extracted.

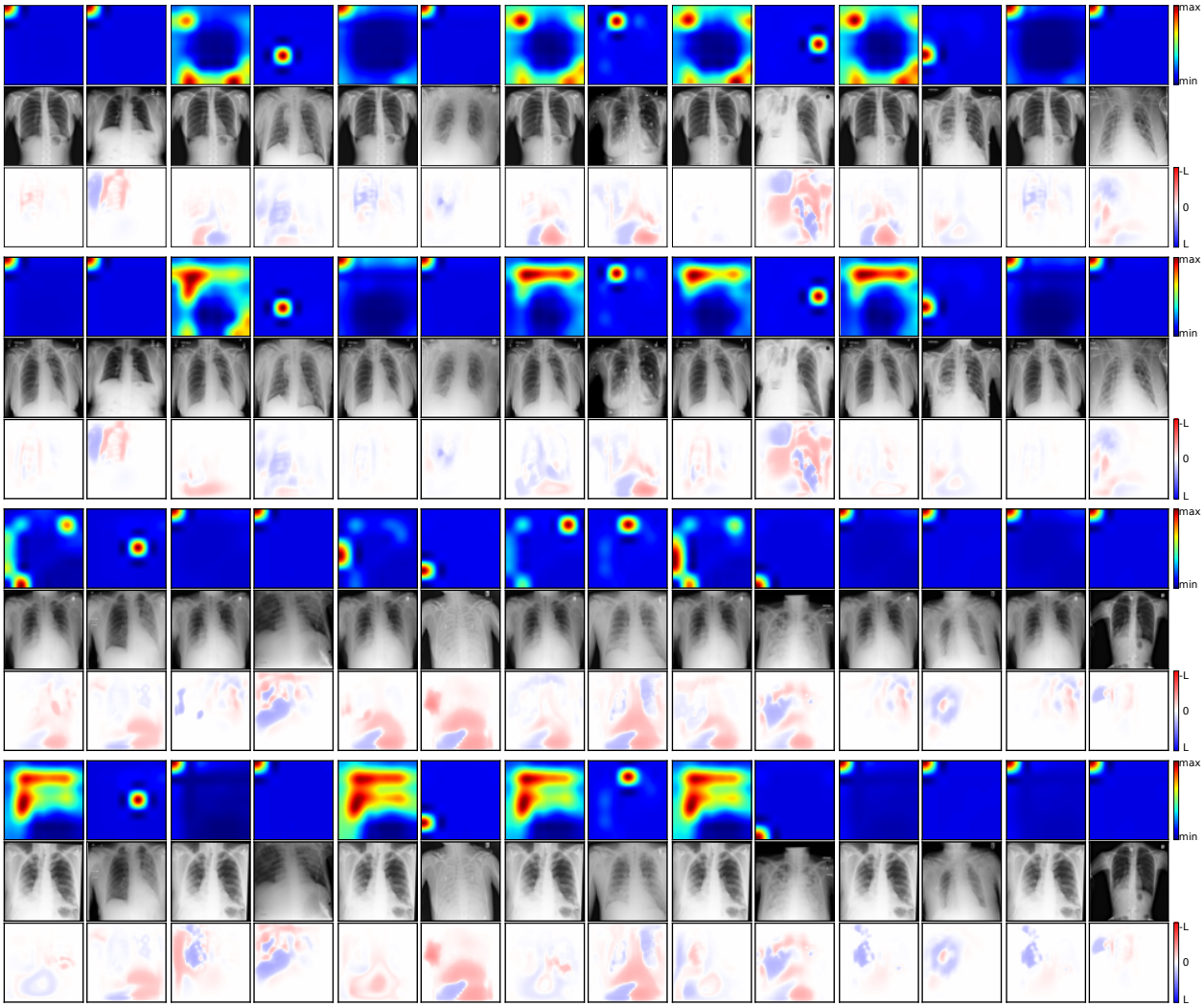

Figure A2: Explanations of the forward pass of a sample with **ConvNet** trained for pneumonia detection on **RSNA**. We explain the layout of each explanation in the top left with formal notation (left to right, top to bottom): (1) explanation of ProtoPNet for the occurrence of a prototype within the test image; (2) explanation of ProtoPNet for the activation of an image, from which a prototype was extracted; (3) test image; (4) training image, from which the prototype was extracted; (5) explanation of ProtoPFaith for the occurrence of a prototype within the test image; (6) explanation of ProtoPFaith about the image, from which the prototype was extracted.

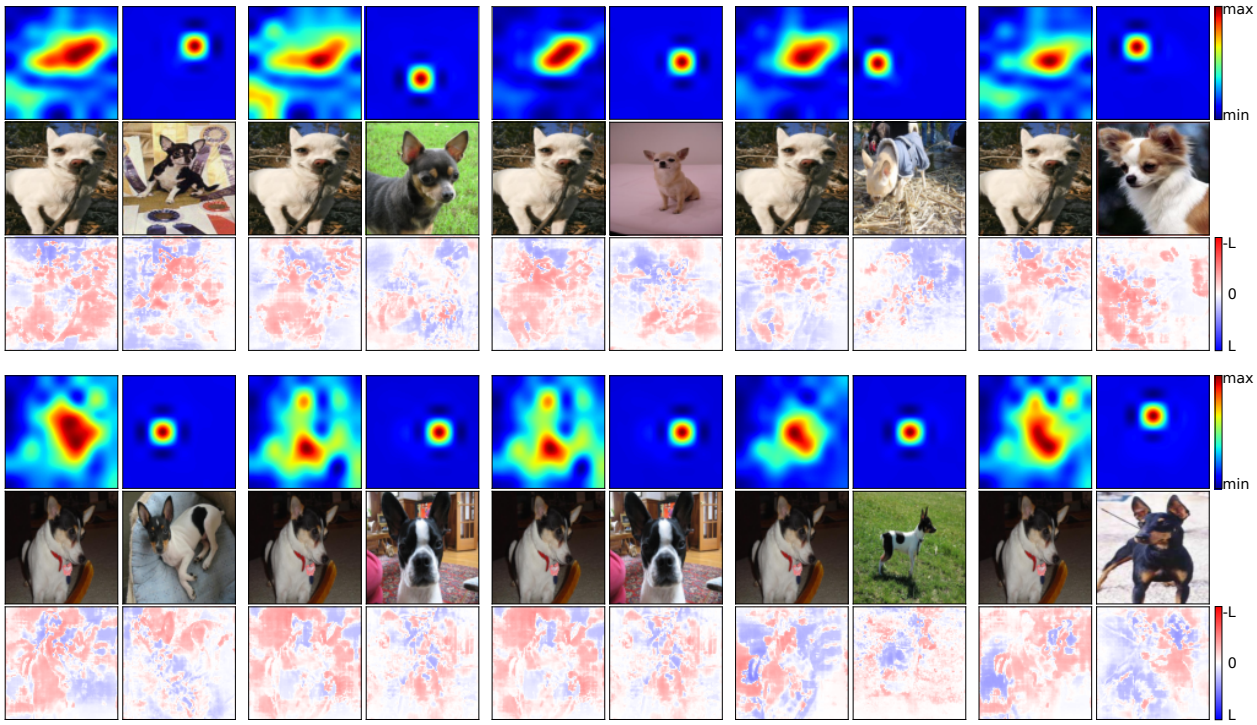

Figure A3: Explanations of the forward pass of a sample with **Wide-ResNet50** trained for **dog classification**. We explain the layout of each explanation in the top left with formal notation (left to right, top to bottom): (1) explanation of ProtoPNet for the occurrence of a prototype within the test image; (2) explanation of ProtoPNet for the activation of an image, from which a prototype was extracted; (3) test image; (4) training image, from which the prototype was extracted; (5) explanation of ProtoPFaith for the occurrence of a prototype within the test image; (6) explanation of ProtoPFaith about the image, from which the prototype was extracted.

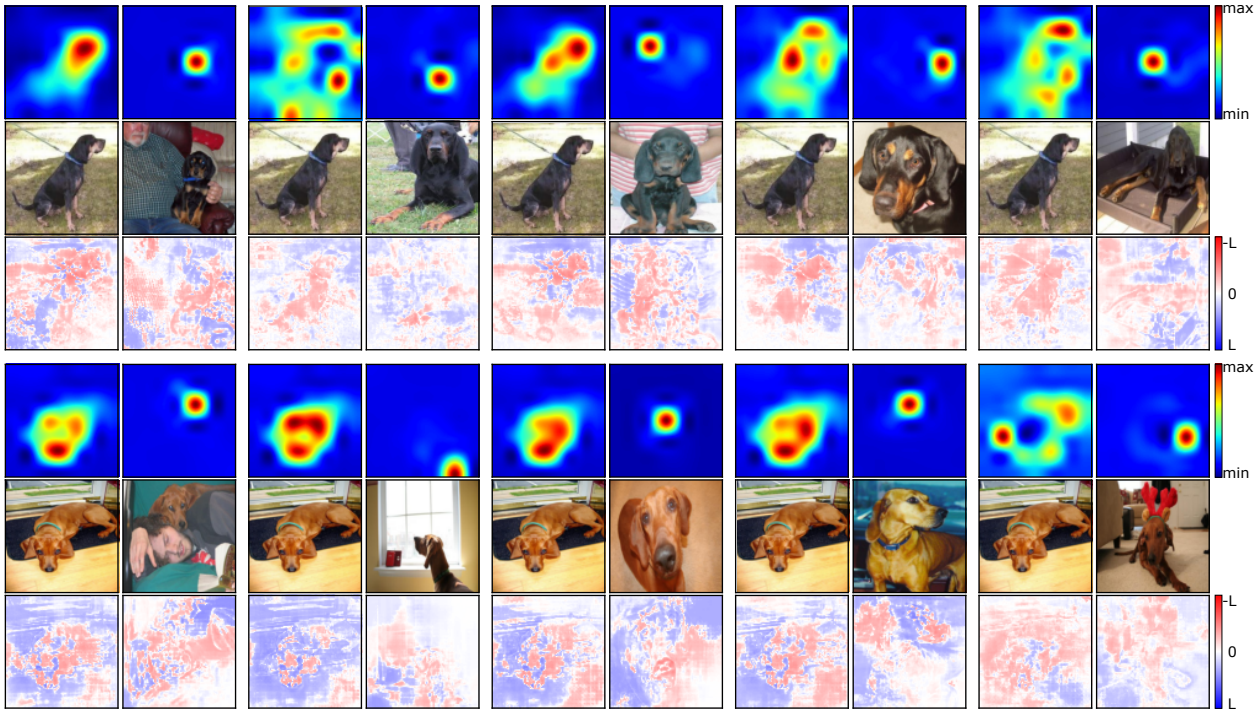

Figure A4: Explanations of the forward pass of a sample with **ResNeXt50** trained for **dog classification**. We explain the layout of each explanation in the top left with formal notation (left to right, top to bottom): (1) explanation of ProtoPNet for the occurrence of a prototype within the test image; (2) explanation of ProtoPNet for the activation of an image, from which a prototype was extracted; (3) test image; (4) training image, from which the prototype was extracted; (5) explanation of ProtoPFaith for the occurrence of a prototype within the test image; (6) explanation of ProtoPFaith about the image, from which the prototype was extracted.

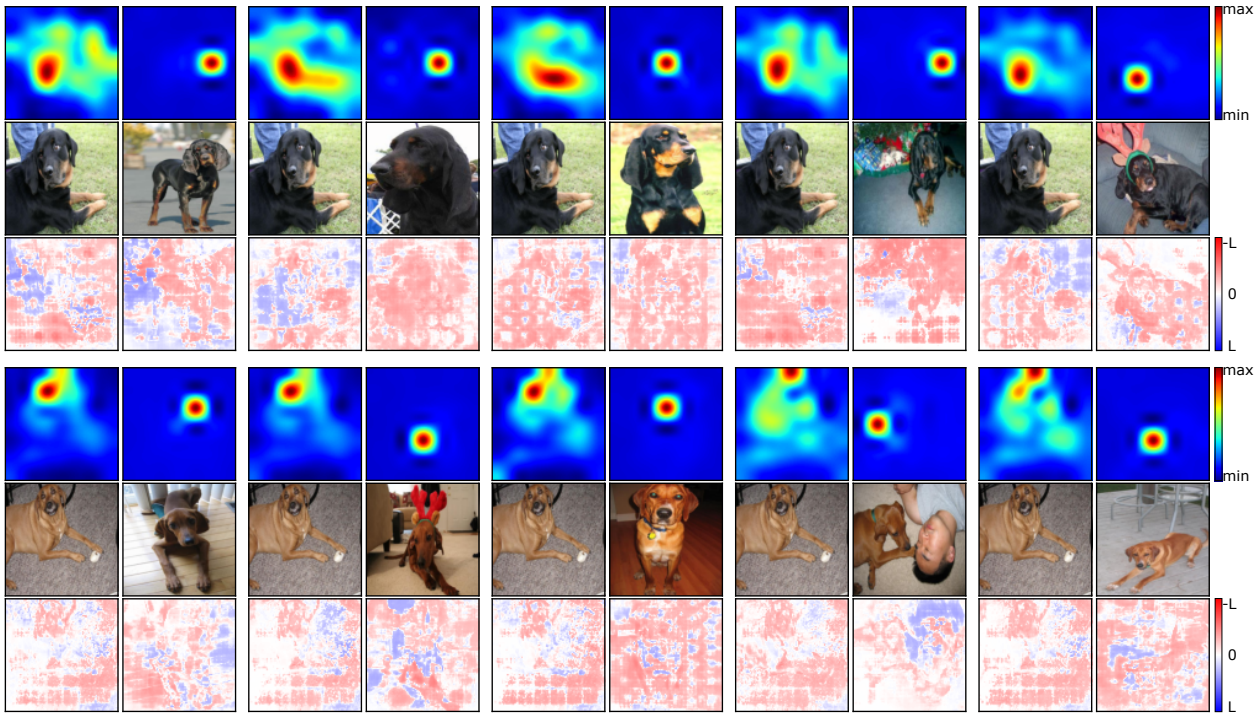

Figure A5: Explanations of the forward pass of a sample with **ResNet50** trained for **dog classification**. We explain the layout of each explanation in the top left with formal notation (left to right, top to bottom): (1) explanation of ProtoPNet for the occurrence of a prototype within the test image; (2) explanation of ProtoPNet for the activation of an image, from which a prototype was extracted; (3) test image; (4) training image, from which the prototype was extracted; (5) explanation of ProtoPFaith for the occurrence of a prototype within the test image; (6) explanation of ProtoPFaith about the image, from which the prototype was extracted.

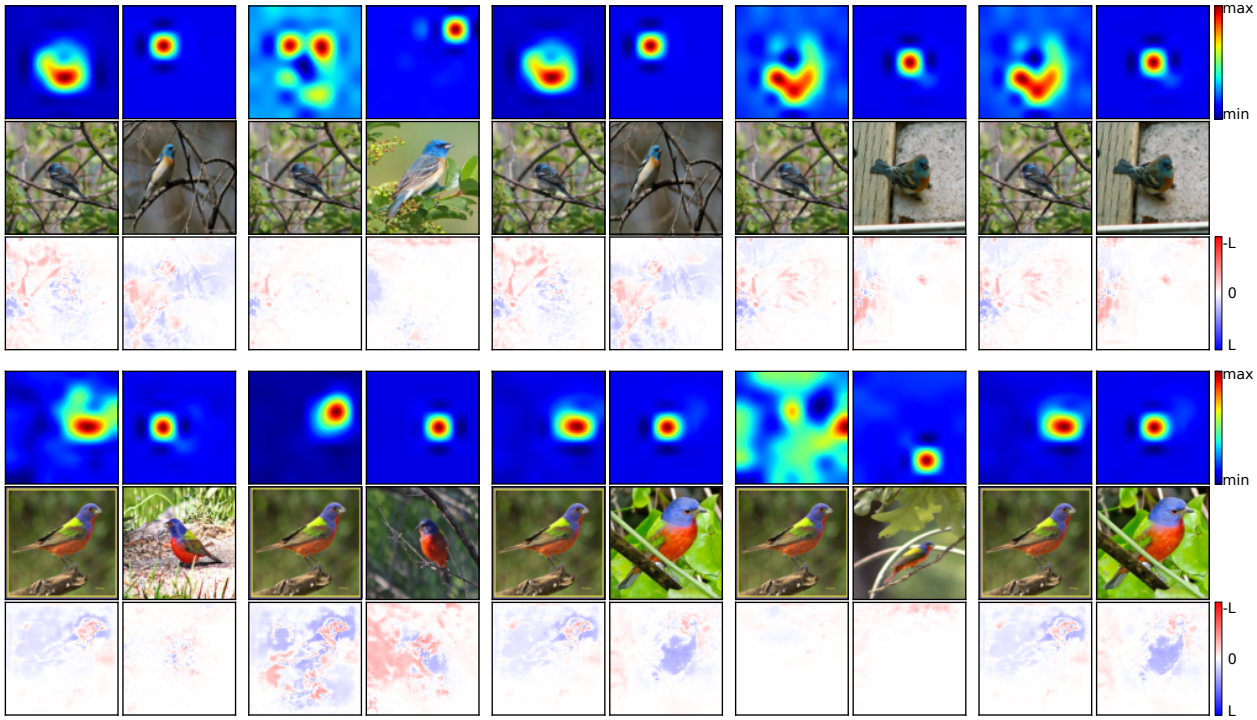

Figure A6: Explanations of the forward pass of a sample with **Wide-ResNet50** trained for **bird classification**. We explain the layout of each explanation in the top left with formal notation (left to right, top to bottom): (1) explanation of ProtoPNet for the occurrence of a prototype within the test image; (2) explanation of ProtoPNet for the activation of an image, from which a prototype was extracted; (3) test image; (4) training image, from which the prototype was extracted; (5) explanation of ProtoPFaith for the occurrence of a prototype within the test image; (6) explanation of ProtoPFaith about the image, from which the prototype was extracted.

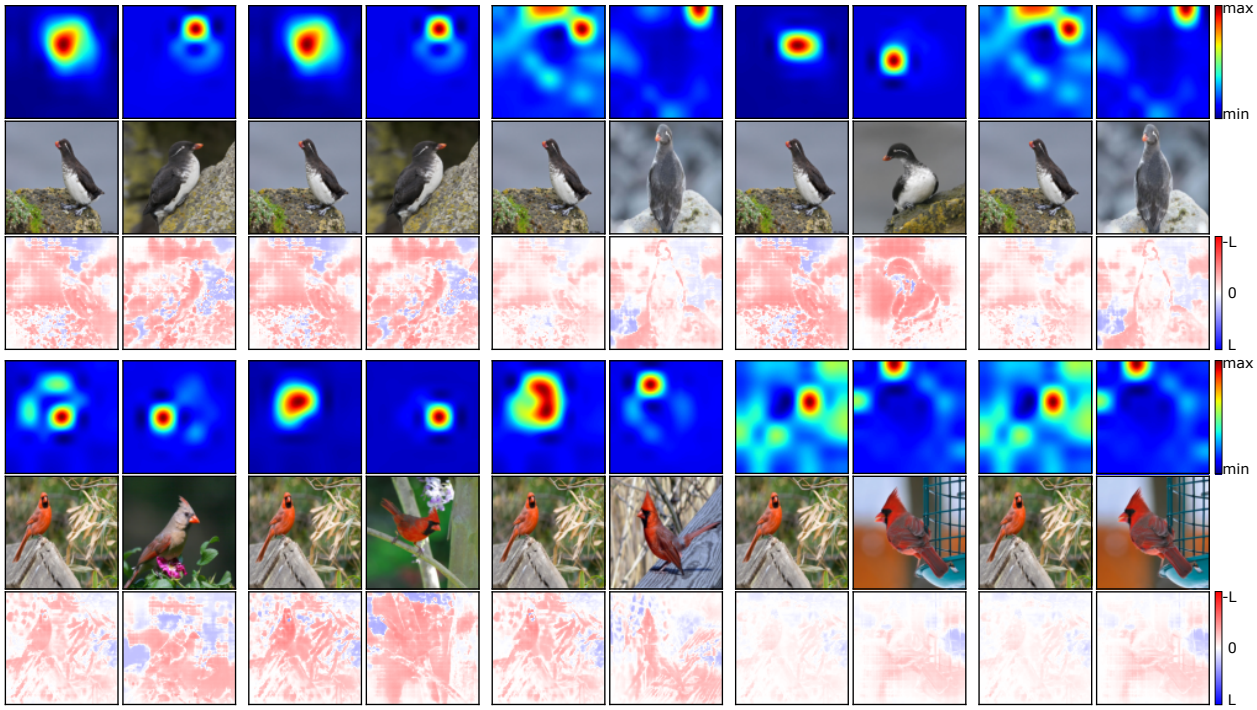

Figure A7: Explanations of the forward pass of a sample with **ResNeXt50** trained for **bird classification**. We explain the layout of each explanation in the top left with formal notation (left to right, top to bottom): (1) explanation of ProtoPNet for the occurrence of a prototype within the test image; (2) explanation of ProtoPNet for the activation of an image, from which a prototype was extracted; (3) test image; (4) training image, from which the prototype was extracted; (5) explanation of ProtoPFaith for the occurrence of a prototype within the test image; (6) explanation of ProtoPFaith about the image, from which the prototype was extracted.

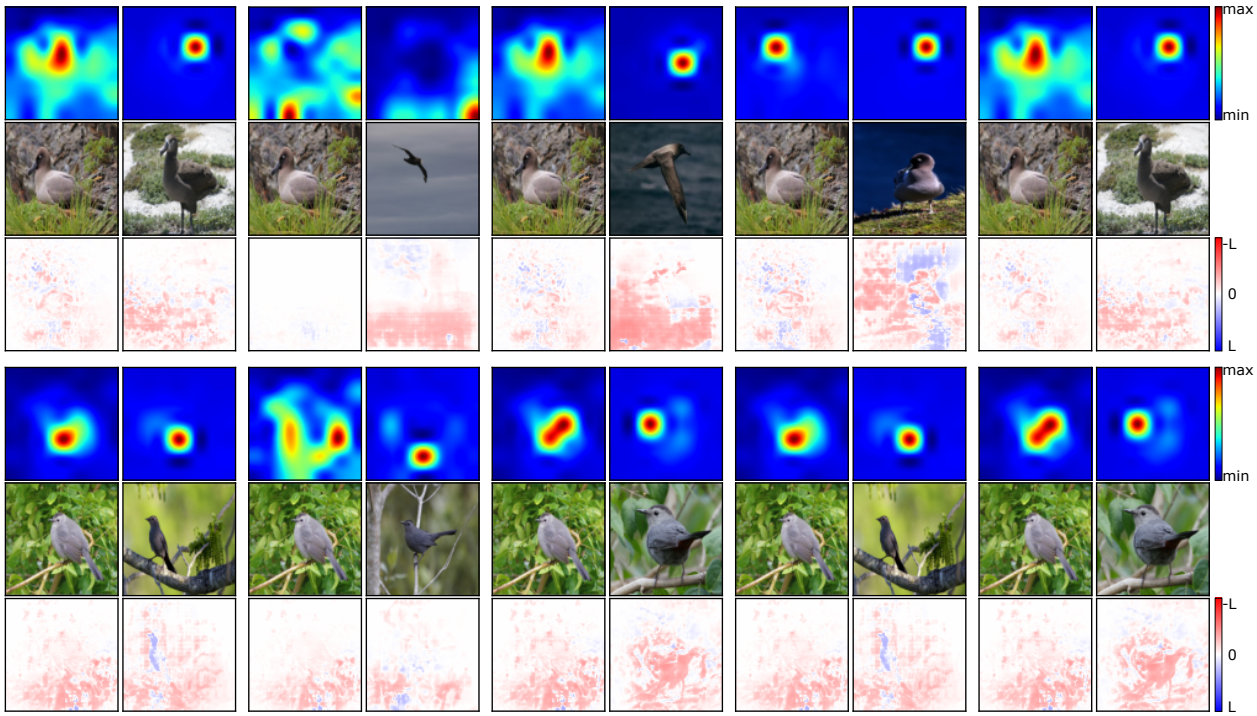

Figure A8: Explanations of the forward pass of a sample with **ResNet50** trained for **bird classification**. We explain the layout of each explanation in the top left with formal notation (left to right, top to bottom): (1) explanation of ProtoPNet for the occurrence of a prototype within the test image; (2) explanation of ProtoPNet for the activation of an image, from which a prototype was extracted; (3) test image; (4) training image, from which the prototype was extracted; (5) explanation of ProtoPFaith for the occurrence of a prototype within the test image; (6) explanation of ProtoPFaith about the image, from which the prototype was extracted.

References

Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pages 272–281. PMLR, 2019.

Chaofan Chen, Oscar Li, Daniel Tao, et al. This Looks Like That: Deep Learning for Interpretable Image Recognition. In *NeurIPS*, volume 32, 2019.

Brendan J Frey and Geoffrey E Hinton. Variational learning in nonlinear gaussian belief networks. *Neural Computation*, 11(1):193–213, 1999.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

D. B. Owen. A table of normal integrals. *Communications in Statistics - Simulation and Computation*, 9(4): 389–419, 1980. doi: 10.1080/03610918008812164. URL `https://doi.org/10.1080/03610918008812164`.