Keep the Faith: Faithful Explanations in Convolutional Neural Networks for Case-Based Reasoning

Tom Nuno Wolf\textsuperscript{1, 2, 3}, Fabian Bongratz\textsuperscript{1, 2, 3}, Anne-Marie Rickmann\textsuperscript{1, 2}, Sebastian Pölstler\textsuperscript{2}, Christian Wachinger\textsuperscript{1, 2, 3}

\textsuperscript{1}Department of Radiology, Technical University Munich, Munich, Germany
\textsuperscript{2}Lab for Artificial Intelligence in Medical Imaging, Ludwig-Maximilians-University, Munich, Germany
\textsuperscript{3}Munich Center for Machine Learning (MCML)

tom_nuno.wolf@tum.de

Abstract

Explaining predictions of black-box neural networks is crucial when applied to decision-critical tasks. Thus, attribution maps are commonly used to identify important image regions, despite prior work showing that humans prefer explanations based on similar examples. To this end, ProtoPNet learns a set of class-representative feature vectors (prototypes) for case-based reasoning. During inference, similarities of latent features to prototypes are linearly classified to form predictions and attribution maps are provided to explain the similarity. In this work, we evaluate whether architectures for case-based reasoning fulfill established axioms required for faithful explanations using the example of ProtoPNet. We show that such architectures allow the extraction of faithful explanations. However, we prove that the attribution maps used to explain the similarities violate the axioms. We propose a new procedure to extract explanations for trained ProtoP-Nets, named ProtoPFaith. Conceptually, these explanations are Shapley values, calculated on the similarity scores of each prototype. They allow to faithfully answer whether prototypes are present in an unseen image and quantify each pixel’s contribution to that presence, thereby complying with all axioms. The theoretical violations of ProtoPNet manifest in our experiments on three datasets (CUB-200-2011, Stanford Dogs, RSNA) and five architectures (ConvNet, ResNet, ResNet50, WideResNet50, ResNeXt50). Our experiments show a qualitative difference between the explanations given by ProtoPNet and ProtoPFaith. Additionally, we quantify the explanations with the Area Over the Perturbation Curve, on which ProtoPFaith outperforms ProtoPNet on all experiments by a factor >10\textsuperscript{3}.

Introduction

With continued progress in deep learning, AI models become ubiquitous in daily life. For many tasks, they are able to outperform humans (Shin et al. 2023). At the same time, it is difficult for humans to understand the decision-making of such complex black-box models with millions of parameters (Adadi and Berrada 2018; Arrieta et al. 2020). This fundamental issue is particularly relevant in decision-critical areas, such as medicine, finance, or justice. To increase transparency, methods have been developed that try to locally approximate the decision-making of neural networks in a post-hoc fashion (Vale, El-Sharif, and Ali 2022). However, such explanations can vary dramatically between explanation techniques and models (Rudin 2019): For example, pixel-wise attribution maps aim at highlighting the contribution of each pixel to the output logit, which is realized in Simonyan, Vedaldi, and Zisserman (2013) via propagation of gradients from the output logit to the input pixels. Such approaches are directly influenced by the model parameters. Alternatively, classifying image latents with a k-nearest-neighbor (kNN) algorithm allows showing the k most similar images as explanations. Although the theoretical motivation of pixel-wise attributions is to mimic the visual cortex, prior studies (Jeyakumar et al. 2020; Kim et al. 2023; Nguyen, Kim, and Nguyen 2021) have concluded that example-based explanations are easiest to understand for humans. While case-based reasoning, as in kNN, answers what contributed to a prediction, it does not allow to explain how the model transformed the image to arrive at its prediction. In other words, the explanations give insight into the decision-making of the model but lack to give insight into the high-dimensional non-linear function that a neural network represents. To compare and analyze explanation techniques from a theoretical point of view, several axioms have been introduced previously (Lundberg and Lee 2017; Sundararajan, Taly, and Yan 2017). In this work, we refer to an explanation method that satisfies all axioms as faithful.

ProtoPNet (Chen et al. 2019) is a prominent implementation of case-based reasoning and has been widely adopted, for image classification (Chen et al. 2019; Donnelly, Barnett, and Chen 2022) and decision-critical tasks, like automated diagnosis from chest radiography (Kim et al. 2021) and dementia-diagnosis (Wolf, Pölstler, and Wachinger 2023). It linearly classifies similarities between trainable class-representative feature vectors (prototypes) and latent features of unseen images. Importantly, its inherently interpretable architecture allows the extraction of explanations on a pixel-level. The model’s decision-making is explained by visualizing image crops that each prototype represents next to pixel-level explanations of the unseen image, as seen in Fig. 1, leading to the type of explanations that humans prefer (Jeyakumar et al. 2020; Kim et al. 2023; Rudin 2019). Therefore, its pixel-level explanations can, supposedly, be incorporated to answer the question how it came to the con-
Figure 1: In ProtoPNet (Chen et al. 2019), an image $\mathcal{I}$ is classified by feeding a vector of distance scores $s_{p_k^c}$ through a linear layer (omitted for clarity). Each distance represents the minimum distance between a trainable prototype $p_k^c$ to each spatial latent feature vector in $z$. An explanation in ProtoPNet is the up-scaled distance map of a prototype $\|p_k^c - z_{i,j}\|^2_2$ overlayed onto the input image. In this work, we demonstrate that these explanations do not adhere to established explainability axioms and, thus, cannot be considered faithful representations of the underlying processes. We propose to resolve this shortcoming of ProtoPNet with faithful explanations based on decomposing distance scores in terms of Shapley values w.r.t. the input image.

Our key contributions are as follows:

- We empirically demonstrate that there are substantial differences, both qualitative and quantitative, between the explanations extracted with ProtoPNet and ProtoPFAith.
- Explanations of ProtoPFAith outperform ProtoPNet on the AOPC score by a factor $>10^3$ for all experiments.

Axiomatic Evaluation of Explanations for Case-Based Reasoning

First, we evaluate explanations found in the literature with respect to the axioms required for faithful explanations. Then, we investigate the architecture of case-based reasoning with the example of ProtoPNet and show why some axioms are violated.

Related Work: Pixel-Level Explanation Methods for CNNs. Typically, explanations ought to represent each input feature’s effect on the prediction. Lundberg and Lee (2017) and Sundararajan, Taly, and Yan (2017) independently proposed a set of useful axioms that a deep learning attribution method should fulfill, which can be summarized as:

- **Sensitivity**: If there is a change in the value of a feature and the prediction, the attribution of that feature should not be zero.
- **Implementation Invariance**: Attributions for two models whose predictions are identical for all inputs should be identical.
- **Completeness**: Attributions of two inputs should add up to the difference in the model output.
- **Dummy**: If the prediction of a model is independent of a feature, its attribution should always be zero.
- **Linearity**: If a model $f$ is a linear combination of other models ($f = af_1 + bf_2$), the attributions of $f$ should follow the same linear combination.
- **Symmetry-Preserving**: If the prediction of a model is identical when replacing two input features with one another, the attribution of both features should be equal.
Gradient-based methods are characterized by a backward pass of the neural network used to calculate the gradient of a prediction with respect to input features, which can be used to create feature attribution maps (Simonyan, Vedaldi, and Zisserman 2013). However, as demonstrated by Sundararajan, Taly, and Yan (2017), gradient-based feature attribution methods violate either the Sensitivity axiom due to vanishing gradients in the ReLU activation, concerning methods like Baehrens et al. (2010), Simonyan, Vedaldi, and Zisserman (2013), Springenberg et al. (2014), Zeiler and Fergus (2014), or Implementation Invariance if the gradient is computed in a discrete fashion, as in Binder et al. (2016) and Shrikumar et al. (2016). Moreover, Shah, Jain, and Netrapalli (2021) showed that gradient-based methods tend to suffer from feature leakage, i.e., the contribution of unrelated features to the prediction. Activation-based methods (Chattopadhyay et al. 2018; Desai and Ramaswamy 2020; Fu et al. 2020; Jiang et al. 2021; Selvaraju et al. 2017; Wang et al. 2020; Zhou et al. 2016), on the other hand, leverage neuron activations in (usually deep/final) convolutional layers to assess the importance of input regions for the network output. Even though being widely established in practice, these methods typically violate Sensitivity and/or Completeness (Fu et al. 2020), which renders their applicability to decision-critical tasks questionable.

Different from gradient- and activation-based methods, perturbation-based methods mask or alter an input image feature and calculate the difference to the output of the original input. Existing work proposed to occlude (Zeiler and Fergus 2014), marginalize with a sliding window (Zinggraf et al. 2017), randomly perturb (Petsiuk, Das, and Saenko 2018), or occlude parts of an input image with perturbation space-exploration (Fel et al. 2022). In Dabkowski and Gal (2017), Fong and Vedaldi (2017), and Ribeiro, Singh, and Guestrin (2016), the black-box predictor is approximated by an interpretable model locally and super-pixel explanations summarized to the global input.

Another perturbation-based approach is based on Shapley values (Shapley 1953), which originate from cooperative game theory; each feature is treated as a player in a game and contributes to the final prediction. Removing a player \( i \) from all possible coalitions \( S \subseteq P \) of a set of players \( P \), i.e., marginalizing a player, yields its contribution \( \psi_i \) to the set function \( \hat{f} : P \to \mathbb{R} \):

\[
\psi_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} \left( \hat{f}(S \cup \{i\}) - \hat{f}(S) \right).
\]

In contrast to other attribution methods, Shapley values satisfy all axioms and, thus, provide faithful explanations (Covert, Lundberg, and Lee 2020; Lundberg and Lee 2017; Sundararajan, Taly, and Yan 2017; Zheng et al. 2022). Specifically, they satisfy Completeness, as the sum of contributions of all players equals the prediction:

\[
\sum_i \psi_i = \hat{f}(P) - \hat{f}(\emptyset).
\]

When aiming to compute Shapley values for an input image \( I \) of a DNN, \( \hat{f}(S) \) signifies the output of the DNN when all pixels not in \( S \) are replaced by a baseline value. Approximating Shapley values has gained a lot of attention (Ancona, Oztiireli, and Gross 2019; Lundberg and Lee 2017; Shrikumar, Greenside, and Kundaje 2017; Strumbelj and Kononenko 2014; Sundararajan, Taly, and Yan 2017; Wang et al. 2022), as their exact calculation grows exponentially with the number of input features, with DASP (Ancona, Oztiireli, and Gross 2019) outperforming all other approximation methods in terms of approximation error.

Preliminary: Interpretability with ProtoPNet-like Architectures. ProtoPNet, outlined in Fig. 1, implements case-based reasoning as a function \( f(I) = (F \circ Q \circ Z \circ V)(I) \) mapping an input image \( I \in \mathbb{R}^{H \times W \times C} \) to a set of output logs. More precisely, a CNN encoder \( V : \mathbb{R}^{H \times W \times C} \to \mathbb{R}^{H' \times W' \times C'} \), extract features and is typically pre-trained on the desired task \((H, W, C)\) the spatial height, width, and channel dimension of the input image; \((H', W', C')\) the target feature map height, width, and channel dimensions. The feature extractor \( Z : \mathbb{R}^{H' \times W' \times C'} \to \mathbb{R}^{H'' \times W'' \times L} \) maps to a latent feature map \( z \), which matches the channel-size \( L \) of prototypes, and consists of two \( 1 \times 1 \) convolutions and nonlinearities. The prototype module \( Q : \mathbb{R}^{H'' \times W'' \times L} \to \mathbb{R}^{K \times C} \) extracts a distance vector \( s \) for \( K \) prototypes per class \( C \). It consists of the minimum distances \( s_{p_k} \) of the squared L2-norm of each prototype to all spatial latent feature vectors \( z_{i,j} \) in \( z \):

\[
s_{p_k}(I) = \min_{i=1,...,H',j=1,...,W'} \|p_k^i - z_{i,j}\|^2_2
\]

\[
= \min_{i=1,...,H',j=1,...,W'} \sum_{l=1}^{L} (p_k^i,l - z_{i,j,l})^2,
\]

with \( p_k^i \in \mathbb{R}^{1 \times 1 \times L} \) and \( z = (Z \circ V)(I) \in \mathbb{R}^{H'' \times W'' \times L} \). Lastly, the classification layer \( F : \mathbb{R}^{K \times C} \to \mathbb{R} \) is implemented with a single linear layer.

Importantly, prototypes \( p_k^i \) are trainable parameters (vectors) of the network. After a certain number of iterations, each prototype is replaced by the closest (squared L2-distance) latent feature vector \( z_{i,j} \) extracted from all samples of the training set that are of class \( c \). Thus, a prototype \( p_k^i \) represents exactly one class-representative latent feature vector \( z_{i,j} \) from a training image. An unseen image is classified by feeding the distance vector \( s \) (see Fig. 1) to the classification layer \( F \).

The distance map \( \|p_k^i - z_{i,j}\|^2_2 \) is utilized to extract pixel-level explanations. First, the maximum distance of the distance map is selected, which is globally defined for all possible inputs if the last layer of the feature extractor \( Z \) is a bounded non-linearity. The maximum distance is subtracted by the values of the distance map\(^\dagger\). Then, the flipped distance map is up-scaled to the input image size and overlaid with the input image, forming an attribution map. A prototype is visualized by extracting the image crop around the 95%-percentile region of the corresponding attribution map.

\(^\dagger\)Originally, Chen et al. (2019) used a log-activation instead, which we discuss in detail in the next section.
The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

Theoretic Evaluation of the Faithfulness of ProtoPNet

Explanations. The decision-making of ProtoPNet is modeled as a linear function \( F \), which maps distances \( s_{p_k} \) to class logits. A linear mapping satisfies all introduced axioms (Štrumbelj and Kononenko 2014). Therefore, the model prediction, i.e., the decision-making of the model, is faithful with respect to the distance vector \( s \). However, ProtoPNet adds a log-activation to the prototype module, i.e., \( Q'(s_{p_k}(I)) = \log((s_{p_k}(I) + 1)/(s_{p_k}(I) + \epsilon_{\alpha})) \). This introduces non-linearity between the distance maps (which are used to yield attribution maps) and the classification. Therefore, the Linearity axiom is not preserved.

As explained in the previous section, attribution maps are the result of up-scaling distance maps from the latent space to the image space. This process implies a spatial relationship between distances in latent space and image space, as the distances of prototypes to latent feature vectors are calculated over the spatial dimensions in the latent space. In established CNN architectures used in ProtoPNet (VGG (Simonyan and Zisserman 2015), ResNet (He et al. 2016), DenseNet (Huang et al. 2017)), the locality of latent features is, however, lost after a few layers, i.e., the receptive field of each latent feature is the whole input image. We now prove by counterexample that there is no general spatial dependency between latent feature maps and the input image space:

**Proof.** Suppose a CNN consists of two convolutional layers (one input and output channel with linear activation) with kernel weights \( \theta_1 \in \{0, 1\}^{3 \times 3 \times 1} \) and \( \theta_2 \in \{0, 1\}^{3 \times 3 \times 1} \), as depicted in Fig. 2. Suppose the first layer has a stride and padding of one, and the second layer has a stride of two and padding of one. Both kernel weights are identical, with zeros everywhere beside the top left value, which is one. Feeding an exemplary input of size \( 3 \times 3 \), consisting of zeros with the exception of the top left feature set to one, to this network yields an output of size \( 2 \times 2 \), which consists of zeros and the bottom right feature activated at one. Upscaling this latent feature map to the input dimension, as done in ProtoPNet, creates an attribution map that suggests the bottom right part of the input to be relevant. However, the only feature that activated the bottom right latent feature is the top left feature in the input space. Thus, there is no spatial relation between the features of the input space and the latent space.

As a result, features that do not affect the latent features, and should therefore be treated as dummies, have a high attribution. Therefore, the Dummy axiom is not fulfilled.

Additionally, ProtoPNet explanations violate the axiom Completeness, as each distance \( s_{p_k} \) is exactly the distance of one latent feature vector of the latent feature map, i.e., \( \min_{c=0, \ldots, H', j=0, \ldots, W'} \| p_{k}^c - z_{c,j} \|_2^2 \). Therefore, the sum of pixel-level attributions does not equal the distance. For the same reason, the attribution maps violate the Dummy axiom. Lastly, thresholding the attribution map with the 95%-percentile violates Sensitivity: As the percentile is chosen heuristically, it suppresses the attributions of some input features that have non-zero attribution. In contrast, we will show that ProtoPFaith calculates attribution maps with respect to the minimum distance \( s_{p_k} \) faithfully, which manifests in our experiments.

In summary, the decision-making of ProtoPNet requires a linear activation to be faithful, while the explanations provided on an image-level break the overall faithfulness of the model, as they break Completeness, Dummy, and Sensitivity. Next, we introduce ProtoPFaith to extract faithful explanations from ProtoPNet, rendering it feasible for decision-critical tasks.

Methods

As described in the previous section, the explanations given by ProtoPNet are only faithful if Linearity of the classification w.r.t. minimum distances \( s_{p_k} \) can be restored, and attribution maps are faithful to changes in the minimum distances themselves. We showed that the attribution maps are not faithful. In contrast, Shapley values are guaranteed to satisfy all required axioms, albeit their exact calculation requires exponentially many model evaluations. To this end, DASP (Ancona, Oztireli, and Gross 2019) has shown to approximate true Shapley values with only few model evaluations, rendering it feasible for high-dimensional image inputs while outperforming competing methods in terms of approximation error. We show below how to convert ProtoPNet with this framework. Our proposed method to extract explanations, named ProtoPFaith, is an important step towards a transparent ProtoPNet, allowing to faithfully explain the case-based decision-making referenced by Chen et al. (2019) as "this looks like that". The source code is available at https://github.com/ai-med/KeepTheFaith.

Leveraging Faithful Explanations in ProtoPNet. We propose to restore Linearity of the classification of the distance vector \( s \) by dropping the log-activation introduced by ProtoPNet (note that the minimum of the squared L2-norm is zero). Instead of showing up-scaled log-activations of distance maps and the related weight of the classification layer, we need to calculate a contribution score \( \Psi_k \) of a similarity to the log-probability:

\[
\log(P(y = c | I)) = \log \left( \frac{\exp \sum_{k=1}^K -a_{c,k}s_{p_k}(I)}{\sum_{c=1}^C \exp \sum_{k=1}^K -a_{c,k}s_{p_k}(I)} \right) = \log \left( \frac{\exp \sum_{k=1}^K -a_{c,k}s_{p_k}(I)}{R} \right) = \sum_{k=1}^K -a_{c,k}s_{p_k}(I) - \log R
\]

Figure 2: An up-sampled latent feature map itself cannot serve as a spatial indicator of pixel attribution in the general case.
\[
\frac{1}{K} \sum_{k=1}^{K} \Psi_k, \text{ with } \Psi_k = -a_{c,k} s_{p^k}(I) - \frac{\log R}{K},
\]

where \(a_{c,k}\) are the coefficients of the linear layer \(F\) and \(y\) is the target label of the image \(I\). Note that we use the contribution score to compare the contribution of prototypes of the desired class, i.e., high contribution scores contribute more than low contribution scores.

In ProtoPNet, we are interested in the portion of an image that is representative of a prototype. Therefore, we propose to utilize the implementation for case-based reasoning introduced by ProtoPNet to extract Shapley values via the minimum distance \(s_{p^k}\). After network training, precisely one image exists in the training set from which a prototype arises. As a result, we can visualize a prototype faithfully by calculating the Shapley values w.r.t. the distance between this training image and the prototype (distance is 0 for this image). The resulting explanations highlight the pixels that are compressed into a prototype.

Likewise, we calculate the Shapley values of pixels w.r.t. the squared L2-distance of a prototype and visualize them for unseen test images. As the vector of minimum distances \(s\) is fed through the final linear layer only, we can use the Linearity axiom to transform similarity-based Shapley values into attribution w.r.t. the model prediction. This would be the weighted sum of all attribution maps. To allow quantitative comparison between pixel attributions of different models and prototypes, we opt for a bounded non-linearity before calculating the distances. Furthermore, a bounded non-linearity, such as ReLU1 (Eq. 4), helps for faster model convergence and allows us to derive the first and second-order moments analytically, which is required for DASP.

**Approximation of Shapley Values with DASP.** Recently, advances in uncertainty propagation for DNNs (Gast and Roth 2018) were applied to approximate Shapely values in DASP (Ancona, Oztiireli, and Gross 2019) with a few network evaluations. This is desirable when the number of input features is high, enabling efficient approximation of Shapley values of a whole input image.

DASP calculates the contribution \(\psi_{i,d}\) of the \(i\)-th feature to a coalition \(S_d\) of fixed size \(d\) by modeling each \(S_d\) as an aleatoric uncertainty, which is propagated through a probabilistic network to yield an expected value \(E_d[\psi_{i,d}]\). Then, the expectation of a Shapley value is estimated as:

\[
E[\psi_i] = \frac{1}{|p|} \sum_{d=0}^{|p|-1} E_d[\psi_{i,d}],
\]

(3)

If a DNN can be converted into a probabilistic model, each \(E_d[\psi_{i,d}]\) can be calculated with a single forward pass. To this end, closed-form solutions for mean and variance need to be derived. While they were summarized for standard DNN layers in Ancona, Oztiireli, and Gross (2019), they are not yet readily available for the prototype module \(Q\) and the ReLU1 non-linearity. Hence, we derive them in the following. Notably, the converted probabilistic model does not need to be trained, as the trained weights of a ProtoPNet are re-used.

**Derivation of Lightweight Probabilistic Layers for ProtoPNet.** We analytically derive expectation \(E\) and variance \(V\) of a layer input mean \(\mu\) and variance \(\sigma^2\) for ReLU1 for \(E(\mu, \sigma) = \int g(x)\left(\frac{x}{\sigma}\right)^{1/2} \phi\left(\frac{x-\mu}{\sigma}\right) dx\) and variance \(V(\mu, \sigma) = \int (g(x) - E(\mu, \sigma))^2 \frac{1}{2\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} dx\), where \(\phi\) the standard Gaussian probability density function \(\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\), the corresponding cumulative distribution function \(F(x) = \int_{-\infty}^x \phi(t) dt\) (Frey and Hinton 1999), and \(g(x) = \text{ReLU1}(x)\) defined as:

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

(4)

The expectation and variance are (full derivation in Sec. A.1\(^2\), which can easily be extended to any bound other than 1):

\[
E(\mu, \sigma) := \bar{\mu} = \sigma \left( \phi \left(-\frac{\mu}{\sigma}\right) - \phi \left(\frac{1-\mu}{\sigma}\right) \right) + \mu \left( \Phi \left(\frac{1-\mu}{\sigma}\right) + 1 - \Phi \left(\frac{1-\mu}{\sigma}\right) \right)
\]

(5)

\[
V(\mu, \sigma) = (\mu^2 - 2\mu\bar{\mu} + \sigma^2 + 2\bar{\mu} - 1) \Phi \left(\frac{1-\mu}{\sigma}\right) - (\mu^2 - 2\mu\bar{\mu} + \sigma^2) \Phi \left(-\frac{\mu}{\sigma}\right) - (\mu\sigma - 2\bar{\mu}\sigma + \sigma) \phi \left(\frac{1-\mu}{\sigma}\right) + (\mu\sigma - 2\bar{\mu}\sigma) \phi \left(-\frac{\mu}{\sigma}\right) + \bar{\mu}^2 - 2\bar{\mu} + 1
\]

(6)

Further, we derive expectation and variance of \(s_{p^k}\) (see Eq. 2) as follows: As seen in Ancona, Oztiireli, and Gross (2019), we assume independence of the input Gaussian signals \(z_{i,j,l}\). Thus, we can represent each latent feature vector \(z_{i,j}\) as a multivariate Gaussian with diagonal covariance matrix \(\Sigma_{i,j} \in \mathbb{R}^{L \times L}\) consisting of the individual means \(\mu_{i,j}\) and variances \(\sigma_{i,j}^2\), i.e., \(z_{i,j} \sim \mathcal{N}(\mu_{i,j}, \Sigma_{i,j})\), \(\mu_{i,j} \in \mathbb{R}^L\). We can then introduce a new multivariate random variable \(Y_{i,j}\):

\[
Y_{i,j} = z_{i,j} - p^k \rightarrow Y_{i,j} \sim \mathcal{N}(\mu_{i,j} - p^k, \Sigma_{i,j})
\]

Now, we calculate the squared L2-norm in Eq. 2 as:

\[
Z_{i,j} = \sum_{d=1}^L Y_{i,j}^2 = Y_{i,j}^T Y_{i,j}, \text{ which allows calculating mean and variance via quadratic forms of random variables (Baldessari 1967)}:
\]

\[
E[Z_{i,j}] = \text{trace}(\Sigma_{i,j}) + Y_{i,j}^T Y_{i,j} = \mu_{i,j}
\]

(7)

\[
V[Z_{i,j}] = 2 \text{trace}(\Sigma_{i,j}) + 4\mu_{i,j}^T \Sigma_{i,j} \mu_{i,j} = \sigma_{i,j}
\]

(8)

\[
Z_{i,j} \sim \mathcal{N}(\mu_{i,j}, \sigma_{i,j}^2).
\]

(9)

Finally, we extract the minimum distance with max-pooling over all negated \(Z_{i,j}, i = 1, \ldots, H', j = 1, \ldots, W'\), for which mean and variance are given in Ancona, Oztiireli, and Gross (2019).

\[\text{available at https://arxiv.org/abs/2312.09783.}\]
Experiments and Results

We evaluate ProtoPFaith on datasets similar to the ones used by ProtoPNet and its adaptations (Chen et al. 2019; Donnelly, Barnett, and Chen 2022; Kim et al. 2021): CUB-200-2011 (Wah et al. 2011), Stanford Dogs (Aditya et al. 2011), and a subset of RSNA3 (Shih et al. 2019) consisting of pneumonia and healthy samples only. Tab. 1 reports the classification performance of individual experiments, Sec. A.2 lists training details, and Sec. A.3 and Sec. A.4 present additional results.

Qualitative Evaluation. Fig. 3 visualizes explanations of image predictions from the test sets. Each predicted image was classified correctly by the model. Therefore, we visualize the attributions of the prototypes of that class and the corresponding contribution scores $\Psi_k$, which allows us to evaluate which prototype contributed most to the prediction (greater $\Psi_k$).

The model learned duplicate prototypes when trained on CUB-200-2011 and Stanford Dogs. As seen in almost all explanations of prototypes, attribution maps of ProtoPNet appear focused on a small location of an image. Its prototype activations for the test image are less sparse, except for CUB-200-2011, in which some attributions comprise background, while the prototype appeared to capture the animal only. There are no duplicate prototypes for the pneumonia class on the RSNA dataset. However, almost all prototype explanations of ProtoPNet are located in the background of their corresponding image. In contrast, the prototypes found by ProtoPFaith comprise the body and head for most prototypes of birds. For dogs, the attributions of prototypes are not as apparent, but always focus on the head of the animal around the face. Additionally, the explanations given by ProtoPFaith contain large portions of the background. While the log-activations of ProtoPNet are not helpful for pneumonia on the RSNA dataset, explanations of ProtoPFaith capture parts of the lung, with healthy parts indicating a dissimilarity to the prototype. However, all prototypes in RSNA encode general anatomy like the heart and spine. Attributions of the test image are similar for most prototypes. Additional results are presented in Sec. A.3.

Quantitative Evaluation. We evaluate the explanations given by ProtoPNet and extracted with ProtoPFaith based on the squared L2-distance $s_{p_k^c}$. For each prototype of a model, we select the input image that this prototype originates from (note that per definition of the training procedure, the squared L2-distance $s_{p_k^c}$ between each prototype and its input image equals 0). We iteratively remove the most relevant features (according to the attribution map of this prototype), as proposed in Tomsett et al. (2020) as the Area Over the Perturbation Curve (AOPC):

$$\text{AOPC}(Q \circ Z \circ V) = \frac{1}{C + K + T - 1} \sum_{c=1}^{C} \sum_{k=1}^{K} \sum_{t=1}^{T} s_{p_k^c}(\xi(p_k^c)) - s_{p_k^t}(\xi(p_k^t)),$$

where $\xi(p_k^c)$ denotes a mapping from prototypes to the corresponding index of an image in the training set, and $s_{p_k^t}$ indicates the $t$ most important features removed. It is expected that removing the most important features first leads to a faster decrease in the AOPC (negative values) if explanations are more meaningful. Tab. 1 demonstrates that removing features according to ProtoPFaith yields a decrease in AOPC that differs in several orders of magnitude from removing features according to ProtoPNet. Thus, the explanations given by ProtoPFaith are more accurate and discriminative than the original explanations given by ProtoPNet.

Discussion

As seen in Fig. 3, the explanations of ProtoPNet and ProtoPFaith are very different. While the explanations of ProtoPNet appear sparse, the close approximation of Shapley values given by ProtoPFaith indicates that prototypes learned by the network indeed comprise most of the input domain. The introduction of granularity on a pixel-level prohibits extraction of crops of the input image that are believed to represent a prototype by ProtoPNet. While the proposed explanations are harder to interpret, they are faithful to how the model transformed the input into a prototype and what lead to the classification. For medical applications like pneumonia detection, granularity can be beneficial: When the actual disease is behind ribs, attributions of ProtoPFaith do not comprise the ribs, as seen in Fig. 3. The background explanations found by ProtoPNet would not allow a clinician to learn anything about the model’s decision-making. Enlightened by theoretical derivation and the real-world explanations of ProtoPFaith, we can infer that the features encoded in the prototype are indeed just mapped to an arbitrary position in latent space. During training of ResNet on RSNA (see Fig. A1), the prototypes seemed to collapse for each class. This model achieved a test BAcc of 79.8% nevertheless. This is possible because the loss introduced by Chen et al. (2019) does not enforce prototypes of a single class to be distant from one another. With a typical black-box, identifying such unexpected issues would be impossible. Finally, human-understandable explanations, as given by ProtoPNet, are typically designed to fulfill a desired property like sparsity or locality. However, opting for transparency as in ProtoPFaith involves a trade-off, which likely applies to any explainability method. Hence, future work needs to address the question of how to accomplish explanations that are both human-understandable and faithful.

Limitations ProtoPFaith is currently only applied to single-label image classification but can be extended to multi-label classification as done by XProtoNet (Kim et al. 2021), in which prototypes only contribute to the prediction of their own class. This allows the calculation of Shapley values w.r.t. the output logit, yielding attribution maps
Table 1: Balanced Accuracy (BAcc) and AOPC scores for models visualized in Fig. 3. See Tab. A1 and Tab. A2 for full results.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Validation BAcc</th>
<th>Test BAcc</th>
<th>AOPC ProtoPNet</th>
<th>AOPC ProtoPFaith</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUB-200-2011</td>
<td>ResNet50</td>
<td>72.8 ± 0.5</td>
<td>72.0 ± 0.6</td>
<td>-0.000446</td>
<td>-3.721184</td>
</tr>
<tr>
<td>Stanford Dogs</td>
<td>ResNet50</td>
<td>84.1 ± 0.3</td>
<td>83.4 ± 0.6</td>
<td>-0.001767</td>
<td>-1.850909</td>
</tr>
<tr>
<td>RSNA</td>
<td>ConvNet</td>
<td>74.0 ± 10.1</td>
<td>72.6 ± 10.8</td>
<td>-0.001360</td>
<td>-8.816450</td>
</tr>
</tbody>
</table>

Figure 3: Explanations of the forward pass of: (top) a bird with ResNet50 trained on CUB-200-2011; (middle) a dog with ResNet50 trained on Stanford Dogs; (bottom) 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. We grayed out duplicate prototypes, which are identical feature vectors. \( \Psi_k \) denotes the prototype contribution towards log-probability. See Sec. A.4 for more visual results.
in image space for each output logit. Using this attribution map, we would be able to infer the attribution map of each prototype via using the **Linearity** property reversely, which would speed up the computation of attribution maps (see Sec. A.2).

### Conclusion

We elucidated the conceptual and theoretical problems of explanations for case-based reasoning with ProtoPNet. While this work focuses on the improvement of the transparency of ProtoPNet, it can easily be applied to similar case-based reasoning architectures such as ProtoTrees (Nauta, Van Bree, and Seifert 2021) or XProtoNet (Kim et al. 2021). The results highlighted that explanations given by ProtoPNet are merely well-defined rather than giving insight into how the model transformed an image. We showed that the theoretical flaws can be overcome by estimating Shapley values of the similarity score with minor adaptations of the architecture. Quantitatively, the explanations of ProtoPFaith outperformed the explanations of ProtoPNet on the AOPC by a factor >10^3 in all experiments.

### Acknowledgements

This research was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation). The authors gratefully acknowledge the computational and data resources provided by the Leibniz Supercomputing Centre (www.lrz.de).

### References


