Uncovering Unique Concept Vectors through Latent Space Decomposition

Anonymous authors
Paper under double-blind review

Abstract

Interpreting the inner workings of deep learning models is crucial for establishing trust and ensuring model safety. Concept-based explanations have emerged as a superior approach that is more interpretable than feature attribution estimates such as pixel saliency. However, defining the concepts for the interpretability analysis biases the explanations by the user's expectations on the concepts. To address this, we propose a novel post-hoc unsupervised method that automatically uncovers the concepts learned by deep models during training. By decomposing the latent space of a layer in singular vectors and refining them by unsupervised clustering, we uncover concept vectors aligned with directions of high variance that are relevant to the model prediction, and that point to semantically distinct concepts. Our extensive experiments reveal that the majority of our concepts are readily understandable to humans, exhibit coherency, and bear relevance to the task at hand. Moreover, we showcase the practical utility of our method in dataset exploration, where our concept vectors successfully identify outlier training samples affected by various confounding factors. This novel exploration technique has remarkable versatility to data types and model architectures and it will facilitate the identification of biases and the discovery of sources of error within training data.

1 Introduction

Understanding and explaining the inner workings of deep learning models is challenging, yet of outmost importance in high-risk applications where reliable and intuitive explanations are crucial for decision making. Model validation plays a vital role in ensuring trustworthy predictions and to avoid actions that may negatively impact society.

Because of their ease of use and understandability, the use of user-defined queries to inquire about the relevance of high-level concepts such as objects, shapes, and textures, has shown great promise in model validation. These concept-based queries are addressed through explainability methods that leverage concept vectors, representing vectors in the latent space of a layer that are representative of a concept (Kim et al., 2018). Methods further developing on concept vectors were proposed for multiple applications and tasks (Goyal et al., 2019; Graziani et al., 2020; Koh et al., 2020; Chen et al., 2020). However, concept vectors are generally obtained as a response to user-defined queries and are, as such, biased towards the user's knowledge and expectations, a phenomenon known as the experimenter bias Rosenthal & Fode (1963). Users may bias the analysis by only querying the model for some concepts while neglecting others. The exhaustive coverage of all possible concepts, besides, is unfeasible and in some domains such as biology or chemistry, the concepts may be unclear or difficult to define. Because of this, model interpretability with concept vectors is, as of now, limited to concepts that are well-identifiable and easy to label.

Achieving broader interpretability with concept vectors requires a reverse engineering approach that focuses on automating concept identification. In this case, the only assumption made on the concepts is that they are relevant for the models' downstream task. While a few methods have ventured in the clustering of imaging representations based on pixel similarities (Ghorbani et al., 2019) , only limited analyses were conducted on the factorization of layer activations (Raghu et al., 2017; McGrath et al., 2022). These methods aim at finding a local coordinate system of the latent manifold that meaningfully explains how the network represents the training data. In this work, we propose a we propose a comprehensive framework for concept discovery that can be applied across multiple architectures, tasks, and data types. Our novel approach involves the automatic discovery of concept vectors by decomposing a layer's

latent space into singular values and vectors. By analyzing the latent space along these new axes, we identify vectors that correspond to semantically *unique* and distinguishable concepts.

To evaluate the effectiveness of our approach, we apply it to the task of natural image classification. The discovered concepts align with human-understandable interpretations and demonstrate consistency with previous research findings. For instance, texture and object parts emerge as prominent in vision models, reflecting the work of Kim et al. (2018) on recognizing zebra stripes and Ghorbani et al. (2019) on detecting vehicle parts to classify police vans. Furthermore, user evaluation studies confirm that the discovered concepts are easy to understand, and show that they provide valuable insights about the model's decision making process.

2 Related Work

This work focuses on post-hoc interpretability, hence it aims at explaining already trained models. Despite having acknowledged limitations (Rudin, 2019; Adebayo et al., 2018), post-hoc approaches allow us to gather important insights on de-facto standard models for vision tasks such as Inception (Szegedy et al., 2016) and ResNet (He et al., 2016), which are largely reused in real-world applications.

Concept activation vectors (CAVs) (Kim et al., 2018), in particular, introduced the hypothesis that linear combinations of units carry information on user-defined high-level concepts. Ghorbani et al. further extended the method to remove the need for user-defined queries (Ghorbani et al., 2019). The algorithm identifies image segments that cluster in the latent space, and uses CAVs to determine if the direction of a cluster is a relevant concept for the model. The identified vectors, however, depend on the quality of the image segments, which is not guaranteed for medical images (Graziani et al., 2021). Moreover, CAVs are unlikely to be orthogonal in the latent space. Counterintuitively, two CAVs for unrelated concepts may be closely aligned in the latent space. To alleviate this, (Chen et al., 2020) proposed to train a concept whitening layer that ensures the orthogonality of the concept vectors in its latent space.

On a parallel line, an analysis of deep networks based on SVD was proposed by (Raghu et al., 2017), where canonical correlation is used to test the similarity of two compressed representations of different model layers. Here the activations of a single neuron represent a vector in the canonical basis of the latent space. Following this line, non-negative matrix factorization was used to identify meaningful vectors to interpret AlphaZero (McGrath et al., 2022). Our work takes inspiration from all of these works to introduce a novel method that does not rely on user-defined concepts or pixel segmentation, that does not require the training of additional concept whitening layers, and that leads to independent vectors pointing to human-understandable concepts.

3 Methods

Based on the observations of Kim et al., we assume that directions in the latent space of a layer can identify the concepts used by a model for inference (Kim et al., 2018). However, while they rely on user-defined concept examples to find the concept vectors, here we aim to reverse-engineer the problem. Our objective is to identify concept vectors that directly emerge as relevant directions impacting the model decision function. We further assume, as suggested by (Chen et al., 2020), that concept vectors should be orthogonal to each other to maximize the separability of the concepts. the objective of our method is to identify the set of orthonormal vectors in the latent space that carry most of the information, i.e. that most impact the model's prediction.

The method consists of three phases:

1. variance alignment via the singular value decomposition (SVD) of the activations of a layer;
2. ranking of the singular vectors based on directional output sensitivity;
3. selection of the top ranked vectors and refinement of the direction to isolate *unique* concepts.

3.1 Notation

We consider the prediction function of a neural network $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$ from an m-dimensional input vector to an n-dimensional output vector. We assume the model was already trained using a dataset consisting of N labeled

pairs of input data points and labels $\{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^m \times \mathbb{R}^n$. Given an arbitrary layer l, the neural network can be seen as a composition of a feature extractor $\phi^l : \mathbb{R}^m \to \mathbb{R}^d$ and a downstream predictor $\psi^l : \mathbb{R}^d \to \mathbb{R}^n$, i.e. $f(x) = \psi^l(\phi^l(x))$. In the following, to simplify notation, and since we consider a single layer at a time, we drop the superscript identifying the layer ls. Furthermore, for a given input x_i, we use the shorthand $\phi_i = \phi(x_i)$. We are interested in finding M orthonormal vectors $u_1, .., u_M \in \mathbb{R}^d$.

In the following, we present the method for $n = 1$ and for densely connected layers. The method can be extended to $n > 1$, for example multi-class classification by applying the same construction to each of the outputs. Similarly, for convolutional layers, a pooling operation is introduced to obtain a d-dimensional representation. Details about both extensions are given in Section 3.4.

3.2 Step 1: SVD

Step one aims at identifying the vectors that best summarize the encoding of the dataset in the latent space of a layer l (with dimension d). To this end, we apply SVD to the entire layer's response to the input dataset. Such matrix $\Phi \in \mathbb{R}^{d \times N}$ is obtained by column-stacking the latent representations ϕ_i, with $i = 1, \ldots, N$. By applying SVD, we obtain:

$$\Phi = U \Sigma V^T \tag{1}$$

Note, $U \in \mathbb{R}^{d \times d}$, $V^T \in \mathbb{R}^{N \times N}$, and Σ is a diagonal matrix of singular values $\sigma_1, \ldots, \sigma_d$. The left singular vectors are the columns of the matrix U, and they align with the variance in Φ. The singular values rank the directions from the largest to the lowest observed variance.

3.3 Step 2: Sensitivity-based ranking

At this point, we are only making use of the feature extractor ϕ, and we are not considering whether the singular vectors are actually used by the downstream predictor. This means that the vectors found in the previous step might not be particularly relevant to the downstream predictive task. This second step evaluates the sensitivity of the output function along the directions of the singular vectors. The sensitivity represents the impact of perturbations of the feature representation of each input along the direction of the singular vector. As in (Kim et al., 2018), this is computed as the derivative of the model output along the direction of the singular vectors. This operation is computed for all the singular vectors in U. Consider the gradient of the downstream predictor ψ (Section 3.1) with respect to the latent space, which we denote $\nabla_\phi \psi_i$ for input ϕ_i. To compute the directional derivatives, we rotate the gradients to align with the singular vectors in U: $\tilde{\nabla}_\phi \psi_i = U^T \nabla_\phi \psi_i$. At this point, it is important to note that an input data point may have large gradients for low-activation features, and vice-versa, high activation values may be annihilated by close-to-zero gradients. Therefore, we consider the joint impact of gradients and activations together. This operation is fundamental, as it reduces the gradient noise that can be derived by small gradient values at the end of model training. Let

$$\tilde{\phi}_i = U^T \phi_i \tag{2}$$

denote the coefficients of ϕ_i rotated to align with the singular vectors. We consider the (element-wise) product between the coefficients of the rotated activations $\tilde{\phi}_i$ and of the rotated gradients $\tilde{\nabla}_\phi \psi_i$:

$$\tilde{g}_i = \tilde{\nabla}_\phi \psi_i \odot \tilde{\phi}_i \tag{3}$$

Finally, to compute the ranking, we consider the overall importance of a singular vector to the prediction. This is simply given by the sample mean of the g_i over all inputs i in the dataset: $\tilde{g} = \frac{1}{N} \sum_i \tilde{g}_i$. Note, $\tilde{g} \in \mathbb{R}^d$. For simplicity of the notation, we drop the tilde in the rest of the paper. Simply put, this step replaces the conventional ranking of the singular vectors given by the values in Σ with a new ranking based on the value of the coefficients of the rotated gradients.

3.4 Extension to Vision Classification Tasks

Let us demonstrate how to extend Steps 1 and 2 to vision classification tasks. We consider a convolutional neural network (CNN) $f : \mathbb{R}^{h' \times w' \times c'} \to \mathbb{R}^K$ classifying an input image in $n = K$ classes. Particularly, $f_{i,k}$ is the predicted probability of input x_i belonging to class k , namely $p(y = k|x = x_i)$. N_k is the number of samples in class k, with

$N = \sum_{k=1}^{K} N_k$. For convolutional layers, the feature extraction is $\phi^l : \mathbb{R}^{h' \times w' \times c} \rightarrow \mathbb{R}^{h \times w \times d}$ and it maps an input image x_i to d feature maps of width w and height h.

To compute the SVD of Step 1, we aggregate the spatial information by a global average pooling operation, hence $\Phi \in \mathbb{R}^{d \times N}$, where d is the number of channels. More precisely, by pooling, we reduce each $\phi(x_i) \in \mathbb{R}^{h \times w \times d}$ to a d-dimensional vector.

Step 2 is also modified. For each class k in a K-classification task, we compute a separate $g_k \in \mathbb{R}^d$. To obtain the ranking, we further evaluate how each g_k compares to the values obtained for the other classes. For instance, we compare g_k to the distribution of the g_k^- obtained for all the input data points that are not of class k, hence for all the K classes except k.

Formally, for each class k we compute the average of the projection of g_k on the singular vectors:

$$g_k = \frac{1}{N_k} \sum_{i, y_i = k} U^T g_{i,k}, \tag{4}$$

where $g_{i,k} \in \mathbb{R}^d$ is the global average pooling of $\phi(x_i) \odot \nabla_\phi \psi_{i,k}$[1]. We then compute the sample mean and standard deviation of the values obtained for all the rest of the data, namely for all classes except k. The mean is obtained, for instance, by averaging over all the samples x_i such that $y_i \neq k$:

$$g_k^- = \frac{1}{N - N_k} \sum_{i, y_i \neq k} U^T g_{i,k'}. \tag{5}$$

Similarly, the variance is computed as:

$$\sigma_k^{-2} = \frac{1}{N - N_k} \sum_{i, y_i \neq k} (U^T g_{i,k'} - g_{k'}^-)^2. \tag{6}$$

Finally, the importance scores for each singular vector in U are given by:

$$z_k = \frac{g_k - g_k^-}{\sigma_k^-} \tag{7}$$

where $z_{k,j}$ is the importance score of the j-th column in U. This measure is then used to obtain a class-specific ranking, from which we retain the first M positions to identify M concept vectors.

3.5 Step 3: Identification of unique concept vectors

This step focuses on isolating directions that act as pointers to *unique* concept representations. Note, by *unique concept* we refer to a distinct and individual idea or notion that is well-distinguishable from others. Therefore, we aim at isolating a specific and well-defined characteristic or pattern that is relevant to the task at hand and semantically distinguishable from other identified concepts.

At this stage, the singular vectors found by SVD provide no guarantee on whether they are pointing to unique nor human-understandable concepts. Neural networks are known to represent more features than they have neurons (Elhage et al., 2022), and this causes unique features to not be disposed alongside orthogonal axes. To account for this, we propose an additional step, namely the disentanglement of singular value directions into unique concept vectors that possess a distinct semantic interpretation. This refinement process employs an unsupervised clustering methodology that was previously investigated by O'Mahony et al. (2023) on individual neuron directions. First, we align the latent representations in accordance with the directions indicated by the singular vectors. Next, we isolate the representations exhibiting the highest coefficients along each direction. Hierarchical clustering is then employed to identify clusters of inputs with minimal intra-cluster distances. The optimal number of clusters is determined based on this analysis and a distance threshold parameter is employed to control the granularity of the concepts. To further enhance the quality of the resulting concept vectors, we employ k-means clustering with the determined number of clusters. Additionally, we

[1]Note, the pooling is here applied after the element-wise product of the features maps and the gradients

remove any outliers and insignificant clusters that contain an insufficient number of samples, for instance fewer than five images. Candidate concept vectors are calculated as the vectors pointing to the centroids of the newly identified clusters, namely by taking the mean activations of the respective clusters.

Human-interfacing is the last action of this step, necessary to interpret the discovered knowledge. The simplest way to obtain insights about a candidate vector is by data analysis and visualization. Depending on the data type and the model architecture, multiple approaches can be used to analyze the data. The simplest approach is to visualize the input data corresponding to the representations laying in the region pointed by the candidate concept vector. In dense models, the candidate vectors can also be used as feature importance estimates. The elements of a candidate vector u are used as weights that are multiplied element-wise to the feature importance values obtained by back-propagating the model's gradients all the way to the input. Section 3.6 discusses more in detail how to obtain concept visualizations and interpretations in detail.

3.6 Concept Maps and Segmentation Masks

The main challenge of concept discovery is the visualization and interpretation of the discovered information itself. As (McGrath et al., 2022) discuss, the discovered information may not always be interpretable nor associable with a human-understandable concept, and research is needed in this direction. A concept vector discovered by our method can be interpreted as a latent discriminant direction for the downstream task. The variance of the activations aligns with the concept vectors, and feature perturbations along these directions have the largest impact on the output function. Visualizing how a concept activates given a certain input can be particularly beneficial to the interpretation of the discovered information in human-understandable terms. We define concept activation maps as the visualization of the model's response, for a given concept, to a certain input. The computation of such maps is straightforward for convolutional networks. For a discovered concept direction $u \in \mathbb{R}^d$, and an input image x_i, we can visualize the layer's activation responding to the discovered concept as the sum of the d feature maps for each channel $\phi_{i,1}, \ldots, \phi_{i,d}$ weighed by the concept vector coefficients. In other words, one can take the feature maps of each channel and weight them by the coefficient values of the concept vector[2]. Note, if we were to consider a concept that is fully aligned with the n-th feature map (with $n < d$, e.g.. $u = (0, ..., 0, 1, 0, ..., 0)$ (non-zero only for u_n), the visualization for x_i would correspond to the n-th feature map itself. Concept segmentation masks can be directly derived by the concept maps as in (Bau et al., 2017), namely by retaining the input pixel values with concept activation higher than the 80-th percentile in the concept maps.

3.7 Dataset Exploration and Outlier Detection

We showcase the utility of concept discovery for dataset exploration, demonstrating how the discovered concept vectors can be used to find input samples that are mislabeled or that contain confounding factors. The data is rotated to align in the space spanned by the singular vectors and anomalous samples are identified based on the statistical dispersion of the data For instance, we isolate 10% of the representations that fall outside of the interquartile range of the representation coefficients for the training data. The inputs corresponding to these representations are flagged for further inspection, as they are outliers that may contain artefacts or potentially misleading confounding factors.

4 Results

This section presents the comprehensive evaluation of our concept discovery approach across various tasks and models. Our experiments focus on widely adopted models with publicly available pre-trained weights, which have been extensively studied in the field of interpretability. Specifically, we present the results obtained using Inception V3 (IV3) (Szegedy et al., 2016) with pretrained weights on the ImageNet ILSVRC2012 dataset (Russakovsky et al., 2015) (Russakovsky et al., 2015). In addition, we demonstrate the versatility of concept discovery beyond image-based applications by showcasing its effectiveness in classification tasks involving tabular data (details provided in the Supplementary Materials).

[2]Being the concept vector a singular vector, it has norm of one.

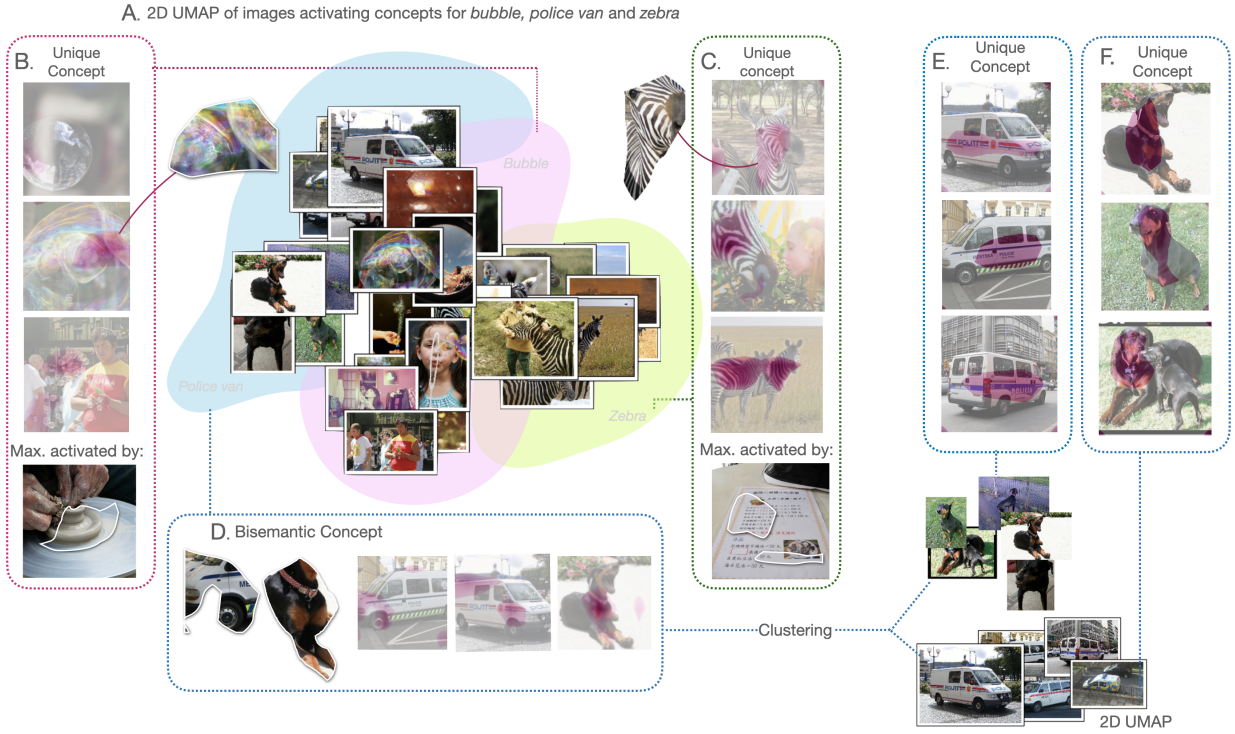

Figure 1: A. 2D UMAP of images activating the discovered concept vectors for *bubble*, *police van* and *zebra*. The images form three seperate clusters. We visualize concept segmentation masks for the images in each of the clusters. B. Concept segmentation masks and maximally activating input image for the class *bubble*. C. Concept segmentation masks for *zebra*. The concepts discovered in B. and C. are *unique* concepts. D. A bisemantic concept is identified for the class *police van*, activating for both vehicle parts and dog paws. After the refinement through clustering, the concept is split in two separate vectors representing either the graphics of the police logo or the dog fur. E. Unique concept representing the police log on the van structure. F. Unique concept deriving from the disentanglement of the polysemantic concept in D.

4.1 Confirmation of existing findings in discovered concepts

Our method automatically identifies concepts associated with image categories. Our results are in line with specific findings that were already discussed by previous research (Kim et al., 2018; Ghorbani et al., 2019), specifically for the object categories of *zebra, police van and bubble*. Additional results for other classes than the selected ones and for ResNet50 are provided in the Supplementary Materials A and an integral analysis of the concepts discovered for ImageNet classes will be made accessible online.

We identify three orthogonal vectors as candidate concept vectors (i.e. one for each of the analyzed classes, with $M = 1$). The discovered concepts encompass a diverse range of visual elements, including patterns observed in the coat of *zebra*, graphical features and tires associated with a *police van*, and glossy reflections as seen in a *bubble*, as demonstrated in Figure 1. The maximally activating images are either visually similar or exaggerate the motifs shown by the activation maps. The maximally activating input for *bubble*, for example, exaggerates the glossy-like reflections and the round shapes generally observed in soap bubbles. The text of the menu restaurant contains black and white stripes, which are also in *zebra* images. Importantly, these illustrations were obtained without any prior annotation of the concepts as detailed in Section 3.6.

Upon closer examination of the *police van* class (Figure1D, our analysis reveals an interesting observation. The candidate concept vector identified for this class exhibits responses to multiple unrelated features. In addition to activating for van tires and police graphics, it also responds to dog chests, which may seem unexpected at first. However, this phenomenon is not entirely unexpected and it can be attributed to the concept of *superposition*, which has been observed in CNNs (Olah et al., 2020). Superposition refers to the ability of neural networks to represent more features than the number of neurons they possess. This results in the presence of *polysemantic neurons*, which exhibit activation patterns that are semantically unrelated. Previous research highlighted the existence of such *polysemantic* neurons that activate simultaneously for car hoods and animal paws (Olah et al., 2020; O'Mahony et al., 2023), aligning with our findings. Our method demonstrates its capability to successfully disentangle the two activations associated with the *police van* class by identifying two distinct clusters that refine the concept direction into separate vectors, as illustrated in Figures 1E and F. In this process, the orthogonality of the direction is compromised to achieve unique concept directions for each activation. The disentanglement of the two patters allows us to capture the distinct semantic meanings associated with each activation, providing a more refined and comprehensive understanding of the underlying concepts.

4.2 Coherence and human interpretability of discovered concepts

To assess the coherence and human interpretability of the discovered concepts, we conducted a human evaluation test following the methodology outlined in previous works (Ghorbani et al., 2019). The test involved 30 participants, and the details of the evaluation test can be accessed at the following link: `https://forms.gle/MJ63G984ERvozuF38`. The evaluation comprised three main experiments, namely (i) intruder detection (ii) concept meaningfulness and (iii) inter-user agreement. Participants were familiarized with the test format by a prior introduction with an exemplar question and the corresponding correct answer.

In the intruder detection experiment (i) illustrated in Figure 2A, participants were presented with ten questions, each featuring a visualization of four concept segmentation masks associated with a specific concept vector, and one random mask from a different concept vector. The task was to identify the outlier image that conceptually differed from the others. Participants successfully recognized the intruders with average accuracy of 0.88. Concept meaningfulness (ii) was measured by asking participants to label the concepts based on the visualization of the concept segmentation masks for three images. Despite the small image segments depicted the concept segmentation masks, all participants accurately labeled all the images, indicating that the concept segmentation masks were easily interpretable and could be associated with specific concepts. Inter-user agreement (iii) was evaluated by requesting participants to agree or disagree with concept labels provided by other participants. Out of the 30 users, inter-user agreement exceeded 91% for all questions. Notably, concepts identified for dog breeds showed relatively lower agreement scores, as users disagreed on the specific type of animal depicted in the concept segmentation masks.

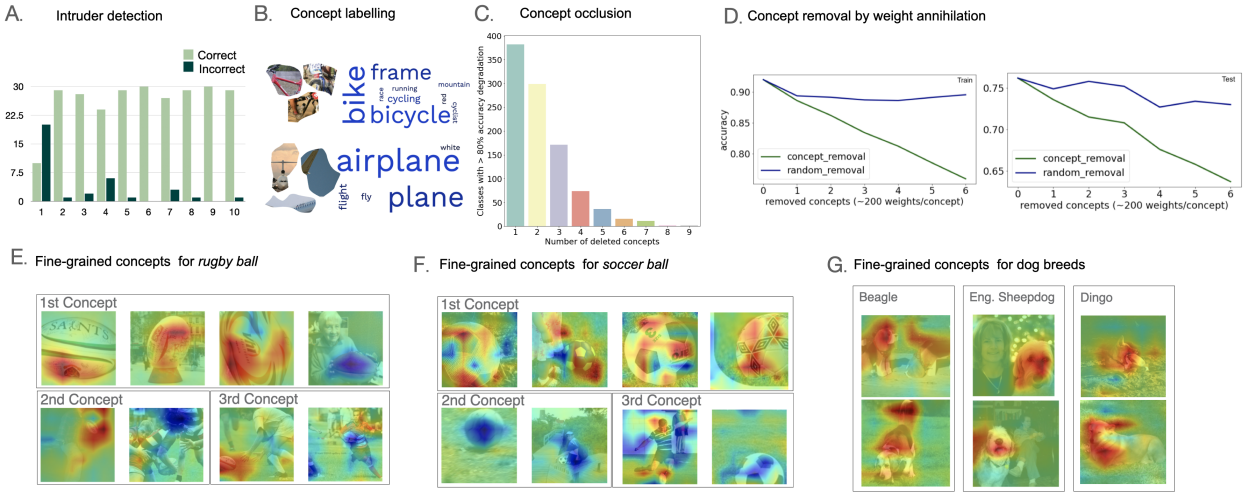

Figure 2: A. Detailed results of the intruder detection experiment with human participants. Out of ten classes, participants were able to detect with very high accuracy the intruding concept. B. Wordclouds of the concept labelling results for two concepts. The font size corresponds to the word frequency. C. Concept importance measured in terms of the impact of removing concepts in the input space by occlusion, namely by masking out areas with high values in the concept activation masks. Performance is evaluated on all ImageNet classes while gradually removing concepts starting from the most important. D. Concept importance measured as the impact of removing concepts in the latent space by annihilating the model weights that are linked to the concept. Performance is evaluated on all ImageNet classes while gradually removing concepts starting from the most important. E. Fine-grained concept activation maps for the first, second and third concept vectors differentiating rugby from soccer balls. The map adopts a symmetric colorbar with red for positive-, blue for negative- and green for zero-valued activations. Images are sorted from the largest absolute projection values on the concept vector. F. Fine-grained concept activation maps with symmetric colorbar for the concept activation maps on ImageWoof categories.

4.3 Quantitative evaluation of concept relevance

We measure the significance of the discovered concept vectors by quantifying the impact of concept removal using two different approaches, namely occlusion in the input space and weight annihilation at the intermediate layer.

Following the methodology of (Ghorbani et al., 2019), occlusion is applied to the input pixels with high values in the concept map of each concept, specifically targeting the pixels with values above the 80th percentile. We quantify the smallest destroying concepts (SDC) as the smallest number of concepts to remove in order to see a performance degradation on at least 80% of the dataset. Starting with the most important concept, we gradually remove concepts while monitoring the drop in performance. As depicted in Figure 2C, we observe a degradation of at least 80% in the prediction accuracy when the first concept is removed for nearly 400 classes. The performance degradation extends to almost all ImageNet classes (962 out of 1000) when we remove the first five concepts. Supplementary Figure B.8 in the Supplementary Materials provides a comparison the model predictions before and after the removal of the SDC for each class.

One limitation of SDC is is the introduction of a distribution shift in the modified images due to the replacement of pixels in the input space. It becomes challenging to disentangle whether the prediction change is solely due to removing the concept or the distribution shift (Hooker et al., 2019). To address this concern, we conduct an additional analysis where concepts are dropped directly inside the layer, comparing the change in prediction accuracy against dropping random directions. To drop a concept, we retain the layer weights with low concept vector components and we set the remaining weights to zero. For example, to remove the first concept of the class *lionfish*, we set to zero 204 parameters out of the original $2,048$ (keeping the remaining $1,844$ weight values untouched), based on the 80-th percentile of the components of the concept vector for that class. We compare the impact of this ablation against dropping random weights to disentangle the effect of removing a specific concept direction from randomly setting weights to zero.

Figure 2D shows the accuracy drop observed when removing the concept directions from the first most important to the fifth, computed on the training images and 1000 validation images. Removing the first five concept directions causes an accuracy drop from 0.920 to 0.785 on train data, and from 0.762 to 0.676 on test data. The same drop cannot be observed instead when setting random weights to zero (i.e. accuracy 0.89 on train and 0.734 on test data).

4.4 Fine-grained analyzes

Concept discovery can be conducted in a fine-grained manner, focusing on specific subsets of classes. This targeted approach enables the identification of concepts that are tailored and relevant to a smaller group of similar classes. Such granular analyses are particularly valuable for classes that pose classification challenges or hold specific importance for interpretability analysis.

We consider a subset of ten dog breeds classes in ImageNet, referred as ImageWoof (Howard). This subset presents a significant challenge as the differences between the dog breeds are subtle (more details in the Supplementary Materials B). Our method identifies one concept for each class, totalling ten concepts. For *Dingo, Beagle and Old English sheepdog* we visualize the images that maximally activate the concept for each class, hence with the largest projection on the concept vector (Figure 2G). High activation values are around the head for all dogs, and confusing factors are ignored (e.g. the woman in *sheepdog*). Each dog has a very distinctive head shape (the *dingo* has straight ears, the beagle long bent ears and the *sheepdog* has long black and white fur) and for this reason, different concept vectors are found for each class. The *beagle* also shows a diffused attention that includes the tail and paws of the dogs. This is probably to distinguish this class from the *English foxhound*.

By running concept discovery on pairs of classes, we identify concepts that provide informative insights into what distinguishes one class from another. For instance, Figures 2E and F showcase the concepts that differentiate a *Soccer ball* from a *Rugby ball*. In this analysis, we set $M = 3$ and present three discovered concepts for each class in order of importance. The most important concept for *Rugby ball* emerges to be the shape of the ball, as shown by the activations around the ball in Figure 2E. On the other hand, the concept activation maps for *Soccer ball* identify the knitted pattern of the ball as the most important concept, and the soccer ball shape emerges only as the second most important concept. We can also see the attention to the context, particularly in the third concept. The maps activate on the players on the field, showing sensitivity to the player's jerseys and socks. Some of these findings align with the hypotheses found in (Ghorbani et al., 2019) about the relevance of jerseys to classify images in the *basketball*

category. Thus, jerseys and socks may introduce confounding factors in the classification of these categories. Finally, in the inputs for the *rugby* class, the maps also activate on the players' hands holding the ball, or on the act of tackling, whereas in the *soccer* images the maps activate on the acts of kicking the ball.

4.5 Concept Uniqueness

We evaluate the uniqueness of the concepts identified by our method, showing that it is possible to disentangle superposition and identify vectors that point to unique concepts. Firstly, it is important to note that the discovered concepts do not align with individual model units. To quantify the alignment, we compute the cosine similarity between the discovered concept vectors for each category in ImageNet (considering $M = 1$ and a total of 100 concept vectors) and the basis vector of the latent space. The cosine similarity is consistently lower than 0.5 for various layers such as *Mixed_7b*, *Mixed_5b*, *Mixed_5d*, *Mixed_6c*, and *Mixed_7c*. A cosine similarity close to zero indicates the independence of the two directions, suggesting that the concept vectors and the latent space vectors are not aligned. In contrast, if the vectors were aligned, the absolute value of the cosine similarity would be close to one.

Next, we examine the outcomes of applying the disentanglement method proposed in O'Mahony et al. (2023). Specifically, we compare how many distinct clusters are suggested by the disentanglement method when applied to individual neurons as opposed to the singular vectors identified by singular value decomposition in our method. This analysis is informative on whether the basis of the singular vectors can facilitate capturing inherently distinct concept representations, i.e. *unique* concepts. Figure 3A, for example, illustrates how three separate clusters are identified as unique concepts starting from one of the singular vectors. The disentanglement outcome provides us with 3 distinct concepts for babies, teddy bears, and necklaces. By applying this approach to all of the singular vector directions within layer *Mixed_7b*, we quantify how many singular vector directions were already pointing to unique clusters and how many required disentanglement. As shown by Figures 3B and C, most of the singular vectors in our method already point to unique concept directions, and only a few require disentanglement. As opposed to individual neuron directions, fewer directions show polysemanticity. This means that the singular value decomposition step in our method is a necessary approach that facilitates the identification of unique concepts. The remaining singular vector directions that are not unique are then disentangled by our clustering method in distinct concepts.

4.6 Effective Outlier Detection

Our method successfully identifies 23% (equivalent 0.2% of the ImageNet training set) of the training images as outliers based on their representation in the latent space. Among these images, 184 (1.8% of the training set) are misclassified by the model. Note, the flagged images were seen during training, hence they are challenging training examples that confuse the model rather than improving its robustness. This observation is supported by the lower top-1 accuracy on the flagged images at 0.90 compared to 0.93 on the rest of the training images.

Furthermore, when compared to random neurons or random directions, our method demonstrates higher accuracy in identifying these challenging training examples. The accuracy on the flagged images obtained through our method is 0.90, whereas random neurons or vectors achieve an accuracy of 0.92. This reaffirms the notion that the concept vectors discovered by our method offer more informative directions for dataset exploration than random alternatives.

Figure 4 shows some examples of the outliers identified by our method. These images present notable variations in style and resolution (e.g., the first two images from the left) or contain confounding factors. For example, in the third image from the left, multiple labels (shovel and lawn mower) are equally correct. Additional examples are shown in the Supplementary Materials.

5 Discussion

The automatic discovery of the latent concepts used by complex models is a challenging research direction that is still under-explored. We presented a novel approach that is simple yet effective to automatically discover concept vectors that point at image clusters with a unique semantic meaning. A key strength in our method is that it does not require manual annotation or supervision. The automated discovery process saves significant time and effort compared to traditional manual labeling methods. Besides, we demonstrate that the discovered concepts are relevant for model performance by evaluating the impact of concept removal. As expected, removing the concepts shows a significant

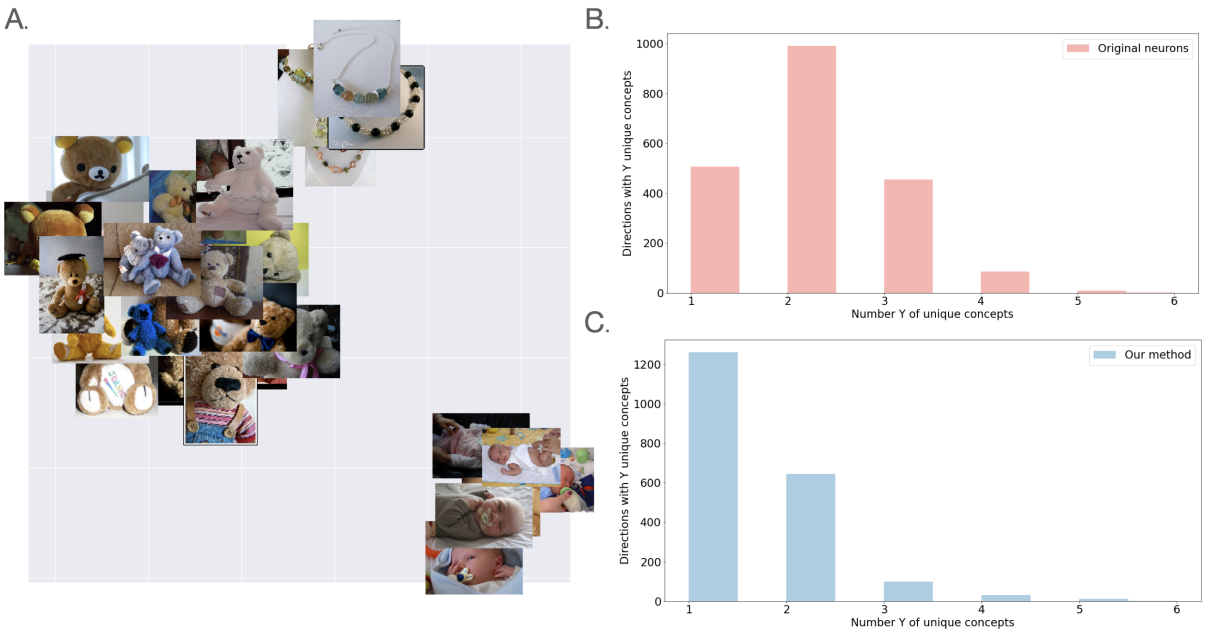

Figure 3: A. UMAP of the images maximally activating a polysemantic singular vector. Hierarchical clustering is used to identify the optimal number of clusters, i.e. three. Three concept vectors pointing to the centroids of each cluster are thus derived to obtain unique pointers to concepts. B. Number of clusters identified by hierarchical clustering for the images maximally activating individual neurons. The high number of neurons with at least two clusters shows that the directions are highly polysemantic, confirming existing research. C. Number of clusters identified by hierarchical clustering for the images that maximally activate our singular vectors. The high number of directions with a single cluster shows that the singular vectors are more likely to identify unisemantic concepts.

Figure 4: A. Results of dataset exploration. Our method identifies training images with distinct issues, such as style shift to drawings (1st image), poor resolution (2nd), multiple correct labels (3rd) and optical illusions (4th).

degradation of the model's prediction accuracy. The uniqueness of the discovered concepts is another noteworthy aspect of our approach. We demonstrate that our concept vectors can more appropriately capture independent and distinct visual attributes than individual neuron directions.

It is important to note that our approach is not without limitations. The reliance on the input data used to identify the concepts is a main limitation. If the network fails to capture relevant concepts during training, our method may not be able to discover them effectively. Additionally, the evaluation of concept meaningfulness relies on human participants, which finally introduces some degree of subjectivity. Notwithstanding, our method proves to be useful in detecting input images with confounding issues, showing that relevant insights can be gathered by looking at the concept directions. The flagged outliers highlight images that exhibit variations in style, resolution, or contain confounding factors. This information is valuable for improving model robustness and understanding the limitations and challenges associated with specific classes or image characteristics. Ultimately, we believe that this work can impact the identification of patterns in fields where deep learning models are accelerating knowledge discovery such as chemistry and biology.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Chirag Agarwal, Satyapriya Krishna, Eshika Saxena, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. OpenXAI: Towards a transparent evaluation of model explanations. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL `https://openreview.net/forum?id=MU2495w47rz`.

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*, pp. 6541–6549, 2017.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:1907.07165*, 2019.

Mara Graziani, Vincent Andrearczyk, Stephane Marchand-Maillet, and Henning Müller. Concept attribution: Explaining cnn decisions to physicians. *Computers in Biology and Medicine*, 123:103865, 2020. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2020.103865.

Mara Graziani, Iam Palatnik de Sousa, Marley MBR Vellasco, Eduardo Costa da Silva, Henning Müller, and Vincent Andrearczyk. Sharpening local interpretable model-agnostic explanations for histopathology: Improved understandability and reliability. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 540–549. Springer, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Jeremy Howard. imagenette. URL `https://github.com/fastai/imagenette/`.

Kareem L Jordan and Tina L Freiburger. The effect of race/ethnicity on sentencing: Examining sentence type, jail length, and prison length. *Journal of Ethnicity in Criminal Justice*, 13(3):179–196, 2015.

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*, pp. 2668–2677. PMLR, 2018.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pp. 5338–5348. PMLR, 2020.

Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Martin Wattenberg, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in alphazero. *Proceedings of the National Academy of Sciences*, 119(47):e2206625119, 2022.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5, 03 2020. doi: 10.23915/distill.00024.001.

Laura O'Mahony, Vincent Andrearczyk, Henning Müller, and Mara Graziani. Disentangling neuron representations with concept vectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3769–3774, 2023.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Robert Rosenthal and Kermit L. Fode. The effect of experimenter bias on the performance of the albino rat. *Behavioral Science*, 8(3):183–189, 1963. doi: https://doi.org/10.1002/bs.3830080302. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/bs.3830080302`.

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, 2019.

Cynthia Rudin, Caroline Wang, and Beau Coker. The age of secrecy and unfairness in recidivism prediction, 11 2018.

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.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

A Additional Results on IV3 and ResNet50

Note: In our study, we apply the methodology to the 1000 classes in ImageNet. To ensure computational feasibility and accommodate our infrastructure capabilities, we employ an undersampling technique. Specifically, we retain only one out of every ten images from the ImageNet training dataset. This strategic choice enables us to manage memory requirements effectively and streamline the computational process. Nevertheless, we demonstrate how concepts can also be obtained by utilizing the complete set of available images for select classes, providing a more granular analysis in Section 4.4. Throughout the experiments, unless explicitly stated otherwise, we primarily focus on the concatenation layer named *Mixed 7b*. This layer is situated deep within the model, close to the end, and is expected to capture complex and high-level concepts. However, we emphasize that similar analyses can be conducted at different depths, layers, and even on entirely different architectures, as showcased in Appendix A.

We provide additional results for 16 classes, namely *airliner, clock, corkscrew, albatross, border collie, road sign, flamingo, mushroom,artichoke, hammerhead shark, screwdriver, iPod, tench fish, suspension bridge, umbrella* and cucumber. The concept segmentation masks for the first most important concept are shown in Figure E.12. The concepts were resized to fit in the square, but they originally appear at multiple scales in the input images.

Results obtained on ResNet50 (at *layer_4.0.add*) are shown in Figures E.13 and A.5. The visualizations point to concepts that are similar to those highlighted in IV3 visualizations, such as the fish fins, the zebra stripes, the glass-like reflections and van tires and logos. Figure A.5 shows the input images in the dataset that have the largest projection value on the concept vector for the four analysed classes. We can see the striped pattern emerging again for the class zebra, although in this case the color of the stripes seems to be given less importance. For the class *bubble*, the shape of the bird belly and their repeated presence may share a similarity with the images in the bubble class.

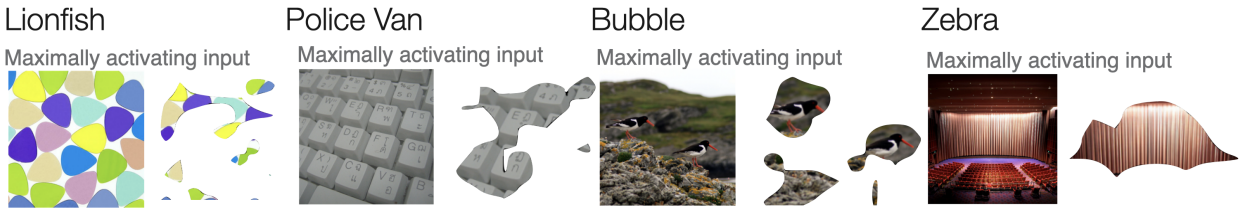

Figure A.5: Results on ResNet50. Visualization of the inputs with the largest projection value on the concept vectors for the classes *lionfish, police van, bubble and zebra*. Next to the input images we visualize the concept segmentation masks.

B Additional Results on ImageWoof

The ImageWoof dataset is a subset of the following ImageNet classes: *Australian terrier, Border terrier, Samoyed, Beagle, Shih-Tzu, English foxhound, Rhodesian ridgeback, Dingo, Golden retriever, Old English sheepdog*. An input image for each class is shown in Figure E.11. This dataset is challenging to classify because the classes represent multiple dog breeds with subtle differences. As Figure B.6 shows, the confusion matrix obtained on *training* images also presents misclassified samples.

Figure B.7 shows additional concept maps for the dog breeds ($M = 1$).

We show the prediction change before and after the removal of the smallest number of destroying concepts on ImageWoof in Figure B.8. No prediction change would appear as a diagonal line, whereas in our case we can clearly see that the prediction becomes random after removing the concept. The SDC is 1 for *Shih-Tzu, Rhodesian ridgeback, Beagle, Australian terrier and Golden retriever*, 2 for the *Old English sheepdog*, 3 for *Samoyed* and *Dingo* and 4 for the *Border Terrier*.

C Additional Results from the User Evaluation

Figure C.9 reports in detail the performance of the end users on each question in the evaluation study part (i).

D Benchmark results on COMPAS

Here we demonstrate the applicability of concept discovery to non-imaging applications such as COMPAS. The model is, in this case, a densely connected feed-forward network. The COMPAS system uses criminal records and demographic features of nearly 19 thousand defendants to predict the likelihood of a new offense. Previous studies largely discussed whether race, gender and age of the defendants are sensible variables affecting the system's fairness Jordan & Freiburger (2015); Rudin et al. (2018). While we agree that a multilayer perceptron is unnecessarily complex for this task Rudin (2019), we recognize its importance as a contradictory task in the model interpretability literature.

Figure D.10 shows three discovered concepts for this task ($M = 3$). Note, gradient backpropagation is used to visualize the importance of each input feature for the concept vector. The values in the plot represent the importance given to the input features to project any input on the concept vector. The three concepts are a linear combination of multiple input features, and high relevance is given to the age of the defendant (age, in the first plot), the number of prior arrests (no_priors, in all plots), and the length of previous detentions (length_stay, two plots on the right).

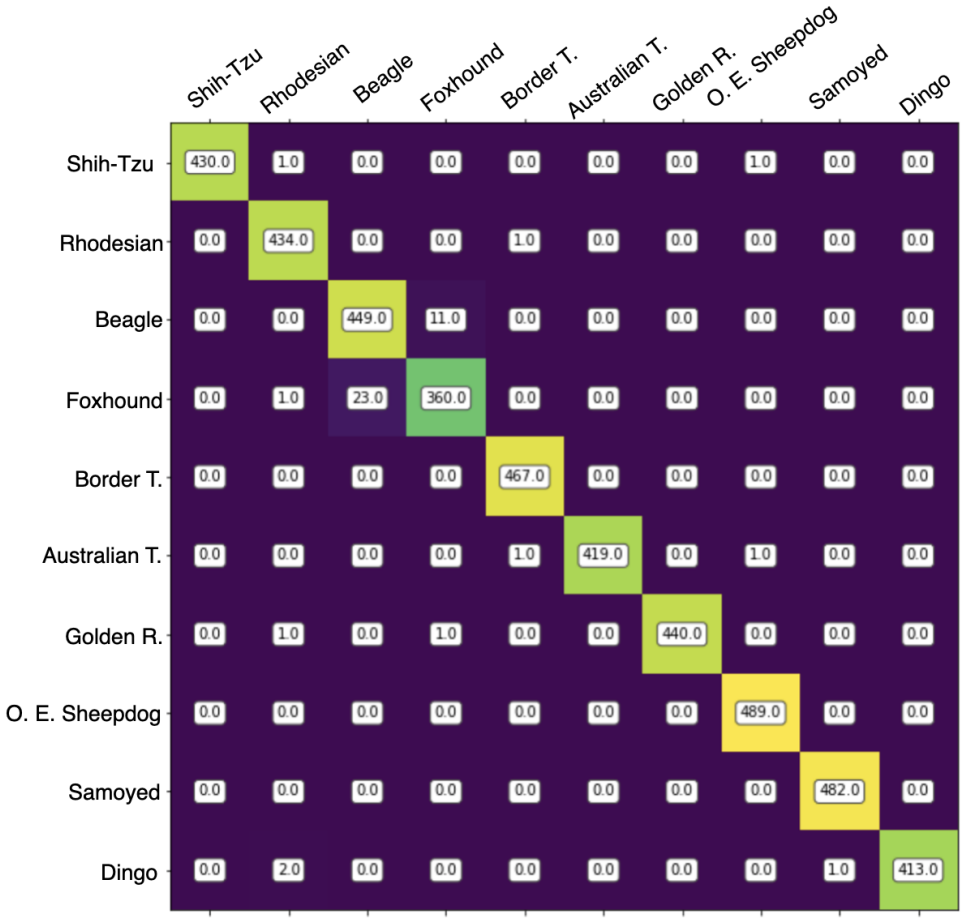

Figure B.6: Detail of the IV3 confusion matrix on ImageWoof classes.

The concept scores in the plot give a global explanation of the model. Local explanations about single inputs can be obtained by multiplying the scores with the feature values of each input. We benchmark the faithfulness of our explanations against other explainability methods by the Prediction Gap on Important feature perturbation (PGI) and Unimportant feature perturbation (PGU), as in Agarwal et al. (2022). High PGI and low PGU values are desirable. The former indicates that the explanation identifies important features, while the latter that the method ignores unimportant ones. We compute the PGI and PGU following the benchmark code [3], obtaining 0.27 PGI and 0.0026 PGU. These values are comparable with other standard explainability methods, outperforming the benchmark results obtained by gradient-based approaches and LIME, as shown in the Appendix Table 1.

Table 1 benchmarks our method in the OpenXAI benchmark Agarwal et al. (2022). The results in the table are taken, except for concept discovery, from the online leaderboard of the benchmark at `open-xai.github.io`. The hyperparameters of the explainer were set as illustrated in the benchmark instructions, i.e. {`protected_class:1`; `positive_outcome:1`; `perturbation_std:0.3`}.

E Additional Visualizations

[3] `github.com/AI4LIFE-GROUP/OpenXAI/blob/main/OpenXAI%20quickstart.ipynb`

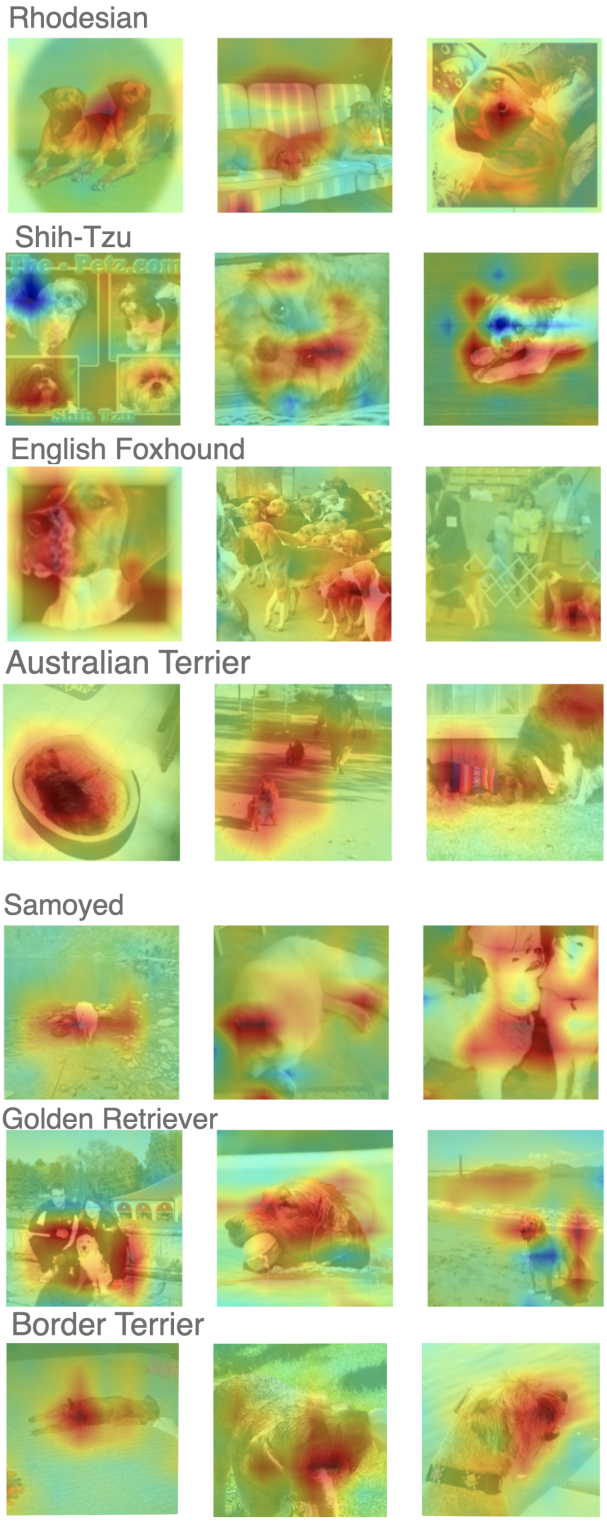

Figure B.7: Concept maps for the classes in ImageWoof. The images are ranked in order of their projection value on the concept vector (largest to lowest).

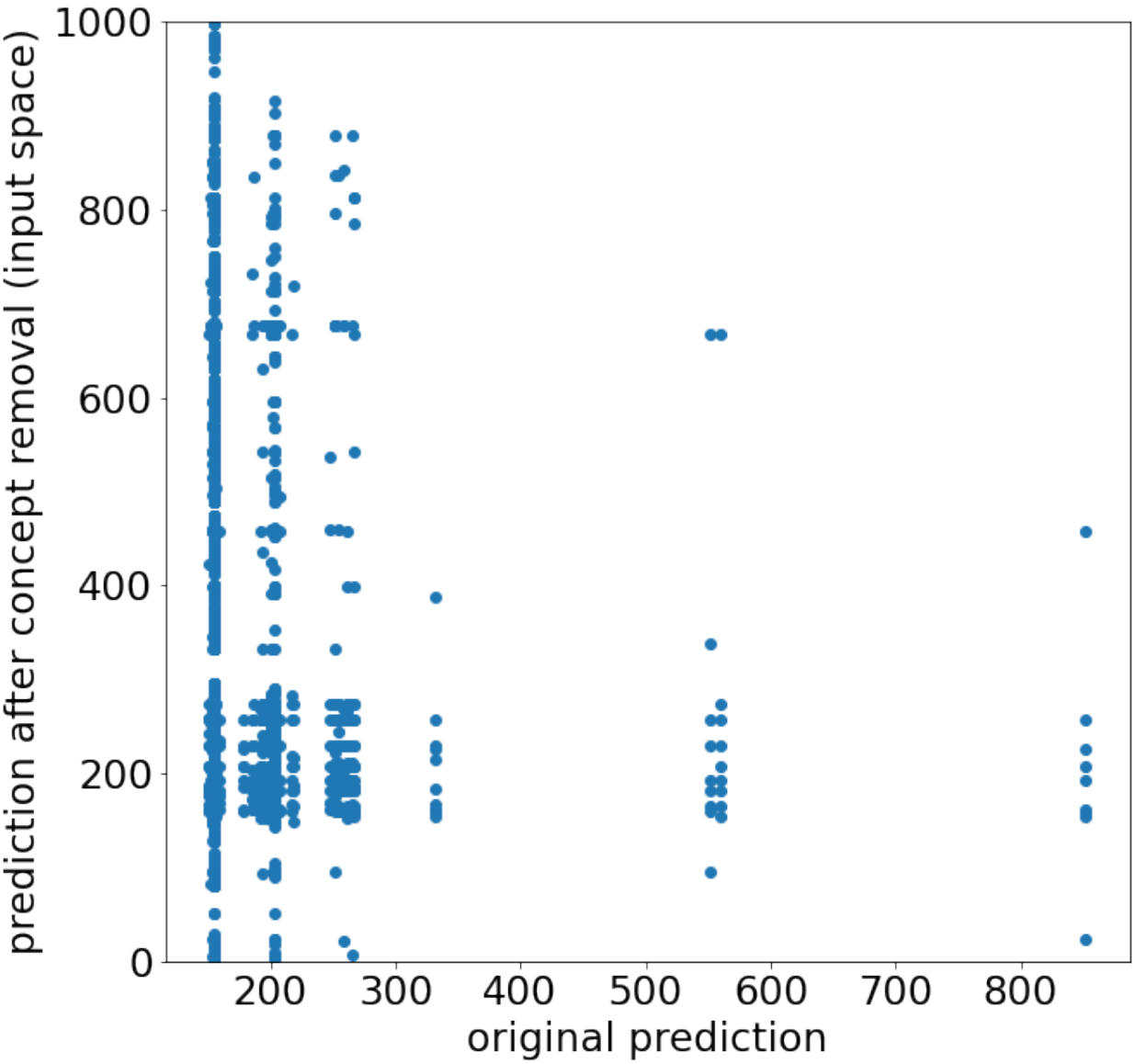

Figure B.8: IV3 predictions on ImageWoof before (original) and after concept removal in the input space.

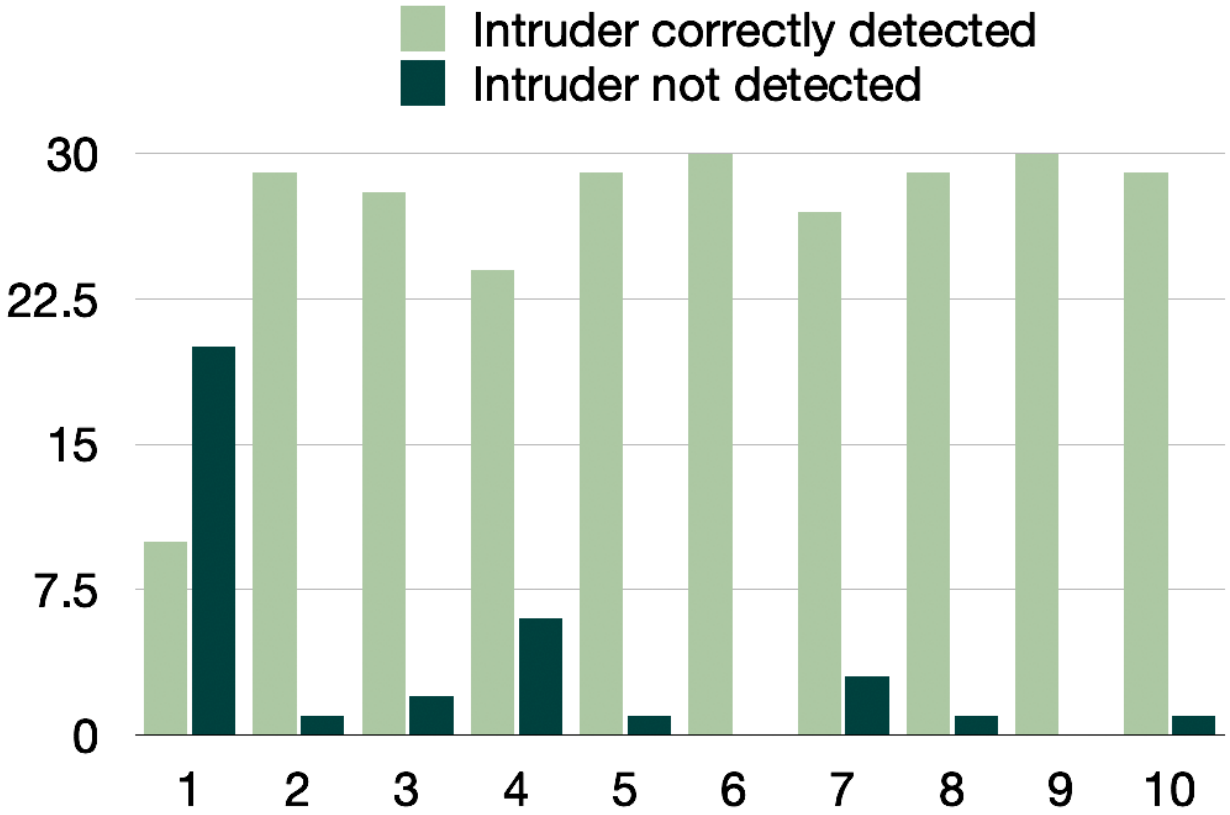

Figure C.9: Detailed performance on part i. of the human evaluation study.

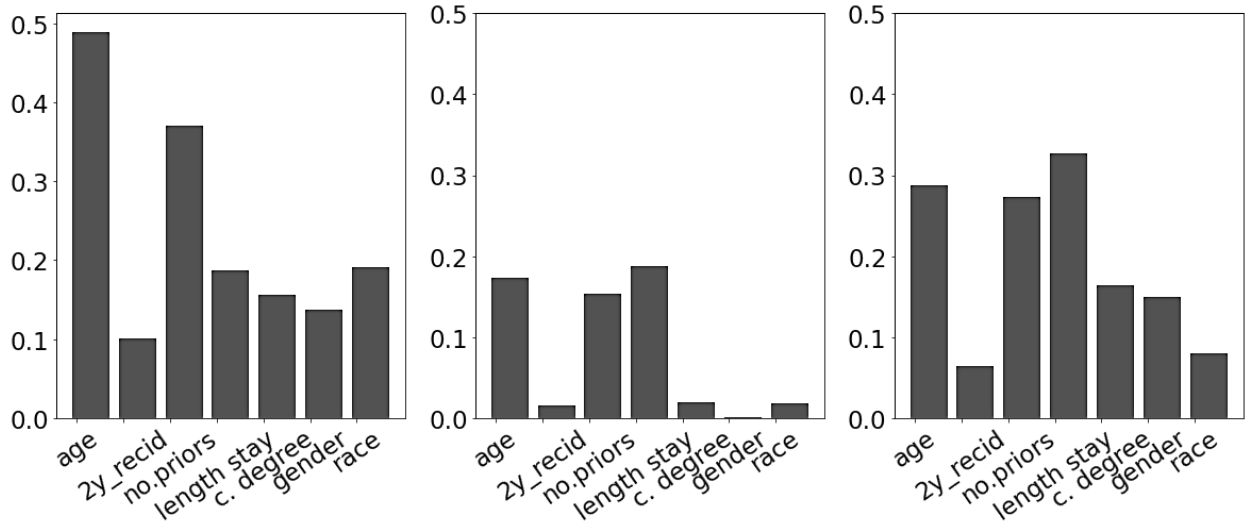

Figure D.10: Top 3 concepts discovered on COMPAS. The bar values illustrate the feature importance values to project each input on the discovered concept vector.

Table 1: Explanation faithfulness for COMPAS.

Method	PGI ↑	PGU ↓
LIME	0.232	0.247
Vanilla Gradient	0.240	0.240
Integrated Gradient	0.240	0.238
Gradient x Input	0.254	0.216
SHAP	0.274	0.194
SmoothGrad	**0.324**	0.106
concept discovery	0.268	**0.0026**

Figure E.11: Dog breeds in ImageWoof.

Figure E.12: IV3 concepts for additional classes. We show the resized segmentation masks.

Figure E.13: Results on RestNet50. Segmentation masks of the concept vectors in *layer4.add* of ResNet50 for the same classes and input images in Figure 1.