ProtoArgNet: Interpretable Image Classification with Super-Prototypes and Argumentation

Hamed Ayoobi [1], Nico Potyka[2], Francesca Toni[1]

[1]Department of Computing, Imperial College London, United Kingdom
[2]Cardiff University, United Kingdom
h.ayoobi@imperial.ac.uk, potykan@cardiff.ac.uk, f.toni@imperial.ac.uk

Abstract

We propose *ProtoArgNet*, a novel interpretable deep neural architecture for image classification in the spirit of prototypical-part-learning as found, e.g., in ProtoPNet. While earlier approaches associate every class with multiple prototypical-parts, ProtoArgNet uses *super-prototypes* that combine prototypical-parts into a unified class representation. This is done by combining local activations of prototypes in an MLP-like manner, enabling the localization of prototypes and learning (non-linear) spatial relationships among them. By leveraging a form of *argumentation*, ProtoArgNet is capable of providing both supporting (i.e. 'this looks like that') and attacking (i.e. 'this differs from that') explanations. We demonstrate on several datasets that ProtoArgNet outperforms state-of-the-art prototypical-part-learning approaches. Moreover, the argumentation component in ProtoArgNet is customisable to the user's cognitive requirements by a process of sparsification, which leads to more compact explanations compared to state-of-the-art approaches.

1 Introduction

Deep neural models are successful in many tasks (LeCun, Bengio, and Hinton 2015), including image classification (our focus). However, they tend to be mostly inscrutable black-boxes. In high-stakes settings, interpretability is crucial and interpretable models are advocated, especially if they achieve comparable performance (Rudin 2019).

Prototypical-part-learning for image classification amounts to learning prototypical-parts of classes in images by introducing a *prototype layer* between a *convolutional backbone* and a *classifier* (Chen et al. 2019). Prototypical-parts are latent representations of patches in images, like the beak or tail of a bird (see Figure 1 (a)). The prototype layer determines the similarity between prototypical-parts and patches in the latent space that the convolutional backbone maps to. While some prototypical-parts may be meaningless for humans, the same can be said about some of the latent features learnt by black-box models (Jo and Bengio 2017). The transparency of protypical-part approaches allows detecting if a decision has been made based on meaningful patterns or statistical artefacts.

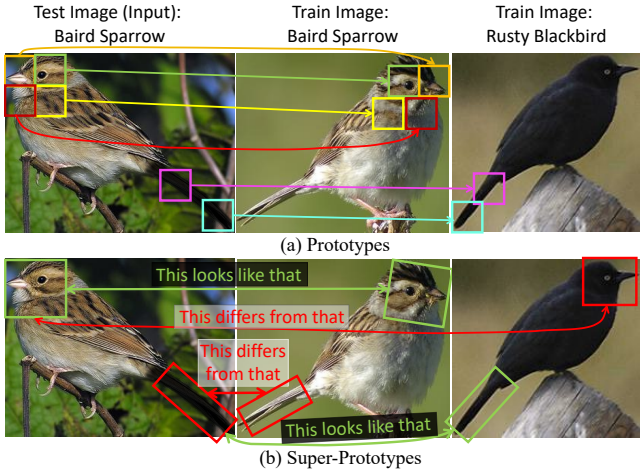

Figure 1: Conventional prototypes (a) versus the proposed super-prototypes (b) for a test image in the CUB dataset (Wah et al. 2011) with the tail intentionally coloured black. Class-specific super-prototypes encode spatial correlation between prototypical-parts by combining the low-level prototypes. They provide both 'this looks like that' and 'this differs from that' explanations.

We propose *ProtoArgNet* (Section 4, overviewed in Figure 2), a novel interpretable deep neural architecture for image classification in the spirit of prototypical-part-learning. Similar to ProtoPShare (Rymarczyk et al. 2021) and ProtoTrees (Nauta, van Bree, and Seifert 2021), ProtoArgNet shares prototypes among classes. However, while existing prototypical-part-learning approaches associate every class with multiple prototypical parts, ProtoArgNet summarizes them in a single *super-prototype* per class. Intuitively, the super-prototype combines local activations of prototypes to encode spatial relationships amongst them (see Figure 1 (b) for an illustration). As we will demonstrate in the experiments with the SHAPES dataset (Hu et al. 2017), these relationships are essential for some classification tasks but state-of-the-art prototypical-part-learning approaches are unable to capture them. This localization of prototypical-parts can be particularly useful in medical diagnosis (Kim et al. 2021) where the model can predict the location of disease indica-

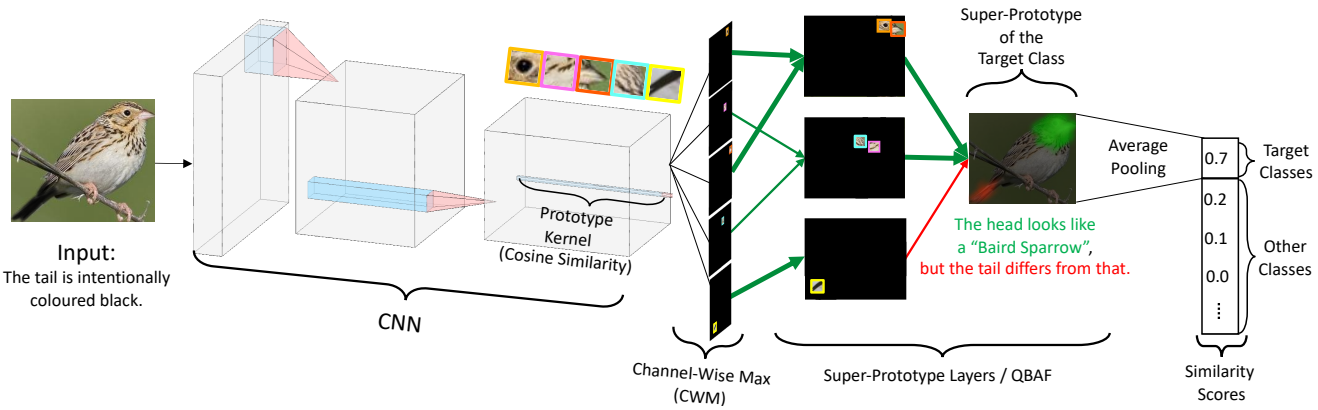

Figure 2: Architecture of ProtoArgNet (details in Section 4), illustrated with a sample from the CUB dataset with the tail intentionally coloured black.

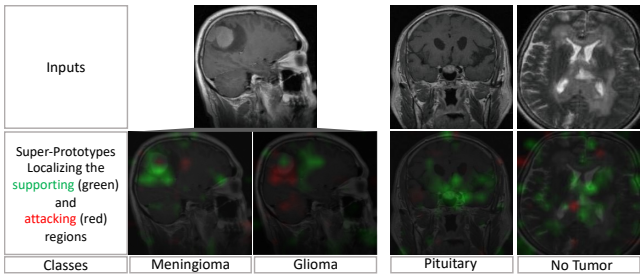

Figure 3: Sample inputs from the Brain Tumor MRI dataset (Nickparvar 2021) (top row), and corresponding super-prototypes by ProtoArgNet (bottom row), localizing the regions supporting (green overlay) and attacking (red overlay) the corresponding classes (details in Section 6).

tors without requiring masks for the training data. Figure 3 shows an application of ProtoArgNet to an MRI scan for brain tumor diagnosis (Nickparvar 2021).

The super-prototype layers in ProtoArgNet can capture non-linear relationships, similar to a Multi-Layer Perceptron (MLP). However, instead of operating on individual neurons, ProtoArgNet operates on activation maps (Section 4). Since MLPs can, in particular, learn logical functions like disjunction and XOR, ProtoArgNet can also learn classes that cannot be captured by atomic spatial patterns (Section 6). To address the lack of interpretability of large MLPs, ProtoArgNet applies the SpArX methodology of (Ayoobi, Potyka, and Toni 2023) to translate MLPs to *quantitative bipolar argumentative frameworks* (QBAFs) (Potyka 2021), a well-known form of *argumentation* (Atkinson et al. 2017). The 'Arg' in ProtoArgNet refers to the use of QBAFs. ProtoArgNet is customisable to user cognitive needs by sparsifying the MLP/QBAF component. The sparse QBAF explains the mechanics of the underlying MLP in terms of the roles played by prototypes towards super-prototypes through the hidden (clusters of) activation maps. In the QBAFs, the 'arguments' (amounting, in ProtoArgNet, to channel-wise maxes, clusters of hidden activation maps in the MLP, and super-prototypes) can 'attack' or 'support' other 'arguments' (shown with red and green arrows in Figure 2), with a dialectical strength in line with activations in the MLP.

In summary, we make the following main contributions:

- We propose *super-prototypes*, which are class-specific combinations of prototypical-parts that allow capturing spatial relationships between them.
- We present *ProtoArgNet*, a novel prototypical-part-learning approach integrating super-prototypes and QBAFs for improved performance and interpretability.
- We show experimentally that ProtoArgNet outperforms the state-of-the-art prototypical-part-learning models ProtoPNet (Chen et al. 2019), ProtoTree (Nauta, van Bree, and Seifert 2021), ProtoPShare (Rymarczyk et al. 2021), ProtoPool (Rymarczyk et al. 2022), TesNet (Wang et al. 2021), Deformable ProtoPNet (Donnelly, Barnett, and Chen 2022), ST-ProtoPNet (Wang et al. 2023) and PIP-Net (Nauta et al. 2023) in terms of classification accuracy, explanation complexity, and the ability to encode and detect (non-linear) spatial relationships in images[1].

2 Related Work

Explaining the outputs of image classifiers is well-studied in the literature. Post-hoc explanation approaches like feature attribution methods (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017; Dejl et al. 2023), attention maps (Sattarzadeh et al. 2021) or counterfactual explanations (Goyal et al. 2019)) aim at explaining black-box models. We focus instead on developing an *interpretable* model based on *prototypical-part-learning* (Snell, Swersky, and Zemel 2017) and *argumentation* (Ayoobi, Potyka, and Toni 2023; Leofante et al. 2024).

Prototypical-parts have been introduced in *ProtoPNet* (Chen et al. 2019). This learns prototypical-parts as subpatches of the output of a convolutional backbone. A prototype layer associates each class with m prototypes and de-

[1]The code is available at https://github.com/H-Ayoobi/ProtoArgNet_AAAI. Additional information is available in the supplementary material at (Ayoobi, Potyka, and Toni 2024b).

termines the maximum similarity between patches in the input image and prototypes. The classification is then made by logistic regression based on the individual similarity values. ProtoPNet has been extended in different directions. For example, *ProtoPShare* (Rymarczyk et al. 2021) allows sharing prototypes among classes, while *ProtoTree* (Nauta, van Bree, and Seifert 2021) replaces the classification part with a soft decision tree. *PIP-Net* (Nauta et al. 2023) and *ProtoPool* (Rymarczyk et al. 2022) further refine the training procedure to improve the plausibility of prototypes. To disentangle the latent space, *TesNet* (Wang et al. 2021) learns prototypes on Grassman manifold. *ST-ProtoPNet* (Wang et al. 2023) introduces support prototypes that lie near the classification boundary to improve classification. *Deformable-ProtoPNet* (Donnelly, Barnett, and Chen 2022) proposes spatially flexible prototypes that can capture pose variations.

ProtoArgNet differs from these state-of-the-art approaches in that it uses super-prototypes and MLPs/QBAFs, based on a novel architecture. Unlike other approaches, ProtoArgNet integrates both positive (supporting) and negative (contrasting) reasoning through its underlying QBAF. This is achieved with a single super-prototype per class, while TesNet (Wang et al. 2021) requires hundreds of prototypes for similar reasoning.

ProtoArgNet uses a form of argumentation (Atkinson et al. 2017), to explain super-prototypes. Specifically, ProtoArgNet extends *SpArX* (Ayoobi, Potyka, and Toni 2023; Mihailescu et al. 2023), originally defined for MLPs with tabular data only, and ProtoSpArX (Ayoobi, Potyka, and Toni 2024a), using argumentation in the context of prototypical-part-learning with images. Several argumentation-based forms of explainability have been proposed in recent years (Cyras et al. 2021). Other works combine argumentation and image classification, e.g. (Albini et al. 2020; Sukpanichnant et al. 2021) for explaining the outputs of CNNs and (Ayoobi et al. 2021a,b, 2019, 2023; Ayoobi and Rezaeian 2020; Ayoobi et al. 2022) to obtain an interpretable image classifier, but without prototypical-parts.

3 Preliminaries

We build up on SpArX (Ayoobi, Potyka, and Toni 2023), a post-hoc explanation method that aims at generating structurally faithful explanations for MLPs. SpArX exploits that MLPs can be understood as a special case of Quantitative Bipolar Argumentation Frameworks (QBAFs) (Potyka 2021). QBAFs are graphical reasoning models, where nodes represent *abstract arguments* and edges represent *attack* or *support* relations between the arguments. Every argument in a QBAF is associated with an *initial strength* and reasoning algorithms determine a *final strength* (representing an acceptability degree) for every argument, based on its initial strength and the final strength of its attackers and supporters.

Arguments in QBAFs are abstract entities. What makes them arguments is that they are in dialectical relationships with each other. Roughly speaking, in order to transform an MLP into a QBAF, neurons can be associated with arguments, their biases can be transformed into initial strength values and their connection weights into intensity values of attackers and supporters. The translation guarantees that the activations of neurons in the original MLP correspond to the final strength values of arguments in the QBAF (Potyka 2021). While this correspondence allows representing MLPs faithfully by QBAFs, it does not add much interpretability because the QBAF has the same size as the MLP. Thus, SpArX sparsifies the network by clustering nodes with similar activations and representing each cluster by a single argument (Ayoobi, Potyka, and Toni 2023).

In this work, we extend SpArX to make ProtoArgNet interpretable and explainable. An illustration is given in Figure 2: activation maps in the super-prototype layers of ProtoArgNet are treated as arguments, alongside the Channel-Wise Maxes (CWMs) that localize the prototypes, which serve as the input features for the super-prototypelayers in our architecture, as presented next.

4 ProtoArgNet

Figure 2 shows the architecture of ProtoArgNet. ProtoArgNet consists of a convolutional backbone f with weights W^{conv}, a prototype layer $\mathcal{P}$, a Channel-Wise Max ($\mathcal{CWM}$) layer, and a Super-Prototype layer $\mathcal{SP}$ mapped onto a QBAF for interoperability and explainability purposes. We discuss each component in turn, assuming that inputs are images and the classification task amounts to predicting a class in the set K ($|K| \geq 2$).

4.1 Prototypes

Let $z = f(x)$ be the convolutional output for an input image x, where the output tensor z has shape $H \times W \times D$ with height H, width W and D channels. This output tensor serves as input to the prototype layer, $\mathcal{P}$, which represents prototypical-parts. $\mathcal{P}$ consists of N prototypes $P = \{p_i\}_{i=1}^{N}$ with shapes $H_1 \times W_1 \times D$. As usual, we use $H_1 = W_1 = 1$. For each prototype $p_i \in P$ and every $1 \times 1 \times D$ sub-tensor $z_{h,w,.}$ of z, the prototype layer $\mathcal{P}$ computes the cosine similarity $\mathcal{SM}^i_{h,w,.} = \frac{p_i \cdot z_{h,w,.}}{\|p_i\| \|z_{h,w,.}\|}$ and summarizes the similarity values in a matrix $\mathcal{SM}^i$ of dimension $H \times W$. Intuitively, a similarity map $\mathcal{SM}^i$ indicates how similar the prototypical-part p_i is to patches of the input image x in the latent space.

Compared to the commonly used approach of computing L2 distance and converting it to similarity (as in ProtoPNet, ProtoPShare, and ProtoTrees), cosine similarity is scale-invariant and thus more easily interpretable. We implemented $\mathcal{SM}$ using the 2D convolution operator $*$. It generates $\mathcal{SM}^i$ by convoluting the normalized convolutional output $\hat{z} = \frac{z}{\|z\|} = \left[\frac{z_j}{\|z_j\|}\right]_{z_j \in z}$ with a normalized prototype kernel $\hat{p_i} = \frac{p_i}{\|p_i\|}$, $\mathcal{SM}^i = \hat{z} * \hat{p_i}$. Since cosine similarity is used for the prototype layer, the values in similarity maps can be both positive and negative in the range $[-1, 1]$. The output dimensions of the prototype layer are $H \times W \times N$.

4.2 Channel-Wise Max

The Channel-Wise Max layer aims to localize and extract the maximum value from each similarity map, while ensuring that only one prototype is activated at each location across all similarity maps. $\mathcal{CWM}$ takes the similarity maps as input. For each similarity map, it determines the maximal

value and sets all non-maximal values to 0. Formally, for every similarity map $\mathcal{SM}^i$, the channel-wise max filter creates a new map $\mathcal{CWM}^i$ of the same dimension. To do so, it determines the maximal value among the entries $\mathcal{SM}^i_{h,w}$, retains the highest value $s^i_{max} = \max_{1\leq h\leq H} \max_{1\leq w\leq W} \mathcal{SM}^i_{h,w}$ within the map and assigns zero to the remaining elements:

$$\mathcal{CWM}^i_{h,w} = \begin{cases} \mathcal{SM}^i_{h,w} & \text{if } s^i_{max} = \mathcal{SM}^i_{h,w}; \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

It may happen that two distinct maps, $\mathcal{CWM}^i$ and $\mathcal{CWM}^j$, have a maximal activation at point (h, w), which would make it more challenging to interpret the subsequent layers. To avoid this, we consider only the maximally activated prototype at each position (h, w). To make this choice differentiable during training, we apply the softmax (Bridle 1990) to each position (h, w) ranging over $\mathcal{CWM}^1, \ldots, \mathcal{CWM}^N$.

$$\mathcal{CWM}^i_{h,w} = \frac{\mathrm{e}^{(\mathcal{CWM}^i_{h,w}/T)}}{\sum_{j=1}^{N} \mathrm{e}^{(\mathcal{CWM}^j_{h,w}/T)}} \quad (2)$$

During training, we gradually decrease the temperature parameter T from 1 to 0.

After training, we replace softmax with the maximum to ensure the activation of at most one prototype per location.

4.3 Super-Prototypes and Similarity Scores

The super-prototype module takes the $\mathcal{CWM}s$ as input and provides a single similarity score per class. To do so, it generalizes the mechanics of MLPs, but whereas MLPs operate on scalars, the super-prototype module operates on matrices. The input matrices are the maps $\mathcal{CWM}^1, \ldots, \mathcal{CWM}^N$, and in the first layer of the super-prototype module they are combined affinely to form new matrices of the same dimension. After applying an activation function, the matrices in this layer can then again be combined to form matrices in the next layer analogously to MLPs. To describe this formally, let A^l_i range over the matrices in layer l. We let $A^0_i = \mathcal{CWM}^i$, and, for $l > 0$,

$$A^l_i = \sigma\left(\left(\sum_{j=1}^{N_{l-1}} w^l_{ji} \cdot\ A^{l-1}_j\right) + b^l_i\right) \quad (3)$$

where $N_0 = N$ is the number of prototypes and, for $l > 0$, N_l is the size of layer l, b^l_i a bias matrix, and σ the activation function.

Like an MLP, the super-prototype layers can have various configurations regarding the number $\mathcal{L}$ of hidden layers, the number $\mathcal{H}$ of hidden activation maps at each layer, and the activation function σ used at each hidden layer, hence we refer to it as Super-prototype MLP (SMLP).

The output layer provides a single super-prototype per class $k \in K$. Each super-prototype $\mathcal{SP}_k$ is defined as:

$$\mathcal{SP}_k = \sum_{j=1}^{|K|} w^{\mathcal{SP}}_{jk} \cdot\ A^{\mathcal{L}}_j. \quad (4)$$

In the final step, a single similarity score ss_k is computed for each super-prototype by summing up the values in $\mathcal{SP}^k$:

$$ss^k = \sum_{1\leq h\leq H, 1\leq w\leq W} \mathcal{SP}^k_{h,w}. \quad (5)$$

Note that Equation 3 can be efficiently implemented by employing convolutions with kernel shapes $1 \times 1 \times N_{l-1}$, followed by an activation function. Similarly, Equation 4 can be implemented using convolutions of shape $1 \times 1 \times \mathcal{H}$.

4.4 Super-Prototypes Layers to QBAFs

Since the SMLP mimics what an MLP does, it can be converted to a QBAF, similar to the approach followed in SpArX (Ayoobi, Potyka, and Toni 2023), by first sparsifying the SMLP and then translating it to a QBAF (c.f., Section 3). SpArX sparsifies an MLP by merging similar neurons. Since we have activation maps instead of single neurons, we have to redefine the distance function in SpArX. Given an input x and two activation maps A^l_i and A^l_j with height H and width W, our distance function is defined as:

$$\delta(A^l_i, A^l_j) = \sum_{x'\in\Delta'} \pi_{x',x} \sqrt{\sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \left(A^l_i - A^l_j\right)^2_{h,w}}, \quad (6)$$

where Δ' denotes a sample neighborhood of x and $\pi_{x',x}$ is a similarity function that assigns a higher weight to neighbors x' closer to the input x (Ayoobi, Potyka, and Toni 2023).

The obtained QBAF can explain how prototypes reason for or against a particular class. To illustrate the idea, consider the (sparsified) 1-hidden layer-SMLP/QBAF in Figure 2. It can be interpreted as follows:

- The prototypes of the beak and eye support with high intensity the top-most hidden activation map, the prototypes corresponding to the neck and the upper wing support the middle hidden activation map, and the prototype of the tail supports the bottom hidden activation map.
- The top and middle hidden activation maps (arguments) strongly support the super-prototype of the target class "Baird Sparrow" which forms the head, while the bottom hidden activation map attacks it. This leads to a super-prototype with positive values for the head (green overlay) and negative values for the tail (red overlay).

Overall, this interpretation indicates that the predicted class for the input image is supported by the bird's head that looks like a "Baird Sparrow" and is attacked by the tail differing from that, while also pointing to the reasoning of the SMLP in terms of the prototypes used.

5 Training ProtoArgNet

ProtoArgNet is trained end-to-end and does not require a prototype pruning stage as some approaches do (e.g. ProtoPNet (Chen et al. 2019), ProtoPShare (Rymarczyk et al. 2021), ProtoTrees (Nauta, van Bree, and Seifert 2021), and PIP-Net (Nauta et al. 2023)). For the i^{th} data point in a dataset of size n, with the data point belonging to class label $y_i \in K$ (where K is the set of class labels), the target class super-prototype should obtain a high similarity score

ss^{y_i}. Moreover, the corresponding similarity scores for the super-prototypes of other classes should be low. Simultaneously, the output of the classifier should be 1 for the target class y_i and 0 for the other classes. These two objectives are aligned and can be implemented by a single loss function $L_{\mathcal{SP}}$. Additionally, we would like the prototypes learned by the model to be dissimilar to each other to encourage diversity by incorporating a dissimilarity loss $L_{\mathcal{DS}}$. The *total loss function* that we aim to minimize is:

$$Loss = L_{\mathcal{SP}} + \alpha L_{\mathcal{DS}} \tag{7}$$

where $L_{\mathcal{SP}}$ is the Cross-Entropy loss, α is a constant ($\alpha = 0.1$ is used in the experiments) and $L_{\mathcal{DS}}$ is defined as

$$L_{\mathcal{DS}} = \sum(|P.P^{\intercal} - I_N|) \tag{8}$$

where P is the matrix of all normalized prototypes, I_N is the identity function of size $N \times N$, and $|.|$ is the absolute value function. Note that by definition of $P.P^{\intercal}$, the entry at position (i, j) contains the dot product of the i-th and j-th normalized prototypes. All elements of the main diagonal are equal to 1. The non-diagonal elements correspond to the cosine similarities between pairs of prototypes and are 0 if and only if the prototypes are orthogonal. Hence, the loss term will be minimal if all prototypes are orthogonal, thus encouraging diversity among them.

We minimize our loss function using the AdamW optimizer (Loshchilov and Hutter 2019). The trainable parameters are the convolutional weights W^{conv}, prototypes $\mathcal{P}$, hidden layers weights W^l, and super-prototype weights $W^{\mathcal{SP}}$.

$$\min_{W^{conv},\mathcal{P},W^l,W^{\mathcal{SP}}} Loss(W^{conv}, \mathcal{P}, W^l, W^{\mathcal{SP}}). \tag{9}$$

After training, we perform a *projection step* analogous to ProtoPNet (Chen et al. 2019). That is, we replace each learnt prototype with the latent representation of the closest image patch from the training data. This allows associating each latent prototype with an image space representation (see the image patches in Figure 2 for an illustration).

6 Experiments

We compared ProtoArgNet to the state-of-the-art prototypical-part-learning models ProtoPNet (Chen et al. 2019), ProtoTrees (Nauta, van Bree, and Seifert 2021), ProtoPShare (Rymarczyk et al. 2021), ProtoPool (Rymarczyk et al. 2022), TesNet (Wang et al. 2021), Deformable ProtoPNet (D-ProtoPNet) (Donnelly, Barnett, and Chen 2022), ST-ProtoPNet (Wang et al. 2023) and PIP-Net (Nauta et al. 2023). Our experiments (set-up in Section 6.1) evaluate the classification performance (Section 6.2), the sparsification process (Section 6.3), the role of each layer on the model's performance by an ablation study (Section 6.4), the ability to encode and detect spatial relationships in the input (Section 6.5) and the complexity of explanations drawn from ProtoArgNet (Section 6.6). As usual, we use top-1 accuracy (the standard accuracy) as the performance measure. We also perform a qualitative evaluation (Section 6.7).

Method	Accuracy			
	CUB	Cars	Brain	SHAPES
ProtoPNet	79.2 ± 0.1	86.1 ± 0.1	97.4 ± 0.2	50.6 ± 0.7
ProtoPShare	74.7 ± 0.2	86.4 ± 0.2	97.7 ± 0.1	50.2 ± 0.8
ProtoPool	80.3 ± 0.2	88.9 ± 0.1	98.3 ± 0.2	49.7 ± 0.6
TesNet	83.0 ± 0.2	88.5 ± 0.2	98.2 ± 0.1	50.6 ± 0.3
D-ProtoPNet	83.4 ± 0.1	88.6 ± 0.2	98.4 ± 0.2	50.1 ± 0.5
ST-ProtoPNet	83.6 ± 0.2	88.7 ± 0.2	98.7 ± 0.3	50.2 ± 0.4
ProtoTrees	82.2 ± 0.7	86.6 ± 0.2	98.0 ± 0.3	50.1 ± 0.7
PIP-Net	82.0 ± 0.3	86.5 ± 0.3	97.5 ± 0.3	50.3 ± 0.6
ProtoArgNet	$\mathbf{85.4 \pm 0.2}$	$\mathbf{89.3 \pm 0.3}$	$\mathbf{99.5 \pm 0.3}$	$\mathbf{99.8 \pm 0.1}$

Table 1: Accuracy of ProtoArgNet and other methods on the CUB, Cars, Brain, and SHAPES datasets. SHAPES is used for the evaluation of spatial correlation between prototypical-parts. (Best accuracy in **bold**)

For all experiments, we have used CUB (Wah et al. 2011) and Cars (Krause et al. 2013), which are the standard benchmarks for prototypical-part-learning models. To emphasize the importance of localizing specific regions in images that either support or attack the target class, we utilized the Brain Tumor MRI dataset (Nickparvar 2021). Additionally, we demonstrate ProtoArgNet's capability to identify spatial relationships that are undetectable by other approaches by applying it to (an adaptation to binary classification of) the SHAPES dataset (Hu et al. 2017).

6.1 Experimental Setup

Following the usual protocol (Chen et al. 2019), the input images are resized to 224×224. We set the number of prototypes N to 1024.[2] For training the model, we set the batch size to 32 and the number of epochs to 1000. The convolutional backbone was ResNet-50 (He et al. 2016) pre-trained using ImageNet (Deng et al. 2009). The choices of batch size, number of training epochs, convolutional backbone and pre-trained weights are aligned with previous prototypical-part-learning approaches. The SMLP had 1 hidden layer, 400 hidden activation maps and GELU (Hendrycks and Gimpel 2023) activation functions[3].

6.2 Classification Performance

The first two columns in Table 1 show that ProtoArgNet outperforms (in terms of classification accuracy) all baselines on the CUB, Cars and Brain datasets.

6.3 Sparsification of QBAF

To evaluate the tradeoff between sparsity and performance, we evaluated the accuracy under $40\%, 80\%$, and 94% sparsification ratios (240, 80, and 24 activation maps remaining).

[2]The performance of ProtoArgNet with various choices for N (512, 1024, 2048) is reported in the supplementary material in (Ayoobi, Potyka, and Toni 2024b).

[3]The performance of ProtoArgNet employing various MLP configurations, encompassing 1 to 5 hidden layers and a range of hidden activation maps (50, 100, 200, 400, 600), is detailed in the supplementary material in (Ayoobi, Potyka, and Toni 2024b).

Sparsification Ratio	Datasets			
	CUB	Cars	Brain	SHAPE
0.4	85.4	89.3	99.5	99.8
0.8	85.4	89.3	99.5	99.8
0.94	85.1	88.8	99.4	99.8

Table 2: Classification accuracy of ProtoArgNet with different sparsification ratios/datasets. As in SpArX(Ayoobi, Potyka, and Toni 2023), the local explanations with various ratios do not compromise classification accuracy.

Super-Prototypes	Prototype Layer	Classifier	Accuracy	
			CUB	Cars
—	L2	Fixed	79.5	86.4
—	L2	SMLP	81.5	86.9
—	Cosine	Fixed	81.7	87.4
—	Cosine	SMLP	81.9	88.0
✓	L2	Fixed	81.4	87.6
✓	L2	SMLP	82.7	88.3
✓	Cosine	Fixed	83.5	88.9
✓	Cosine	SMLP	**85.4**	**89.3**

Table 3: Ablation study with different prototype layers and classifiers with respect to the super-prototypes.

As Table 2 shows, the classification accuracy remained unchanged up to the 93% ratio. At 94% sparsification, the accuracy starts dropping. However, we can see that the size of the SMLP can often be reduced significantly without affecting its performance negatively. This is in line with the experiments on MLPs in (Ayoobi, Potyka, and Toni 2023).

6.4 Ablation study

Ablation studies on CUB and Cars (excluding the Brain and SHAPES datasets, which have not been used originally for the baseline methods) in Table 3 show that ProtoArgNet achieves the best accuracy when employing a cosine similarity prototype layer with an SMLP with one hidden layer. Notably, ProtoArgNet surpasses the performance of state-of-the-art methods (see Table 1) even when utilizing a fixed logistic regression layer, instead of SMLP super-prototype layers (but performs best with the SMLP).

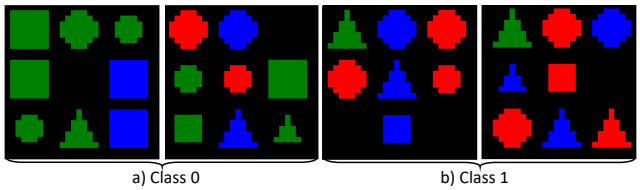

Figure 4: Examples from (our binary adaptation of) the SHAPES dataset. Class 1 contains images with a triangle in the leftmost column and a circle in the rightmost column of the same row or vice versa. Class 0 is when these conditions are not met.

6.5 Localization and Spatial Correlations

Figure 3 shows some super-prototypes for randomly selected examples from the Brain Tumor MRI dataset. These images showcase the regions that are either supporting or attacking the corresponding classes. For instance, when examining the leftmost input image, a radiologist would find that the 'Meningioma' (benign tumour) class has the highest probability. She can then look into the plausibility of the decision process by looking at the corresponding super-prototype (leftmost super-prototype). The prototypical-parts associated with green-highlighted regions should be indicative of 'Meningioma' (benign tumour), while those in the red-highlighted regions should be contraindicative. To understand the significance of the red-highlighted area, the radiologist can compare it with the super-prototypes of other classes where the same regions are highlighted in green. The second super-prototype from the left, associated with the Glioma (malignant tumour) class, highlights these regions in green, suggesting that further examination of that region may be necessary.

To assess whether different image classification methods can localize the prototypical-parts and encode spatial relationships between them, we adapted the SHAPES dataset (Hu et al. 2017) as a benchmark. We randomly generated synthetic images containing 3×3 grids of circles, triangles, and squares in different colours (red, green, and blue). An image is assigned to Class 1 if a triangle is located in the leftmost column and a circle is located in the rightmost column of the same row or vice versa, with a circle in the leftmost column and a triangle in the rightmost column of the same row[4], and Class 0 otherwise. The resulting dataset comprises 10,000 224×224 images with balanced binary class labels. Figure 4 shows examples of images in the dataset.

The last column in Table 1 compares the accuracy of the baselines for this SHAPES dataset. ProtoArgNet, with an accuracy of $99.7\% \pm 0.2\%$, significantly outperforms all other approaches (whose accuracy is around 50%). This can be explained by noting that these models only look at the presence of prototypes in images, but are unable to infer information from their relative position. ProtoArgNet addresses this limitation by using channel-wise max and super-prototypes, which enable the model to infer the spatial correlation of different prototypical-parts in the image when needed for classification.

6.6 Cognitive Complexity of Explanations

We measure the *cognitive complexity* of an explanation of a prototypical-part-learning approach by the number of activated prototypes per input x_i. We consider a prototype p_j to be activated if the maximum value in its similarity map (in $\mathcal{SM}_j$) exceeds a threshold of $\tau = 0.1$ (after normalizing the absolute value of the similarity scores to the range [0, 1]).

ProtoArgNet retains the 7×7 spatial dimensions of the convolutional output from the ResNet50 backbone, applying the maximum function across $\mathcal{CWM}s$ during inference. This ensures that at most a single prototype is activated per

[4]This criterion can be customized to reflect the user's preferences, e.g. the dataset could assign disjunction of multiple criteria.

Figure 5: Examples of ProtoArgNet explanations for the CUB and Cars dataset. The first row shows input images. The second row shows the super-prototypes of the target classes provided to the user as explanations. The third row shows the corresponding localized activated prototypical-parts for each super-prototype visualized by following the attack and support relations in the QBAF. The last two rows show the corresponding supporting prototypical-parts and attacking prototypical-parts.

spatial location for each input. Thus, the theoretical upper bound for activated prototypes in ProtoArgNet is $7 \times 7 = 49$ per example. In contrast, the upper bound for activated prototypes in other methods equals the total number of prototypes (see Table 4). Notably, ProtoArgNet uses these 49 prototypes to construct the super-prototypes of all classes. Further, Table 4 reports the Average number of Activated Prototypes per example ($\#AAP$), confirming that ProtoArgNet has lower cognitive complexity than other methods.

Method	# AAP / Upper Bound	
	CUB	Cars
ProtoPNet	1147.84/2000	1059.49/2000
ProtoPShare	182.58/400	159.20/480
ProtoPool	95.19/202	70.64/195
ProtoTrees	103.78/202	76.31/195
TesNet	1093.54/2000	1014.82/2000
D-ProtoPNet	594.23/1200	548.69/1176
ST-ProtoPNet	1167.41/2000	1086.57/1960
PIP-Net	211.83/495	213.46/515
ProtoArgNet	**24.57/49**	**8.42/49**

Table 4: Comparing average numbers of activated prototypes ($\#AAP$) per sample and upper bound of activated prototypes. A lower number means lower cognitive complexity.

6.7 Qualitative Evaluations

The super-prototypes of the target classes in the bottom row of Figure 5 illustrate the local explanations generated for a few data instances from the CUB and the Cars datasets. The top row shows the input images fed to ProtoArgNet. To interpret the super-prototypes, one could trace the attack and support relations in the QBAF to localize the prototypical-parts in the super-prototypes as in the third row. The green overlay on the super-prototypes highlights the regions in the input image that support the classification, while the red areas identify the attacked or unsupported portion of the input. For example, the super-prototype of the left-most image in Figure 5 can be interpreted as the bird's neck and the wing resembling the target class while the belly and flank differ from the observed target class instances in the training set.

7 Conclusion

ProtoArgNet is a novel prototypical-part-learning approach. It utilizes super-prototypes that combine multiple prototypical-parts to a single class representation that can take account of spatial relationships between individual parts. Using an MLP structure for the super-prototypes layers allows ProtoArgNet to capture non-linear relationships, while applying the SpArX methodology allows interpretable argumentative reading of the MLP as a QBAF. Experiments show that ProtoArgNet outperforms state-of-the-art prototypical-part-learning approaches in terms of accuracy, cognitive complexity, and the ability to learn spatial relationships between prototypical-parts.

Future work includes expanding ProtoArgNet's capabilities further to encompass multi-modal data. Further, we plan to deploy ProtoArgNet in the medical domain and explore the automatic generation of human-readable interpretations of the super-prototypes and QBAFs for explanatory purposes, including interactive explanations (Cyras et al. 2021).

Acknowledgements

This research was partially funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 101020934, ADIX) and by J.P. Morgan and by the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme.

References

Albini, E.; Lertvittayakumjorn, P.; Rago, A.; and Toni, F. 2020. DAX: Deep Argumentative eXplanation for Neural Networks. *CoRR*, abs/2012.05766.

Atkinson, K.; Baroni, P.; Giacomin, M.; Hunter, A.; Prakken, H.; Reed, C.; Simari, G. R.; Thimm, M.; and Villata, S. 2017. Towards Artificial Argumentation. *AI Mag.*, 38(3): 25–36.

Ayoobi, H.; Cao, M.; Verbrugge, R.; and Verheij, B. 2019. Handling Unforeseen Failures Using Argumentation-Based Learning. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 1699–1704.

Ayoobi, H.; Cao, M.; Verbrugge, R.; and Verheij, B. 2021a. Argue to Learn: Accelerated Argumentation-Based Learning. In *20th IEEE International Conference on Machine Learning and Applications (ICMLA)*.

Ayoobi, H.; Cao, M.; Verbrugge, R.; and Verheij, B. 2021b. Argumentation-Based Online Incremental Learning. *IEEE Transactions on Automation Science and Engineering*, 1–15.

Ayoobi, H.; Kasaei, H.; Cao, M.; Verbrugge, R.; and Verheij, B. 2022. Local-HDP: Interactive open-ended 3D object category recognition in real-time robotic scenarios. *Robotics and Autonomous Systems (RAS)*, 147: 103911.

Ayoobi, H.; Kasaei, H.; Cao, M.; Verbrugge, R.; and Verheij, B. 2023. Explain What You See: Open-Ended Segmentation and Recognition of Occluded 3D Objects. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 4960–4966.

Ayoobi, H.; Potyka, N.; and Toni, F. 2023. SpArX: Sparse Argumentative Explanations for Neural Networks. In Gal, K.; Nowé, A.; Nalepa, G. J.; Fairstein, R.; and Radulescu, R., eds., *European Conference on Artificial Intelligence (ECAI)*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, 149–156. IOS Press.

Ayoobi, H.; Potyka, N.; and Toni, F. 2024a. Argumentative interpretable image classification. In *Proceedings of the 2nd International Workshop on Argumentation for eXplainable AI co-located with the 10th International Conference on Computational Models of Argument (COMMA 2024)*, 3–15. CEUR Workshop Proceedings.

Ayoobi, H.; Potyka, N.; and Toni, F. 2024b. ProtoArgNet: Interpretable Image Classification with Super-Prototypes and Argumentation [Technical Report]. arXiv:2311.15438.

Ayoobi, H.; and Rezaeian, M. 2020. Swift distance transformed belief propagation using a novel dynamic label pruning method. *IET Image Processing*, 14(9): 1822–1831.

Bridle, J. S. 1990. Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. In Soulié, F. F.; and Hérault, J., eds., *Neurocomputing*, 227–236. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-76153-9.

Chen, C.; Li, O.; Tao, D.; Barnett, A.; Rudin, C.; and Su, J. K. 2019. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32.

Cyras, K.; Rago, A.; Albini, E.; Baroni, P.; and Toni, F. 2021. Argumentative XAI: A Survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*.

Dejl, A.; Ayoobi, H.; Williams, M.; and Toni, F. 2023. CAFE: Conflict-Aware Feature-wise Explanations. arXiv:2310.20363.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Donnelly, J.; Barnett, A. J.; and Chen, C. 2022. Deformable ProtoPNet: An Interpretable Image Classifier Using Deformable Prototypes. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10255–10265.

Goyal, Y.; Wu, Z.; Ernst, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Counterfactual visual explanations. In *ICML*. PMLR.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hendrycks, D.; and Gimpel, K. 2023. Gaussian Error Linear Units (GELUs). arXiv:1606.08415.

Hu, R.; Andreas, J.; Rohrbach, M.; Darrell, T.; and Saenko, K. 2017. Learning to Reason: End-to-End Module Networks for Visual Question Answering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 804–813.

Jo, J.; and Bengio, Y. 2017. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*.

Kim, E.; Kim, S.; Seo, M.; and Yoon, S. 2021. XProtoNet: Diagnosis in Chest Radiography with Global and Local Explanations. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15714–15723.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3D Object Representations for Fine-Grained Categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, 554–561.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*, 521(7553): 436–444.

Leofante, F.; Ayoobi, H.; Dejl, A.; Freedman, G.; Gorur, D.; Jiang, J.; Paulino-Passos, G.; Rago, A.; Rapberger, A.; Russo, F.; Yin, X.; Zhang, D.; and Toni, F. 2024. Contestable AI Needs Computational Argumentation. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning*, 888–896.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101.

Lundberg, S. M.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In *NeurIPS*.

Mihailescu, I.; Weng, A.; Sharma, S.; Ghitu, M.; Grewal, D.; Chew, K.; Ayoobi, H.; Potyka, N.; and Toni, F. 2023. PySpArX - A Python library for generating Sparse Argumentative eXplanations for neural networks. In *Proceedings of the 39th International Conference on Logic Programming (ICLP 2023)*, 336–336. Open Publishing Association.

Nauta, M.; Schlötterer, J.; van Keulen, M.; and Seifert, C. 2023. PIP-Net: Patch-Based Intuitive Prototypes for Interpretable Image Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2023)*. Computer Vision Foundation / IEEE.

Nauta, M.; van Bree, R.; and Seifert, C. 2021. Neural Prototype Trees for Interpretable Fine-Grained Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, 14933–14943. Computer Vision Foundation / IEEE.

Nickparvar, M. 2021. Brain Tumor MRI Dataset.

Potyka, N. 2021. Interpreting Neural Networks as Quantitative Argumentation Frameworks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-21)*.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1(5): 206–215.

Rymarczyk, D.; Struski, L.; Górszczak, M.; Lewandowska, K.; Tabor, J.; and Zieliński, B. 2022. Interpretable Image Classification With Differentiable Prototypes Assignment. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, 351–368. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-19774-1.

Rymarczyk, D.; Struski, L.; Tabor, J.; and Zielinski, B. 2021. ProtoPShare: Prototypical Parts Sharing for Similarity Discovery in Interpretable Image Classification. In Zhu, F.; Ooi, B. C.; and Miao, C., eds., *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 1420–1430. ACM.

Sattarzadeh, S.; Sudhakar, M.; Plataniotis, K. N.; et al. 2021. Integrated Grad-CAM: Sensitivity-Aware Visual Explanation of Deep Convolutional Networks via Integrated Gradient-Based Scoring. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Sukpanichnant, P.; Rago, A.; Lertvittayakumjorn, P.; and Toni, F. 2021. Neural QBAFs: Explaining Neural Networks Under LRP-Based Argumentation Frameworks. In *AIxIA 2021 - Advances in Artificial Intelligence - 20th International Conference of the Italian Association for Artificial Intelligence, Virtual Event, December 1-3, 2021*, volume 13196 of *Lecture Notes in Computer Science*, 429–444. Springer.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. Caltech-UCSD Birds-200-2011 (CUB-200-2011). Technical Report CNS-TR-2011-001, California Institute of Technology.

Wang, C.; Liu, Y.; Chen, Y.; Liu, F.; Tian, Y.; McCarthy, D.; Frazer, H.; and Carneiro, G. 2023. Learning Support and Trivial Prototypes for Interpretable Image Classification. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2062–2072.

Wang, J.; Liu, H.; Wang, X.; and Jing, L. 2021. Interpretable Image Recognition by Constructing Transparent Embedding Space. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 875–884.