SPARROW: Semantically Coherent Prototypes for Image Classification

Stefan Kraft[1,4]
stefan.kraft@stz-softwaretechnik.de

Klaus Broelemann[2]
klaus.broelemann@schufa.de

Andreas Theissler[3]
https://orcid.org/0000-0003-0746-0424

Gjergji Kasneci[4,2]
gjergji.kasneci@uni-tuebingen.de

[1] IT-Designers Group
Esslingen am Neckar, GER

[2] SCHUFA Holding AG
Wiesbaden, GER

[3] Aalen University of Applied Sciences
Aalen, GER

[4] Data Science & Analytics Research
The University of Tübingen
Tübingen, GER

Abstract

Current prototype-based classification often leads to prototypes with overlapping semantics where several prototypes are similar to the same image parts. Also, single prototypes tend to activate highly on a mixture of semantically different image parts. This impedes interpretability since the nature of the connections between the parts is unknown. We propose a framework that is comprised of two key elements: (i) A novel method which leads to semantically coherent prototypes and (ii) an evaluation protocol which is based on part annotations and allows to quantitatively compare the explanatory capacity of prototypes from different methods. We demonstrate the viability of our framework by comparing our method to a standard prototype-based classification method and show that our method is capable of producing prototypes of superior interpretability.

1 Introduction

Recent research has called for ML models that are interpretable by design [20] rather than post-hoc explanations on black-box models. However, in image processing most explanation approaches rely on activation and saliency maps, which can be highly misleading and sometimes even lead to spurious saliency maps [1].

Prototype classification, that is classifying samples based on their similarity to prototypical samples in a latent space, aims to achieve this kind of interpretability [20]. However, there is a performance-interpretability trade-off with regard to the number of prototypes [6]. While the classification performance typically increases with an increasing number of prototypes, the explanation quality for a local prediction diminishes; even more so if either prototypes do not relate to specific semantic concepts or if they overlap on the same semantic concepts.

This work complements the literature on prototype classification by suggesting a novel framework for prototype quality: `SPARROW` (uniquenessS – sPArsity – naRROWness). `SPARROW`

© 2021. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

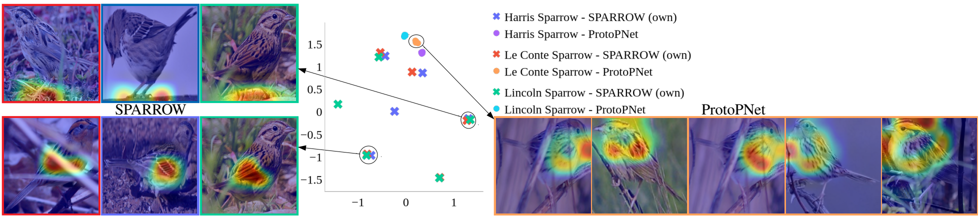


Figure 1: **Comparison of prototypes** for ProtoPNet [6] and our own method: SPARROW. The latter encourages semantically coherent prototypes (*e.g.* legs, wing) while ProtoPNet selects class prototypes (*e.g.* Le Conte Sparrow). The center panel shows a combined t-SNE visualization of the latent prototype spaces based on a cosine distance measure. Samples that represent the prototypes are displayed (left: SPARROW, right: ProtoPNet) and connected to their coordinates in t-SNE space. The superior interpretability of prototypes from SPARROW will be quantified globally by means of our novel evaluation protocol.

enables researchers and practitioners alike to learn semantically coherent prototypes and evaluate their explanatory capacity. It is inspired by well understood interpretability principles: (1) **Sparsity** – Sparse explanations allow humans to understand how a few different concepts jointly form a model prediction [20]. (2) **Narrowness** – An explanation should capture a concept as narrowly as possible. *E.g.*, saliency areas covering larger image parts can be ambiguous and the observer can only guess what the pivotal factor in favor of the model's decision was [20]. (3) **Uniqueness** – The overlapping of concepts should be minimized.

We showcase the effectiveness of SPARROW – in terms of a quantifiable performance (instead of just visual inspection) and in terms of semantically coherent prototype generation – in comparison to state-of-the-art prototype generation [6].

ProtoPNet We briefly introduce ProtoPNet [6] on which the SPARROW prototype learning method is based. ProtoPNet utilizes a set of loss components (cf. table 1) to jointly learn neural network weights and weights of prototype vectors. It calculates similarities between parts (patches) of latent space sample embeddings and prototypes. Similarity scores are then used in a weighted superposition in a final fully-connected layer to derive class prediction scores. Since the similarities are not derived post-hoc and the last layer is simple and transparent, ProtoPNet qualifies as being intrinsically interpretable. During training, fixed and evenly distributed class identities for prototypes are enforced. In the end, prototypes are projected onto the latent space patch of a training sample which they are most similar to. Thus, prototypes can naturally represent a part of a sample image and the latent space self-activation maps of prototypes can be upsampled and visualized in the input space. The same can be done for activation maps between prototypes and test samples so that the reason for their similarity can be visually inspected. Details about the model and training process are available in the supplementary material which also contains a table of notation.

Case Study We want to demonstrate that semantically coherent prototypes lead to less ambiguous visual interpretability. Figure 1 shows upsampled latent space self-activation maps of prototypes from ProtoPNet [6] and our method (SPARROW). While it is difficult for the competitor to put semantic labels on the self-activation maps of prototypes, because

Component	Goal	Type	Weight
$\mathcal{L}^{\text{CrsEnt}}$	Classification performance	CE	1
$\mathcal{L}^{\text{Clst}}$	Cluster samples around prototypes	Agg	1
$\mathcal{L}^{\text{Sep}}$	Separate clusters by class	Agg	0
$\mathcal{L}^{\text{AS}*}$	Decorrelate prototypes	log-loss	1
$\mathcal{L}^{\text{PSD}*}$	Keep prototypes close to a sample	log-loss	100

Table 1: **Overview of loss components**. Components marked with * are our own addition to ProtoPNet [6]. CE: Cross Entropy, Agg: Aggregate function. Our choice of weights is discussed in section 3.

they are neither narrow nor unique, the prototypes from `SPARROW` are easier to interpret and less ambiguous. *E.g*. the activations of the bottom three images on the left seem to focus on the right wing while we could not make a similarly clear statement for the prototypes from ProtoPNet. Additionally, the t-SNE visualization in the middle shows different samples in cosine distance space. We consider cosine similarity as it serves as a natural measure to quantify correlations between prototypes. Prototypes from ProtoPNet are very similar to one another since they are closely clustered while our prototypes are further apart, indicating that they are pointing to different concepts. As Adebayo *et al*. demonstrated [1], simply judging two methods from a handful of samples is not sufficient to compare them with respect to their general capacity to produce semantically meaningful explanations, which makes the need for our `SPARROW` quantitative evaluation protocol all the more urgent.

Related Work *Interpretable Models by Design*: In a seminal position paper, Rudin [20] argued in favour of building predictive models that are explainable by design. Recent work followed this suggestion focusing on rendering neural networks interpretable using piecewise linear functions [2] or prototypes [6, 10, 25]. Our present work is most similar to the work of [6]. However, it differs from [6] by introducing novel loss components that encourage unique and narrow semantically coherent prototypes.

Interpretability measures: To the best of our knowledge other works in the field of prototype classification have not evaluated the interpretability of latent space prototype activations quantitatively but instead rely on qualitative visual assessments. In the broader field of predictions with convolutional neural networks (CNNs) Bau *et al*. [4] quantify the alignment of CNN units with various pixel-wise labeled concepts by performing binary segmentation on activation maps and calculating the intersection over union with labeled concepts. We also match activation maps with labels but use samples with keypoint part annotations instead of pixel-wise annoation maps. They proceed to count the number of distinct visual concepts that are matched per CNN layer. We follow a similar goal by measuring the completeness of part annotations that are matched by prototypes over all samples. Zhang *et al*. [24] extended this work by adding a location instability metric which measures the degree to which the inferred position of a CNN activation pattern in the input space varies with respect to a set of landmark positions over different images. We calculate a prototype focus measure which does not use other landmark positions but instead determines how consistent prototypes are in recurrently matching the same parts over all samples.

2 SPARROW

Method for Learning Semantically Coherent Prototypes In order to learn semantically coherent prototypes, we extend ProtoPNet [6] with two additional loss components. First, we add an angular similarity (*AS*) based loss which aids in decorrelating the prototypes:

$$\mathcal{L}^{\mathrm{AS}} = -\frac{1}{C}\sum_{v=1}^{C} \max_{i,j\in\mathcal{I}_v} \log(1-\mathrm{AS}(\mathbf{p}_i,\mathbf{p}_j)), \tag{1}$$

where $\mathcal{I}_v$ denotes the set of prototype indices of intra-class prototypes and C is the number of classes. The *AS* between prototypes $\mathbf{p}_i$ and $\mathbf{p}_j$ is [22] $\mathrm{AS}(\mathbf{p}_i,\mathbf{p}_j) = 1-\frac{1}{\pi}\arccos(\mathrm{CS}(\mathbf{p}_i,\mathbf{p}_j))$ with $i,j\in\mathcal{I}_v$ and the cosine similarity (*CS*) is [19] $\mathrm{CS}(\mathbf{p}_i,\mathbf{p}_j) = \frac{\mathbf{p}_i^T\cdot\mathbf{p}_j}{\|\mathbf{p}_i\|\|\mathbf{p}_j\|}$. The *AS* can be interpreted as a probability which allows to calculate a log-loss. As our initial experiments have shown, taking the maximum in eq. 1 reduces the variance of the angular similarity between intra-class prototype combinations as compared to other aggregate functions like average or sum. We also found that the optimization typically results in one prototype per class being close to samples of the class while the other prototypes become outliers in latent space. In order to fulfill the requirement by Chen *et al.* [6] to keep prototypes sufficiently close to samples in latent space, we implement a second loss component:

$$\mathcal{L}^{\mathrm{PSD}} = -\frac{1}{m}\sum_{j=1}^{m}\log(1-\frac{\mathrm{PSD}_j(X,\mathbf{p}_j)}{\mathrm{dist}_{\max}}). \tag{2}$$

Here, m denotes the number of prototypes, $\mathrm{dist}_{\max}$ is the maximum possible distance in latent space and the term that we call prototype-sample distance (*PSD*) is taken from Li *et al.* [10] as $\mathrm{PSD}_j(X,\mathbf{p}_j) = \min_{\mathbf{z}\in\mathrm{patches}(f(\mathbf{x}_\mu))}\|\mathbf{p}_j-\mathbf{z}\|^2$. It is calculated over all samples of the current batch, i.e. $\mathbf{x}_\mu\in X$ and $\mu\in\mathrm{batch}([1,\dots,n])$ with the set of training samples X of length n. Both new loss components are added to the total loss with static weights (cf. table 1).

Evaluation Protocol We propose an evaluation protocol which leverages information from part annotations to provide *explanatory capacity* estimates. On a high level, our approach is based on matching part annotations with activation masks. For each sample we assume that there are T keypoint annotations of sample parts available. The activation masks stem from the latent space activations of samples by prototypes which have the same class identity as the samples. These activation maps are upsampled to the input space and cropped to masks by an activation threshold. Choices for this threshold will be discussed in section 3. We show schematically annotated parts and prototype activation masks in figure 2. In the following we will refer to matches between activation masks and part annotations as “matches between prototypes and parts". If a prototype does not match any part, we select the closest part coordinate to the activation mask as a match. This is done because some activation masks are too narrow to match any part. For a more detailed explanation of the matching procedure we refer to the supplementary material. We are now ready to derive evaluation measures.

The decorrelation measure quantifies to which degree prototypes with the same class identity activate highly only on non-overlapping annotated part semantics. *E.g.* two prototypes that both activate highly on the wing of a bird have a low decorrelation score. Highly decorrelated prototypes typically lead to narrower saliency maps and therefore enable less ambiguous interpretations.

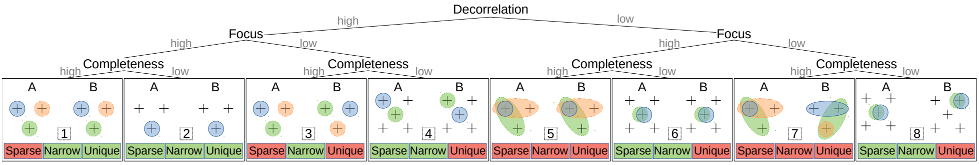


Figure 2: **Relation between SPARROW measures (tree nodes) and explainability principles (leafs).** Each leaf contains parts (crosses) of two schematic samples (A and B), where the parts in A and B are the same at the respective positions. Overlayed are schematic prototype masks (blue, green, orange) which depict a typical situation corresponding to the specific value scale (high, low) of SPARROW measure scores. The bottom of each leaf displays the leaf number and the consequence for each principle (green: fulfilled, red: not fulfilled).

The decorrelation of prototypes can be determined by first counting per sample the number of parts that are matched by u prototypes. We call this count $\text{cnt}_{i,u}^{\text{decorr}}$, where $u \in [1,\ldots,|\mathcal{P}_{c_i}|]$ with the number of prototypes $|\mathcal{P}_{c_i}|$ in class c_i of the sample with index i. In figure 2, leaf 5, we would have $\text{cnt}_{\text{idx}(A),1}^{\text{decorr}} = 2$, $\text{cnt}_{\text{idx}(A),2}^{\text{decorr}} = 0$ and $\text{cnt}_{\text{idx}(A),3}^{\text{decorr}} = 1$. Perfect decorrelation would require $\text{cnt}_{i,u>1}^{\text{decorr}} = 0$ for every sample index i. In addition, the higher the number of prototypes which match the same part becomes the more correlated this set of prototypes becomes. Motivated by this idea we define the first measure.

Definition 1 (Prototype decorrelation) *The prototype decorrelation (ptd) is defined as the normalized weighted sum over the number of times that annotated parts are matched by prototype activation masks over all samples:*

$$\text{ptd} = \sum_{i,u} \frac{(|\mathcal{P}_{c_i}| + 1 - u) \cdot \text{cnt}_{i,u}^{\text{decorr}}}{N \cdot \tilde{T}_i \cdot |\mathcal{P}_{c_i}|}, \tag{3}$$

where N is the total number of samples and $\tilde{T}_i$ is the number of parts for sample i which are matched by at least one prototype ($\tilde{T}_i > 0$ as discussed before). $\text{ptd}_{\text{max}} = 1.0$ is achieved when no part is matched by more than one prototype. Then, $\text{cnt}_{i,u=1}^{\text{decorr}} = \tilde{T}_i$, and $\text{cnt}_{i,u>1}^{\text{decorr}} = 0$ for all i. On the other hand, ptd_{min} arises when all parts are matched by all prototypes for every sample. Then, $cnt_{i,u=|P_{c_i}|} = \tilde{T}_i$ and $cnt_{i,u\neq|P_{c_i}|} = 0$ for all i. If we neglect pruning, i.e. $|\mathcal{P}_{c_i}| = \frac{m}{C}$ with the number of prototypes m and classes C for all c_i, we arrive at $\text{ptd}_{\text{min}} = \frac{C}{m}$. Since it only makes sense to look into the decorrelation of prototypes per class for $m > C$ (where typically $m \gg C$), ptd_{min} will typically be a small number. This result also means that for only one prototype per class *ptd* yields a perfect decorrelation score which would be expected since it is defined per class.

The prototype focus level follows the goal of determining how consistent prototypes are in recurrently matching the same part over all samples. A prototype that matches the head of a bird in one picture but the tail in another would not be well focused. Prototypes with little focus either consistently represent multiple part semantics (*e.g.* always the breast and the belly of a bird) or represent semantics that are not purely part-related (they might *e.g.* focus on color or texture) or a mixture of both. These cases will be further discussed in section 4.

For each prototype and all T types of annotated parts we count the number of times that this prototype globally (i.e. over all samples) matches this part type. We normalize this quantity by dividing each count by the total number of matches between this prototype and any part in any sample. We call it $\text{frac}_{j,k}^{\text{focus}}$, where $j \in [1,\ldots,m]$ and $k \in [1,\ldots,T]$ with the total number of prototypes m and parts T. $\sum_k \text{frac}_{j,k}^{\text{focus}} = 1$ for each prototype with index j due to normalization. A highly focused prototype would have very sparse values in $\text{frac}_{j,k}^{\text{focus}}$. In order to quantify this concept over all prototypes we suggest looking at the top-1 matched part for each prototype which leads to the distribution $\text{frac}_{j}^{\text{focus-top-1}} = \max_k(\text{frac}_{j,k}^{\text{focus}})$. We derive the following measure:

Definition 2 (Prototype focus) *The prototype focus (ptf) is defined as the median of the distribution of normalized global match counts of prototypes with their top-1-matched part:*

$$\text{ptf} = \text{median}_j(\text{frac}_{j}^{\text{focus-top-1}}). \tag{4}$$

For samples A and B in figure 2, leaf 7, we can find $\text{frac}_{\text{idx(blue)}}^{\text{focus-top-1}} = \frac{2}{3}$, $\text{frac}_{\text{idx(green)}}^{\text{focus-top-1}} = \frac{1}{2}$ and $\text{frac}_{\text{idx(orange)}}^{\text{focus-top-1}} = \frac{1}{3}$, so that $\text{ptf} = \frac{1}{2}$. For leaf 1 instead, we find $\text{ptf} = 1$.

The completeness of the description of a sample by prototypes is the last concept that we present. It quantifies how fully samples are described by prototypes in terms of the annotated parts. A sample would be completely described by prototypes if all its parts were matched by at least one prototype. This is a useful concept to track in order to balance the trade-off between the sparsity of explanations and a full description of samples by annotated part semantics that are deemed important by domain experts. This trade-off exists since it is easiest to maximize completeness by increasing the number of prototypes. This point is discussed in more detail at the end of this section.

To derive the completeness measure, we start by counting how many parts are matched by prototypes per sample. We name this count $\text{cnt}_i^{\text{comp}}$ and it is $\text{cnt}_i^{\text{comp}} \in [1,\ldots,T]$ for each sample with index i and with the total number of parts T. For example in figure 2, leaf 4, we have $\text{cnt}_{\text{idx}(A)}^{\text{comp}} = 2$. Since it is clear that samples with lower numbers of captured parts by prototypes are worse with respect to the completeness of the sample description we can define the completeness measure as follows.

Definition 3 (Completeness of sample description) *The completeness of sample description (sac) is defined as the normalized sum over the number of annotated parts which are matched by at least one prototype activation mask over all samples:*

$$\text{sac} = \sum_i \frac{\text{cnt}_i^{\text{comp}}}{N \cdot T}. \tag{5}$$

For the samples A and B in leaf 4 (figure 2) this yields $\text{sac} = \frac{2+2}{2\cdot 5} = 0.4$. We see that $\text{sac}_{\text{max}} = 1.0$ which requires all parts in all samples to be captured by prototypes. In practice this may not be possible if not all parts are visible for every sample and in this case are not annotated. The theoretical minimum score is $\text{sac}_{\text{min}} = \frac{1}{T}$ which would happen if every sample had exactly one part matched by all prototypes ($\text{cnt}_i^{\text{comp}} = 1$ for all i). If it is the goal to maximize both *ptd* and *sac* measures, we suggest to use the following overall measure.

Definition 4 (Decorrelation-completeness balance) *The decorrelation-completeness balance (dcb) between the ptd and sac measures is defined as their harmonic mean as:*

$$\text{dcb} = 2 \cdot \frac{\text{ptd} \cdot \text{sac}}{\text{ptd} + \text{sac}}. \tag{6}$$

Take note that in order to achieve a perfect *dcb* score it is necessary that the number of prototypes per class is exactly as high as the number of annotated parts per sample, which means that (without pruning) the total number of prototypes must be $m = C \cdot T$. Otherwise, since a prototype is always matched with at least one part, a perfect *sac* score could only be achieved by a reduced *ptd* score.

Finally, we look at the relation between the SPARROW measures and the explainability principles introduced in section 1. We can see at the bottom of figure 2 that the only way to fulfill all principles is to achieve high *ptd* and *ptf* scores and a low *sac* score (leaf 2) which would require to have only one prototype per class. One might argue that leaf 1 is actually preferable, i.e. favoring completeness over sparsity of explanations. There are valid arguments for this choice. For once, it is generally known that there is a trade-off between sparsity and classification performance [21]. Additionally, in the computer vision domain it is not believed that fewer pixels generally constitute a better explanation [21]. However, in the case of explanations by prototypes we argue that the similarity of a sample to prototypes has to be verified by the end user separately for each prototype. Cognitive science indicates that people rarely expect complete explanations but are typically content with a few presented causes for a decision [15]. Grasping many explanations at the same time is difficult for humans since their mental capacity is limited to process 7 ± 2 items at once [14]. In [18] predictions are explained with hierarchical prototype trees and the authors share the belief that smaller trees with fewer prototypes are easier to interpret. In the end - although we speculate that sparse explanations may be preferable and one will typically want to achieve a situation in between those depicted in leaf 1 and leaf 2 - we follow the assessment of [16] in that the importance of sparsity should be measured by empirical evaluation which we plan to perform in the future.

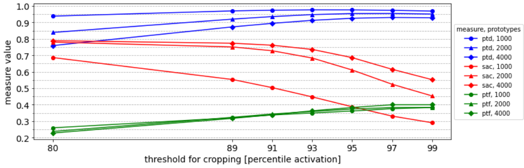
3 Experiments

In the following, “ProtoPNet" denotes the model described in section 1. “SPARROW (own)" includes the additional novel loss components discussed in section 2. For additional details about the experiments, *e.g.* the dataset, data preprocessing, hyperparameters, training hardware or time demand we refer to the supplementary material.

Dataset The subsequent experiments are performed on the Caltech-UCSD Birds-200-2011 dataset (CUB) [23]. This dataset is selected since it allows for a direct comparison to the results from Chen *et al.* [6].

Hyperparameter Tuning We tuned the loss component weights by a combination of random search and Bayesian Optimization to the values reported in table 1.

Another important hyperparameter is the choice of the threshold for activation masks (cf. section 2). This choice is driven by the preference if only the most salient similarities should be kept (high threshold) or if also minor similarities should be taken into account (low threshold). We followed Chen *et al.* [6], who selected the 95-th percentile, in opting for a high threshold. However, we wanted to analyze the effect that varying the threshold in the high value range has on the SPARROW measures. This is shown in figure 3 (a). We can see that *sac* is most sensitive to the threshold whereas *ptd* and *ptf* vary to a lesser extent. We recommend to chose the 95-th percentile when comparing a method to other works but we emphasise that other choices are valid as well.

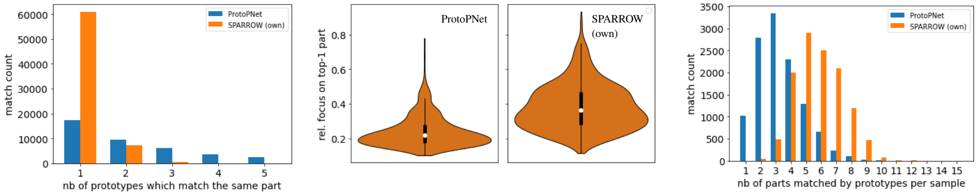


(a)

ProtoPNet		Own	
Prototypes	Accuracy	Prototypes	Accuracy
1000	**73.4 ± 0.3**	1000	70.3 ± 1.1
2000	73.3 ± 1.5	2000	**74.7 ± 0.7**
4000	73.7 ± 1.4	4000	**75.9 ± 1.4**

(b)

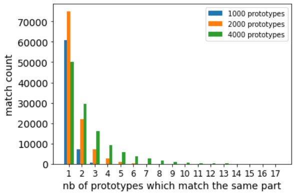
Figure 3: **(a) Dependency of SPARROW measures on the cropping threshold** for activation masks. **(b) Comparison of the average accuracy** of ProtoPNet with our own method.



(a) Distributions leading to *ptd* measures for two models

(b) Distributions leading to *ptf* measures for two models

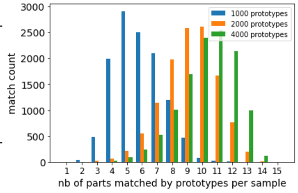
(c) Distributions leading to *sac* measures for two models



(d) Distrs. leading to *ptd* measures the SPARROW model

Model	Prototypes	Measures			
		ptd	*sac*	*dcb*	*ptf*
ProtoPNet	1000	78.4	22.1	34.5	21.7
SPARROW (own)	1000	**97.6**	38.8	55.5	36.2
SPARROW (own)	2000	95.3	61.1	74.5	37.4
SPARROW (own)	4000	92.5	**68.6**	**78.8**	**38.3**

(e) SPARROW evaluation measures



(f) Distrs. leading to *sac* measures for the SPARROW model

Figure 4: **Comparing the explanatory capacity of prototypes.** Top row: Distributions leading to the interpretability measures for 1000 prototypes and different models (from left to right): Prototype decorrelation (*ptd*), prototype focus (*ptf*) and completeness of sample description (*sac*). Bottom row: Left and right report results for the SPARROW model for various numbers of prototypes. Results for the SPARROW measures are shown in the table.

Classification Performance We compare the classification performance of our method with ProtoPNet as shown in figure 3 (b). The performance of the two models is overall comparable. ProtoPNet shows better performance for a low number of prototypes while our SPARROW method shows the potential to perform better for higher numbers of prototypes.

Evaluation with SPARROW We start by comparing ProtoPNet with our method, where both methods use 1000 prototypes. The table in figure 4 (e) shows that there is a significant improvement in all four measures. This implies that prototypes produced by our method are at the same time more decorrelated from each other, more focused towards single semantic concepts while representing more semantic concepts in samples than ProtoPNet. This is reflected in the distributions from which the measures are derived. There are significantly more prototypes that match only one part per sample and significantly fewer prototypes overlapping at high numbers of same-sample matches (cf. figure 4 (a)) which leads to the higher *ptd* score. The distribution of our run in figure 4 (b) is shifted towards the perfect score of 1.0 when compared with ProtoPNet. This leads to the higher *ptf* score which is the median (white dot) of the distributions. Finally the distribution in figure 4 (c) for our method is shifted to the right when compared with ProtoPNet towards the theoretical goal of the *sac* measure of all counts being at mark 15. This comes as a bit of a surprise since our method was not designed to increase the completeness of the sample description. For just 1000 prototypes (i.e. 5 prototypes per class) it is actually more useful to look at this distribution instead of the *dcb* measure since a perfect *dcb* score is not possible (cf. discussion after definition 4). We can see that the maximum of the distribution is at mark 5 which seems ideal since counts at marks < 5 indicate an imperfect decorrelation and counts at marks > 5 an imperfect focus of the prototypes.

Up next, we compare the models based on our method for different numbers of prototypes. The distributions (d) and (f) again show additional details for the calculation of the measures. From the table in figure 4 (e) we see that an increased number of prototypes leads to a decreased decorrelation (*ptd*) and an increased completeness of sample description (*sac*) which we would intuitively have expected. This shows that it made sense to define the decorrelation-completeness balance measure (*dcb*) as the harmonic mean between *ptd* and *sac*. Based on *dcb* alone, 4000 prototypes seem optimal. However, if *ptd* or sparsity of explanations is considered most important, a lower number of prototypes might be preferable (cf. the discussion about sparsity at the end of section 2). Looking at the *ptf* measure we see an increased focus with an increasing number of prototypes. 1000 prototypes may not have been enough to capture all important semantics for optimizing the classification score so that the focus of prototypes was more “washed out" to compensate for this.

We can conclude that the quantification of the explanatory capacity of prototypes by means of the SPARROW evaluation protocol confirms the improved visual interpretability of prototypes from our method when compared with ProtoPNet (cf. case study in section 1).

4 Scope and Future Extensions

The SPARROW measures are guided by well-known principles of interpretability (cf. section 1). They are derived from latent space activations of a CNN which are then used as prototypes in the prediction following the ProtoPNet method [6]. Since ProtoPNet is model agnostic in the sense that it allows to use arbitrary convolutional base networks so is our evaluation protocol. A limitation that remains is the reliance on part semantics in the form of keypoint annotations. SPARROW is therefore best used if there is domain-specific indication that an image classification dataset contains part-related concepts which are well suited for a corresponding prediction task. SPARROW then allows to optimize models to contain semantically coherent prototypes representing those concepts. Such models are expected to be useful for human experts like ornithologists or physicians for whom it is a common strategy to explain class predictions based on part-related semantics [6] in a case-based reasoning fashion [13, 21]. Examples for suitable tasks and datasets which already contain keypoint part annotations are the prediction of animal species like birds [23] or tigers [11]. Unfortunately not many datasets currently contain ground truth annotations at the part level. Oftentimes there is no way around employing domain experts to perform the labeling. This issue is not just symptomatic of our proposed evaluation protocol but is frequently found in the area of evaluating conceptual representations and *e.g.* lead [4] to release the Broden dataset which contains diverse conceptual annotations.

For other tasks like human pose estimation [3, 8] concepts that relate to a combination of parts (*e.g.* the relative position of joints to each other) are deemed important [5]. SPARROW could still be applied to such tasks but it would make sense to relax the definition of the prototype focus measure to tolerate prototypes which recurrently focus on the same group of parts instead of single parts. Apart from keypoint part annotations many datasets also use other types of concept annotations like pixel-wise binary masks or bounding boxes as well as attributes or relations between those annotations which are especially useful to label higher level concepts [4, 9]. In order to match similarity maps of samples and prototypes with pixel-wise annotations, the approach in [4] can be used. Also, harder tasks like semantic image retrieval are generally expected to require models to learn higher level semantic concepts [7, 12]. We therefore want to evolve SPARROW to be applicable to different types of possibly spatially or semantically overlapping concept annotations on different levels of abstraction. This would require to adapt the prototype focus measure. However, if *e.g.* bounding boxes are restricted around relatively small image parts, they could be converted to keypoint part annotations and it would make sense to apply SPARROW in its current form.

An interesting extension would be to incorporate the importance of visual characteristics (*e.g.* texture, shape, hue) about part-related concepts that prototypes represent into SPARROW. This knowledge can be obtained without additional human annotations as was shown by [17]. Also pruning of prototypes which focus on background semantics [6] should be helpful because such prototypes do not generalize well and distort the results of SPARROW.

Currently, the SPARROW evaluation protocol encourages single prototypes per part per class. An alternative would be to enable sharing prototypes between classes. This could potentially lead to a better global interpretability. We might for example find inter-class prototypes for short and long bird legs which could be useful to discriminate between different bird species. We believe that it comes down to the goal (*e.g.* knowledge discovery or decision support) which approach is preferable. For example end users might be confused if a sample from which a prototype originates has a different class identity than test samples that are explained by being similar to this prototype. This might diminish trust in the system.

Finally, we plan to break down the global analysis from SPARROW to the local prototype level. This way, SPARROW could be used to find prototype candidates for pruning. Additionally it could be used in concept discovery in an expert-in-the-loop approach to annotate new concepts and evaluate prototypes iteratively. For this purpose, prototype activations in samples that do not match any annotated concept should be investigated closely instead of automatically selecting the nearest part as a match, as it is currently done. In this respect the location instability measure [24] should prove useful to find new part-related concepts – something that the prototype focus measure cannot accomplish.

5 Conclusion and Outlook

The rapidly growing number of available prototype classification methods calls for standardized and efficient ways to assure the quality of a new technique in comparison with other approaches on various datasets. Quality assurance is a key aspect of explainable models since those explanations determine how well humans can understand the model predictions. In this work, we presented SPARROW, an own prototype generation method and a benchmarking protocol for the standardized and transparent comparison of prototype based classification methods. In the explainability field, SPARROW bears the potential to help researchers and practitioners alike to efficiently derive more realistic and use-case-driven prototype models and assure their quality through extensive comparative evaluations. We hope that this work contributes to further advances in explainability research.

Acknowledgements

We thank Mr. Martin Pawelczyk for many helpful discussions and comments. We also thank Ms. Elizabeth Baker for proofreading.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems (NeurIPS) 31*.

[2] David Alvarez-Melis and Tommi S Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems (NeurIPS) 31*, 2018.

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.

[6] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. In *Advances in neural information processing systems (NeurIPS)*, 2019.

[7] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[8] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *bmvc*, volume 2, page 5. Citeseer, 2010.

[9] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[10] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[11] Shuyuan Li, Jianguo Li, Hanlin Tang, Rui Qian, and Weiyao Lin. Atrw: A benchmark for amur tiger re-identification in the wild. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2590–2598, 2020.

[12] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European conference on computer vision*, pages 852–869. Springer, 2016.

[13] Cindy Marling, Stefania Montani, Isabelle Bichindaritz, and Peter Funk. Synergistic case-based reasoning in medical domains. *Expert systems with applications*, 41(2): 249–259, 2014.

[14] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[15] Tim Miller, Piers Howe, and Liz Sonenberg. Explainable ai: Beware of inmates running the asylum or: How i learnt to stop worrying and love the social and behavioural sciences. *arXiv preprint arXiv:1712.00547*, 2017.

[16] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682*, 2018.

[17] Meike Nauta, Annemarie Jutte, Jesper Provoost, and Christin Seifert. This looks like that, because... explaining prototypes for interpretable image recognition. *arXiv preprint arXiv:2011.02863*, 2020.

[18] Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14933–14943, 2021.

[19] Hieu V Nguyen and Li Bai. Cosine similarity metric learning for face verification. In *Asian conference on computer vision*, pages 709–720. Springer, 2010.

[20] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019.

[21] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *arXiv preprint arXiv:2103.11251*, 2021.

[22] Anshumali Shrivastava and Ping Li. Improved asymmetric locality sensitive hashing (alsh) for maximum inner product search (mips). In *31st Conference on Uncertainty in Artificial Intelligence, UAI 2015*, 2015.

[23] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[24] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[25] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.