Understanding the (Extra-)Ordinary: Validating Deep Model Decisions with Prototypical Concept-based Explanations

Maximilian Dreyer[1], Reduan Achtibat[1], Wojciech Samek[1,2,3,†], Sebastian Lapuschkin[1,†]
[1] Fraunhofer Heinrich Hertz Institute, [2] Technical University of Berlin,
[3] BIFOLD – Berlin Institute for the Foundations of Learning and Data
[†]corresponding authors: {wojciech.samek | sebastian.lapuschkin}@hhi.fraunhofer.de

Abstract

Ensuring both transparency and safety is critical when deploying Deep Neural Networks (DNNs) in high-risk applications, such as medicine. The field of explainable AI (XAI) has proposed various methods to comprehend the decision-making processes of opaque DNNs. However, only few XAI methods are suitable of ensuring safety in practice as they heavily rely on repeated labor-intensive and possibly biased human assessment. In this work, we present a novel post-hoc concept-based XAI framework that conveys besides instance-wise (local) also class-wise (global) decision-making strategies via prototypes. What sets our approach apart is the combination of local and global strategies, enabling a clearer understanding of the (dis-)similarities in model decisions compared to the expected (prototypical) concept use, ultimately reducing the dependence on human long-term assessment. Quantifying the deviation from prototypical behavior not only allows to associate predictions with specific model sub-strategies but also to detect outlier behavior. As such, our approach constitutes an intuitive and explainable tool for model validation. We demonstrate the effectiveness of our approach in identifying out-of-distribution samples, spurious model behavior and data quality issues across three datasets (ImageNet, CUB-200, and CIFAR-10) utilizing VGG, ResNet, and EfficientNet architectures. Code is available at https://github.com/maxdreyer/pcx.

1. Introduction

Deep Neural Networks (DNNs) showcase remarkable performance in tasks such as medical diagnosis [7] and autonomous driving [19]. The significance of understanding and validating Machine Learning (ML) models becomes particularly pronounced in such safety-critical applications. Notably, DNNs have been shown to learn shortcuts that stem from spurious data artifacts, such as watermarks [29]. They further provide unreliable predictions when faced with samples from unrelated data domains, commonly referred to as Out Of Distribution (OOD) samples. In both scenarios, model predictions may be rooted in incorrect reasoning, potentially leading to severe consequences when these models are deployed in real-world applications.

The field of eXplainable Artificial Intelligence (XAI) has emerged to demystify the inner workings of black-box models and offer insights into their decision-making processes. XAI methods generally fall into the categories of *global* and *local*. While global techniques study model behavior on the class-wise or dataset-wise level, local XAI renders explanations at the instance level, facilitating an understanding of input feature relevance for specific prediction outcomes.

When deploying ML models, the importance of transparency and safety for *single* decisions often takes precedence, rendering global XAI insufficient on its own [24]. While local XAI techniques offer the potential for model prediction validation, they often rely heavily on human assessment to understand and interpret model behavior. The labor-intensive process of human assessment, coupled with the potential for human bias [16], hinders the practical implementation of these systems in critical applications. Therefore, a need for a more efficient and reliable means of understanding and validating safety of ML models remains.

In this work, we address this challenge and propose a novel concept-based XAI framework named Prototypical Concept-based Explanations (PCX), that signals and reveals deviations from expected model behavior by providing meaningful and more objective explanations, reducing the need for (possibly biased) human interpretation. Concretely, for any prediction, PCX communicates the differences and similarities to the expected model behavior via (automatically discovered) prototypes. Here, prototypes are representative predictions, that summarize the global model behavior in condensed fashion. To guarantee high interpretability throughout, we build upon the latest progress made in concept-based XAI, offering explanations in terms of human-understandable concepts [1, 18], applicable to

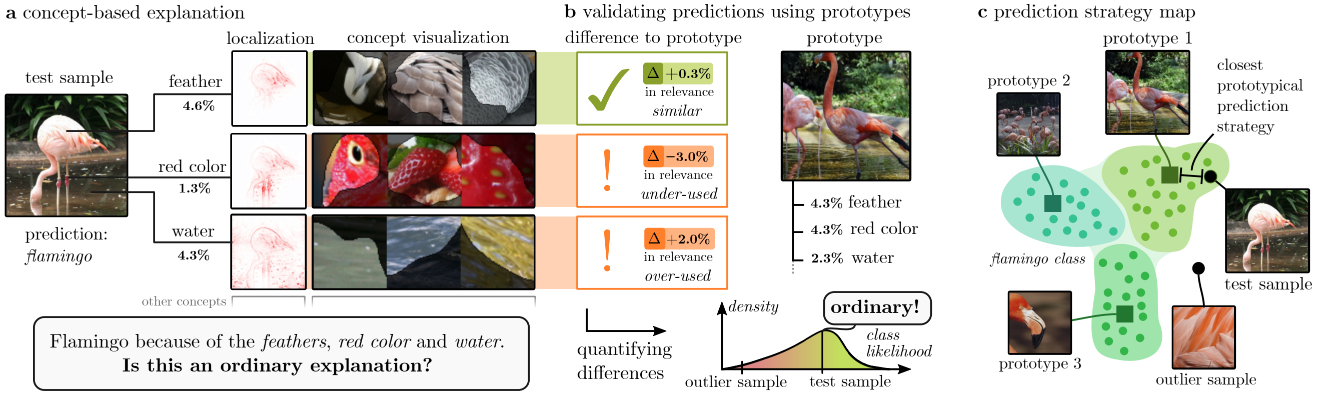

Figure 1. Using the PCX framework: By contrasting a prediction with the prototypical prediction strategy, the stakeholder can understand how (un-)ordinarly the model behaves. (**a**): A flamingo prediction is based on concepts like "feather", "red color" and "water". While recent concept-based XAI methods provide relevance scores, localization heatmaps, and visualizations for each concept, it remains unclear whether such composition of used concepts is expected. (**b**): Comparing against prototypes enables to understand to what extend concepts are similar (*e.g*., "feather"), underused (*e.g*., "red color"), or overused (*e.g*., "water"). These differences can be quantitatively measured to assess the degree of an outlier prediction. (**c**): PCX allows to automatically identify outliers, or, alternatively, the closest prototypical prediction strategy. Prototypes are hereby automatically discovered, summarizing the global model behavior in condensed fashion.

any DNN architecture in a post-hoc manner. Notably, PCX can hereby not only highlight which concepts are used, but also which ones are *not* used. For the flamingo in Fig. 1a, *e.g*., we learn that the red color concept, typically present for the prototype in Fig. 1b, is underrepresented. Quantifying differences in latent feature use allows for an objective means to identify typical prediction strategies, *e.g*., for positively validating predictions, issuing warnings otherwise.

Contributions This work introduces PCX, an intrinsically explainable framework for validating DNN predictions that combines both class-wise *and* instance-wise prediction strategies. We show how PCX allows to

1. *locally* study predictions on the concept-level by leveraging state-of-the-art concept-based XAI techniques.
2. *globally* understand (dis-)similarities in prediction strategies within and across classes via prototypes. We further validate prototypes w.r.t. metrics such as faithfulness, stability and sparseness, demonstrating the superiority of concept relevance scores over activations.
3. *glocally* quantify and understand (un-)usual concept use by a model for individual predictions by comparing these to prototypes. We showcase PCX for detecting spurious model behavior, data quality issues and OOD samples.

2. Related Work

We now present an overview of related work in concept-based XAI, prototypical explanations and OOD detection.

2.1. Concept-based Explanations

Contrary to traditional local feature attribution methods that investigate the importance of input-level features, concept-based XAI methods study the function (concept) of latent representations in a specific layer of a DNN. Here, either single neurons [6], directions, *i.e*., Concept Activation Vectors (CAVs) [26], or feature subspaces [47] are investigated.

Early XAI works study how these concepts are used for *global* decision making, *e.g*., the concepts most relevant for an output class [26]. Recent works also generalize local feature attribution methods to compute importance scores of concepts for *individual* predictions [1, 18], bringing concept-based explanations to the instance level.

Whereas instance-wise concept-based explanations enable new levels of insight, they can be overwhelming and complex for a stakeholder to process, as hundreds of concepts might exist and need to be studied for each instance [1]. Hence, other works [10, 17] illustrate the advantage of visualizing local decisions in a global embedding, as also shown in Fig. 1c. With PCX, we extend this idea by introducing prototypes. This further reduces the need for human interpretation as it allows to compare individual (local) prediction strategies with prototypical (global) ones.

2.2. Prototypes for Explanations

Prototypes represent example predictions that summarize the global model behavior in condensed fashion, rendering them especially valuable for large and complex datasets. While there is a large group of works focusing on using prototypes for (robust) classification, *e.g*., [31, 51], few works use prototypes for XAI. The works of ProtoAttend [4] and [13] increase interpretability of DNNs by anchoring decisions on prototypical samples. Whereas both require modification of the DNN architecture as well as additional training, [13] highlights similar features between test sample and prototype. In contrast, PCX is *post-hoc* applicable and communicates similarities as well as *differences* via human-understandable concepts.

Prototypical parts are widely used in interpretable models [11, 34, 35]. For example, ProtoPNet [11], one of the pioneering works, dissects images into prototypical parts for each class, subsequently classifying images by consolidating evidence from prototypes. It is important to note, that whereas these works study prototypical *parts*, we define a prototype as a *prediction strategy*.

2.3. Out-of-Distribution Detection

A popular safety task in DNN deployment is OOD detection, catching data samples that are not of the training distribution. One line of research focuses on the fact that OOD samples often result in uncertain predictions, rendering the softmax output values already highly informative [22, 32]. Other works leverage, *e.g.*, the latent activations and measure divergence from typical patterns [30]. Notably, OOD methods are not intrinsically interpretable. Hence, first works introduce post-hoc concept-based explanations for OOD detectors [12, 42]. As PCX is rooted in concept-based XAI, it inherently provides interpretable OOD detection.

3. Methods

PCX is based on recent concept-based XAI techniques, which are introduced in Sec. 3.1. Using concept-based explanations, we define and compute prototypical explanations, discussed in Sec. 3.2, to which we then compare individual predictions as described in Sec. 3.3.

3.1. Concept-based Explanations

A model's prediction is the result of successive layer-wise feature operations, where the intermediate latent features of each layer are described by the activations of its neurons. Given a sample $\mathbf{x}$ of dataset $\mathcal{X}$, the latent activations $\mathbf{a}(\mathbf{x}) \in \mathbb{R}^n$ in a specific layer with n neurons can be viewed as a point in a vector space (activation space) that is spanned by n canonical basis vectors (one for each neuron).

Then, we can assign a concept c to each neuron, or more generally, also to a superposition of neurons describing a direction in latent space via a CAV $\mathbf{u}^c \in \mathbb{R}^n$. How the latent activations $\mathbf{a}(\mathbf{x})$ are decomposed into a linear combination of m (chosen) CAVs summarized by $\mathbf{U} = (\mathbf{u}^1, \dots, \mathbf{u}^m) \in \mathbb{R}^{n \times m}$ is described by the transformation

$$\mathbf{a}(\mathbf{x}) = \mathbf{U}\boldsymbol{\nu}^{\text{act}}(\mathbf{x}) \tag{1}$$

from concept space to activation space. Here, $\boldsymbol{\nu}^{\text{act}}(\mathbf{x}) \in \mathbb{R}^m$ summarizes the activation (contribution) of each of the m concepts. Depending on the choice of the set of CAVs, the decomposition might only be approximated, *e.g.*, when using non-negative matrix factorization [17]. For simplicity and to ensure exact reconstruction as in Eq. (1), we study the concepts of individual neurons in this work, *i.e.*, choose $\mathbf{u}^c = \mathbf{e}^c$ with canonical basis vectors $\mathbf{e}^c_i = \delta_{ci}$ with Kronecker delta δ leading to a direct mapping $\boldsymbol{\nu}^{\text{act}}(\mathbf{x}) = \mathbf{a}(\mathbf{x})$.

Concept Relevance Scores In order to attain an understanding of how concepts are *used* for individual samples $\mathbf{x}$ and prediction outcomes y_k, we require relevance scores $\nu^{\text{rel}}_c(\mathbf{x}|y_k)$ of each concept c. Concept relevance scores can be computed using various established feature attribution methods such as Input×Gradient [43] or Layer-wise Relevance Propagation (LRP) [5] (see [17] for an overview). When studying each neuron's concept, concept relevances $\nu^{\text{rel}}_c(\mathbf{x}|y_k)$ are directly given by applying a feature attribution method and aggregating the relevances in the latent space instead of the input space.

Concept Localizations Furthermore, we localize individual concepts in the input via heatmaps as shown in Fig. 1a. Specifically, we leverage the CRP framework [1] that enables concept-specific heatmaps by restricting the backward pass of feature attribution methods (with LRP by default).

Concept Visualizations Several works have proposed techniques to visualize concepts of latent representations [37]. We adhere to the recently proposed Relevance Maximization approach [1]. This technique explains concepts by exemplifying them, selecting reference samples that most accurately represent the functionality of a neuron. These reference samples highlight the input components most *relevant* to a specific concept, as shown in Fig. 1a.

3.2. Finding Prototypes

During training, a model learns to extract and use features from the input data to fulfill its training task. If we are to collect the *presence* of such features (given by latent activations), we can measure feature distributions that are characteristic for specific classes. The works of [30, 53] model such distribution via multivariate Gaussian distributions.

In our framework, we collect *relevances* instead of activations of features, which describe how features are utilized by the model in making specific predictions. Thus, relevances naturally filter out irrelevant activations and amplify useful features for the model's class prediction. As a result, relevances provide more precise and specific information regarding the encoded classes, as also illustrated in Fig. 2 (*bottom*) using UMAP embeddings for eight ImageNet class of feline species. Further, instead of assuming a single Gaussian distribution for each class, we model a class distribution via a multivariate Gaussian Mixture Model (GMM). This is motivated by the fact, that multiple sub-strategies can exist, *e.g.*, flamingos photographed from different perspectives or distances as shown in Fig. 1c.

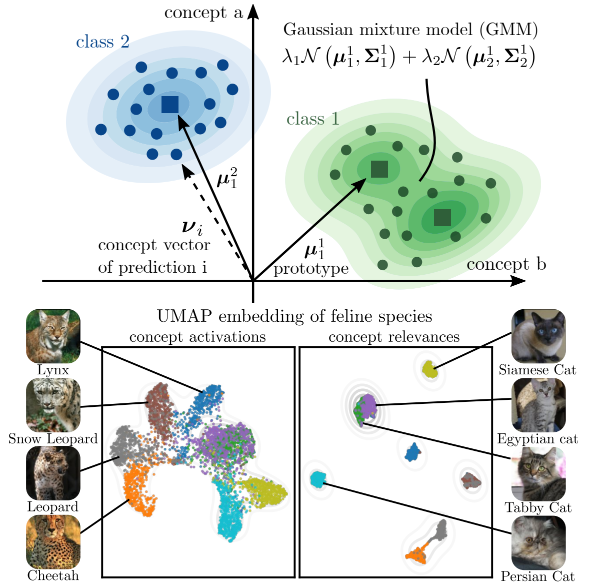

Figure 2. Intuition behind modeling prototypes: (*top*): In concept space, each dimension represents the relevance or activation of a concept. We assume, that concept vectors $\boldsymbol{\nu}$ of a specific class are forming distinct clusters that can be approximated by a mixture of Gaussian distributions (GMM). (*bottom*): Concept relevances (LRP ε-rule) result in more disentangled UMAP embeddings compared to activations. Shown are eight feline ImageNet classes (differently color-coded) for the VGG-16's last convolutional layer.

Concretely, as illustrated in Fig. 2 (*top*), we model the distribution p of concept attributions for each class k as

$$p^k = \sum_i \lambda_i^k p_i^k = \sum_i \lambda_i^k \mathcal{N}(\boldsymbol{\mu}_i^k, \boldsymbol{\Sigma}_i^k) \tag{2}$$

with $\lambda_i^k \geq 0$ and $\sum_i \lambda_i^k = 1$ to ensure that all probabilities add up to one. Here, $\boldsymbol{\mu}_i^k$ and $\boldsymbol{\Sigma}_i^k$ correspond to the means and covariance matrices of each Gaussian.

The probability density function of each Gaussian p_i^k is further given as

$$p_i^k(\boldsymbol{\nu}) = \frac{1}{(2\pi)^{\frac{n}{2}} \det(\boldsymbol{\Sigma}_i^k)^{\frac{1}{2}}} e^{-\frac{1}{2}\left(\boldsymbol{\nu}-\boldsymbol{\mu}_i^k\right)^\top \left(\boldsymbol{\Sigma}_i^k\right)^{-1} \left(\boldsymbol{\nu}-\boldsymbol{\mu}_i^k\right)} . \tag{3}$$

Having specified the number of Gaussians, GMMs then naturally provide prototypes, *i.e*., one for each Gaussian as in [25, 39]. In fact, we receive a mean and covariance matrix for each prototype, which with Eq. (3) allows to measure how likely a new point belongs to a prototype, further detailed in the following section. Note, that as the mean does not necessarily correspond to a training sample, we show the closest sample for illustration purposes.

To summarize, PCX requires a pre-processing step to find prototypes, as outlined in Fig. 3: For the training samples of each class, we compute concept relevance vectors on which a GMM is fitted to provide the prototypes.

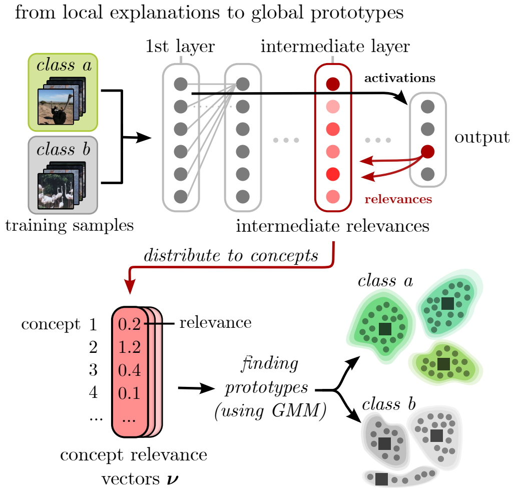

Figure 3. Pre-processing pipeline of PCX : DNN predictions are generated over training samples of a specific class. We further compute concept relevance scores for each prediction, representing prediction strategies. By fitting GMMs on the concept relevance vectors, we find prototypical prediction strategies.

3.3. Quantifying the (Extra-)Ordinary

Having modeled class prediction strategies via GMMs allows to compute the likeliness of new sample (prediction) to correspond to class k directly via the log-likelihood L^k

$$L^k(\boldsymbol{\nu}) = \log p^k(\boldsymbol{\nu}) . \tag{4}$$

Then, a prediction with concept relevances $\boldsymbol{\nu}$ corresponds most likely to the class given as $\text{argmax}_k L^k(\boldsymbol{\nu})$.

Analogously, we can also assign a test prediction to the most likely prototypical prediction strategy ρ^* given by

$$\rho^*(\boldsymbol{\nu}) = \text{argmax}_{k,i} \log p_i^k(\boldsymbol{\nu}) . \tag{5}$$

Other popular metrics that can be used to assign samples to prototypical predictions include the Mahalanobis distance and Euclidean distance, as further discussed in Appendix B.

Understanding (Dis-)Similarities A high likelihood as in Eqs. (4) and (5) directly results from small deviations in concept relevance scores between test and prototype prediction, given by the difference of concept relevance vectors

$$\boldsymbol{\Delta}_i^k(\boldsymbol{\nu}) = \boldsymbol{\nu} - \boldsymbol{\mu}_i^k . \tag{6}$$

Thus, we can understand, which concepts are over- and underused, corresponding to high and low entries in $\boldsymbol{\Delta}_i^k(\boldsymbol{\nu})$, respectively, or similar (with small entries in $|\boldsymbol{\Delta}_i^k(\boldsymbol{\nu})|$). It is to note that it is also possible to include information from the covariance matrices as in Eq. (3), allowing for an understanding of which concept *combinations* are (un-)usual, further described in Appendix D.1.

4. Experiments

We address the following research questions:

1. **(Q1)** What global insights can we gain with prototypes?
2. **(Q2)** How can we evaluate prototypes?
3. **(Q3)** How can we use prototypes to validate predictions and ensure safety?

Experimental Setting We use ResNet-18 [20], VGG-16 [44] and EfficientNet-B0 [46] architectures on ImageNet [40], CUB-200 [48] and CIFAR-10 [27]. Whereas models on ImageNet are pre-trained from the PyTorch model zoo, we train models on CUB-200 and CIFAR-10. Details on datasets and training are given in Appendix A.

4.1. (Q1) Prototypical Concept-based Explanations

Class prototypes allow us to inspect and understand the global prediction strategies of our model. Concretely, we can study the concepts most relevant for each prototype, with concept visualizations and localizations available for easier understanding of the concepts, as shown in Fig. 1a.

Comparing Class Strategies To gain a global understanding at one glance, we visualize the similarity in prediction strategies between all classes via a similarity matrix in Fig. 4a. Here, we compute the cosine similarity between class prototypes (one per class). Concretely, the prototypes of the VGG-16 model for the first 20 ImageNet classes are shown when using LRP-ε for concept relevances in layer `features.28`. There are apparent clusters with similar prediction strategies, *e.g.*, for fish and bird species.

At this point, we can compare individual class strategies, such as the Brambling and Robin bird species (more examples in Appendix D). Both seem to have similar concepts, as indicated by a similarity of 80 %. Whereas both show orange-brown color in parts, they differ in a "gray-white spotted" texture (indicating Brambling) and the combination of brown, white and black patches (indicating Robin).

Prototypes for Data Quality and Annotation When examining *multiple* prototypes per class, we gain a post-hoc understanding of a model's sub-strategies for decision-making, including, *e.g.*, prototypes for different types of hens, or habitats for the ice bear class. This in turn shows promise for large-scale data annotation by assigning samples to the respective prototypes. While studying prototypes, we encounter a multitude of problems within the ImageNet [40] (train) dataset such as wrong labels, poor data quality, and correlating features (shortcuts). All following and additional examples are provided in Appendix C.6.

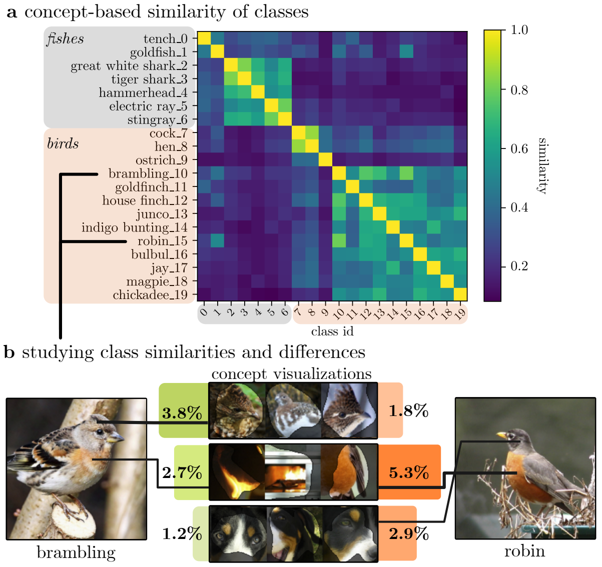

Figure 4. Prototypes allow for a global understanding of class prediction (dis-)similarities. (**a**) Similarity matrix of the first 20 ImageNet class prototypes. We can identify distinct clusters for fishes and bird species. (**b**) Unraveling the (dis-)similarities of the Brambling and Robin prototype: Whereas both are similar in terms of orange-brown color in parts, they differ , *e.g.*, in a "gray-white spotted" texture (indication for Brambling).

Wrong Labels: We identify prototypes for objects not corresponding to the actual label, likely due to a similar naming, including tigers for "Tiger Cat", buses for "passenger car" and Leopard Lacewing butterflies for "lacewing".

Correlating Features: We reveal various correlations that the model exploits for shortcut-learning, distinctly visible in prototypes, including cats in cartons or buckets, dogs with tennis balls, and white wolves or lynxes behind fences.

Poor Data Quality: We further observe data quality issues, such as large numbers of low-resolution images that result in dedicated "blur" concepts relevant for, *e.g.*, detecting milk cans, red-breasted merganser or ties. Also, objects are cropped out of images in some cases due to, *e.g.*, data augmentation, as for the "pickelhaube" class, stressing the need to inspect data after applying augmentation.

4.2. (Q2) Evaluating Prototypes

To evaluate prototypical explanations, three established XAI evaluation criteria in literature [17, 21] are applied, namely *faithfulness*, *stability* and *sparseness*. We further introduce a *coverage* measure and perform outlier detection.

Faithfulness: Local XAI methods compute importance scores of features, which can either be input features or latent features. One of the most popular evaluation methods to check an explanation's faithfulness to the model is to perform feature deletion [36]. Concretely, the most important features (according to a chosen feature attribution method) are removed successively, *i.e.*, set to a baseline value, and the change in model confidence is measured. A faithful ex-

Table 1. Evaluating different attribution methods for concept relevance scores used for prototypes. We show results on ImageNet for 20 classes using (VGG | ResNet | EfficientNet) architectures averaged over all layers, where higher (↑) values are better and best are bold.

	Faithfulness (↑)	Stability (↑)	Sparseness (↑)	Coverage (↑)	Outlier Detection (↑)
LRP (ε-rule) [5]	12.2 | **14.2** | 7.4	91.7 | 90.6 | 96.1	**37.1** | **36.6** | **37.0**	**56.4** | **66.5** | **71.3**	**70.9** | **78.8** | **82.8**
Input×Gradient [43]	12.2 | **14.2** | 6.7	91.8 | 90.7 | 84.1	**37.1** | **36.6** | 35.5	56.2 | 66.3 | 50.8	70.4 | 78.3 | 72.9
LRP (composite) [33]	**12.6** | 13.6 | **7.5**	98.1 | 99.0 | 99.0	21.0 | 22.8 | 14.0	42.3 | 56.5 | 50.2	65.6 | 73.3 | 68.2
GuidedBackProp [45]	12.0 | 13.0 | 6.0	98.7 | **99.3** | 85.9	31.1 | 30.9 | 31.8	43.6 | 59.3 | 53.2	66.2 | 74.8 | 73.5
Activation (max)	11.9 | 12.5 | 6.3	**99.3** | 99.2 | **99.1**	7.1 | 4.9 | 9.8	27.5 | 39.5 | 36.1	54.3 | 57.7 | 57.3
Activation (mean)	11.1 | 13.1 | 5.9	98.7 | 98.8 | 92.8	11.4 | 12.2 | 24.0	24.8 | 41.6 | 36.1	55.8 | 60.8 | 60.2

planation is assumed to result in a strong confidence drop, when the most important features are removed. In our case, the most relevant concepts according to the nearest class prototype are removed (*i.e.*, set to zero activation), and the change in the class output logit measured. To receive a final score, the Area Under the Curve (AUC) is computed.

Stability: To evaluate stability, we compute five prototypes on k-fold subsets of the data ($k = 10$ as default). We then map prototypes together using a Hungarian loss function [28] and measure the cosine similarity between vectors.

Sparseness: We compute the cosine similarity between the absolute value of the prototype vector (*i.e.*, centroid $\boldsymbol{\mu}$) and the unit vector, which represents a uniform distribution of concept attributions. The less similar a prototype is to the unit vector, the more sparse, and easier to interpret.

Coverage: To measure how well prototypes model the underlying distributions and are suitable to assign correct prediction strategies, we introduce the *coverage* metric. The task is to correctly assign sample predictions from a hold-out set correctly to known sub-strategies using Eq. (5). We therefore compute eight prototypes on concept attributions from eight (animal) classes of the same family. Such a setting is illustrated in Fig. 2 (*bottom*) for feline species. Details on the groups of classes are given in Appendix C.

Outlier Detection: GMMs do not only allow to assign samples to prototypes, but also to detect outliers. We adhere to the same setting as for the *coverage* metric, but now measure how well we can detect outliers (from other classes of the same family) using Eq. (4). Concretely, the AUC is measured when plotting the true positive rate over false positive rate under a varying detection threshold.

In the following, we evaluate the influence of various degrees of freedom of our approach. This includes the choice of the underlying attribution method to compute concept relevance scores, and the number of prototypes used to fit the GMM. Note, that the concept basis $\mathbf{U}$, as in Eq. (1), is also variable. We refer to [17] for a thorough comparison of various techniques to compute concept bases. Further note, that we average the evaluation scores computed over multiple model layers (layers are detailed in Appendix A).

4.2.1 Evaluating Concept Attribution Methods

We compare (modified-)gradient-based attribution methods to compute concept *relevances*, including LRP variants [5, 33], Input×Gradient [43], GuidedBackProp [45] and *activation* with max- and mean-pooling (details in Appendix C). Notably, we refrain from using other popular methods such as SHAP and GradCAM due to their inefficiency or inapplicability, as discussed in Appendix C.

When comparing the results in Tab. 1 (standard errors reported in Appendix C) for models on ImageNet, it is apparent that relevance scores (computed via local XAI methods) are not only more faithful than activation values, but also more sparse, similarly observed in [15]. Further, relevances lead to better coverage and outlier detection scores, indicating higher disentanglement of distributions, as also observed in Fig. 2 (*bottom*). Generally, higher-level layers result in better scores, as shown in Appendix D.

Overall, LRP (ε-rule) relevances result in high faithfulness, sparseness, coverage and outlier detection scores, by still providing stable prototypes. Thus, in the following, we use LRP (ε-rule) relevances.

4.2.2 Varying the Number of Prototypes

Increasing the number of prototypes allows for a more fine-grained, but also more complex understanding. Interestingly, *faithfulness* does not significantly increase with the number of prototypes, as shown in Fig. C.1 of Appendix C. It could be expected, that the closer a prototype to the actual sample (which is more probable for a larger number of prototypes), the higher the faithfulness score. Apparently, this is true when removing the first concepts, but for later stages the summarizing (global) effect of few prototypes seems to be favorable, as further detailed in Appendix C.

There are, however, clear trends regarding *stability*, that is decreasing, and *sparseness*, which is increasing. Further, as can be expected, *coverage* and outlier detection scores are improving as well. All trends are depicted in detail in Fig. C.1 of Appendix C. We also provide qualitative examples for different prototype numbers in Appendix D.2.

4.2.3 Using GMMs for Improved Clustering

The k-means clustering algorithm represents a simpler alternative to GMMs for finding prototypes that is not based on covariance estimation. In fact, k-means is commonly used as a starting point to fit GMMs [38]. Compared to k-means, GMMs lead to improved coverage and outlier detection scores, as further shown and discussed in Appendix C.3. Notably, the covariance information is especially improving outlier detection for small numbers of prototypes.

4.3. (Q3) Validating Predictions

In this section, we leverage prototypes to validate predictions, reveal spurious model behavior, and identify OOD samples in a human-interpretable, yet automated, manner.

To achieve these goals, we employ a two-step approach: First, we compute the class likelihood as in Eq. (3), which provides a quantitative measure of how unusual a sample is to the model. This score allows to objectively assess spurious predictions and OOD samples in Secs. 4.3.1 and 4.3.2, respectively. To provide detailed human-understandable information, we secondly proceed to compute the difference between concepts relevance values with the prototypes, allowing to identify which features are over- or underused in the context of the given sample.

Spotting Differences: In Fig. 1a, a sample is predicted as "flamingo" with a class likelihood slightly below the ordinary. We begin by studying the deviation of relevance values in Fig. 1b between sample and prototype. The sample shows strong relevance on the water concept, suggesting an unusual amount of water in the background. Additionally, the comparatively lower relevance on the redness concept indicates that the depicted flamingo lacks the expected level of redness. Notably, we can also analyze deviations to prototypes of other classes, offering counterfactual insights.

Aligning to Prototypical Strategies: The sample closely aligns with prototype 1, representing flamingos standing in water. By understanding the underlying prototypes we can thus not only understand the data (or domain) as per the model's perception, but also to identify similar data instances, *e.g.*, more flamingos in water, useful for data annotation purposes. We further want to remark the idea of tracking prototypes during training, possibly giving insights into challenges such as (detecting) data drift [23].

4.3.1 Use Case: Spurious Model Behavior

Several works in the field of XAI have tackled the problem of revealing spurious model behavior. The usual approach is to study outliers, *e.g.*, in local explanations [3, 29, 41] or latent representations [8, 9, 49]. Whereas PCX allows to find and study outlier predictions, we also want to highlight the study of ordinary, *i.e.*, prototypical predictions.

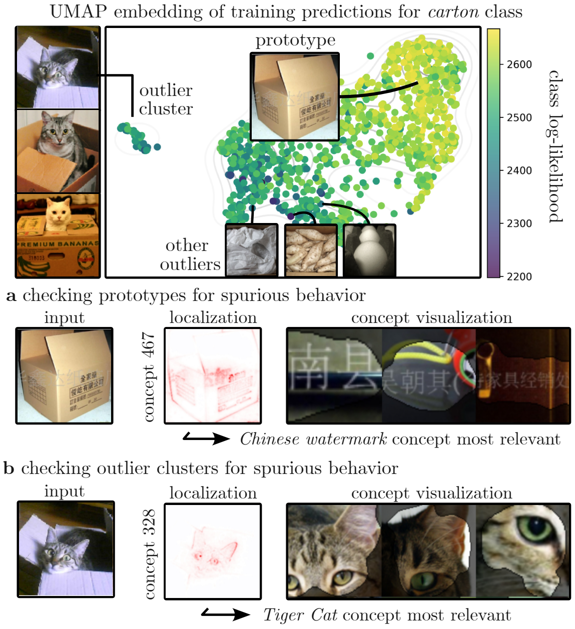

Figure 5. Revealing spurious model behavior with PCX: (**a**) Firstly, we examine the characteristic concepts of each prototype to find spurious concepts. As shown, a spurious Chinese watermark concept is most relevant for the prototype of the "carton" class. (**b**) Secondly, clusters of training predictions that deviate strongly from prototypes can be studied for spurious behavior. For the "carton" class, we reveal a cluster of Tiger Cats in cartons, that lead to the model using cat features to predict the carton class.

An illustrative example is given by the ImageNet "carton" class, as shown in Fig. 5a. Here, we reveal that most characteristic for the class prototype is a gray "Chinese watermark" concept which is overlaid over the entire image (best observed in digital print).

Furthermore, our analysis extends beyond individual outlier samples to include the study of entire outlier clusters within the training set. These outlier clusters represent instances that significantly deviate from the norm and often reveal surprising and unexpected model strategies. In Fig. 5b, we discover an outlier cluster consisting of samples depicting cartons with cats inside, leading the model to utilize cat-related features to increase its prediction confidence. Notably, the cat cluster receives its own prototype as we increase the number of prototypes (see Appendix D).

This example underlines the value of examining prototypes to ensure safety. Notably, once examined, they allow for each new prediction to be automatically validated and understood when assigned to a prototype (and no outlier).

4.3.2 Use Case: Out-of-Distribution Detection

In Sec. 4.2, we use PCX to detect outlier predictions that deviate from prototypes. In the following, we investigate the

Table 2. OOD detection results for (VGG|ResNet|EfficientNet) models trained on CUB-200. Higher AUC scores are better with best bold.

	LSUN	Places365	Textures	ImageNet	Average
MSP [22]	99.2 | 98.8 | 98.9	91.2 | 93.9 | 88.4	89.2 | 91.6 | 89.7	85.3 | 90.7 | 87.0	92.0
Energy [32]	**100.0** | 99.8 | 47.7	97.3 | 96.1 | 84.4	94.7 | 95.0 | 63.9	89.7 | 93.2 | 87.2	87.4
Mahalanobis [30]	16.9 | 74.4 | 53.1	80.3 | 95.9 | 90.0	92.1 | 96.9 | 95.6	89.4 | 95.7 | 89.6	80.8
PCX-E (ours)	99.9 | 99.8 | 99.8	95.9 | 97.4 | 94.9	98.7 | 98.9 | 98.6	93.2 | 95.7 | 92.7	97.1
PCX-GMM (ours)	**100.0** | **99.9** | **99.9**	**97.9** | **98.5** | **96.1**	**99.3** | **99.3** | **98.8**	**95.9** | **97.2** | **93.6**	**98.0**

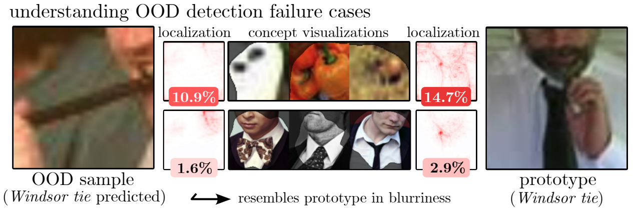
understanding OOD detection failure cases
localization
concept visualizations
localization
10.9%
14.7%
1.6%
2.9%
OOD sample
(*Windsor tie* predicted)
resembles prototype in blurriness
prototype
(*Windsor tie*)

Figure 6. Understanding why an OOD detection is classified as in-distribution: For the model, the blurry OOD sample is similar to a "Windsor tie" prototype due to the high relevance of blurring (top concept). This suggests a potential flaw in the model as it relies on a blur concept rather than the actual "tie-like" concept.

effectiveness of our approach for detecting OOD samples, and compare against established and dedicated methods in literature, namely MSP [22] based on softmax probabilities, Energy [32] and Mahalanobis [30]. The task is to detect samples from unrelated datasets such as LSUN [52], iSUN [50], Textures [14], SVHN [54] and Places365 [55]. For PCX, we perform OOD detection by measuring the likelihood for the predicted class using Eq. (4) (PCX-GMM), and alternatively also compute the Euclidean distance to the closest prototype of the predicted class (PCX-E). To evaluate OOD detection performance, we report the AUC when plotting the true positive rate over false positive rate under a varying detection threshold. PCX and Mahalanobis are hereby based on features of the last convolutional layer.

For the models trained on CUB-200, PCX is most effective for OOD detection (results given in Tab. 2). Note that we exclude bird species from ImageNet here. For CIFAR-10 and ImageNet models, the Energy method performs slightly better on average, as shown in Appendix D.4.

Importantly, PCX is intrinsically explainable (contrary to other dedicated OOD detection methods), allowing to understand why OOD samples are (falsely) classified as in-distribution. In Fig. 6, a sample from LSUN (predicted as "Windsor tie" by a VGG-16 on ImageNet) is similar to a class prototype, because of a "blur" concept relevant for both. This reveals that the model has learned to associate blurred images with the "Windsor tie" class, as many low-resolution training samples exist. Thus, by understanding OOD failure cases, we can reveal flaws of the model itself.

5. Limitations and Future Work

PCX relies on estimating covariances for GMMs which becomes unstable with few data points. To automatically specify the number of prototypes, studying approaches as in [2, 35] is of interest. Further, how to choose a concept basis (*i.e.*, $\mathbf{U}$) with optimal human-interpretability is still an open question in concept-based XAI literature. Improving concepts will also further increase the usefulness of PCX.

6. Conclusion

PCX is a novel concept-based XAI framework that brings prototypes to local explanations, providing more objective and informative explanations of DNNs in a post-hoc manner. Concept-based prototypes hereby enable to study the model behavior on the whole training data efficiently and in great detail, allowing to understand sub-strategies and issues with the data. As PCX bases prototype extraction on GMMs, we receive effective quantitative measures for in- and outlier detection. By assessing the difference to the prototypical model behavior, PCX reduces the reliance on human assessment and allows for a scalable analysis of large sets of predictions. We demonstrate the value of our method in detecting spurious behavior by studying not only outliers, but also inliers, *i.e.*, the prototypes. Further, PCX shows not only effective for OOD detection, but is simultaneously interpretable, revealing flaws of the model itself through missed OOD detections. This work firstly introduces post-hoc concept-based prototypes, showcasing XAI's potential for broader applicability in ML validation and safety.

Acknowledgements

We thank Johanna Vielhaben for her valuable feedback. This work was supported by the Federal Ministry of Education and Research (BMBF) as grant BIFOLD (01IS18025A, 01IS180371I); the German Research Foundation (DFG) as research unit DeSBi (KI-FOR 5363); the European Union's Horizon Europe research and innovation programme (EU Horizon Europe) as grant TEMA (101093003); the European Union's Horizon 2020 research and innovation programme (EU Horizon 2020) as grant iToBoS (965221); and the state of Berlin within the innovation support programme ProFIT (IBB) as grant BerDiBa (10174498).

References

[1] Reduan Achtibat, Maximilian Dreyer, Ilona Eisenbraun, Sebastian Bosse, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. From attribution maps to human-understandable explanations through concept relevance propagation. *Nature Machine Intelligence*, 5(9):1006–1019, 2023. 1, 2, 3

[2] Kelsey Allen, Evan Shelhamer, Hanul Shin, and Joshua Tenenbaum. Infinite mixture prototypes for few-shot learning. In *International conference on machine learning*, pages 232–241. PMLR, 2019. 8

[3] Christopher J Anders, Leander Weber, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Finding and removing clever hans: using explanation methods to debug and improve deep models. *Information Fusion*, 77:261–295, 2022. 7

[4] Sercan O Arik and Tomas Pfister. Protoattend: Attention-based prototypical learning. *J. Mach. Learn. Res.*, 21:210:1–210:35, 2020. 2

[5] 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, 6

[6] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017. 2

[7] Titus J Brinker, Achim Hekler, Alexander H Enk, Joachim Klode, Axel Hauschild, Carola Berking, Bastian Schilling, Sebastian Haferkamp, Dirk Schadendorf, Tim Holland-Letz, et al. Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task. *European Journal of Cancer*, 113:47–54, 2019. 1

[8] Kirill Bykov, Mayukh Deb, Dennis Grinwald, Klaus Robert Muller, and Marina MC Höhne. Dora: Exploring outlier representations in deep neural networks. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023. 7

[9] Kirill Bykov, Laura Kopf, Shinichi Nakajima, Marius Kloft, and Marina MC Höhne. Labeling neural representations with inverse recognition. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 7

[10] Gromit Yeuk-Yin Chan, Enrico Bertini, Luis Gustavo Nonato, Brian Barr, and Claudio T Silva. Melody: Generating and visualizing machine learning model summary to understand data and classifiers together. *arXiv preprint arXiv:2007.10614*, 2020. 2

[11] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 3

[12] Jihye Choi, Jayaram Raghuram, Ryan Feng, Jiefeng Chen, Somesh Jha, and Atul Prakash. Concept-based explanations for out-of-distribution detectors. In *International Conference on Machine Learning*, pages 5817–5837. PMLR, 2023. 3

[13] Penny Chong, Ngai-Man Cheung, Yuval Elovici, and Alexander Binder. Toward scalable and unified example-based explanation and outlier detection. *IEEE Transactions on Image Processing*, 31:525–540, 2021. 2

[14] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. 8

[15] Maximilian Dreyer, Reduan Achtibat, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. Revealing hidden context bias in segmentation and object detection through concept-specific explanations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3828–3838, 2023. 6

[16] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019. 1

[17] Thomas Fel, Victor Boutin, Mazda Moayeri, Rémi Cadène, Louis Bethune, Mathieu Chalvidal, Thomas Serre, et al. A holistic approach to unifying automatic concept extraction and concept importance estimation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2, 3, 5, 6

[18] Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2711–2721, 2023. 1, 2

[19] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020. 1

[20] 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, 2016. 5

[21] Anna Hedström, Leander Weber, Daniel Krakowczyk, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M-C Höhne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, 24(34):1–11, 2023. 5

[22] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2016. 3, 8

[23] T Ryan Hoens, Robi Polikar, and Nitesh V Chawla. Learning from streaming data with concept drift and imbalance: an overview. *Progress in Artificial Intelligence*, 1:89–101, 2012. 7

[24] Yan Jia, John McDermid, Tom Lawton, and Ibrahim Habli. The role of explainability in assuring safety of machine learning in healthcare. *IEEE Transactions on Emerging Topics in Computing*, 10(4):1746–1760, 2022. 1

[25] Ruijin Jiang and Zhaohui Cheng. Mixture gaussian prototypes for few-shot learning. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 902–908. IEEE, 2021. 4

[26] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 2

[27] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[28] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6

[29] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10: 1096, 2019. 1, 7

[30] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018. 3, 8

[31] Cheng-Lin Liu and Masaki Nakagawa. Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition. *Pattern Recognition*, 34(3):601–615, 2001. 2

[32] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020. 3, 8

[33] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019. 6

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

[35] Meike Nauta, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. Pip-net: Patch-based intuitive prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2744–2753, 2023. 3, 8

[36] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Computing Surveys*, 55(13s): 1–42, 2023. 5

[37] Anh Nguyen, Jason Yosinski, and Jeff Clune. Understanding neural networks via feature visualization: A survey. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 55–76. Springer, Cham, 2019. 3

[38] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 7

[39] Cristiano Pereira and George Cavalcanti. Prototype selection: Combining self-generating prototypes and gaussian mixtures for pattern classification. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 3505–3510. IEEE, 2008. 4

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5

[41] Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020. 7

[42] Laya Rafiee Sevyeri, Ivaxi Sheth, Farhood Farahnak, and Shirin Abbasinejad Enger. Transparent anomaly detection via concept-based explanations. *arXiv preprint arXiv:2310.10702*, 2023. 3

[43] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017. 3, 6

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[45] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015. 6

[46] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 5

[47] Johanna Vielhaben, Stefan Bluecher, and Nils Strodthoff. Multi-dimensional concept discovery (mcd): A unifying framework with completeness guarantees. *Transactions on Machine Learning Research*, 2023. 2

[48] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-UCSD birds 200. 2010. 5

[49] Shirley Wu, Mert Yuksekgonul, Linjun Zhang, and James Zou. Discover and cure: Concept-aware mitigation of spurious correlation. In *Proceedings of the 40th International Conference on Machine Learning*. JMLR.org, 2023. 7

[50] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015. 8

[51] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3474–3482, 2018. 2

[52] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 8

[53] Mert Yuksekgonul, Linjun Zhang, James Zou, and Carlos Guestrin. Beyond confidence: Reliable models should also consider atypicality. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[54] Netzer Yuval. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 8

[55] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Advances in neural information processing systems*, 27, 2014. 8