Interpretability Benchmark for Evaluating Spatial Misalignment of Prototypical Parts Explanations

Mikołaj Sacha[1, 2], **Bartosz Jura**[3, 4], **Dawid Rymarczyk**[1,2,6], **Łukasz Struski**[1], **Jacek Tabor**[1], **Bartosz Zieliński**[1,5]

[1]Faculty of Mathematics and Computer Science, Jagiellonian University
[2]Doctoral School of Exact and Natural Sciences, Jagiellonian University
[3] Łukasiewicz Research Network – Poznań Institute of Technology, 6 Ewarysta Estkowskiego St., 61-755, Poznań, Poland
[4] Faculty of Management and Social Communication, Jagiellonian University
[5]IDEAS NCBR
[6]Ardigen SA
mikolaj.sacha@doctoral.uj.edu.pl

Abstract

Prototypical parts-based networks are becoming increasingly popular due to their faithful self-explanations. However, their similarity maps are calculated in the penultimate network layer. Therefore, the receptive field of the prototype activation region often depends on parts of the image outside this region, which can lead to misleading interpretations. We name this undesired behavior a spatial explanation misalignment and introduce an interpretability benchmark with a set of dedicated metrics for quantifying this phenomenon. In addition, we propose a method for misalignment compensation and apply it to existing state-of-the-art models. We show the expressiveness of our benchmark and the effectiveness of the proposed compensation methodology through extensive empirical studies.

Introduction

The lack of insights into the reasons behind model predictions is a major limitation of current deep learning-based systems, particularly in high-stake decision fields like medicine and autonomous driving (Rudin 2019). As a result, eXplainable Artificial Intelligence (XAI) has gained significant attention in recent years, with two main branches of research being extensively developed: post hoc and self-explainable methods (Rudin 2019).

The post hoc approaches assume that an explainer model needs to be developed to explain the predictions of a classic deep neural network. However, this approach may be biased and unreliable (Adebayo et al. 2018a). That is why self-explainable methods were introduced, such as prototypical parts-based methods (Chen et al. 2019). They contain built-in interpretability components and provide interpretation along with the prediction.

Prototypical parts-based networks, such as ProtoPNet (Chen et al. 2019), utilize feature-matching learning theory (Rosch 1975) to identify important image parts by comparing them with reference patterns from training data. However, despite its ability to provide highly faithful explanations, this approach has known shortcomings, such as ambiguity of prototypical parts (Nauta et al. 2020) or non-resistance to image modifications, like JPEG compression (Hoffmann et al. 2021). Consequently, tools such as PRP (Gautam et al. 2023) have been introduced to improve interpretability.

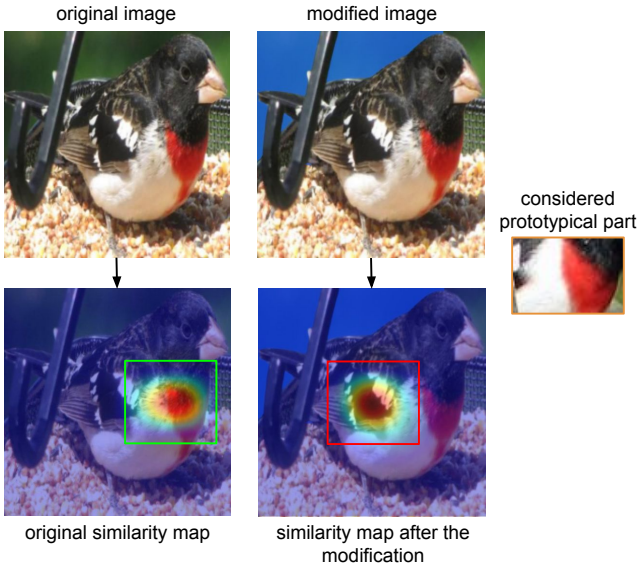

Figure 1: The receptive field of the prototypical part activation region often depends on parts of the image outside this region. In this example, the activation region (green bounding box) depends on the upper left background outside the bounding box. Therefore, after changing this part of the background (as in the modified image), the location of the activation region also changes (red bounding box). Such behavior is unwanted because it misleads explanations.

In this paper, we identify another risk related to the fact that prototypical parts similarity maps are calculated in the penultimate network layer. Therefore, the receptive field of the prototypical part activation region often depends on parts of the image outside this region. It can result in misleading explanations because users usually identify the activation region with the receptive field, while, as presented in Fig. 1,

this assumption is only sometimes fulfilled.

To assess the explanation misalignment of prototypical parts-based methods, we introduce an interpretability benchmark with a set of dedicated metrics. It adversarially modifies the input image to reduce the high activation of a prototypical part by changing only the image area where a particular prototypical part is almost inactive. We decided to use adversarial modification due to its flexibility which allows us to choose the modified pixels and strength of the modification without altering the input image too much. We use the original and modified images to compute easy-to-interpret explanation misalignment metrics.

Except for the interpretability benchmark, we propose a novel compensation methodology preserving the spatial relationship between the prototypical part activation region and its receptive field. It is based on a novel loss function computed for the image passed twice through the network, with and without a mask, and masking-based augmentation.

Finally, we provide an extensive experimental evaluation of our compensation methodology using state-of-the-art prototypical parts methods, showing the effectiveness of our benchmark and the constructed compensation method[1].

Our contributions can be summarized as follows:

- We systematize the limitations of the basic visualization of prototypical parts activation, which can lead to misleading explanations.
- We propose an interpretability benchmark for measuring the spatial misalignment of the prototypical part activation region and its receptive field to assess the reliability of explanations.
- We introduce a novel compensation methodology for this misalignment that can be used with any prototypical parts-based model.

Related Works

Explainable Artificial Intelligence Methods used to explain deep learning models can be classified into post hoc and self-explainable (Rudin 2019). Post hoc methods assume that the reasoning process is hidden within a black box model, and a new explainer model must be created to reveal it. Some of those methods generate saliency maps (Rebuffi et al. 2020; Selvaraju et al. 2019; Simonyan, Vedaldi, and Zisserman 2014) or use Concept Activation Vectors (CAV) to construct explanation with user-friendly concepts (Chen, Bei, and Rudin 2020; Ghorbani et al. 2019; Kim et al. 2018; Yeh et al. 2020). Others provide counterfactual examples (Abbasnejad et al. 2020; Niu et al. 2021) or analyze the network's reaction to image perturbations (Basaj et al. 2021; Fong, Patrick, and Vedaldi 2019; Ribeiro, Singh, and Guestrin 2016). The post hoc methods are easy to implement as they do not interfere with the architecture. However, they may produce biased and unreliable explanations (Adebayo et al. 2018b). Therefore, considerable effort has been devoted to designing self-explainable models (Alvarez Melis and Jaakkola 2018; Brendel and Bethge 2019) that make the internal decision process visible for the user. Many interpretable solutions use attention mechanisms (Liu et al. 2021; Zheng et al. 2017, 2019) or exploit the activation space (Guidotti et al. 2020; Puyol-Antón et al. 2020), such as adversarial autoencoders. However, the most recent approaches are built on an interpretable method introduced in (Chen et al. 2019) (ProtoPNet) using a hidden layer of prototypical parts to discover visual concepts.

Multiple self-explainable methods enhance ProtoPNet (Chen et al. 2019). TesNet (Wang et al. 2021) constructs the latent space on a Grassman manifold. PIP-Net (Nauta et al. 2023) redefines the prototypical parts layer to allow out-of-distribution data detection. ProtoVAE (Gautam et al. 2022) leverages a variational autoencoder with prototypical parts. ProtoPShare (Rymarczyk et al. 2021), ProtoTree (Nauta et al. 2021), ProtoKNN (Ukai et al. 2023), and ProtoPool (Rymarczyk et al. 2022) reduce the number of prototypical parts used in the classification. ProtoPShare introduces data-dependent merge-pruning that discovers prototypical parts of similar semantics and joins them. ProtoTree uses a soft neural decision tree that may depend on the negative reasoning process and is extended to a visual transformer by (Kim, Nam, and Ko 2022). ProtoPool proposes differentiable prototypical parts to class assignments while ProtoKNN adapts distance-based classifiers to prototypical parts. At the same time, more alternative approaches organize the prototypical parts hierarchically (Hase et al. 2019) to classify input at every level of a predefined taxonomy or transform prototypical parts from the latent space to data space (Li et al. 2018). Moreover, prototype-based solutions are widely adopted in various fields such as medical imaging (Afnan et al. 2021; Barnett et al. 2021; Kim et al. 2021; Rymarczyk et al. 2023a; Singh and Yow 2021), time-series analysis (Gee et al. 2019), graphs analysis (Rymarczyk, Dobrowolski, and Danel 2023; Zhang et al. 2022), semantic segmentation (Sacha et al. 2023), deepfake detection (Trinh et al. 2021), zero-shot learning (Xu et al. 2020), and continual learning (Rymarczyk et al. 2023b).

As the number of published prototypical parts-based methods grows, the community starts to contemplate the correct ways of comparing them, using not only accuracy. For example, (Hoffmann et al. 2021) investigates ProtoPNets interpretability and discovers a semantic gap between similarity in input and latent space. At the same time, (Etmann et al. 2019; Zhang and Zhu 2019; Tsipras et al. 2018) highlight connections between the explainability of machine learning models and their adversarial robustness. However, according to our knowledge, no systematic benchmark has been proposed for a comprehensive comparison of prototypical parts-based models, such as the one we propose.

Adversarial Attacks are strategies designed to identify perturbations capable of modifying the predictions of a machine-learning model. These perturbations can be extremely subtle, often imperceptible to the human eye (Balda, Behboodi, and Mathar 2020; Huang et al. 2017).

One such method is white-box attacks, representing the most straightforward approach, in which the attacker possesses complete knowledge of the model's parameters, en-

[1]Code available at: https://github.com/gmum/interpretability-benchmark

abling them to leverage gradient information for crafting adversarial examples (Goodfellow, Shlens, and Szegedy 2014).

On the other hand, black-box attacks present a significantly more challenging scenario. In these attacks, the attacker lacks information about the model's parameters and does not have access during the training stage. Consequently, gradient information cannot be utilized to create malicious examples (Bhambri et al. 2019).

Untargeted attacks are another type of adversarial attack, aiming to manipulate pixel intensities to reduce the confidence of the original class prediction until it is no longer the dominant one (Degirmenci, Ozcelik, and Yazici 2022).

The final category is targeted attacks, which aim to perturb the input that leads the model to misinterpret the input as the attacker's specified target class. These attacks seek to make the model classify the input according to the attacker's desired class (Miller, Xiang, and Kesidis 2020).

Preliminaries

Prototypical Parts Network

To make this work self-contained, in this section, we describe the ProtoPNet model (Chen et al. 2019), which introduces the prototypical parts layer. The following paragraphs include the architecture, inference, and basic visualization. We provide the ProtoPNets' training schema in Supplement.

Architecture. Prototypical parts networks (Chen et al. 2019) consist of a backbone convolutional network f, a prototypical part layer g, and a fully connected layer h. The prototypical part layer g consists of K prototypical parts $p \in \mathbb{R}^D$ per class, whose assignment is coded in the fully connected layer h. If the prototypical part p is assigned to class c, then the weight between them equals 1. Otherwise, it is set to -0.5. We will denote the set of all prototypical parts as P and the set of prototypical parts of class c as P_c.

Inference. Given an input image x, its representation $f(x)$ of shape $H \times W \times D$ is generated with a backbone f. The H and W represent height and width after the last convolutional layer, where D is its depth. Then, each prototypical part p is compared to each of $H \times W$ representation vectors $z_j \in f(x)$ to find the maximum similarity (i.e. the maximal activation of this prototypical part on the input image)

$$g_p(x) = \max_{z_j \in f(x)} sim(p, z_j), \tag{1}$$

where

$$sim(p, z_j) = \log \frac{|z_j - p|_2 + 1}{|z_j - p|_2 + \eta} \tag{2}$$

and $\eta \ll 1$. Suppose s of dimension $W \times H$ refers to the similarity map generated for the whole $f(x)$ representation

$$s = sim(p, f(x)). \tag{3}$$

In that case, the final prediction is obtained by pushing similarity values through the fully connected layer h.

Visualization. Visualization of the regions corresponding to prototypical parts is obtained with similarity maps calculated by the layer g before max pooling, upscaled from $H \times W$ to the resolution of the input image, and overlayed.

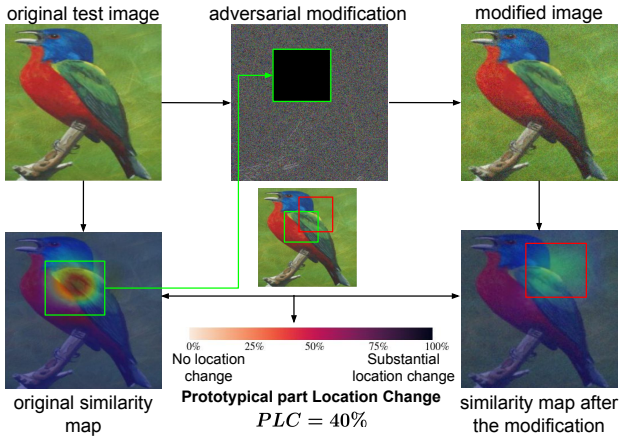

Figure 2: *Prototypical part Location Change* (PLC), similarly to the remaining metrics, is a two-step process. The first step is similar for all metrics. It calculates the similarity map for the maximally activated prototypical part and adversarially modifies the image outside the activated region (outside the green bounding box). As a result, the activation region can change (red bounding box). The second step differs between metrics. In the case of PLC, it measures the location change of activation before and after modification (difference between green and red bounding box, respectively).

To further simplify the visualization, the authors of ProtoPNet (Chen et al. 2019) take the 90th percentile of this upscaled similarity map and draw a bounding box around the highest activation values to mark the prototypical part.

Spatial Misalignment of Explanations

Definition 0.1. Let us consider a similarity map $s_i = sim(p_i, f(x))$ of the prototypical part p_i for an input image x (as defined in Eq. (3)). Moreover, let m_i be a binarized mask obtained by interpolating s_i to the input resolution and assigning a positive mask value to pixels with an activation value above the 90th percentile. We define the spatial misalignment of explanations as:

$$\Delta = \max_{\bar{x}_i} \|sim(p_i, f(x)) - sim(p_i, f(\bar{x}_i))\|, \tag{4}$$

where $\bar{x}_i$ correspond to any x modification outside mask m_i.

Observe that if $\Delta = 0$, then changes outside the mask do not impact the explanation. Therefore, they are spatially aligned. However, the higher Δ, the larger the misalignment.

Interpretability Benchmark

This section introduces a benchmark for evaluating the spatial misalignment of prototypical part explanations. It consists of three metrics that can be considered complementary to the performance metrics by the community researchers.

The central idea is to analyze differences in prototypical part activations (similarity maps) obtained for the original and adversarially modified image. Modifications are made only outside the activation region of the original image. Therefore, in the case of perfect alignment, there should be no difference between prototypical part activations obtained

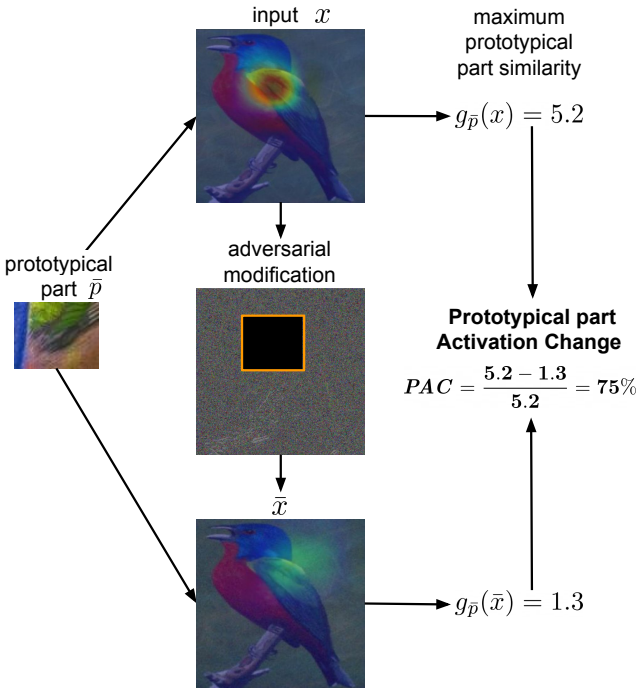

Figure 3: *Prototypical part Activation Change* (PAC) measures the relative difference between the maximum activation of the prototypical part before and after adversarial modification. In this example, the activation of the prototypical part drops substantially (by 75%), which indicates a misalignment of the prototypical part explanation.

for both images. Otherwise, there can be differences in explanation location, activation, and ranking, considered in the following metrics.

Adversarial Modification. To formally describe our benchmark, let us assume that X is a test set of images used to quantify the spatial misalignment of the model. Each $x \in X$ is passed through the backbone convolutional network f and compared to all prototypical parts using Eq. (2).

Let $\bar{p}$ be a prototypical part with the largest activation on x, i.e. $\bar{p} = \arg\max_{p \in P} g_p(x)$. We calculate the similarity map $\bar{s} = sim(\bar{p}, f(x))$ and interpolate it bilinearly to the input resolution. Then, following the visualization method from (Chen et al. 2019), we construct a bounding box $b(x)$ defined as the smallest rectangular region containing all activation above the 90th percentile (see green bounding box in Fig. 2). This region is presented to the user as the one activated by the prototypical part $\bar{p}$.

We propagate the gradient from $\bar{p}$ back to the input image using the *projected gradient descent* (PGD) method (Papernot et al. 2018) to generate an adversarially modified version of the input image $\bar{x}$ (see modified image in Fig. 2) with the goal to minimize the $g_{\bar{p}}(x)$. However, in contrast to standard PGD, we only modify the input pixels outside $b(x)$ (see adversarial modification in Fig. 2).

If PGD manages to decrease $g_{\bar{p}}(x)$, then we deal with misalignment, as it is possible to modify activations inside $b(x)$ by modifying the region outside of it. This phenomenon

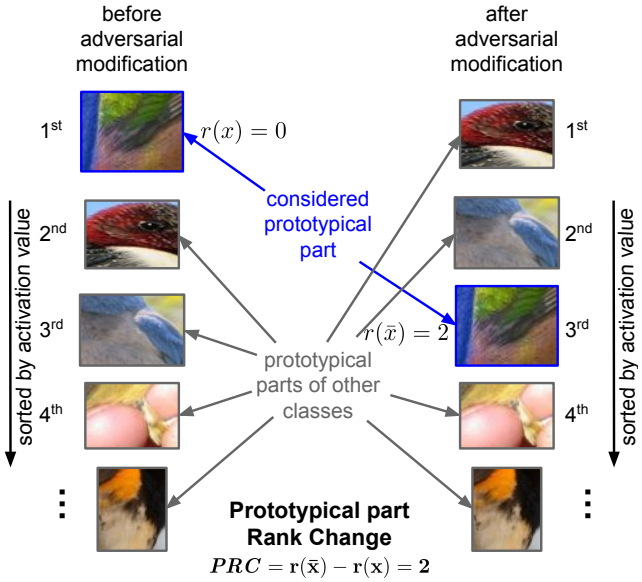

Figure 4: *Prototypical part Rank Change* (PRC) corresponds to the difference in ranking of the prototypical part activations before and after modification. The ranking calculates the number of prototypical parts from the classes other than the class of the considered prototypical part with greater maximum similarity. PRC close to 0 indicates that the explanation is spatially aligned.

is quantified using specialized *spatial misalignment metrics*, which are introduced in the following paragraph.

Spatial Misalignment Metrics. In this paragraph, we describe our metrics for evaluating the spatial misalignment of prototypical parts explanations. They all operate on the original image x, the prototypical part p_i, the adversarially modified image $\bar{x}_i$, and two similarity maps (s_i and $\bar{s}_i$).

The first metric, *Prototypical part Location Change* (PLC), corresponds to the change in the explanation location (see Fig. 2)

$$PLC = 1 - \mathbb{E}_{x \in X} \frac{|b(x) \cap b(\bar{x}_i)|}{|b(x) \cup b(\bar{x}_i)|}, \tag{5}$$

where $b(x)$ corresponds to the minimal rectangular region covering the binarized mask obtained by assigning a positive mask value to pixels with an activation value above the 90th percentile. It quantifies how much the explanation region can be relocated due to changes made by adversarial modification. $PLC = 0$ indicates that the region location remains unchanged. However, a high PLC value suggests a significant shift in the explanation location.

The second metric, *Prototypical part Activation Change (*PAC*)*, corresponds to the relative difference between the maximum activation of the prototypical part before and after adversarial modification (see Fig. 3)

$$PAC = \mathbb{E}_{x \in X} \frac{g_{p_i}(x) - g_{p_i}(\bar{x}_i)}{g_{p_i}(x)}. \tag{6}$$

It quantifies the impact of adversarial modification on prototypical part activation. $PAC = 0$ indicates no activation

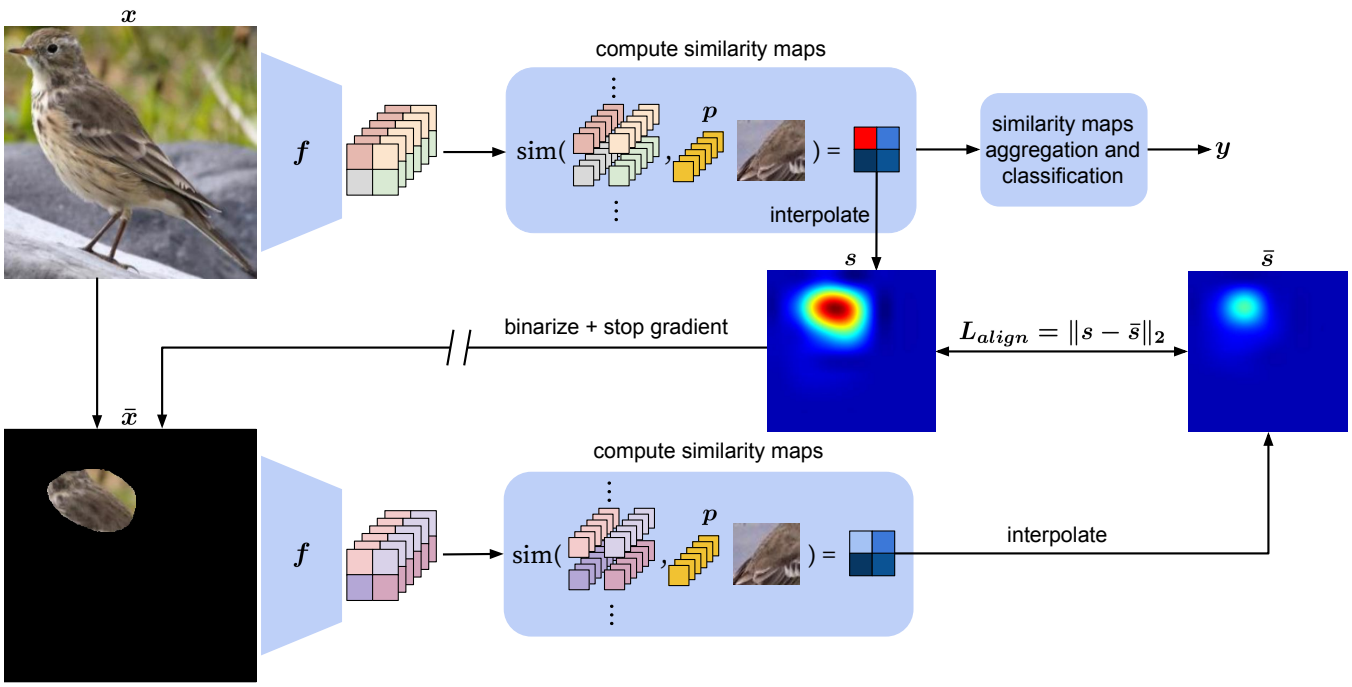

Figure 5: The spatially-aligned training aims to maximize the alignment of activation maps of a selected prototypical part between the original image and the image with only the highest activated fragment visible to the model. As shown in the picture, the training step on a single image consists of two passes of the model. During the first pass, we compute the model output together with the intermediate similarity maps to the prototypical parts. In the second pass, we randomly select a prototypical part from the ground-truth class and use its similarity map to mask the image ($\bar{x}$). This way, we obtain a new similarity map ($\bar{s}$), which we compare with the original one, obtaining loss L_{align}.

change, while higher PAC values correspond to high activation changes.

The third metric, *Prototypical part Rank Change*, corresponds to the difference in ranking of the prototypical parts activations (see Fig. 4)

$$PRC = \mathbb{E}_{x \in X}[r(\bar{x}_i) - r(x)], \tag{7}$$

where $r(x) = |\{p \in P \setminus P_k \colon g_p(\bar{x}_i) > g_{p_i}(\bar{x}_i)\}|$ calculates the number of prototypical parts p from the classes other than the ground truth class of x (here noted as class k) with maximum activation greater than this obtained by $\bar{p}$. $PRC = 0$ means that $\bar{p}$ remains the most activated prototypical part. However, the higher the PRC, the more prototypical parts from other classes become increasingly important, indirectly indicating misalignment.

As the final metric, we define the *Accuracy Change* (AC), which is equal to the difference between the accuracy obtained for the original and modified images, expressed in percentage points.

Misalignment Compensation

In this section, we propose a compensation methodology to prevent spatial misalignment of prototypical parts explanations. It is a general strategy as its only assumption is that the model calculates the prototypical parts similarity map over the full input image at some point in its pipeline. Hence, this strategy can be used with all state-of-the-art models based on

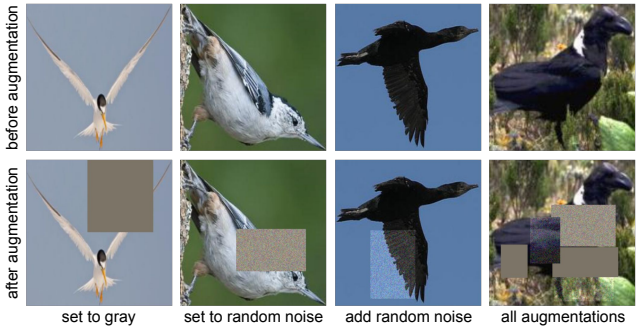

Figure 6: Sample images without (top row) and with (bottom row) the masking augmentation. Each of the left three columns presents only one type of masking, while the column on the right combines all of them. The number and types of masking augmentations are randomized. The masking augmentation steers the prototypical part model towards learning more locally-focused prototypical parts.

prototypical parts. The main idea behind our training strategy is to enforce the alignment by passing the image through the network twice: the original image and the image with the area outside the activation region masked.

To formalize our approach, let us assume that the considered prototypical part model consists of a backbone convolutional network f used to calculate the similarity map $s_i = sim(p_i, f(x))$ to a prototypical part p_i, as defined

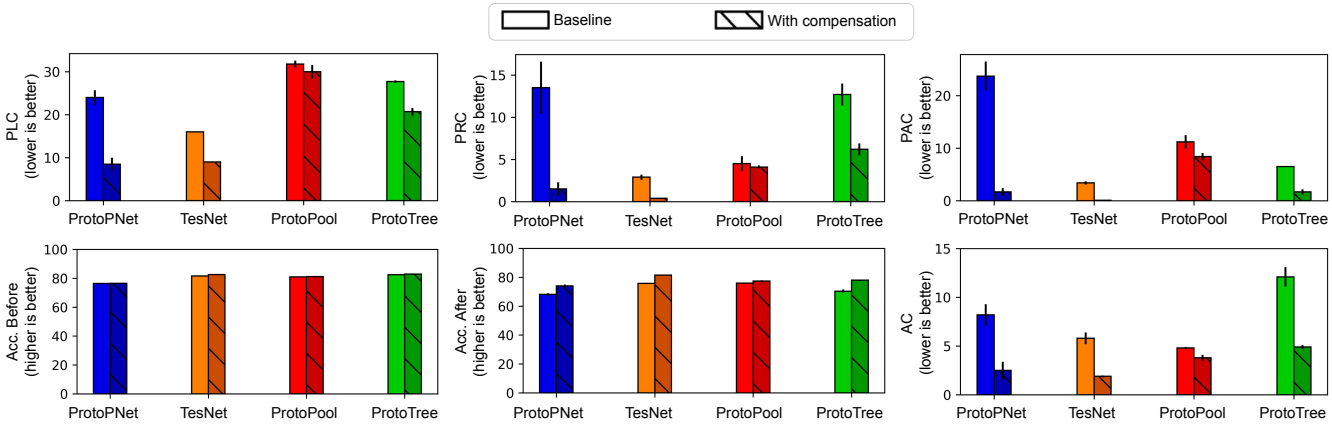

Figure 7: Comparison of the spatial misalignment metrics between baseline prototype-based models (non-hatched bars) and the best variants achieved using the spatial misalignment compensation methods (hatched bars).

in Eq. (3). Similar to the standard prototypical parts-based approaches, the similarity maps are aggregated and classified, as presented in Fig. 5. Additionally, we interpolate bilinearly the detached s_i to the input resolution and binarize it so that positive values correspond to activation value above the 90th percentile, obtaining m_i. This mask is used to generate a masked input $\bar{x_i} = x \cdot m_i$ that is used in the second pass of the model. By passing $\bar{x_i}$ trough f and comparing it to the prototypical part p_i, we obtain a new similarity map $\bar{s_i} = sim(\bar{p_i}, f(x))$.

For aligned explanation, $\bar{s_i}$ should be very similar to s_i because the area outside the activation region should not influence the final results. Therefore, we introduce a spatial alignment loss function, which penalizes the model for not fulfilling this condition

$$L_{align} = \|s_i - \bar{s_i}\|_2, \tag{8}$$

We weight this loss component with λ_{align}.

Masking Augmentation. To further prevent explanation misalignment, we consider a special type of augmentation during model training (Fig. 6). For every training image, we apply masking with a given probability. For the modified samples, we randomly select a number of rectangular regions together with their widths, heights, and locations. We randomly modify each region, either by adding noise or by replacing the region with gray color or random noise.

Experimental Setup

In this section, we discuss the experimental setup of our spatial misalignment compensation. We describe other details related to training the compared models in Supplement.

Masking Augmentation Setup. For the variants of the trained models that employ masking augmentation, we apply it during all phases of the training, augmenting each sampled training image with the probability of 50%. We sample the number of modified image regions between 1 and 6. For each region, we sample the augmentation type out of the three options. The width and the height of each region are randomly selected between 0.1 and 0.5 of the image width and height, respectively. The location of each region is randomly selected from each possible location that is fully within the image. All random values are sampled from the uniform probability on the respective intervals.

Spatial Misalignment Benchmark Setup. To evaluate the spatial misalignment of the tested models, we perform the spatial misalignment test on each image from the test set of CUB-200-2011 dataset (Wah et al. 2011). For each image, we select the top activated prototypical part in the image for a given model and modify it adversarially according to the procedure described in **??** . We use the following parameters for the *projected gradient descent* function used within the benchmark: maximum total perturbation: 0.4; maximum perturbation within one iteration: 0.01; number of iterations: 40.

What is the impact of using the bounding boxes instead of the mask? As the heatmap-based mask covers a smaller area, it makes the adversarial modification stronger, resulting in higher values for the proposed metrics. We have chosen bounding boxes because they are commonly used in explanation visualizations for Prototypical Parts models.

Results

What is the level of the spatial misalignment for vanilla models and how it can be decreased with our compensation method? Fig. 7 illustrates the values for spatial misalignment metrics and classification accuracy (in percents) achieved by the ProtoPNet, TesNet, ProtoPool, and ProtoTree models, when trained with and without spatial-misalignment compensation on the CUB-200-2011 dataset. We show the results for baseline models and the best variants achieved with our compensation methods. The implementation of our training technique yields a notable enhancement in the robustness of the explanations, as gauged by the proposed metrics, as well as increased stability in prediction accuracy across all tested prototypical parts-based models. More details are in the Supplement.

Relative improvements, relative to each model's baseline, exhibit varying degrees of prominence. Notably, the highest

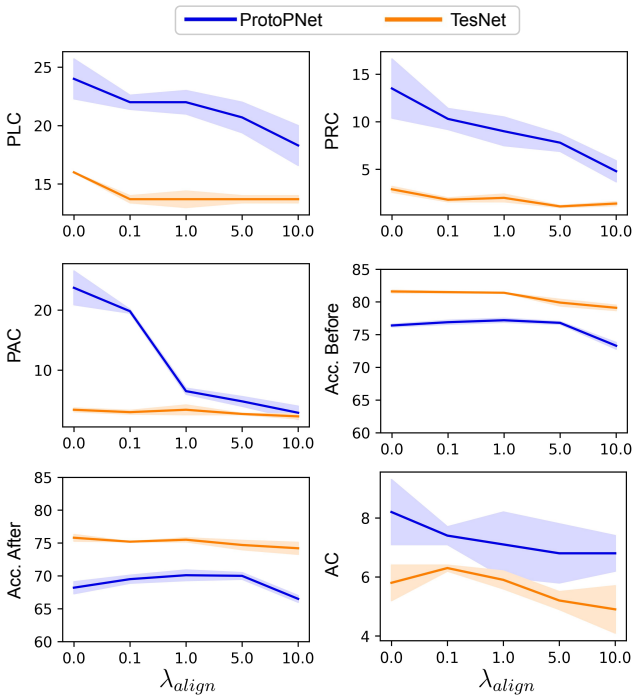

Figure 8: Ablation on the value of λ_{align} for ProtoPNet and TesNet models, trained without the masking augmentation.

gain is observed for ProtoPNet, whereas ProtoPool demonstrates the least one. This can be related to the ProtoPool's specific focal similarity function, which is designed to generate salient explanations. Comparatively, the enhancements for ProtoTree align more closely with those observed for ProtoPNet which correlates with the findings of (Nauta et al. 2023) emphasizing ProtoTree's limitations in capturing atomic parts of objects as prototypical parts. In the case of TesNet, its basic version presents robust interpretations, particularly evident when considering the PAC metric. A comprehensive analysis of TesNet's robustness is provided in the subsequent paragraphs.

Furthermore, ProtoPNet's explanations exhibit the highest susceptibility to spatial misalignment (with the exception of the PLC metric), while TesNet's interpretations show the least vulnerability to misalignment.

In terms of computations, masking augmentation has minimal impact on model training. On the other hand, computing the spatial alignment loss necessitates an additional pass, leading to an average 40% increase in training time. Potentially, calculating the loss using only a subset of the training dataset or specific image subregions may address it.

How do the explanations differ between the baselines and improved models? In Fig. 10, we show the results of the spatial misalignment benchmark for the baseline ProtoPNet model, as well as for the ProtoPNet model trained with $\lambda_{align} = 10$ with and without the masking augmentation technique. The examples were selected at random from the test set of the CUB-200-2011 dataset. We observe that the activation maps of the baseline model are diminished by the test, while the model trained with the spatial-aligning loss is

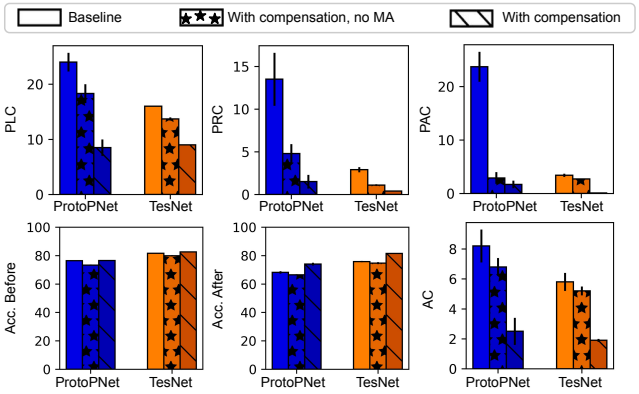

Figure 9: Ablation on the spatial-alignment compensation without and with the masking augmentation technique.

robust to the modification of the image area outside the high-activation bounding box. Better spatial alignment of our improved models is also indicated by the results of the metrics shown below the images in the figure. We present more such examples in the Supplementary Materials.

What is the optimal weight for spatial alignment loss? To investigate the optimal value of the weighting factor for spatial alignment loss, we trained the models varying the value of $\lambda_{align} \in \{0.0, 0.1, 1, 5, 10\}$. For this ablation, we turned off the masking augmentation technique. The results presented in Fig. 8 show that a larger value of the λ_{align} weight allows obtaining more spatially-aligned explanations, as evidenced by the decreasing metric's values, with observable improvements for large weights ($\lambda_{align} \geq 1$). We provide more detailed results in the Supplement.

What is the gain from using masking augmentation? Fig. 9 shows how training with and without masking augmentation (MA) influences the spatial misalignment metrics and the model's accuracy. We observe that, while applying the compensating loss improves the values of metrics, the additional usage of masking augmentation combined with the compensating loss yields the best results. We provide more detailed results in Supplementary Materials.

Does the spatial misalignment compensation generalize to other model backbones and datasets? To evaluate the generalization of our approach, we conduct experiments using the ProtoPNet (Chen et al. 2019) and TesNet (Wang et al. 2021) models with VGG16 (Simonyan, Vedaldi, and Zisserman 2014) backbone (instead of ResNet), as well as we benchmark the approach using the Stanford Cars dataset (Krause et al. 2013). With the VGG16 backbone, both models show enhanced spatial misalignment metrics, likely attributed to the narrower receptive field of VGG. Moreover, similar trends for spatial misalignment are observed on the Stanford Cars dataset, mirroring the behavior on CUB. More comprehensive results are in the Supplement.

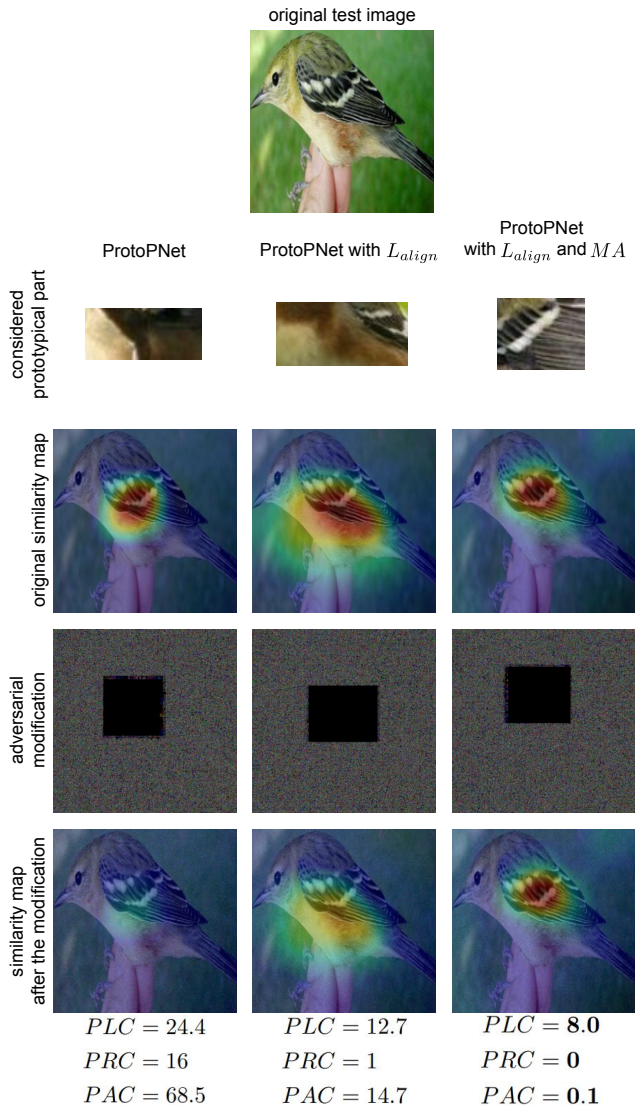

Figure 10: Comparison of the spatial misalignment benchmark results on the baseline ProtoPNet model and the variants with the spatial-aligning loss ($\lambda = 10$) and with the spatial-aligning loss enhanced by the additional masking augmentation technique. Our variants achieve better consistency between the similarity map before and after modifying the image, indicating better spatial alignment.

Why is TesNet so robust? In order to investigate what makes TesNet so robust to the adversarial modifications of our benchmark, we trained it without its specific loss terms and applied the benchmark. Specifically, we trained TesNet 1) without the subspace orthogonality loss (λ_{orth}), and 2) without both the λ_{orth} and the subspace-separation loss (λ_{ss}). Results of these experiments are provided in Supplementary Materials. Classification accuracy as well as metrics for these two models are comparable to those obtained for the baseline TesNet model. PAC metric tends even to be slightly better than for the baseline TesNet model. These results might suggest that it is the prototype similarity function used by TesNet, i.e., projection of the latent space patches onto the prototype vectors (instead of a function of L^2-distance as used by ProtoPNet), that is primarily responsible for the superior performance of TesNet in our benchmark, as compared to results obtained by the ProtoPNet model.

Conclusions

In this article, we discuss the limitations of prototypical parts-based methods, such as ProtoPNet, caused by the misalignment between input and representation space. To address this issue, we propose an interpretability benchmark that measures this misalignment and introduce a novel compensation methodology. Experimental evaluations show the adequacy of the proposed benchmark and the effectiveness of the compensation methodology. With the proposed spatial misalignment benchmark, we can automatically assess the accuracy of explanations before presenting them to the user, thus avoiding potential misinformation.

We hope this benchmark will improve the faithfulness of visualizations generated by the prototypical parts-based models and strengthen research dedicated to the automatic assessment of models' interpretability.

Acknowledgements

This research was partially funded by the National Science Centre, Poland, grants no. 2021/41/B/ST6/01370 (work by Mikołaj Sacha and Jacek Tabor), 2022/45/N/ST6/04147 (work by Dawid Rymarczyk), 2020/39/D/ST6/01332 (work by Łukasz Struski), and 2022/47/B/ST6/03397 (work by Bartosz Zieliński). The research of Bartosz Jura was carried out within the research project "Bio-inspired artificial neural network" (grant no. POIR.04.04.00-00-14DE/18-00) within the Team-Net program of the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund. This paper has been also supported by the Horizon Europe Programme (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237. Moreover, Dawid Rymarczyk received an incentive scholarship from the funds of the program Excellence Initiative – Research University at the Jagiellonian University in Kraków. Finally, some experiments were performed on servers purchased with funds from a Priority Research Area (Artificial Intelligence Computing Center Core Facility) grant under the Strategic Programme Excellence Initiative at Jagiellonian University.

References

Abbasnejad, E.; Teney, D.; Parvaneh, A.; Shi, J.; and Hengel, A. v. d. 2020. Counterfactual vision and language learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10044–10054.

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018a. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31.

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018b. Sanity Checks for Saliency Maps. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Afnan, M. A. M.; Liu, Y.; Conitzer, V.; Rudin, C.; Mishra, A.; Savulescu, J.; and Afnan, M. 2021. Interpretable, not black-box, artificial intelligence should be used for embryo selection. *Human Reproduction Open*.

Alvarez Melis, D.; and Jaakkola, T. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Balda, E. R.; Behboodi, A.; and Mathar, R. 2020. Adversarial examples in deep neural networks: An overview. *Deep learning: algorithms and applications*, 31–65.

Barnett, A. J.; Schwartz, F. R.; Tao, C.; Chen, C.; Ren, Y.; Lo, J. Y.; and Rudin, C. 2021. IAIA-BL: A Case-based Interpretable Deep Learning Model for Classification of Mass Lesions in Digital Mammography. *arXiv preprint arXiv:2103.12308*.

Basaj, D.; Oleszkiewicz, W.; Sieradzki, I.; Górszczak, M.; Rychalska, B.; Trzcinski, T.; and Zielinski, B. 2021. Explaining Self-Supervised Image Representations with Visual Probing. In *International Joint Conference on Artificial Intelligence*.

Bhambri, S.; Muku, S.; Tulasi, A.; and Buduru, A. B. 2019. A survey of black-box adversarial attacks on computer vision models. *arXiv preprint arXiv:1912.01667*.

Brendel, W.; and Bethge, M. 2019. Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. In *International Conference on Learning Representations*.

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. In *NeurIPS*, 8930–8941.

Chen, Z.; Bei, Y.; and Rudin, C. 2020. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12): 772–782.

Degirmenci, E.; Ozcelik, I.; and Yazici, A. 2022. Effects of Un targeted Adversarial Attacks on Deep Learning Methods. In *2022 15th International Conference on Information Security and Cryptography (ISCTURKEY)*, 8–12. IEEE.

Etmann, C.; Lunz, S.; Maass, P.; and Schönlieb, C.-B. 2019. On the connection between adversarial robustness and saliency map interpretability. *arXiv preprint arXiv:1905.04172*.

Fong, R.; Patrick, M.; and Vedaldi, A. 2019. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2950–2958.

Gautam, S.; Boubekki, A.; Hansen, S.; Salahuddin, S. A.; Jenssen, R.; Höhne, M. M.; and Kampffmeyer, M. 2022. ProtoVAE: A Trustworthy Self-Explainable Prototypical Variational Model. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Gautam, S.; Höhne, M. M.-C.; Hansen, S.; Jenssen, R.; and Kampffmeyer, M. 2023. This looks more like that: Enhancing self-explaining models by prototypical relevance propagation. *Pattern Recognition*, 136: 109172.

Gee, A. H.; Garcia-Olano, D.; Ghosh, J.; and Paydarfar, D. 2019. Explaining deep classification of time-series data with learned prototypes. In *CEUR workshop proceedings*, volume 2429, 15. NIH Public Access.

Ghorbani, A.; Wexler, J.; Zou, J. Y.; and Kim, B. 2019. Towards Automatic Concept-based Explanations. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Guidotti, R.; Monreale, A.; Matwin, S.; and Pedreschi, D. 2020. Explaining Image Classifiers Generating Exemplars and Counter-Exemplars from Latent Representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09): 13665–13668.

Hase, P.; Chen, C.; Li, O.; and Rudin, C. 2019. Interpretable image recognition with hierarchical prototypes. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, 32–40.

Hoffmann, A.; Fanconi, C.; Rade, R.; and Kohler, J. 2021. This looks like that... does it? shortcomings of latent space prototype interpretability in deep networks. *arXiv preprint arXiv:2105.02968*.

Huang, S.; Papernot, N.; Goodfellow, I.; Duan, Y.; and Abbeel, P. 2017. Adversarial attacks on neural network policies. *arXiv*.

Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; Viegas, F.; et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, 2668–2677. PMLR.

Kim, E.; Kim, S.; Seo, M.; and Yoon, S. 2021. XProtoNet: Diagnosis in Chest Radiography with Global and Local Explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15719–15728.

Kim, S.; Nam, J.; and Ko, B. C. 2022. ViT-NeT: Interpretable Vision Transformers with Neural Tree Decoder. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 11162–11172. PMLR.

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.

Li, O.; Liu, H.; Chen, C.; and Rudin, C. 2018. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Liu, N.; Zhang, N.; Wan, K.; Shao, L.; and Han, J. 2021. Visual saliency transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4722–4732.

Miller, D. J.; Xiang, Z.; and Kesidis, G. 2020. Adversarial learning targeting deep neural network classification: A comprehensive review of defenses against attacks. *Proceedings of the IEEE*, 108(3): 402–433.

Nauta, M.; Jutte, A.; Provoost, J.; and Seifert, C. 2020. This Looks Like That, Because... Explaining Prototypes for Interpretable Image Recognition. *arXiv preprint arXiv:2011.02863*.

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*, 2744–2753.

Nauta, M.; et al. 2021. Neural Prototype Trees for Interpretable Fine-grained Image Recognition. In *CVPR*, 14933–14943.

Niu, Y.; Tang, K.; Zhang, H.; Lu, Z.; Hua, X.-S.; and Wen, J.-R. 2021. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12700–12710.

Papernot, N.; Faghri, F.; Carlini, N.; Goodfellow, I.; Feinman, R.; Kurakin, A.; Xie, C.; Sharma, Y.; Brown, T.; Roy, A.; Matyasko, A.; Behzadan, V.; Hambardzumyan, K.; Zhang, Z.; Juang, Y.-L.; Li, Z.; Sheatsley, R.; Garg, A.; Uesato, J.; Gierke, W.; Dong, Y.; Berthelot, D.; Hendricks, P.; Rauber, J.; and Long, R. 2018. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv preprint arXiv:1610.00768*.

Puyol-Antón, E.; Chen, C.; Clough, J. R.; Ruijsink, B.; Sidhu, B. S.; Gould, J.; Porter, B.; Elliott, M.; Mehta, V.; Rueckert, D.; et al. 2020. Interpretable deep models for cardiac resynchronisation therapy response prediction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 284–293. Springer.

Rebuffi, S.-A.; Fong, R.; Ji, X.; and Vedaldi, A. 2020. There and back again: Revisiting backpropagation saliency methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8839–8848.

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.

Rosch, E. 1975. Cognitive representations of semantic categories. *Journal of experimental psychology: General*, 104(3): 192.

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

Rymarczyk, D.; Dobrowolski, D.; and Danel, T. 2023. ProGReST: Prototypical Graph Regression Soft Trees for Molecular Property Prediction. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, 379–387. SIAM.

Rymarczyk, D.; Pardyl, A.; Kraus, J.; Kaczyńska, A.; Skomorowski, M.; and Zieliński, B. 2023a. ProtoMIL: Multiple Instance Learning with Prototypical Parts for Whole-Slide Image Classification. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part I*, 421–436. Springer.

Rymarczyk, D.; Struski, Ł.; 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. Springer.

Rymarczyk, D.; van de Weijer, J.; Zieliński, B.; and Twardowski, B. 2023b. ICICLE: Interpretable Class Incremental Continual Learning. *International Conference on Computer Vision*.

Rymarczyk, D.; et al. 2021. ProtoPShare: Prototypical Parts Sharing for Similarity Discovery in Interpretable Image Classification. In *SIGKDD*, 1420–1430.

Sacha, M.; Rymarczyk, D.; Struski, Ł.; Tabor, J.; and Zieliński, B. 2023. ProtoSeg: Interpretable Semantic Segmentation With Prototypical Parts. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1481–1492.

Selvaraju, R. R.; Lee, S.; Shen, Y.; Jin, H.; Ghosh, S.; Heck, L.; Batra, D.; and Parikh, D. 2019. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2591–2600.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer.

Singh, G.; and Yow, K.-C. 2021. These do not Look Like Those: An Interpretable Deep Learning Model for Image Recognition. *IEEE Access*, 9: 41482–41493.

Trinh, L.; Tsang, M.; Rambhatla, S.; and Liu, Y. 2021. Interpretable and Trustworthy Deepfake Detection via Dynamic Prototypes. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1972–1982.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2018. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*.

Ukai, Y.; Hirakawa, T.; Yamashita, T.; and Fujiyoshi, H. 2023. This Looks Like It Rather Than That: ProtoKNN For Similarity-Based Classifiers. In *The Eleventh International Conference on Learning Representations*.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset.

Wang, J.; et al. 2021. Interpretable Image Recognition by Constructing Transparent Embedding Space. In *ICCV*, 895–904.

Xu, W.; Xian, Y.; Wang, J.; Schiele, B.; and Akata, Z. 2020. Attribute Prototype Network for Zero-Shot Learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

Yeh, C.-K.; Kim, B.; Arik, S.; Li, C.-L.; Pfister, T.; and Ravikumar, P. 2020. On Completeness-aware Concept-Based Explanations in Deep Neural Networks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 20554–20565. Curran Associates, Inc.

Zhang, T.; and Zhu, Z. 2019. Interpreting adversarially trained convolutional neural networks. In *International conference on machine learning*, 7502–7511. PMLR.

Zhang, Z.; Liu, Q.; Wang, H.; Lu, C.; and Lee, C. 2022. ProtGNN: Towards Self-Explaining Graph Neural Networks.

Zheng, H.; Fu, J.; Mei, T.; and Luo, J. 2017. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE international conference on computer vision*, 5209–5217.

Zheng, H.; Fu, J.; Zha, Z.-J.; and Luo, J. 2019. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5012–5021.