MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Pixel-Grounded Prototypical Part Networks

Carmichael, Zachariah; Jones, Lohit, Suhas; Cherian, Anoop; Michael J.; Scheirer, Walter

TR2024-002 January 05, 2024

Abstract

Prototypical part neural networks (ProtoPartNNs), namely PROTOPNET and its derivatives, are an intrinsically interpretable approach to machine learning. Their prototype learning scheme enables intuitive explanations of the form, this (prototype) looks like that (testing image patch). But, does this actually look like that? In this work, we delve into why object part localization and associated heat maps in past work are misleading. Rather than localizing to object parts, existing ProtoPartNNs localize to the entire image, contrary to generated explanatory visualizations. We argue that detraction from these underlying issues is due to the alluring nature of visualizations and an over-reliance on intuition. To alleviate these issues, we devise new receptive field-based architectural constraints for meaning- ful localization and a principled pixel space mapping for ProtoPartNNs. To improve interpretability, we propose ad- ditional architectural improvements, including a simplified classification head. We also make additional corrections to PROTOPNET and its derivatives, such as the use of a validation set, rather than a test set, to evaluate generalization during training. Our approach, PIXPNET (Pixel-grounded Prototypical part Network), is the only ProtoPartNN that truly learns and localizes to prototypical object parts. We demonstrate that PIXPNET achieves quantifiably improved interpretability without sacrificing accuracy.

IEEE Winter Conference on Applications of Computer Vision (WACV) 2024

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Pixel-Grounded Prototypical Part Networks

Zachariah Carmichael[1,2] Suhas Lohit[2] Anoop Cherian[2] Michael J. Jones[2] Walter J. Scheirer[1]
[1]University of Notre Dame
[2]Mitsubishi Electric Research Laboratories
zcarmich@nd.edu, {slohit,anoop.cherian,mjones}@merl.com, walter.scheirer@nd.edu

Abstract

Prototypical part neural networks (ProtoPartNNs), namely ProtoPNet *and its derivatives, are an intrinsically interpretable approach to machine learning. Their prototype learning scheme enables intuitive explanations of the form,* this *(prototype) looks like* that *(testing image patch). But, does* this *actually look like* that*? In this work, we delve into why object part localization and associated heat maps in past work are misleading. Rather than localizing to object parts, existing ProtoPartNNs localize to the entire image, contrary to generated explanatory visualizations. We argue that detraction from these underlying issues is due to the alluring nature of visualizations and an over-reliance on intuition. To alleviate these issues, we devise new receptive field-based architectural constraints for meaningful localization and a principled pixel space mapping for ProtoPartNNs. To improve interpretability, we propose additional architectural improvements, including a simplified classification head. We also make additional corrections to* ProtoPNet *and its derivatives, such as the use of a validation set, rather than a test set, to evaluate generalization during training. Our approach,* PixPNet *(Pixel-grounded Prototypical part Network), is the* ***only*** *ProtoPartNN that truly learns and localizes to prototypical object parts. We demonstrate that* PixPNet *achieves quantifiably improved interpretability without sacrificing accuracy.*

1. Introduction

Prototypical part neural networks (ProtoPartNNs) are an attempt to remedy the inscrutability and fundamental lack of trustworthiness characteristic of canonical deep neural networks [11]. By learning prototypes of object parts, ProtoPartNNs make human-interpretable predictions with justifications of the form: *this* (training image patch) looks like *that* (testing image patch). Since black-box AI systems often obfuscate their deficiencies [26, 44, 66], ProtoPartNNs represent a shift in the direction of transparency. With unprecedented interest in AI from decision-makers in high-stakes industries – *e.g*., medicine, finance, and law [44, 47, 54, 67] – the demand for explainable AI systems is greater than ever. Further motivation for transparency is driven by real-world consequences of deployed black boxes [6, 46, 52] and mounting regulatory ordinance [19, 20, 43, 69].

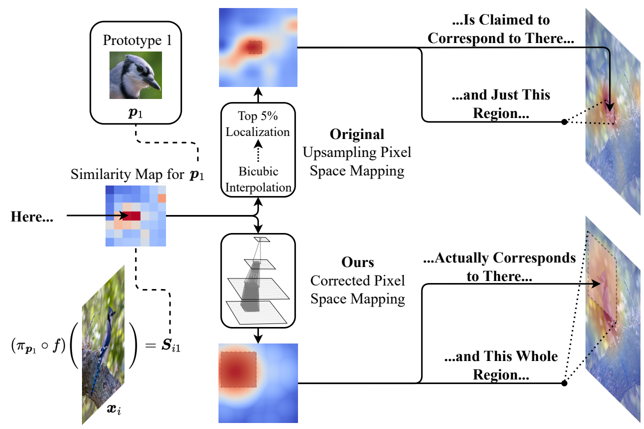


Figure 1. The two primary issues identified with prototype visualization: *here* (this embedded patch) does not correspond to *there* (this image patch), and *this* (prototype) does not correspond to *just that* (test image patch). In the extreme case, *this* can actually correspond to *the entire image* (*i.e*., when the receptive field is 100%).

ProtoPartNNs approach explainability from an intrinsically interpretable lens and offer many benefits over post hoc explanation. Whereas post hoc explainers estimate an explanation, ProtoPartNN explanations are part of the actual prediction process – explanations along the lines of "*this* looks like *that*" follow naturally from the symbolic form of the model itself. This implicit explanation is characteristic of models widely considered to be human-comprehensible [53]. Moreover, ProtoPartNNs enable concept-level debugging, human-in-the-loop learning, and implicit localization [11, 48, 50]. Being independent of the explained model, post hoc explainers have been found to be unfaithful, inconsistent, and unreliable [8, 32, 41, 64] (see Section 2 for expanded discussion).

When misunderstood or used inappropriately, explainable AI (XAI) methods can have unintended consequences [33,41]. This harm arises from unverified hypotheses, whether it is that explanations represent phenomena faithful to the predictor or meaningful properties of the predictor. So, why do we see such hypotheses proliferating throughout both academia and industry [33, 42]? The problem is very human – there is often an over-reliance on intuition that may lead to illusory progress or deceptive conclusions. Whether it is dependence on alluring visualization or behavioral extrapolation from cherry-picked examples, XAI methods often are left insufficiently scrutinized and subject to "researcher degrees of freedom" [42, 60].

Recent evidence indicates that ProtoPartNNs may suffer from these same issues: PROTOPNET and its variants exhibit irrelevant prototypes, a human-machine semantic similarity gap, and exorbitant explanation size [28, 37, 63]. Unfortunately, in our study, we confirm that this is the case – there are several facets of existing ProtoPartNN explanations that do *not* result from the implicit form of the model: image part localization, pixel space grounding, and heat map visualizations. Instead, these are founded on unverified assumptions and an over-reliance on intuition, often justified *a posteriori* by attractive visuals. We demonstrate that, colloquially, *this* does not actually look like *that*, and *here* may not actually correspond to *there* – see Figure 1 for illustration. These issues with ProtoPartNNs are not limited to just PROTOPNET, but to all of its derivatives.

This work aims to elevate the interpretability of ProtoPartNNs by rectifying these facets. In doing so, all aspects of ProtoPartNN explanations are embedded in the symbolic form of the model. Our contributions are as follows:

- We identify that existing ProtoPartNNs based on PROTOPNET do not localize faithfully nor actually localize to image parts, but rather the full image in most cases.
- We propose a novel pixel space mapping based on the receptive fields of an architecture (we guarantee that *here* corresponds to *there*).
- We propose architectural constraints that we efficiently discover through a transfer task to enable true image part localization (*this* looks like *that*).
- We devise a novel functional algorithm for the receptive field calculation of any architecture.
- On several image classification tasks, our approach, PIXPNET, achieves competitive accuracy with other ProtoPartNNs *while maintaining a higher degree of interpretability*, as substantiated by functionally grounded XAI metrics, and being the *only ProtoPartNN that truly localizes to image parts*.

2. Background

In this section, we give a brief background of explainable AI methods, the PROTOPNET formulation, and an overview of PROTOPNET extensions.

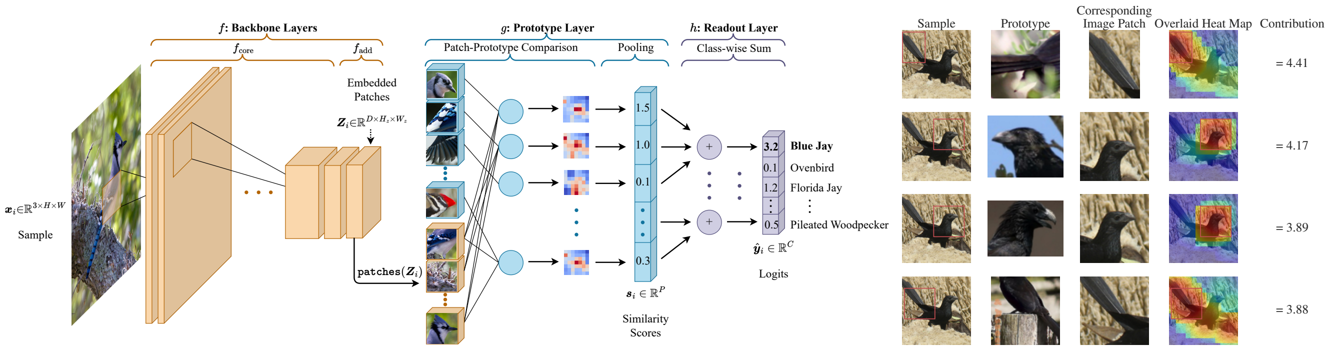
Explainable AI Methods Explainable AI (XAI) solutions can be classified as post hoc, intrinsically interpretable, or a hybrid of the two [59]. Whereas intrinsically interpretable methods are both the explanator and predictor, post hoc methods act as an explanator for an independent predictor. Unfortunately, post hoc explainers are known to be inconsistent, unfaithful, and possibly even intractable [5, 8, 12, 22, 41]. Furthermore, they are deceivable [3, 14, 15, 64] and have been shown to not affect, or even reduce, end-user task performance [31, 32]. While this is the case, post hoc explanations have been shown to possibly increase user trust in AI systems [10], improve end-user performance for some explanation types and tasks [31], and explain black boxes in trustless auditing schemes [9]. However, for high-stakes domains, post hoc explanation is frequently argued to be especially inappropriate [54].

For these numerous reasons, our work concerns intrinsically interpretable machine learning solutions (see [59] for a methodological overview). In particular, we are interested in *prototypical part neural networks* (ProtoPartNNs) [11].

PROTOPNET Architecture Here, we go over the PROTOPNET architecture [11], a type of ProtoPartNN. As much of the formalism overlaps with our approach, Figure 2 can be referred to for visualization of the architecture. Let $\mathbf{D}=\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$ be the data set where each sample $\boldsymbol{x}_i \in \mathbb{R}^{3\times H\times W}$ is an image with a height of H and a width of W, and each label $y_i \in \{1, \ldots, C\}$ represents one of C classes.

A PROTOPNET comprises a neural network backbone responsible for embedding an image. The first component of the backbone is the core f_{core}, which could be a RESNET [25], VGG [61], or DENSENET [29] as in [11]. Proceeding, there are the add-on layers f_{add} that are responsible for changing the number of channels in the output of f_{core}. In PROTOPNET, f_{add} comprises two 1×1 convolutional layers with ReLU and sigmoid activation functions for the first and second layers, respectively. The full feature embedding function is denoted by $f = f_{\text{add}} \circ f_{\text{core}}$. This function gives us our *embedded patches* $f(\boldsymbol{x}_i) = \boldsymbol{Z}_i \in \mathbb{R}^{D\times H_z\times W_z}$ which have D channels, a height of H_z, and a width of W_z.

In PROTOPNET, we are interested in finding the most similar embedded patch $\boldsymbol{z}$ for each prototype. Each prototype can be understood as the embedding of some prototypical part of an object, such as the head of a blue jay as in Figure 2. Each embedded patch can be thought of in the same way – ultimately, a well-trained network will find that the most similar embedded patch and prototype will both be, *e.g.*, the head of a blue jay (*this* prototype looks like *that* embedded patch). This is accomplished using the prototype layer, g. We use the notation $g_{\boldsymbol{p}_j}$ to denote the unit that computes the most similar patch $\boldsymbol{z} \in \texttt{patches}(\boldsymbol{Z}_i)$ to prototype $\boldsymbol{p}_j$. The function $\texttt{patches}(\boldsymbol{Z}_i)$ yields a set of $D \times H_p \times W_p$ embedded patches in a sliding window manner ($H_p = W_p = 1$



(a) PIXPNET (Pixel-grounded Prototypical part Network) architecture.

(b) An example of an explanation.

Figure 2. (a) Our proposed architecture, PIXPNET. (b) An example of an explanation with PIXPNET for a Groove-billed Ani. The following are important deviations from PROTOPNET: the backbone f receptive field is constrained, the readout layer h is simplified, both prototypes and embedded patches truly localize to image parts, and the pixel space mapping is corrected (see Figure 3).

in PROTOPNET). First, the pairwise distances between $\texttt{patches}(\boldsymbol{Z}_i)$ and prototypes $\boldsymbol{P}=\{\boldsymbol{p}_j\}_{j=1}^{P}$ are computed using a distance function φ where $\boldsymbol{p}_j \in \mathbb{R}^{D \times H_p \times W_p}$, H_p is the prototype kernel height, W_p is the prototype kernel width, and P is the total number of prototypes. Each prototype is class-specific and we denote the set of prototypes belonging to class y_i as $\boldsymbol{P}_{y_i} \subseteq \boldsymbol{P}$. Subsequently, a min-pooling operation is performed to obtain the closest embedded patch for each prototype – each prototype (*this*) is "assigned" a single embedded patch (*that*). Finally, the distances are converted into similarity scores using a similarity function v. Putting this process altogether for unit $g_{\boldsymbol{p}_j}$, we have

$$g_{\boldsymbol{p}_j}(\boldsymbol{Z}_i) = v\Big(\min_{\boldsymbol{z} \in \texttt{patches}(\boldsymbol{Z}_i)} \varphi(\boldsymbol{z}, \boldsymbol{p}_j)\Big). \tag{1}$$

We denote the vector of all similarity scores for a sample as $\boldsymbol{s}_i = g(\boldsymbol{Z}_i) \in \mathbb{R}^P$.

The architecture ends with a readout layer h that produces the logits as $\hat{\boldsymbol{y}}_i = h(\boldsymbol{s}_i)$. In PROTOPNET, h is a fully-connected layer with positive weights to same-class prototype units and negative weights to non-class prototype units. Each logit can be interpreted as the sum of similarity scores weighed by their importance to the class of the logit. Note that this readout layer is not reflected in Figure 2. The full PROTOPNET output for a sample is given by $(h \circ g \circ f)(\boldsymbol{x}_i)$.

ProtoPartNN Desiderata and PROTOPNET Variants Many extensions of PROTOPNET have been proposed, some of which make altercations that fundamentally affect the interpretability of the architecture. To differentiate these extensions, we propose a set of desiderata for ProtoPartNNs:

1. *Prototypes must correspond directly to image patches.* This can be accomplished via prototype replacement, which grounds prototypes in human-interpretable pixel space (see Section 4 for details).
2. *Prototypes must localize to image parts.*
3. *Case-based reasoning must be describable by linear or simple tree models.*

Architectures that satisfy all three desiderata are considered to be *3-way ProtoPartNNs* – satisfying fewer diminishes the interpretability of the algorithm.

The idea of sharing prototypes between classes has been explored in PROTOPSHARE [56] (prototype merge-pruning) and PROTOPOOL [55] (differential prototype assignment). In PROTOTREE [51], the classification head is replaced by a differentiable tree, also with shared prototypes. An alternative embedding space is explored in TESNET [72] based on Grassmann manifolds. A ProtoPartNN-specific knowledge distillation approach is proposed in PROTO2PROTO [34] by enforcing that student prototypes and embeddings should be close to those of the teacher. DEFORMABLE PROTOPNET [17] extends the PROTOPNET architecture with deformable prototypes. ST-PROTOPNET [71] learns support prototypes that lie near the classification boundary and trivial prototypes that are far from the classification boundary.

In an attempt to improve PROTOPNET visualizations, an extension of layer-wise relevance propagation [2], Prototypical Relevance Propagation (PRP), is proposed to create more model-aware explanations [23]. PRP is quantitatively more effective in debugging erroneous prototypes and assigning pixel relevance than the original approach.

ProtoPartNN-Like Methods The following papers are inspired by PROTOPNET but cannot be considered to be the same class of model. This is due to not fulfilling the proposed ProtoPartNN desiderata #1 (prototypes must correspond directly to image patches) and/or #3 (case-based reasoning must be describable by linear or simple tree models).

VIT-NET [36] combines a vision transformer (ViT) with a neural tree decoder that learns prototypes. In another transformer-based approach, PROTOPFORMER [73] exploits the inherent architectural features (local and global branches) of ViTs. SEMI-PROTOPNET [65] fixes the readout weights as NP-PROTOPNET [62] does and is used for power distribution network analysis. In SDFA-SA-PROTOPNET [30], a shallow-deep feature alignment (SDFA) module aligns the similarity structures between deep and shallow layers. In addition, a score aggregation (SA) module aggregates similarity scores to avoid learning inter-class information. Unfortunately, each of these networks omits prototype replacement with the typical justification being that doing so improves task accuracy. In addition, VIT-NET has additional layers after g that break the mapping back to pixel space and complicate its case-based reasoning.

3. The Problem with Existing ProtoPartNNs

Despite the many extensions of PROTOPNET, there are still fundamental issues with image part localization, pixel space grounding, and heat map visualizations, which preclude *any existing ProtoPartNN from satisfying all three desiderata* – all ProtoPartNNs violate desideratum #2: prototypes must localize to image parts. The underlying issues with existing ProtoPartNNs arise from 1) their pixel space mapping being reliant on spatial correlation between embedded patches and the input space, which is dubious; 2) their pixel space mapping being receptive field-invariant, arbitrarily localizing to some area in the input. Rather, intrinsically interpretable models should produce explanations *implicit in the symbolic form of the model itself* [53, 59].

As a refresher, the original visualization process involves three steps. First, a single similarity map $\boldsymbol{S}_{ij}{=}\pi_{\boldsymbol{p}_j}(\boldsymbol{Z}_i) \in \mathbb{R}^{H_z/H_p \times W_z/W_p}$ is selected for visualization where $\pi_{\boldsymbol{p}_j}$ gives the similarity map for prototype $\boldsymbol{p}_j$. Each element of $\boldsymbol{S}_{ij}$ is given by $v\left(\varphi(\boldsymbol{z}, \boldsymbol{p}_j)\right)$ where $\boldsymbol{z}{\in}\texttt{patches}(\boldsymbol{Z}_i)$. Subsequently, this map is upsampled from $H_z/H_p{\times}W_z/W_p$ to $H{\times}W$ using bicubic interpolation, producing a heat map $\boldsymbol{M}_{ij}{\in}\mathbb{R}^{H \times W}$. To localize within the image, the smallest bounding box is drawn around the largest 5% of heat map elements – this box is of variable size. While no justification is provided for this approach in the original paper [11], we believe that the intuition is that the embedded patches $\boldsymbol{Z}_i$ maintain spatial correlation with the input. Finally, $\boldsymbol{M}_{ij}$ and the bounding box can be superimposed on the input image for visualization. From here on out, we will refer to this as the *original pixel space mapping*, which is visualized in Figure 3a. It should also be noted that while this pixel space mapping is crucial in establishing interpretability, it is left undiscussed in the vast majority of PROTOPNET extensions. Immediately, we can see several issues with this approach.

Here* Does Not Correspond to *There The original pixel space mapping is based on naive upsampling, which is invariant to architectural details. The approach will always assume that all similarity scores can be mapped to pixel space with a single linear transformation – an embedded patch at position $\langle t_x, t_y \rangle$ is effectively localized to position $\langle t_x{\cdot}W{\cdot}W_p/W_z, t_y{\cdot}H{\cdot}H_p/H_z \rangle$ in pixel space. This assumption of spatial correlation from high to low layers is easy to invalidate. For instance, even a simple latent transpose eradicates this correlation. ***The similarity scores of embedded patches do not determine where the architecture "looked" in the image. Rather, the architecture determines where the similarity scores correspond to in the image.*** Figure 1 demonstrates this discrepancy. Very recently, evidence in [27, 57] strongly corroborates our arguments about poor localization. We correct this pixel space mapping according to the receptive fields of the underlying neural architecture. The original approach also only provides a way to localize a prototype rather than any embedded patch – our method enables us to do so. Our approach is described in detail in Section 4 and we validate its correctness over the original approach in Section 5.

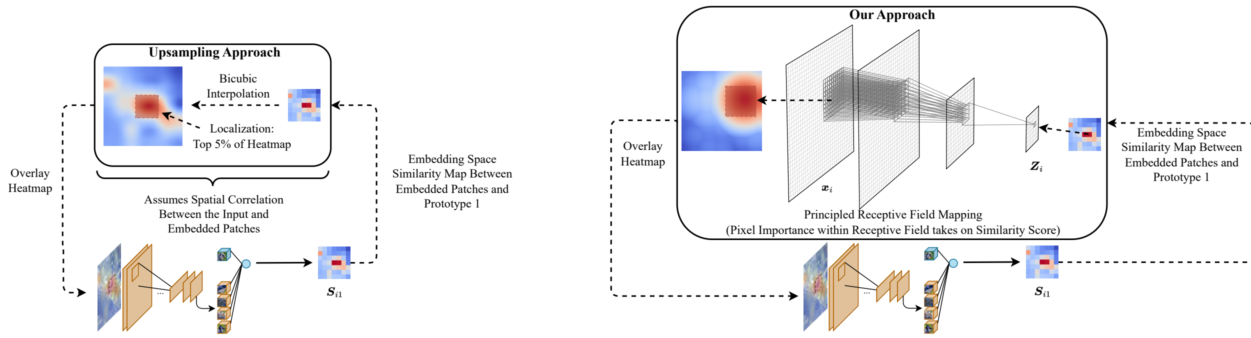
This* Does not Correspond to *Just That PROTOPNET and its derivatives all elect to localize to a small region of the input by drawing a bounding box around the largest 5% of values of heat map $\boldsymbol{M}_{ij}$ as shown in Figure 3a. While this produces alluring visualizations, most of the architectures evaluated in all prior approaches have a mean receptive field of 100% at the embedding layer[1]. ***A mean receptive field of 100% means that every element of the embedding layer output is a complex function of every pixel in the input space. Is it fair to say that only ~5% of the input contributed to some part of a decision?*** Attribution within the input space spanned by a receptive field is unverifiable from both the feature-selectivity and feature-additivity points of view [7, 42]. This issue is visualized in Figure 1 for an architecture with a mean receptive field under 100%. Moreover, while selecting the top 5% of $\boldsymbol{M}_{ij}$ may localize in accordance with its (faulty) intuition, it can actually localize to wildly inaccurate parts of the image (*e.g.*, if multiple top values in $\boldsymbol{S}_{ij}$ are all close), breaking the intuition of the (unfaithful) pixel space mapping. We go on to discuss our solution to this problem in Section 4.

The Allure of Visualization The original pixel space mapping appears to satisfy human intuitions. However, it is not based on well-justified aspects of explainability. Beyond the assumption of spatial correlation and naive localization, bicubic interpolation artificially increases the resolution of maps (see Figure 3a), which leads non-experts to believe that per-pixel attributions are estimated. In our proposed approach, these explanation aspects follow naturally from the symbolic interpretation of the model itself.

4. Fixing ProtoPartNNs

As discussed in Section 3, the underlying issues with ProtoPartNNs arise from 1) the original pixel space mapping being reliant on spatial correlation between embedded patches and the input space, which is dubious; 2) the original pixel space mapping being receptive field-invariant,

[1] The lowest mean receptive field of an evaluated architecture is from VGG19 (~70%) [11].



(a) The original upsampling-based pixel space mapping.

(b) Our proposed receptive field-based pixel space mapping.

Figure 3. Visualization of the original and proposed pixel space mapping approaches.

arbitrarily localizing to some area in the input. Our proposed architecture, PIXPNET (Pixel-grounded Prototypical part Network), is largely based on PROTOPNET but mitigates these issues through symbolic interpretation of its architecture – see Figure 2 for an overview. In this section, we first describe a new algorithm for the calculation of receptive fields, describe our proposed fixes for prototype visualization and localization, and proceed with additional ProtoPartNN corrections and improvements. With the proposed improvements, PIXPNET is the *only ProtoPartNN that truly localizes to image parts*, satisfying all three desiderata.

Receptive Field Calculation Algorithm Before delving into our proposed remedies, we describe our approach to computing receptive fields precisely for any architecture. Our proposed algorithm, `FunctionalRF`, takes a neural network as input and outputs the *exact* receptive field of every neuron in the neural network. Recall that a neuron is a function of a *subset* of pixels defined by its receptive field. `FunctionalRF` represents receptive fields as hypercubes (multidimensional tensor slices). For instance, the slices for a 2D convolution with a 5×5 kernel, stride of 1, and c_{in} channels at output position $3, 3$ would be $\{\{[\![1, c_{\text{in}}]\!], [\![1, 5]\!], [\![1, 5]\!]\}\}$ where $[\![a, b]\!]$ denotes the slice between a and b. We can compute the *mean receptive field* of a layer as the average number of pixels within the receptive field of each hypercube element of a layer output. The algorithm does not rely on approximate methods nor architectural alignment assumptions like other approaches [1, 45]. The full algorithmic details are provided in Appendix C.

Corrected Pixel Space Mapping Algorithm

From Embedding Space to Pixel Space For each prototype $\boldsymbol{p}_j$, we have some $\boldsymbol{z} \in \boldsymbol{Z}_i$ that is most similar. We are interested in knowing where $\boldsymbol{z}$ localizes to in an image $\boldsymbol{x}_i$. With `FunctionalRF` applied to the backbone, we have the precise pixel space region that $\boldsymbol{z}$ is a function of – *this exactly corresponds to that*. This can also be done for any $\boldsymbol{p}_j$ after prototype replacement. Additionally, this process can actually be used to visualize any $\boldsymbol{z} \in \boldsymbol{Z}_i$, unlike the procedure specified in the original pixel space mapping [11]. See Figure 3b for intuition as to how this process works.

Producing a Pixel Space Heat Map In order to compute a pixel space heat map, we propose an algorithm based on `FunctionalRF` rather than naively upsampling an embedding space similarity map $\boldsymbol{S}_{ij}$. Our approach uses the same idea as going from embedding space to pixel space. Each pixel space heat map $\boldsymbol{M}_{ij} \in \mathbb{R}^{H\times W}$ is initialized to all zeros ($\mathbf{0}^{H\times W}$), and corresponds to a sample $\boldsymbol{x}_i$ and a prototype $\boldsymbol{p}_j$. Let $\boldsymbol{M}_{ij}^S$ be the region of $\boldsymbol{M}_{ij}$ defined by the receptive field of similarity score $S \in \boldsymbol{S}_{ij}$. For each S, the pixel space heat map is updated as $\boldsymbol{M}_{ij}^S \leftarrow \max(\boldsymbol{M}_{ij}^S, S)$ where $\max(\cdot)$ is an element-wise maximum that appropriately handles the case of overlapping receptive fields. We take maxima instead of averaging values due to Eq. (1). Again, see Figure 3b for a visualization of this procedure. Further algorithmic details are provided in Appendix D.

Improved Localization & the "Goldilocks" Zone To reiterate, the region localized by a ProtoPartNN is controlled by the receptive field of the embedding layers of f. A fundamental goal of ProtoPartNNs is to identify and learn prototypical object parts. We propose to achieve this by constraining the receptive field of f to a range that yields image parts that are both meaningful and interpretable to humans.

It is well known that the receptive field of a neural network correlates with performance [1, 45] to an extent – too small or large a receptive field can harm performance due to bias-variance trade-offs [39]. We hypothesize that there is a "Goldilocks" zone where the desired receptive field localizes to intelligible image parts without diminishing task performance. To corroborate this, we evaluate various backbone architectures at intermediate layers on ImageNette [21], a subset of ImageNet [13]. The evaluation aims to produce architectures suitable for the backbone of PIXPNET according to the criteria outlined prior. We propose this approach as performance on subsets of ImageNet has been shown to be reflective of performance on the full dataset [16], and ImageNet performance strongly correlates

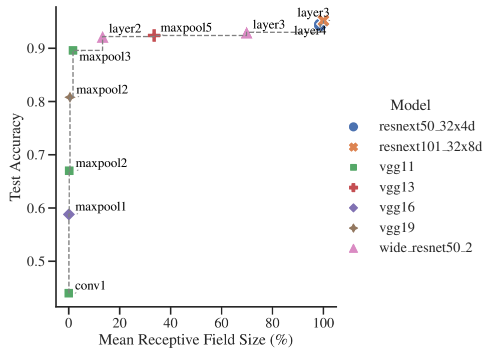


Figure 4. The Pareto front of architectures trained on ImageNette [21] and evaluated at various intermediate layers. This front details the accuracy-localization size trade-offs and informs backbone selection of PIXPNET as in Section 5.

with performance on other vision datasets [38]. We detail the full experiment setup in Appendix F. The Pareto front of mean receptive field and accuracy for the evaluated architectures is shown in Figure 4. This front informs our backbone selection as detailed in Section 5.

Simplified Classification Head While the original fully-connected classification head h is human-interpretable, it has several weaknesses – its explanation size limits its comprehension [37, 63] and it requires an additional training stage, adding up to 100 additional epochs in PROTOPNET[2]. We quantify explanation size in terms of *positive reasoning* and *negative reasoning* about the prediction of a class. For positive reasoning, the number of elements in an explanation with the original fully-connected layer is $2P/C$: one similarity score per class-specific prototype and a positive weight coefficient. However, considering both positive and negative reasoning involves $2P$ total explanation elements.

To address these limitations, we propose to replace the linear layer with a class-wise summation. This operation simply produces the logit of each class as the sum of class-specific similarity scores as $\hat{y}_{ic} = \sum_{j:\boldsymbol{p}_j \in \boldsymbol{P}_c} s_{ij}$ where $\hat{y}_{ic}$ is the logit for class c and s_{ij} is the similarity score for prototype $\boldsymbol{p}_j$. The layer is visualized in Figure 2. Our new parameter-free readout layer removes the additional training stage and comprises only P/C explanation elements for *both* positive and negative reasoning. Substituting our layer in the original PROTOPNET configuration for the CUB-200-2011 dataset [11, 70] reduces the number of explanation elements for a class prediction from 4,000 down to *just 10*.

Other Improvements We also make a few smaller contributions. In prototype replacement, we remove duplicate prototypes (by image or sample) to encourage diversity. If duplicates are found, the next most-similar embedded patch is used in replacement instead. We also reformulate the similarity function v to have lower numerical error (see Appendix G for details) as $v(d){=}\log(\frac{1}{d+\varepsilon}{+}1)$ where ε is mitigates division by zero and the distance $d{=}\varphi(\boldsymbol{z}, \boldsymbol{p}_j)$. While PROTOPNET uses $\varphi(\boldsymbol{z}, \boldsymbol{p}_j){=}\|\boldsymbol{z} - \boldsymbol{p}_j\|_2^2$, we elect to use $\varphi(\boldsymbol{z}, \boldsymbol{p}_j){=}1 - \frac{\boldsymbol{z} \cdot \boldsymbol{p}}{\|\boldsymbol{z}\|_2 \|\boldsymbol{p}\|_2}$ (cosine distance), which has a desirable normalizing factor. This distance is also used in [4, 17, 35, 72]. In implementation, the distances are computed using generalized convolution [11, 24, 49].

Training Our multi-stage training procedure is similar to that of PROTOPNET. The first stage optimizes the full network, except for the readout layer, by minimizing Eq. (2) via stochastic gradient descent

$$\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{\text{xent}}(\hat{\boldsymbol{y}}_i, y_i){+}\lambda_{\text{cls}}\mathcal{L}_{\text{cls}}(\boldsymbol{P}, \boldsymbol{Z}_i){+}\lambda_{\text{sep}}\mathcal{L}_{\text{sep}}(\boldsymbol{P}, \boldsymbol{Z}_i) \quad (2)$$

where $\mathcal{L}_{\text{xent}}$ is the categorical cross-entropy loss function, λ_{cls} and λ_{sep} are auxiliary loss weights, and the auxiliary loss functions, $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{sep}}$, are defined as

$$\mathcal{L}_{\text{cls}}(\boldsymbol{P}, \boldsymbol{Z}_i) = \frac{1}{N}\sum_{i=1}^{N} \min_{\substack{\boldsymbol{p}_j \in \boldsymbol{P}_{y_i} \\ \boldsymbol{z} \in \texttt{patches}(\boldsymbol{Z}_i)}} \varphi(\boldsymbol{z}, \boldsymbol{p}_j) \quad (3)$$

$$\mathcal{L}_{\text{sep}}(\boldsymbol{P}, \boldsymbol{Z}_i) = -\frac{1}{N}\sum_{i=1}^{N} \min_{\substack{\boldsymbol{p}_j \notin \boldsymbol{P}_{y_i} \\ \boldsymbol{z} \in \texttt{patches}(\boldsymbol{Z}_i)}} \varphi(\boldsymbol{z}, \boldsymbol{p}_j). \quad (4)$$

The goal of $\mathcal{L}_{\text{cls}}$ is to ensure that at least one embedded patch of every training image is similar to at least one prototype belonging to the class of the image. In contrast, the goal of $\mathcal{L}_{\text{sep}}$ is to ensure that the embedded patches of every training image are dissimilar from prototypes not belonging to the class of the image.

Subsequently, the prototypes are replaced, which is arguably the most important stage of training as it grounds prototypes in human-comprehensible pixel space. The process involves replacing each prototype $\boldsymbol{p}_j$ with an embedded patch $\boldsymbol{z}$ of a training sample of the same class – the most similar embedded patch replaces the prototype. In the literature, *prototype replacement* is also referred to as prototype "pushing" or "projection." We stick with "replacement" for the sake of clarity. Formally, this update can be written as $\boldsymbol{p}_j \leftarrow \arg\min_{\boldsymbol{z} \in \texttt{patches}(\boldsymbol{Z}_i)} \varphi(\boldsymbol{z}, \boldsymbol{p}_j)$, s.t. $\boldsymbol{p}_j \in \boldsymbol{P}_{y_i}$. Without this update, the human interpretation of prototypes is unclear as prototypes are not grounded in pixel space.

In PROTOPNET and its variants, a third stage optimizes the linear readout layer. However, we do not employ this stage as our readout layer is parameter-free. The multi-stage optimization process can be repeated until convergence.

5. Experiments & Discussion

To validate our proposed approach, PIXPNET, we evaluate both its accuracy and interpretability on CUB-200-2011 [70]. We also show evaluation results on Stanford

[2]In the original PROTOPNET implementation, as well as subsequent extensions, the last layer is optimized 5 times, each for 20 epochs [11].

BBox	D1	D2	D3	Model	f	Expl. Size +	Expl. Size ±	P	MRF	Acc.	±	S_{con}	S_{sta}	Code Avail.	Val. Set
✗	✓	✓	✓	PIXPNET (Ours)	ResNeXt@layer3	**10**	**10**	2000	100	**81.76**	0.2	56.4	**64.7**	✓	✓
✗	✓	✓	✓	PIXPNET (Ours)	VGG19@maxpool5	**10**	**10**	2000	70.4	80.10	0.1	47.6	64.2	✓	✓
✗	✓	✓	✓	PIXPNET (Ours)	VGG16@maxpool5	**10**	**10**	2000	52.5	79.75	0.2	**69.5**	51.6	✓	✓
✗	✓	✓	✓	PIXPNET (Ours)	VGG13@maxpool4	**10**	**10**	2000	9.69	75.32	0.2	66.9	45.0	✓	✓
✗	✓	✗	✓	ST-PROTOPNET [71]	DenseNet161	20	4000	2000	100	80.60	–	–	–	✓	✗
✓	✓	✗	✓	ST-PROTOPNET [71]	DenseNet161	20	4000	2000	100	<u>86.10</u>	0.2	–	–	✓	✗
✓	✓	✗	✓	TESNET [72]	DenseNet121	20	4000	2000	100	84.80	0.2	63.1	<u>66.1</u>	✓	✗
✓	✓	✗	✓	PROTOPOOL [55]	ResNet152	20	404	202	100	81.50	0.1	35.7	58.4	✓	✗
✓	✓	✗	✓	PROTOPNET [11]	DenseNet121	20	4000	2000	100	80.20	0.2	24.9	58.9	✓	✗
✓	✓	✗	✓	PROTO2PROTO [34]	ResNet34	20	4000	2000	100	79.89	–	–	–	✓	✗
✓	✓	✗	✓	PROTOPSHARE [56]	DenseNet161	1200	1200	600	100	76.45	–	–	–	✓	✗
✓	✓	✗	✓	PROTOTREE [30, 51]	DenseNet121	18	404	202	100	73.20	–	21.5	24.4	✓	✗
✗	✗	✗	✓	PROTOPFORMER [73]	DeiT-S	40	8000	4000	100	**84.85**	–	–	–	✓	✗
✗	✗	✗	✗	VIT-NET [36, 73]	CaiT-XXS-24	**8**	30	15	100	84.51	–	–	–	✓	✗
✓	✗	✗	✗	VIT-NET [36]	SwinT-B	<u>10</u>	62	31	100	<u>91.60</u>	–	–	–	✓	✗
✓	✗	✗	✓	SDFA-SA [30]	DenseNet161	20	20	2000	100	86.80	–	<u>73.2</u>	<u>73.5</u>	✓	✗

Table 1. ProtoPartNN results on CUB-200-2011 with ImageNet used for pre-training. Columns D1, D2, and D3 correspond to the three desiderata established in Section 2. Our approach, PIXPNET, is the only method that is a *3-way ProtoPartNN*, satisfying all three desiderata. "BBox" indicates whether a method crops each image using a bounding box annotation. The best results of ProtoPartNNs with and without such annotations are **bold** and <u>underlined</u>, respectively. The table is split based on whether the method meets at least two desiderata. The S_{con} and S_{sta} scores for other methods are taken from [30] and the top reported accuracy score is taken for each method.

Cars [40] in Appendix B. We draw comparisons against other ProtoPartNNs with a variety of measures. We elect to not crop images in CUB-200-2011 by their bounding box annotations to demonstrate the localization capability of PIXPNET. Hyperparameters, software, hardware, and other reproducibility details are specified in Appendix E.

Lastly, upon inspection of the original code base[3], we discovered that the test set accuracy is used to influence training of PROTOPNET. In fact, neither PROTOPNET nor its extensions for image classification that are mentioned in Section 2 employ a validation set in provided implementations. See Appendix H for further details. In our implementation, we employ a proper validation set and tune hyperparameters only according to accuracy on this split.

Accuracy The experimental results in Table 1 show that PIXPNET obtains competitive accuracy with other approaches regardless of whether images are cropped by bird bounding box annotations – *while we trade off network depth for interpretability, we outperform* PROTOPNET *and several of its derivatives*. This is quite favorable as PIXPNET is the only method that truly localizes to image parts.

Interpretability We evaluate the interpretability of our approach with several functionally grounded metrics [18]. See Figure 2b for an example of a PIXPNET explanation.

Relevance Ordering Test (ROT) The ROT is a quantitative measure of how well a pixel space mapping attributes individual pixels according to prototype similarity scores [23]. First, a pixel space heat map $\boldsymbol{M}_{ij}$ is produced for a single sample $\boldsymbol{x}_i$ and prototype $\boldsymbol{p}_j$. Starting from a completely random image, pixels are added back to the random image one at a time in descending order according to $\boldsymbol{M}_{ij}$. As each pixel is added back, the similarity score for $\boldsymbol{p}_j$ is evaluated. This procedure is averaged over each class-specific prototype over 50 random samples. The faster that the original similarity score is recovered, the better the pixel space mapping is. Assuming a faithful pixel space mapping, a network with a mean receptive field of, *e.g.*, 25%, will recover the original similarity score after 25% of the pixels are added back in the worst-case scenario.

We also introduce two aggregate measures of the ROT. First is the area under the similarity curve (**AUSC**) which is normalized by the difference between the original similarity score and the baseline value (similarity score for a completely random image)[4]. Second is the percentage of pixels added back to recover the original similarity score: pixel percentage to recovery (**%2R**).

We compare our pixel space mapping to the original upsampling approach and PRP [23]. However, the PRP implementation only supports RESNET architectures[5], so it is not included in all experiments. The results in Table 2 demonstrate that our pixel space mapping best identifies the most

[3] https://github.com/cfchen-duke/ProtoPNet

[4] AUSC>1 is possible as the maximum possible similarity is unknown.

[5] The hard-coded and complex nature of the PRP code base precludes simple extension to other architectures.

Backbone	MRF	Acc. ↑	PSM	S_{con} ↑	S_{sta} ↑	AUSC ↑	%2R ↓
VGG11 @maxpool4	8.31	72.9	Ours	**65.3**	**48.3**	**0.99**	**11.2**
			Orig.	45.8	44.0	0.90	30.5
VGG13 @maxpool4	9.69	75.3	Ours	**66.9**	**45.0**	**0.97**	**13.0**
			Orig.	48.1	41.8	0.88	84.1
VGG16 @maxpool4	15.7	76.4	Ours	**62.0**	**46.4**	**1.02**	**6.98**
			Orig.	46.8	42.2	0.89	35.5
VGG19 @maxpool4	22.8	77.1	Ours	**60.1**	**42.5**	**0.94**	**21.4**
			Orig.	48.4	41.3	0.80	99.9
VGG13 @maxpool5	33.5	78.1	Ours	**67.0**	**42.5**	**0.90**	**29.5**
			Orig.	43.7	39.9	0.81	99.2
VGG16 @maxpool5	52.5	79.8	Ours	**69.5**	**51.6**	**0.90**	**32.0**
			Orig.	44.1	42.4	0.82	55.5
WRN50 @layer3	69.9	80.1	Ours	**56.4**	**64.7**	**0.93**	**13.0**
			Orig.	56.4	47.6	0.85	39.6
VGG19 @maxpool5	70.4	80.1	Ours	**47.6**	**64.2**	**0.92**	**43.4**
			Orig.	45.8	46.0	0.85	92.9
ResNet18 @layer2	15.4	57.2	Ours	**59.2**	**46.6**	**0.98**	**4.10**
			Orig.	25.2	45.6	0.88	96.8
			PRP	–	–	0.95	25.4
ResNet50 @layer3	69.8	76.6	Ours	47.9	**62.0**	**0.58**	**72.8**
			Orig.	**53.5**	42.7	0.42	97.8
			PRP	–	–	0.34	100.0

Table 2. Evaluation of pixel space mapping (PSM) methods with functionally-grounded interpretability metrics. Methods are compared on PIXPNET with "Goldilocks" zone and RESNET backbones on CUB-200-2011 (no BBox cropping). Our PSM outperforms both the original and PRP PSMs across *all* backbones.

important pixels in an image. Naturally, the mean receptive field correlates with both ROT scores.

Explanation Size Recall from Section 4 that the explanation size is the number of elements in an explanation, *i.e.*, similarity scores and weight coefficients. This number differs when considering positive or negative reasoning. Due to the original classification head being fully-connected, most ProtoPartNNs have large explanation sizes when considering both positive and negative reasoning, as shown in Table 1. In contrast, our explanation size comprises just 10 elements when reasoning about a decision. Our proposed classification head helps to prevent overwhelming users with information, which has been shown to be the case with other ProtoPartNNs [37].

Consistency The consistency metric [30] quantifies how consistently each prototype localizes to the same human-annotated ground truth part. It evaluates both semantic similarity quality and the pixel space mapping to a degree. For a sample $\boldsymbol{x}_i$ with label y_i, the pixel space mapping is computed for each prototype $\boldsymbol{p}_j \in \boldsymbol{P}_{y_i}$. Let $o_{\boldsymbol{p}_j}(\boldsymbol{x}_i) \in \mathbb{R}^K$ be a binary vector indicating which of K object parts are contained within the region localized by the pixel space mapping. Let $u(\boldsymbol{x}_i) \in \mathbb{R}^K$ be a binary vector indicating which of the K object parts are actually visible in $\boldsymbol{x}_i$. A single object part is associated with $\boldsymbol{p}_j$ by taking the maximum frequency of an object part present in the pixel space mapping region across all applicable images. A prototype is said to be consistent if this frequency is at least μ, *i.e.*,

$$S_{\text{con}} = \frac{1}{P}\sum_{j=1}^{P} \mathbb{1}\left[\max\left(\sum_{\boldsymbol{x}_i \in \mathbf{X}_{c_j}} \left(o_{\boldsymbol{p}_j}(\boldsymbol{x}_i) \oslash u(\boldsymbol{x}_i)\right)\right) \geq \mu\right]$$

where $\mathbf{X}_{c_j}$ are samples of the same class allocated to $\boldsymbol{p}_j$, $\oslash$ denotes element-wise division, and $\mathbb{1}$ is the indicator function. To compare with results reported in [30], we change the receptive field size in our pixel space mapping to equal this, as well as set $\mu = 0.8$. A notable weakness of the evaluation approach is that it uses a fixed 72×72 pixel region independent of the architecture. While the approach is not perfect, it allows for reproducible and comparative interpretability evaluation between ProtoPartNN variants.

Results are shown in Tables 1 and 2 for CUB-200-2011, which provides human-annotated object part annotations. We outperform PROTOPNET and many of its variants, as well as the original pixel space mapping (Table 2).

Stability The stability metric [30] measures how robust object part association is when noise is added to an image. Simply, some noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is added to each sample $\boldsymbol{x}_i$ and the object part associations are compared as

$$S_{\text{sta}} = \frac{1}{P}\sum_{j=1}^{P} \frac{\sum_{\boldsymbol{x}_i \in \mathbf{X}_{c_j}} \mathbb{1}\left[o_{\boldsymbol{p}_j}(\boldsymbol{x}_i) = o_{\boldsymbol{p}_j}(\boldsymbol{x}_i + \epsilon)\right]}{|\mathbf{X}_{c_j}|}.$$

Following [30], we set σ=0.2. Results in Tables 1 and 2 support the robustness of PIXPNET compared to other ProtoPartNNs and the original pixel space mapping. There is a marginal decrease in stability as the receptive field lessens.

6. Limitations and Future Work

The receptive field constraint is a design choice and is inherently application-specific, subject to data characteristics and interpretability requirements. Future work should investigate multi-scale receptive fields and automated receptive field design techniques. Nevertheless, we *trade off network depth for significant gains in interpretability with very little penalty in accuracy*. Prior studies have shown that ProtoPartNNs have a semantic similarity gap with humans, prototypes can be redundant or indistinct, and limited utility in improving human performance [27,28,37,63]. Moreover, the consistency and stability evaluation metrics are imperfect. Although we improve upon interpretability over other networks, human studies are needed to understand other facets of interpretability, such as trustworthiness, acceptance, and utility [59]. In the future, architectural improvements should be made, *e.g.*, the enriched embedding space of TESNET, prototype diversity constraints [58, 68, 71], and human-in-the-loop training [48].

References

[1] André Araujo, Wade Norris, and Jack Sim. Computing receptive fields of convolutional neural networks. *Distill*, 2019. https://distill.pub/2019/computing-receptive-fields. 5

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. 3

[3] Hubert Baniecki. Adversarial explainable AI. https://hbaniecki.com/adversarial-explainable-ai/, 2023. Accessed: 2023-01-28. 2

[4] Alina Jade Barnett, Zhicheng Guo, Jin Jing, Wendong Ge, Cynthia Rudin, and M. Brandon Westover. Mapping the ictal-interictal-injury continuum using interpretable machine learning. *arXiv*, 2022. 6

[5] Sebastian Bordt, Michèle Finck, Eric Raidl, and Ulrike von Luxburg. Post-hoc explanations fail to achieve their purpose in adversarial contexts. *ACM Conference on Fairness, Accountability, and Transparency*, 5, 2022. 2

[6] Joy Buolamwini and Timnit Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91. PMLR, Jan. 2018. 1

[7] Oana-Maria Camburu, Eleonora Giunchiglia, Jakob Foerster, Thomas Lukasiewicz, and Phil Blunsom. Can i trust the explainer? verifying post-hoc explanatory methods. In *NeurIPS 2019 Workshop on Safety and Robustness in Decision Making*. arXiv, 2019. 4

[8] Zachariah Carmichael and Walter J. Scheirer. A framework for evaluating post hoc feature-additive explainers. *arXiv*, abs/2106.08376, 2021. 1, 2

[9] Zachariah Carmichael and Walter J Scheirer. Unfooling perturbation-based post hoc explainers. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2023. 2

[10] Selina Carter and Jonathan Hersh. Explainable ai helps bridge the ai skills gap: Evidence from a large bank. *Economics Faculty Articles and Research*, 276, 2022. 2

[11] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan Su. This looks like that: Deep learning for interpretable image recognition. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Neural Information Processing Systems, NeurIPS*, pages 8928–8939, 2019. 1, 2, 4, 5, 6, 7

[12] Guy Van den Broeck, Anton Lykov, Maximilian Schleich, and Dan Suciu. On the tractability of SHAP explanations. In *AAAI Conference on Innovative Applications of Artificial Intelligence, IAAI*, pages 6505–6513. AAAI Press, 2021. 2

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE Computer Society, 2009. 5

[14] Botty Dimanov, Umang Bhatt, Mateja Jamnik, and Adrian Weller. You shouldn't trust me: Learning models which conceal unfairness from multiple explanation methods. *Frontiers in Artificial Intelligence and Applications: ECAI*, 2020. 2

[15] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. *Advances in neural information processing systems*, 32, 2019. 2

[16] Xuanyi Dong and Yi Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. In *International Conference on Learning Representations, ICLR*. OpenReview.net, 2020. 5

[17] Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable ProtoPNet: An interpretable image classifier using deformable prototypes. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 10255–10265. IEEE/CVF, 2022. 3, 6

[18] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv*, 2017. 7

[19] Council of the EU and European Parliament. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, L 119:1–88, 2016. 1

[20] European Commission. Proposal for a regulation of the European Parliament and the Council: Laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union legislative acts. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206, 4 2021. 1

[21] FastAI. Imagenette. https://github.com/fastai/imagenette, 2020. 5, 6

[22] Damien Garreau and Ulrike von Luxburg. Explaining the explainer: A first theoretical analysis of LIME. In Silvia Chiappa and Roberto Calandra, editors, *International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1287–1296. PMLR, Aug. 2020. 2

[23] Srishti Gautam, Marina M.-C. Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. This looks more like that: Enhancing self-explaining models by prototypical relevance propagation. *Pattern Recognition*, 136:1–13, 2023. 3, 7

[24] Kamaledin Ghiasi-Shirazi. Generalizing the convolution operator in convolutional neural networks. *Neural Processing Letters*, 50(3):2627–2646, 2019. 6

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 770–778. IEEE Computer Society, 2016. 2

[26] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Conference on Computer Vision and Pattern Recognition*, pages 15262–15271. IEEE/Computer Vision Foundation, 2021. 1

[27] Robin Hesse, Simone Schaub-Meyer, and Stefan Roth. FunnyBirds: A synthetic vision dataset for a part-based analysis of explainable AI methods. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 1–18. IEEE, 2023. 4, 8

[28] Adrian Hoffmann, Claudio Fanconi, Rahul Rade, and Jonas Kohler. This looks like that... does it? Shortcomings of latent space prototype interpretability in deep networks. In *ICML Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI*, 2022. 2, 8

[29] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2261–2269. IEEE Computer Society, 2017. 2

[30] Qihan Huang, Mengqi Xue, Wenqi Huang, Haofei Zhang, Jie Song, Yongcheng Jing, and Mingli Song. Evaluation and improvement of interpretability for self-explainable part-prototype networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2011–2020, October 2023. 3, 7, 8

[31] Christina Humer, Andreas Hinterreiter, Benedikt Leichtmann, Martina Mara, and Marc Streit. Comparing effects of attribution-based, example-based, and feature-based explanation methods on ai-assisted decision-making. *OSF Preprints*, 2022. 2

[32] Sérgio Jesus, Catarina Belém, Vladimir Balayan, João Bento, Pedro Saleiro, Pedro Bizarro, and João Gama. How can i choose an explainer? an application-grounded evaluation of post-hoc explanations. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, FAccT'21, page 805–815, New York, NY, USA, 2021. Association for Computing Machinery. 1, 2

[33] Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik, editors, *Conference on Human Factors in Computing Systems*, pages 1–14. ACM, Apr. 2020. 2

[34] Monish Keswani, Sriranjani Ramakrishnan, Nishant Reddy, and Vineeth N. Balasubramanian. Proto2Proto: Can you recognize the car, the way I do? In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 10223–10233. IEEE/CVF, 2022. 3, 7

[35] Eunji Kim, Siwon Kim, Minji Seo, and Sungroh Yoon. XProtoNet: Diagnosis in chest radiography with global and local explanations. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 15719–15728. CVF/IEEE, 2021. 6

[36] Sangwon Kim, Jae-Yeal Nam, and ByoungChul Ko. Vitnet: Interpretable vision transformers with neural tree decoder. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 11162–11172. PMLR, 2022. 3, 7

[37] Sunnie S. Y. Kim, Nicole Meister, Vikram V. Ramaswamy, Ruth Fong, and Olga Russakovsky. HIVE: evaluating the human interpretability of visual explanations. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *European Conference on Computer Vision, ECCV*, volume 13672 of *Lecture Notes in Computer Science*, pages 280–298. Springer, 2022. 2, 6, 8

[38] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2661–2671. Computer Vision Foundation / IEEE, 2019. 6

[39] Khaled Koutini, Hamid Eghbal-zadeh, Matthias Dorfer, and Gerhard Widmer. The receptive field as a regularizer in deep convolutional neural networks for acoustic scene classification. In *27th European Signal Processing Conference, EUSIPCO 2019, A Coruña, Spain, September 2-6, 2019*, pages 1–5. IEEE, 2019. 5

[40] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 7

[41] Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. The disagreement problem in explainable machine learning: A practitioner's perspective. *arXiv*, pages 1–46, 2022. 1, 2

[42] Matthew L. Leavitt and Ari Morcos. Towards falsifiable interpretability research. In *NeurIPS Workshop on ML-Retrospectives, Surveys & Meta-Analyses*, pages 1–15. arXiv, 2020. 2, 4

[43] Library of Congress. H.R.6580 - 117th Congress (2021-2022): Algorithmic accountability act of 2022. `https://www.congress.gov/bill/117th-congress/house-bill/6580/text`, 2 2022. 1

[44] Zachary C. Lipton. The mythos of model interpretability. *ACM Queue*, 16(3):30, July 2018. 1

[45] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4898–4906, 2016. 5

[46] Sean McGregor. Preventing repeated real world AI failures by cataloging incidents: The AI incident database. In *AAAI Conference on Innovative Applications of Artificial Intelligence, IAAI*, pages 15458–15463. AAAI Press, 2021. 1

[47] D. Douglas Miller and Eric W. Brown. Artificial intelligence in medical practice: The question to the answer? *The American Journal of Medicine*, 131(2):129–133, 2018. 1

[48] Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. Interpretable and steerable sequence learning via prototypes. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis, editors, *SIGKDD Inter-*

national Conference on Knowledge Discovery & Data Mining, KDD, pages 903–913. ACM, 2019. 1, 8

[49] Keivan Nalaie, Kamaledin Ghiasi-Shirazi, and Modhammad-R. Akbarzadeh-T. Efficient implementation of a generalized convolutional neural networks based on weighted Euclidean distance. In *International Conference on Computer and Knowledge Engineering, ICCKE*, pages 211–216, 2017. 6

[50] Meike Nauta, Annemarie Jutte, Jesper C. Provoost, and Christin Seifert. This looks like that, because... explaining prototypes for interpretable image recognition. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases - International Workshops of ECML PKDD*, volume 1524 of *Communications in Computer and Information Science*, pages 441–456. Springer, 2021. 1

[51] Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 14933–14943. CVF/IEEE, 2021. 3, 7

[52] Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown, Sept. 2016. 1

[53] Tim Räz. ML interpretability: Simple isn't easy. *arXiv*, 2022. 1, 4

[54] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019. 1, 2

[55] Dawid Rymarczyk, Lukasz Struski, Michal Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zielinski. Interpretable image classification with differentiable prototypes assignment. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *European Conference on Computer Vision, ECCV*, volume 13672 of *Lecture Notes in Computer Science*, pages 351–368. Springer, 2022. 3, 7

[56] Dawid Rymarczyk, Lukasz Struski, Jacek Tabor, and Bartosz Zielinski. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In Feida Zhu, Beng Chin Ooi, and Chunyan Miao, editors, *SIGKDD Conference on Knowledge Discovery and Data Mining, KDD*, pages 1420–1430. ACM, 2021. 3, 7

[57] Mikołaj Sacha, Bartosz Jura, Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Interpretability benchmark for evaluating spatial misalignment of prototypical parts explanations. *arXiv*, 2023. 4

[58] Mikołaj Sacha, Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. ProtoSeg: Interpretable semantic segmentation with prototypical parts. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 1481–1492. IEEE/CVF, January 2023. 8

[59] Gesina Schwalbe and Bettina Finzel. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, Jan 2023. 2, 4, 8

[60] Joseph P. Simmons, Leif D. Nelson, and Uri Simonsohn. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11):1359–1366, 2011. 2

[61] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations, ICLR*, 2015. 2

[62] Gurmail Singh and Kin Choong Yow. These do not look like those: An interpretable deep learning model for image recognition. *IEEE Access*, 9:41482–41493, 2021. 3

[63] Poulami Sinhamahapatra, Lena Heidemann, Maureen Monnet, and Karsten Roscher. Towards human-interpretable prototypes for visual assessment of image classification models. *arXiv*, 2022. 2, 6, 8

[64] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling LIME and SHAP: adversarial attacks on post hoc explanation methods. In Annette N. Markham, Julia Powles, Toby Walsh, and Anne L. Washington, editors, *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pages 180–186. ACM, 2020. 1, 2

[65] Stéfano Frizzo Stefenon, Gurmail Singh, Kin Choong Yow, and Alessandro Cimatti. Semi-ProtoPNet deep neural network for the classification of defective power grid distribution structures. *Sensors*, 22(13):4859, 2022. 3

[66] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, pages 1–10, Apr. 2014. 1

[67] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (XAI): toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, 2020. 1

[68] Loc Trinh, Michael Tsang, Sirisha Rambhatla, and Yan Liu. Interpretable and trustworthy deepfake detection via dynamic prototypes. In *Winter Conference on Applications of Computer Vision, WACV*, pages 1972–1982. IEEE, 2021. 8

[69] U.S.-EU TTC. U.S.-EU joint statement of the Trade and Technology Council. `https://www.commerce.gov/news/press-releases/2022/05/us-eu-joint-statement-trade-and-technology-council`, 5 2022. 1

[70] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 6

[71] Chong Wang, Yuyuan Liu, Yuanhong Chen, Fengbei Liu, Yu Tian, Davis McCarthy, Helen Frazer, and Gustavo Carneiro. Learning support and trivial prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2062–2072, October 2023. 3, 7, 8

[72] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *International Conference on Computer Vision, ICCV*, pages 875–884. IEEE/CVF, 2021. 3, 6, 7

[73] Mengqi Xue, Qihan Huang, Haofei Zhang, Lechao Cheng, Jie Song, Minghui Wu, and Mingli Song. ProtoPFormer:

Concentrating on prototypical parts in vision transformers for interpretable image recognition. *arXiv*, 2022. 3, 7