Cause and Effect: Hierarchical Concept-based Explanation of Neural Networks

Mohammad Nokhbeh Zaeem[1] and Majid Komeili[2]

***Abstract*—Artificial Intelligence (AI) has revolutionized many fields. There has been growing concerns around use of so-called black-box models such as neural networks. In this work, we take a step in the interpretability of neural networks by examining their internal representation or neuron's activations against concepts. A concept is characterized by a set of samples that have specific features in common. We propose a framework to check the existence of a causal relationship between a concept (or its negation) and task classes in the activations of intermediate layers. While the previous methods focus on the importance of a concept to a task class, we go further and introduce four measures to quantitatively determine the order of causality. For example, we can quantify to what extent a concept is a necessary condition for a task class. A high necessary score implies that whenever the information about the task class is present in that layer, the information for the concept will also be present. Moreover, we propose a method for constructing a hierarchy of concepts in the form of a concept-based decision tree that can shed light on the activations and how various concepts might be interacting towards predicting output classes. Through experiments, we demonstrate the effectiveness of the proposed method in explaining the causal relationship in the activations of a neural networks.**

I. Introduction

During the past decade, Artificial Intelligence (AI) has revolutionized many fields and we observed a surge in AI applications and AI-based products. AI is an extremely useful tool, but there is a lot we don't know about how it works and how it will behave in new situations. We have witnessed a lot of concerns about adopting such black-box machine learning (ML) models. There is an imperative demand for tools and methods to systematically and scientifically address such concerns. Neural networks, as one of the most promising forms of AI with high performance on classification problems like ImageNet challenge [1], [2] have been criticized for their black-box decision-making process. One of the most important questions asked about neural networks decisions is how a certain *concept* influences the internal representation and eventually the output of a neural network. Here, a concept is a representation of a particular feature and is defined by a set of samples with that feature, against a random set [3]. Fig. 1 shows an example of how concept red can be represented by a set of samples. More broadly, for example, for the task of predicting the job title of a person from their image, a concept can be the colour of uniform (e.g. white or pink), background (e.g. office, ambulance or clinic), or presence of objects (e.g. stethoscope) in the image.

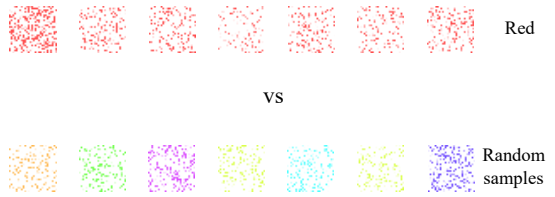

Fig. 1. Concept red is represented by a set of red samples versus a random set.

Concept-based explanation of neural networks is a post-hoc approach to examine the internal representations or *activations* of the neural network against high-level humanly meaningful concepts. The training phase of the original network and the post-hoc explanation phase can be completely separate with separate datasets (one for task classes and one for concept classes) and be done by different people independently. For example, the task of predicting whether the job title of a person is physician can be represented by a dataset of physician images. But, the concept clinic can be represented by a dataset of clinic images and the concept stethoscope can be represented by a dataset of stethoscope images. In the following sections, we will highlight three problem areas and discuss how the proposed methods tries to address them.

A. Linear vs Nonlinear Concepts

The first problem area is where most of the existing concept-based methods assume that a concept, if present in an activation space, is linearly separable from non-concept samples [4], [3], [5], [6]. This assumption, however, does not necessarily hold, especially in the earlier layers of a network where the learned features are often not abstract enough to linearly separate concepts [7] or in later layers when concepts are mixed up to form higher-level concepts. This hinders such methods' ability to track the presence of a concept throughout the layers of the network. Another limitation comes from the assumption that the gradient of a section of a network with respect to the input is a good representation of that section [3], [6], [8]. Such a first-order approximation might be misleading. This issue has been extensively discussed for saliency maps — which are also based on gradient approximations— and have been proven to be misleading [9], [10].

[1] mohammadnokhbehzaeem@carleton.ca
[2]Majid Komeili is with School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Canada. majid.komeili@carleton.ca

B. Correlation vs Causation

Another shortcoming of the previous methods is that most methods yield a score that captures the correlation between concepts and output and cannot give any further details about causation [4], [3], [6]. Following the above example about the classification of job titles from images, the correlation between wearing a lab coat and being classified as a doctor cannot answer questions like "do all images classified as a doctor include lab coats?" or "do all people wearing lab coats are classified as doctors?" This problem is sometimes referred to as causality confusion. In a medical diagnosis setting one may ask questions like, do all patients that are classified as having flu have fever symptoms (fever necessary for predicting flu)? Are all patients with fever classified as having flu (fever sufficient for predicting flu)? Are all patients classified as flu do not have a fever (absence of fever necessary for predicting flu)? Are all patients with fever not classified as flu (fever sufficient for negative of predicting flu)? What are the symptoms that if presented together, the predicted diagnosis will be flu? Note that the goal is not to investigate causal relationships in the training dataset. The goal is to investigate the activations for causal relations "learned" by a neural network. For example, whenever the flu information is present in a layer (i.e. predicted class is flu), the information for fever is also present.

C. Individual Concepts vs Combination of Multiple Concepts

Most approaches are unable to analyze combinations of concepts, especially in cases where the behavior is nonlinear. The linear assumption means that each concept attributes to a small part of the decision-making and their influences add up. For example, consider a situation where symptoms (A and B), or (C and D) are enough for a diagnosis, but not mixed symptoms like (A and C), (B and C), (A and D) and (B and D). Here, a linear model fails because it cannot detect (A and B) and (C and D) pairs without activating, for example, (A and C). Another example is when a task class is equivalent to the exclusive OR of two concepts; such concept relationships cannot be represented by linear models.

To address the above issues, the proposed method uses a non-linear model with a specific structure to detect a concept in a layer's activations. It checks the presence of a concept in a layer's activations by training a concept classifier —a network with the same structure as the task classifier from that layer onward but trained to detect the concept. See Fig. 2 (middle). The output probability of the task classifier is indicative of the presence of the task information in the layer's activations. Likewise, the output probability of the concept classifier is indicative of the presence of the concept information in the layer's activations. The proposed method aggregates the output of the task and concept classifiers on a distribution sample set; a set of samples representing the distribution of the input data. This set is representative of the likely inputs of the network. We propose four metrics to quantify the causal relationship between concept and task in the layer's activations. For example, a high necessary score in a layer indicates that in that layer whenever the information for predicting task is present, information for predicting concept is also present. To address third issue, we propose an approach for constructing a decision tree that can capture nonlinear relations between concepts and provide a hierarchy of concepts that exist in the activations of a layer. Unlike the previous works in [4], [6], which are limited to specific network structures like convolutional layers, the proposed method can be applied to a wide range of network structures.

The main contributions of this paper is as follows:

- We propose a general framework for quantifying causal relationships between a concept (or its negation) and task classes in the activations of a neural networks. While the previous approaches focused on the importance of a concept to a task class, we go further and introduce four measures to quantitatively determine whether the concept (or its negation) is necessary, sufficient, or irrelevant for a task class.
- While the mainstream previous work is based on linear models for representing concepts, we propose to use a non-linear model in the form of a neural network that its structure and initial weights are transferred from an appropriate segment of the original network.
- We propose an approach for constructing a hierarchy of concepts in the form of a decision tree. The decision tree can shed light on the activations and how various concepts might be interacting towards predicting output classes.

A preliminary version of this work has appeared in [11]. Improvements compared to the preliminary version are two fold: first, we have expanded the proposed method to work on multiple concepts and propose a method for constructing a hierarchy of concepts in the form of a decision tree. Second, we have included new experiments as well as additional discussions.

II. Related Work

There have been several works on explaining intermediate activations of a neural network based on human-friendly concepts. Most notably, Kim et al. [3] proposed a percentage measure, called TCAV score, to measure how much a concept interacts with the task classifier. TCAV works based on whether the gradient of the neural network is in the direction of the concept. The direction of the concept is defined as the direction orthogonal to the linear classification decision boundary between concept and non-concept samples. The TCAV score captures the correlation between the network output and the concept and lacks detailed information about the nature of the relationship. Moreover, it assumes that concepts can be represented linearly in the activation space, an assumption that does not necessarily hold [7]. They also represent a section of the network only by its gradient (first-order approximation), which might be misleading. Similar approaches have also been explored in the Net2Vec [12] and Network dissection [4] methods, but they assume that the concepts are aligned with single neurons' activation. The derived works such as RCAV (Robust Concept Activation Vectors) [8] try to address the linearity problem, but the gradient approximation problem still persists.

In another work, Interpretable Basis Decomposition for visual explanation (IBD) [6], authors tried to explain the activations of a neural network by greedily decomposing it into some concept directions. They used the resulting decomposition as an explanation for the image classification task. One of the drawbacks of such an approach is its linear assumption, which comes from the usage of linear decomposition of the gradient in the activation space. Using greedy methods can also potentially result in inaccurate and unstable results. Another limitation of the IBD method [6] is that it can only explain convolutional layers, and therefore for neural networks that include dense layers, they have to modify the network. In their experiments, they have replaced each dense layer with a global average pooling layer and a linear layer.

Concurrent to our work, Singla et al. [13] proposed that instead of linear decomposition, a decision tree can be used to explain the behavior of the neural network. However, they still use logistic regression for concept representation, which, by nature, is linear and has all the problems mentioned earlier.

The linear assumption indicates that a concept in hidden layers corresponds to a vector and the representation of data in each layer is a vector space. Such methods assume that addition, subtraction, scalar product and inner product (as projecting an activation to a concept vector) operations in an activations space are always meaningful. The linear assumption is originated in feature visualization methods. Most feature visualization methods optimize for inputs that maximally activate certain neurons or directions. Early studies on neural network activation space tried to find samples that maximally activate a single neuron, and associate a concept to the neuron. In [14], the authors argued that random linear combinations of neurons may also correspond to interpretable meaningful concepts. The general idea of using a linear classifier to check the information of the intermediate layers originated in [15]. They proposed to use linear probes, which are trainable linear classifiers independent of the network, to gain insight into the network representations. In contrast to what was mentioned in [14], in [4], [16] the authors reported that the basis (each neuron) direction activation is more often corresponding to a meaningful concept than just random vectors. Still feature visualization methods, ignore the distribution of the input data which results in inputs that are not consistent with real samples.

Linear interaction of concepts has been even less studied in feature visualization methods. In [16] the authors showed in some cases the addition of two concepts' activations will result in inputs with both concepts present. But they cast doubt on whether this finding is always true. The linear assumption lacks enough evidence to be considered reliable for being the basis of interpretability methods that try to gain the trust of humans and justify neural network decision-making.

Some other methods have tried to automatically discover new concepts from neural networks, namely Automatic Concept-based Explanations (ACE) [17] and Completeness-aware Concept-Based Explanations (CCE) [18] and others [19], [20], [21], rather than taking a concept as input. ACE tries to automatically extract concepts based on TCAV while CCE extracts concepts based on convolutional layers continuity. CCE also tries to avoid first-order approximation by measuring the importance of concepts using the Shapley score. A drawback of such approaches is that because they are unsupervised, there is no guarantee that the resulting concepts will be human-friendly. It might be challenging for a human expert to draw conclusions based on some machine-generated concepts that are hard or even impossible to understand. Though such methods can help with cases that no principle exists for rational behaviour of the network, in many cases, the experts have a good principle about the problem at hand and the principle's concepts are predefined. So they want to check the consistency of the neural network with the existing principles.

Our work relates to CACE [5] in that, both try to address the shortcoming of TCAV [3] by capturing causal expressions. The CACE method [5] measures the influence of concept by the difference of conditional expected values. This requires highly controllable datasets or very accurate generative models that may not be available in practice. Our method relates to works that define and train neural networks with concept-based explanations in mind [7], [22], [23], [24], though our method explains existing pre-trained neural networks (post-hoc approach).

Our work relates to [25] in that both use a specific visual method to examine the influence of different input features on the output of a machine learning model. But our method goes further and inspects the nature of the relationship and quantifies these visualizations. We also consider high-level human-friendly concepts instead of raw input features.

In the next section, we will present the proposed framework, which simultaneously addresses the linear assumption, the first-order approximation, and the causality confusion issues discussed above.

III. Framework

In this section, we first provide some background information. Then we describe how to determine relevant concepts, i.e. concepts that are present in a layer. Finally we describe how to quantify the causal relationships between the relevant concepts and task in that layer.

A. Background

Logical expressions are usually expressed as a causality clause in the mathematical notation form of $A \Rightarrow B$. In this notation, phenomenon A is the reason for phenomenon B and whenever A happens, B will follow. In fundamental math, concepts are represented by sets. If for a sample in the distribution set, the information for a task class is present in an activation layer, i.e. if the output of the task classifier is larger than a threshold t, then the sample is in the task set. Likewise, if the information for a concept is present in an activation layer, i.e. if the output of the concept classifier is larger than a threshold t, then the sample is in the concept set. Two arbitrary sets are usually demonstrated by a Venn diagram (see the top row of Fig. 3). There can be four possible relations between the two sets. For example Necessary condition can be represented by $T \subseteq C$ meaning that concept is necessary

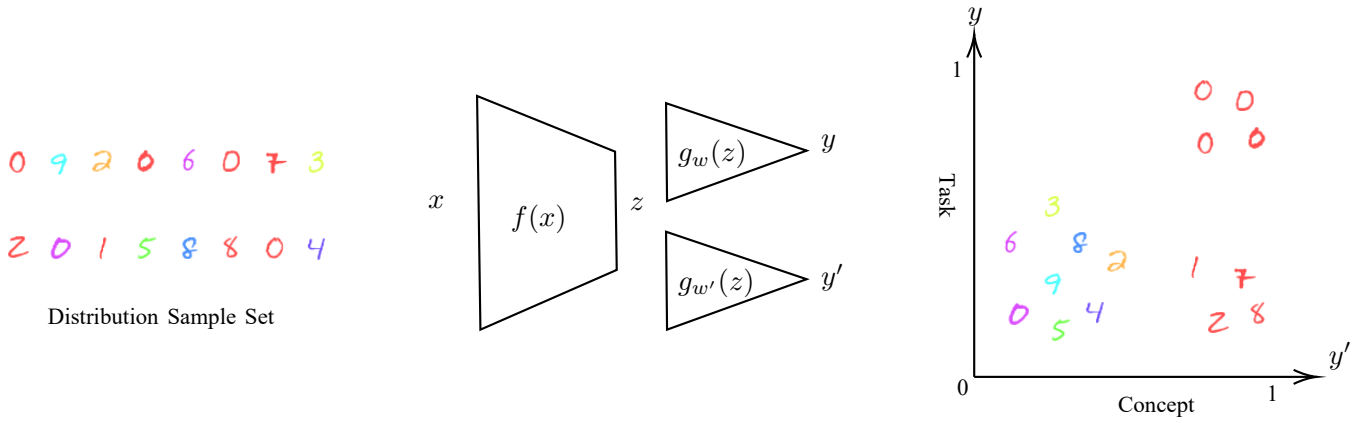

Fig. 2. Left: Distribution sample set. Center: Illustration of the original network as $f(g_w(z))$, the task class network $g_w(z)$ and the concept network $g_{w'}(z)$. Right: Analysis of the relationship between the task class and the concept showing that the concept is a necessary condition for the task class. Here the task class was defined as the classification of digits where each class has a unique colour.

for task. Each of these relations can also be represented as a causality clause. For example necessary would be $T \Rightarrow C$. See the top row of Fig. 3 for other relationships.

B. Determining Relevant Concepts

In our method, we base the explanations on the activations of a certain layer and explain whether and how a concept interacts with a task class based on the activations of the layer. We break the neural network into two sections, the section before the hidden layer (denoted by $f(x)$) and the section after the hidden layer (denoted by $g_w(z)$). w denoted the trainable parameters of the second section, and the whole network can be expressed as $g_w(f(x))$.

We only need two sets of samples for our analysis. (1) Concept set labeled with concept information only (Fig. 1). (2) Distribution sample set, without any labeling (Fig. 2 (left)). Note that access to the training data of the original tasks is not required.

For the sake of explaining the proposed method, let us consider a neural network trained on color-coded handwritten digits. In the training set, a unique color was assigned to each class and the samples within each class were colored accordingly. For instance, all 0's in the training set are red, and all 1's are blue. One would expect the network decision to be influenced by the color as well as the digit itself; but the challenge is how to measure this influence. We aim to determine how a concept, e.g. red, influences the decision-making of this neural network. The first step in our analysis is to check whether a concept is present in a layer. In other words, can a classifier trained on the activations of that layer achieve an acceptable concept detection accuracy? We check if the second section of the network has adequate power and capacity to distinguish the concept set (colour red) from random none-concept samples in that layer. Since detecting a concept is a binary classification, the number of output neurons is adjusted accordingly.

For representing a concept, a set of positive and negative concept examples are used, in our case red samples against other colours (Fig. 1). Then the concept classifier ($g_{w'}(z)$) with the structure of the second section of the neural network ($g_w(z)$) is trained to distinguish the concept from non-concept samples. As a result, we will have a network with identical structures as the task classification but different parameters. w denotes the parameters of the original network trained for task classification (digit classification) and w' denotes the parameters learned to distinguish a concept from non-concept samples (red vs. other colours). $g_w(z)$ is the task classifier, whereas $g_{w'}(z)$ is the concept classifier. For learning $g_{w'}$, we initialize it with the weights of the task classifier i.e. w. Note that the parameters of the first section ($f(x)$) do not change while w' is learned.

Why same structure? Although concepts like color can be learned by a much simpler network structure, more complex concepts like the presence of objects might not be as simple to detect. If a linear model gives high concept detection accuracy in a layer, then there is no need to use a more complex structure. But if it does not, a more complex model is needed. In general a concept classifier would give poor concept detection accuracy for two main reasons: either the concept universally does not exist in that layer, or it is there but it is too complex for the concept classifier to detect it and therefore a more complex structure is needed. But how far should one go and what could be an upper limit to the complexity of the concept classifier? We argue that the structure of the task classifier is an upper limit to what is needed for the concept classifier. Proving that a concept is universally forgotten (i.e. does not exist in an activation layer) is not tractable. However, we don't need to prove the absence in a strict sense. We just need to show that the concept is forgotten up to the capacity and power of the given network. If in a particular layer, a concept cannot be detected by a concept classifier that has the same structure as the task classifier, that concept "effectively" does not exist there. Such a conclusion can be made only if the concept classifier has the same structure as the task classifier or has more capacity than that. Whenever a simple concept classifier (e.g. a linear model) have poor concept detection

accuracy in a layer, instead of searching for alternative options, training the same structure is preferred because we can initialize the weights of the concept classifier using the weights of the task classifier. Both task and concept classifiers are to extract higher level representations from the same layer and from activations of the same set (i.e. distribution sample set). Therefore, such a pre-training facilitates training the concept classifier. If the concept detection accuracy is still poor, we can conclude that the concept effectively does not exist in that layer. Note that only if a concept exists in a layer, we would proceed with quantifying the relationships using the metrics that will be described in the next section. Another advantage of using the same structure is that the structural coherence of the network would be kept intact. In other words, the limitations, powers and local behavior of the network (as initial parameters) are considered in the detection of concepts, by keeping, for example, convolutional activations as convolution representations (with the spatial information preserved).

C. Quantifying Causal Relationships between Concepts and Task Classes

Checking if a set is a subset of another, can be easily done by checking the definition. Since we cannot sample every possible instance in the input space, we only check the relationship on a distribution sample set. The distribution sample set, the samples used for causal analysis, does not have any ground-truth labels. Labels are not needed because we pass the samples of the distribution set through the task and concept classifiers and only the resulting output probability pairs will be needed to quantify the causal relationships (more on this will follow). The set T is a subset of set C if every sample in T is also in C, which is equivalent to C being a necessary condition for T or $T \Rightarrow C$. Checking the negation of this definition is much easier (just checking that no counter-example exists). For this purpose, a scatter plot is generated by evaluating the task classifier and the concept classifier on the distribution sample set, see Fig. 2 (right). Each point in the scatter plot is a sample from the distribution sample set. A counterexample, in this case, is a sample in C and not in T (e.g. a sample classified as 0 and not red). So, the clause correctness corresponds to the case where the top left corner of the scatter plot is empty – equivalently, no counterexamples found in our distribution sample set (Fig. 2). The points in the scatter plots are outputs of the task classifier ($g_w(z)$) and the concept classifier ($g_{w'}(z)$) based on the activations in an intermediate layer. We want to measure the relationship between the predictions of the concept and the task class networks regardless of the true labels. Likewise, other corners of the scatter plot being empty corresponds to other types of relationships between concept and task class (see Fig. 3).

In the following we explain the process for quantifying the necessary condition i.e. the logical expression $T \Rightarrow C$ which is equivalent to $\neg C \Rightarrow \neg T$. As shown in the first column of Fig. 3, ideally when $T \subseteq C$ these two logical expressions are equivalent. However, in a non-ideal situation where there are samples in T that are not in C, the two logical expressions are not equivalent and we need to quantify them separately. $T \Rightarrow C$ is equivalent to the logical expression $\neg T \vee C$ which implies that for the logical expression $T \Rightarrow C$ to be true, $\neg T$ or C must be true. To check $T \Rightarrow C$, we check if there are counter-examples. A counter-example for $T \Rightarrow C$ happens when $\neg(T \Rightarrow C) = T \wedge \neg C$ is true. $\neg C$ can be calculated as $1 - C$ because C is a probability. $\neg C$ is greater than a threshold t is equivalent to C being less than $1 - t$. To reflect this, we compare the task class probability against the threshold t and the concept probability against the threshold $1-t$. Consequently, we can work directly in the scatter plot of T vs C, rather than T vs $\neg C$. Therefore, the logical expression $T \Rightarrow C$ is true for samples outside the region F. The third row of Fig. 3 shows an example for $t = 0.25$. A sample would be in region F, if it is in set T and set $\neg C$. That is the output probability $T > 0.25$ and the output probability $\neg C > 0.25$. The latter is equivalent to $C \leq 0.75$. Putting these two together ($T > 0.25$ and $C \leq 0.75$) defines the region F as shown in the first column of Fig. 3. In order to have a strong necessary score, we want the number of samples that satisfy the logical expression ($T > 0.25$) and ($C \leq 0.75$) be small. That is, the number of samples in the region F (the red region) be small. Now let's focus on the green region. The number of samples in the red region F is small when the number of samples in the TP and TN regions are large. We define the P and R scores to exactly reflect this. For the TP region, we define P as follows:

$$P = \frac{TP}{TP + F}\,, \tag{1}$$

where TP, TN, and F denote the number of samples in the corresponding regions (see Fig. 3). Intuitively, $TP/(TP+F)$ is the percentage of the samples above the line $T = t$ that satisfy $C > 1-t$. This can be expressed as $P(C > 1-t|T > t)$. By looking at this probability, it can be seen that it measures $T \wedge C$. Likewise for the TN region, we define R as follows:

$$R = \frac{TN}{TN + F}\,. \tag{2}$$

$TN/(TN+F)$ is the percentage of samples to the left of the line $C = 1 - t$ that satisfy $T < t$. This can be expressed as $P(T < t|C < 1 - t)$. By looking at this probability, it can be seen that it actually measures $\neg C \wedge \neg T$.

In theory, $T \wedge C$ implies $\neg C \wedge \neg T$ and vice versa. They are equivalent logical expressions. If one is true, the other is also true. However, empirically we rarely have a perfect necessary condition. Consequently, $T \wedge C$ and $\neg C \wedge \neg T$ may not be equivalent. More precisely, P and R can take different values. We combine them by taking the harmonic mean in the same way as F1 score combines Recall and Precision. To show that concept is necessary for task, both of the probabilities corresponding to $T \wedge C$ and $\neg C \wedge \neg T$ must be high i.e. P and R scores must be high. Instead of working with P and R, we use their harmonic mean (i.e. F1).

$$F1 = \frac{2PR}{P + R}\,. \tag{3}$$

Note that P and R are just two quantities we defined above and they are not related to the Precision and Recall measures typically used in binary classification problems.

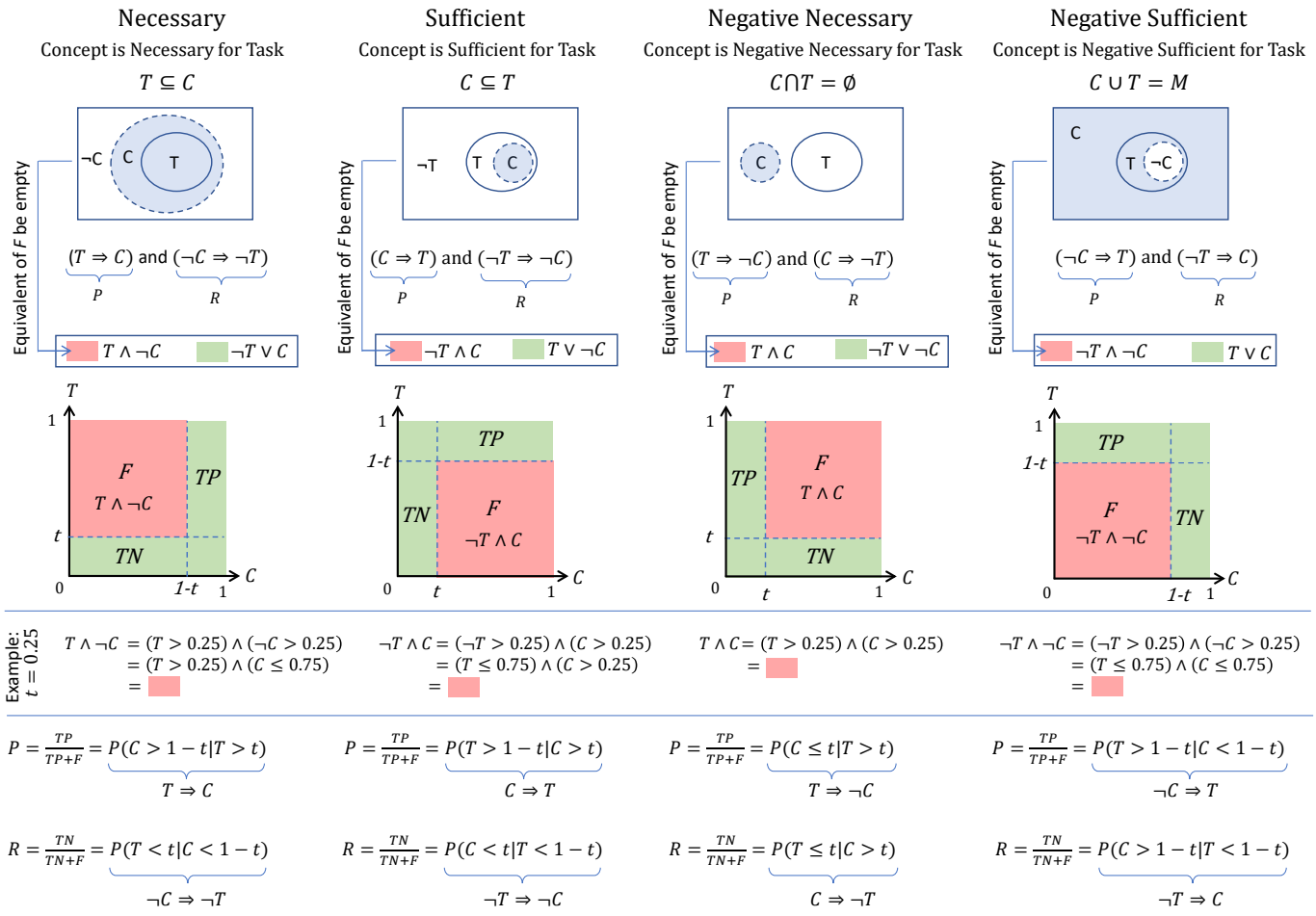

Fig. 3. Summary of the four causal relationships. Each column is for one of the relationships. Top: Venn diagram representation of the relationships and the corresponding scatter plots. Middle: An example for $t = 0.25$. The equations represent the region F. Bottom: probabilistic view of P and R scores.

We transform the prediction of concept and task class networks into binary using a cut-off threshold t. For each threshold t, an F1 score can be calculated. Similar to the notion of the ROC curve, we vary the threshold t and plot the F1 score for different values of t. The strength of a necessary relationship is then calculated as the area under the F1-versus-threshold curve. Intuitively the strongest necessary relationship happens when we have larger F1 scores for smaller values of t. That is, even a weak probability of belonging to the task class would require the concept to be present with a high probability.

The process for calculating P, R and $F1$ scores for the other three relationships is similar and is shown in Fig. 3. However, in practice, it is easier if the other three cases just be converted to a necessary condition and quantified using the same process as the necessary condition. For example, sufficient score can be obtained from the scatter plot of C vs. T (instead of T vs. C) because C being sufficient for T is equivalent to T being necessary for C. Likewise, the negative necessary score can be obtained from the scatter plot of T vs. $1 - C$, and the negative sufficient score can be obtained from the scatter plot of $1 - C$ vs. T.

For simplicity here, we assumed that the thresholds for C and $\neg T$ are the same, but, in general, these thresholds can be considered as threshold t_C for C and threshold t_T for T. In that case, the quantitative curve will be a 3D surface (the F1 score vs. t_C and t_T) and the volume under the curve should be used as the measure of the strength of the relationship.

For each of our experiments, we create four quantification curves based on the scatter plot. Each quantification curve shows the F1 score against different thresholds. We further use the area under the curve (AUC) to summarize each curve into a real-valued score between 0 and 1. An AUC value higher than 0.5 would indicate the existence of evidence for the corresponding relationship. For example, for the concept stereoscope and the task class physician, we expect a large necessary AUC. An AUC value less than 0.5 indicates that there is evidence against the corresponding relationship. For example, for the concept wrench and the task class physician, the necessary AUC is expected to be closer to 0. An AUC value around 0.5 would indicate there is no measurable relationship. For example, for the concept skin colour and the task class physician, we expect to see a necessary AUC of 0.5. In general, when the two variables T and C are probabilistically independent and have uniform distributions, it can be proved

that the AUC will be equal to 0.5 as follows:

$$P = \frac{TP}{TP+F} = P(C > 1-t|T > t) = P(C > 1-t) = t \quad (4)$$

$$R = \frac{TN}{TN+F} = P(T < t|C < 1-t) = P(T < t) = t \quad (5)$$

and then the $F1$ score will be

$$F1 = \frac{2PR}{P+R} = t \quad (6)$$

and the AUC we will be

$$AUC = \int_{t=0}^{1} F1 \, dt = \int_{t=0}^{1} t \, dt = 0.5. \quad (7)$$

D. Concept Hierarchy

Now that we have determined the presence of concepts in the embedding space, we can move on to quantifying the relationships between concepts and construct a hierarchy of the relevant concepts. To this end, we select the concepts that are detected in the embedding space, i.e., the accuracy of their corresponding concept classifier is high, and train a decision tree that tries to approximate the task classifier as an interpretable function of detectable concepts. At each node of the decision tree, we split based on a concept. The decision tree estimates the task classifier and not the ground truth and it will discover what combination of concepts might have been considered for a task class. Note that the decision tree is not unique. It provides "an" explanation rather than "the" explanation. It can be used to see if there are any non-sense interactions in the decision tree which is an interpretable surrogate model to the task classifier.

A decision tree can be constructed in two ways: A single decision tree can be trained to decide about all task classes based on all the present concepts. Alternatively, we can train a decision tree per task class and determine a hierarchy of related concepts per task class. The former provides a higher-level understanding of the multi-class problem as a whole. The latter allows us to isolate and quantify related concepts for each task class more effectively, resulting in a smaller tree. In our experiments, we examined both approaches.

The combination of concept classifier networks and the decision tree can be considered as an approximation of the original network. The accuracy of the decision tree combined with the accuracy of the concepts used for constructing it determine how faithful the combined model is to the original network. Assuming that the concepts are simpler than task classes, it is much easier to verify the accuracy and correctness of the concept classifiers and then explain the original network through the decision tree. Note that the concept classifiers and the decision trees are for explainability purpose, and it is the original network that performs the classification task.

Low faithfulness accuracy (accuracy based on faithfulness to the network outputs and not the true labels) of the resulting decision tree may suggest that the considered concepts are not all the information that the original network is using, and introducing other concepts can potentially make the decision tree more faithful to the task classifier. Regardless, the proposed method can quantify the relation between concepts and task classes and also the interactions between the provided concepts even if the provided concepts do not fully represent all the information the original neural network uses.

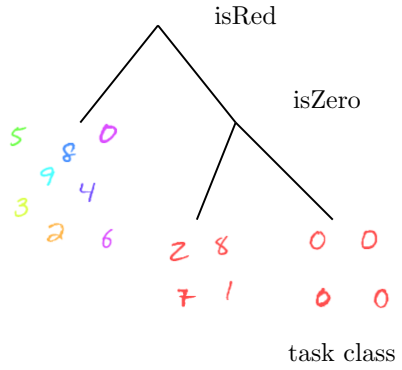

Fig. 4. Analysis of the relationship between the task class and concepts using a decision tree. The decision tree shows that both the concepts of isRed and isZero are necessary for the task class "Red Zero".

As a by-product of the resulting decision tree, each leaf can be considered as a new compound concept discovered by the decision tree. A compound concept consists of all concepts along the path from the root to a leaf. The newly discovered compound concepts can go through the same process as the original concepts and the four measures discussed earlier can be calculated for them. To this end, a dataset consisting of the compound concept (i.e. intersection of the related original concept sets) versus some random samples can be constructed.

In the next section, we will demonstrate the proposed methods over several experiments.

IV. Experiments and Results

In this section, we explore the application of the proposed methods in the evaluation of the relationship of neural network task classes and concepts in a controlled setting, a real-world setting and a medical application. The experiment in a controlled setting explains a neural network with Alexnet structure, and the experiment in a real-world setting uses a pretrained Resnet18. We compare our results with the TCAV [3] and IBD [6] methods, as they are the most related work to the proposed method. We have constructed two controlled datasets to simulate different scenarios where concepts have certain positive or negative relationships with task classes.

A. Coloured MNIST Dataset

MNIST is an image recognition benchmark dataset consisting of ten classes of handwritten digit images [26]. We modify this dataset to add useful or useless additional hints (as colour) to the neural network that will be trained on this data. In this way, we will train various neural networks with desired characteristics in terms of using certain concepts. Each of the ten task classes may correspond with multiple colours. The colour of each digit in a task class is chosen randomly from a set of two colours. We simulate two scenarios: 1) each digit is coloured using its own set of colours, –i.e. the colour concept and task class are fully correlated. We call

this dataset ColorDataset1. 2) all colourings are random (from the same set), i.e., there is no relation between the color concept and task class. We call this dataset ColorDataset2. We choose the colours of each class (its colour set) in a way that no two classes (colour sets) in the colour space can be linearly separated, see Fig. 5. This is to simulate the situation where concepts are not linearly separable analogous to an XOR problem which cannot be solved by a linear classifier. For generating the concept samples (for training the concept classifier), we shuffle pixels of images (to wipe out the image information) and then add colors to the foreground pixels. Each colour set (two colours) is considered a concept.

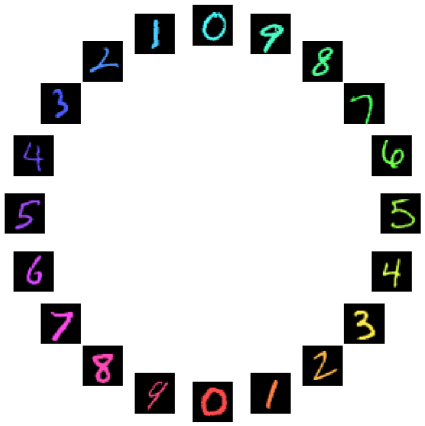

Fig. 5. Illustration of how the two hint colors are chosen for each class in the ColorDataset1. Each class is associated with two different colours that have the highest distance in the colour space.

B. Images with Captions Dataset

In this dataset, we add a hint caption to two classes of the Imagenet dataset [1], [2], namely class dog and class cat. The hint is added as white text on the image (by changing the pixels of the image). Consequently, some pixels of each sample carry extra information about the task class. We consider two scenarios: 1) the caption always reads the same as the image (the word cat for cat images, and dog for dog images). We call this dataset CaptionDataset1. 2) The caption is always a random word and hence does not include any information about the classification task (dogs vs. cats). We call this dataset CaptionDataset2. The captions have random rotation and scaling associated with them. Fig. 6 shows two samples of the CaptionDataset1 images.

For generating the concept samples (caption concept), we shuffled pixels of images (to wipe out the image information) and then added a caption to the resulting shuffled image.

C. Analysis of Causal Relationships between Concepts and Task Classes

In this experiment, we show the effectiveness of the proposed method in detecting the causal relationship between a concept and the task classes of neural networks. We train an Alexnet on each of our datasets. The results for CaptionDataset1 and CaptionDataset2 are shown in Fig. 7 (last

Fig. 6. Two samples from CaptionDataset1.

convolutional layer). Similar results were obtained on the ColorDataset1 and ColorDataset2. On the left side (a), it can be seen that the concept (caption dog) was detected to have a 98% necessary relationship with the class dog. All samples predicted by g_w as the dog class (above the red horizontal line) are predicted by $g_{w'}$ to have the caption concept (i.e. they are on the right of the red vertical line). Here, the red lines show a threshold of $t = 0.5$. Plotting the modified F1 score for different values t gives the necessary quantification curve (blue), which, as expected, has a high AUC (0.98). The results for CaptionDataset2 are shown in Fig. 7 (b). It can be seen that, as we expected, there is no tangible relationship between the dog class and the caption concept. This confirms that the proposed method detects the causal relationship between the caption concept and task classification. More results for other layers are provided in Fig. 8.

Now that we have established that the method can detect the usefulness of the hints, from now on we only use the CaptionDataset1 and ColorDataset1.

D. Informativeness of the Proposed Relationship Measures

In this experiment, we show how the proposed relationship measures can reveal more than just a correlation between a concept and a task class. We examine the third layer of an Alexnet with ten classes, trained on our ColorDataset1 to check the causal relationships between different target classes and a particular concept. In particular, we inspect the concept of the colour set associated with the class of handwritten digit one, which is a shade of blue and a shade of red as shown in Fig. 5. By design, this concept is a necessary condition for class 1 and negative necessary for all other classes. In other words, for class 1, it is necessary to have the concept (if the colours are not present, the task is not classified as class 1), but for other classes, like class 0, it is necessary not to have this concept (if the colours are present, the output will not be class 0). First, we quantitatively measure whether this concept is a necessary condition for class 1. The results are shown in Fig. 9(a). It can be seen that the proposed method successfully calculated a large necessary score i.e. AUC=0.86. Second, we quantitatively measure if this concept is a negative necessary for class 0. The results are shown in Fig. 9(b). It can be seen that the proposed method successfully calculated a large negative necessary score i.e. AUC=0.93.

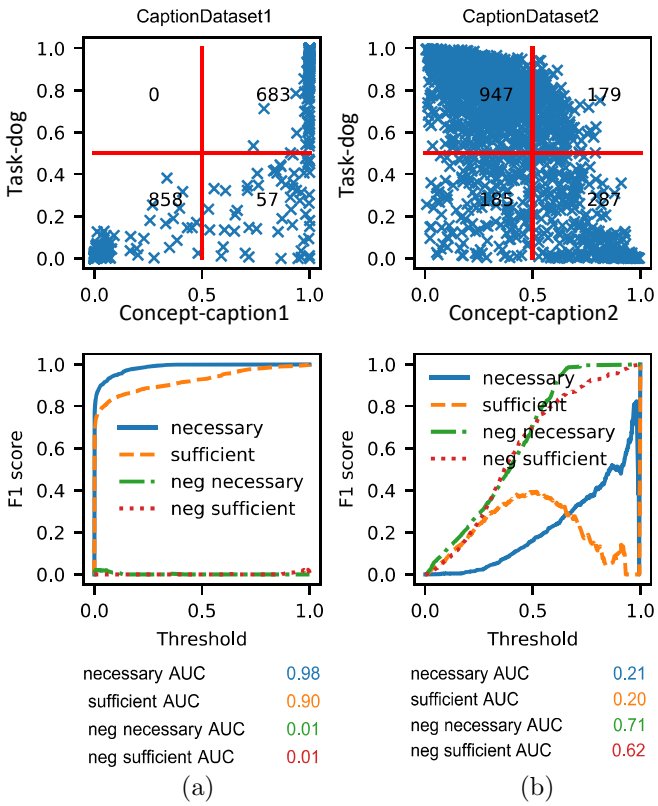

Fig. 7. Comparison of two networks with the same structure, (a) has been trained on CaptionDataset1, (b) has been trained on CaptionDataset2. The AUC of one for the necessary curve shows that the concept is necessary for the task class. The corresponding AUC values are mentioned at the bottom. These results are extracted from the last convolutional layer of a neural network with an Alexnet structure. The correlation coefficient between concept and task of the left side is 0.97, while the right side is -0.64.

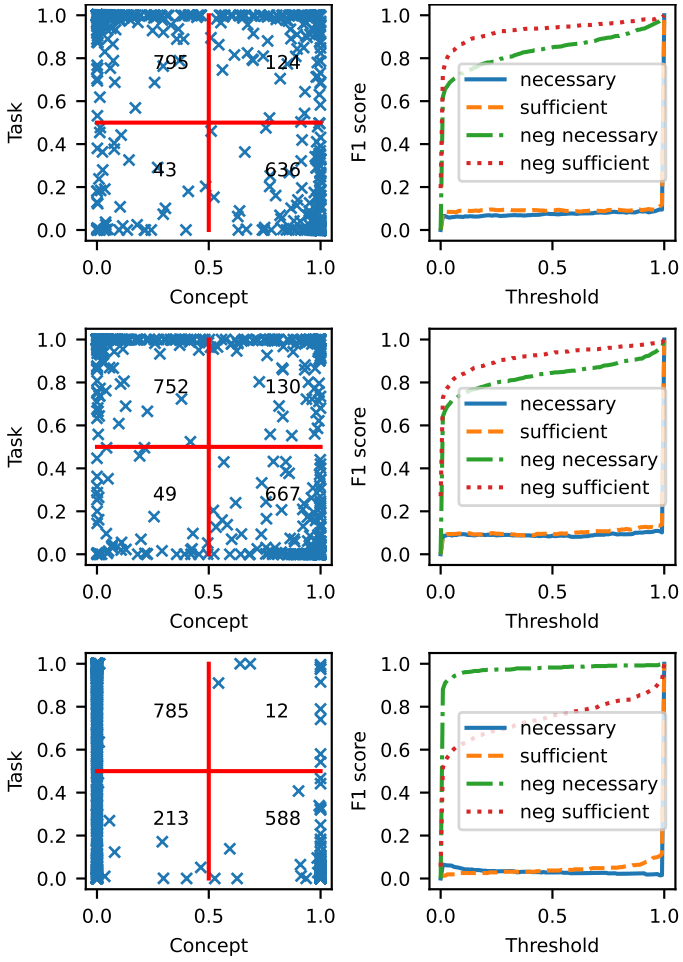

Fig. 8. Results for layers 3, 4, 5 for concept cat and class dog. The correlation coefficient between concept and task are respective -0.84, -0.82, -0.78. The results show that in earlier stages of the neural network, both negative necessary and negative sufficient are good descriptions of the neural network behaviour. But in last stages, the network is more tilted toward negative necessary. Meaning that in fact it is not solely relying on the caption information and some of image information is now dominant enough to help the neural network with the decision.

E. Comparison with a Linear Concept Classifier

As discussed in Section II, many methods use linear classifiers to extract concept information from a middle layer of neural networks [3], [6]. In this experiment, we examine the implications of using a linear classifier to detect concepts in network activations. We check the effectiveness of such an assumption in detecting the caption concept in an earlier layer (third layer) of Alexnet trained on our CaptionDataset1. By design, caption dog is both necessary and sufficient for the task class dog –i.e. all dog images and only dog images have the dog caption. Also, caption cat is a negative necessary and negative sufficient for the task class dog –i.e. it is necessary and sufficient not to have a caption cat to be class dog. In Fig. 10 we have quantified these relationships using two types of concept networks. On the right, we show the results of using a linear concept classifier i.e. we replaced $g_{w'}(z)$ with a linear classifier. On the left, we show the results of using a non-linear concept classifier, in particular, we use $g_{w'}(z)$ as a concept classifier. It can be seen that the proposed method has successfully quantified this relationship where the AUC scores for the negative necessary and negative sufficient conditions are high (78% and 89% respectively). But with a linear classifier, the four relationship scores have almost similar values and the designed relationship cannot be detected. These findings are consistent with those reported in [7].

This experiment shows that the linear classifiers have failed to extract concept information from earlier network layers, where activations usually represent low-level features. This phenomenon might be observed even at later layers of a neural network when the concept is merged with other concepts to create a higher-level concept. (see Fig. 12).

F. Comparison with TCAV

In many methods, the directional derivation of a classifier with respect to activations or input is considered a good representation of the local behavior of the model (first-order approximation) [3], [6]. As it was shown in TCAV [3], the directional derivative can be calculated by the dot product of the concept gradient and the task gradient. These methods use the dot product of a concept classifier gradient and a task classifier gradient as an agreement score evaluated on different class samples (linear assumption). Here, we demonstrate an example of where this score is not accurate. We examine the

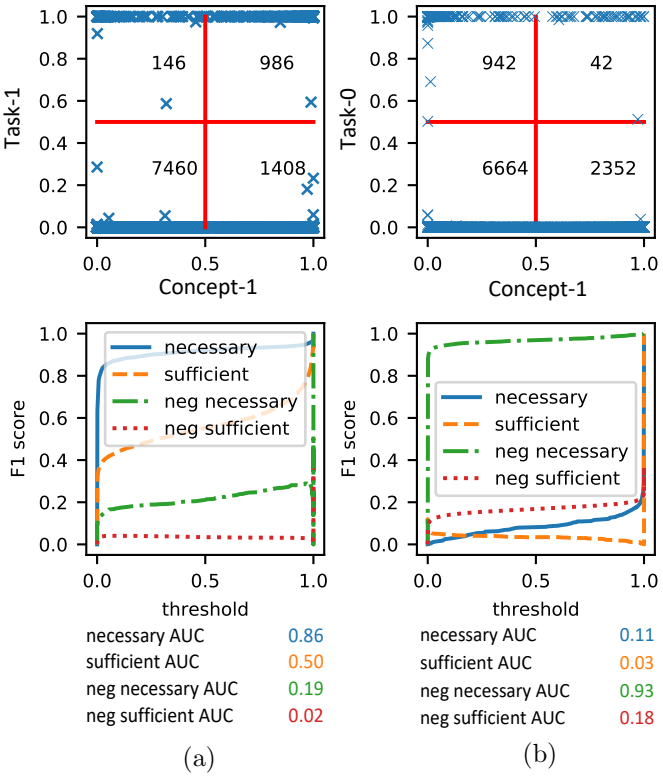

Fig. 9. The causal relationship for the concept of the color set of class one in the ColoredDataset1. (a) this concept is necessary for class one. (b) this concept is a negative necessary for all other classes. For example, results for class zero are shown on the right. The correlation coefficient between concept and task on the left side is 0.57, and it is -0.17 for the right side.

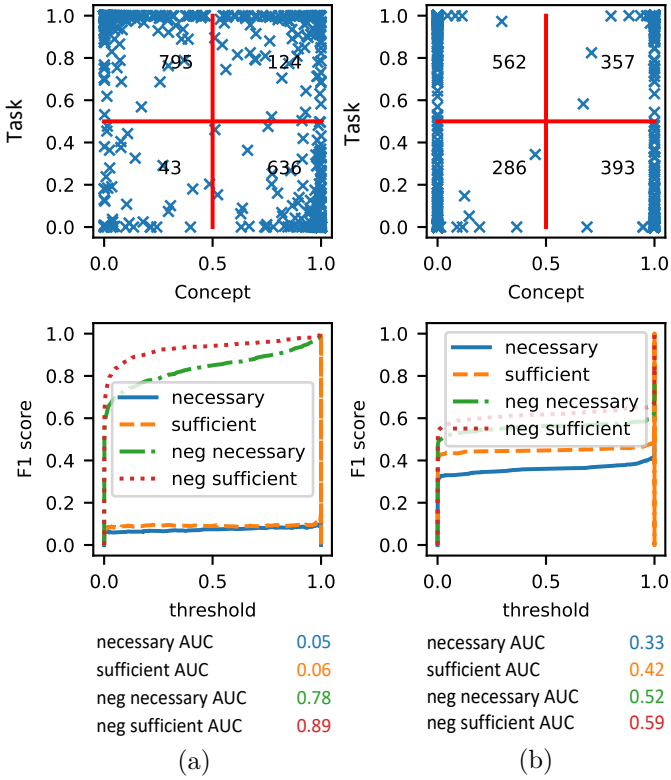

Fig. 10. Comparison with a linear classifier. We consider the concept of caption cat and its relationship to the task class dog. (a) Results of using the proposed concept classifier $g_{w'}(z)$. (b) Results of using a linear concept classifier. By design, the concept is negative necessary and negative sufficient for the task class dog. Unlike the linear model, the proposed method has easily detected these relationships. The accuracy of the nonlinear and linear concept classifiers are 0.84 and 0.6 respectively. The correlation coefficient between concept and task on the left side is -0.84, while for the right side it is -0.19.

relationship between the concept caption cat and the task class dog in the fifth layer of an Alexnet trained on CaptionDataset1. By design, these two pieces of information are inconsistent in the training data, –i.e. no training sample from class dog has caption cat. The distribution sample set consists of dog images with either caption dog or cat, and cat images with either caption dog or cat –i.e. all four possible combinations of cat and dog images with cat and dog captions. Therefore, for half of the distribution sample set, the image class and the added caption match and for the other half, images have a wrong caption. Fig. 11 (a) shows the distribution of directional derivatives between the concept and class using a linear model (similar to TCAV). Though the concept and class are by design inconsistent, the resulting directional derivatives are positive on all samples of the distribution sample set, showing that directional derivatives are not a reliable explanation. Similar results observed using a non-linear classifier $g_{w'}$ (see Fig. 11 (b)). Note that the proposed method successfully captured the correct relationship (negative necessary) with an AUC of 96%.

G. Comparision with the IBD method

Since the IBD method [6] is limited to convolutional networks, in this experiment, we modify our method to be comparable to IBD. We examine the last hidden layer of a Resnet18 trained on the Places365 data set [27] – a data set where each class is a place. This network was the benchmark

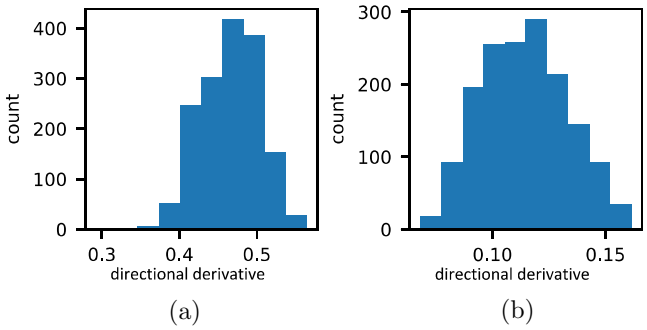

Fig. 11. Histogram of directional derivatives for the caption experiment discussed in Section IV-F using (a) a linear concept classifier, and (b) a nonlinear concept classifier.

of the IBD method. We use the same set of concept classifiers they trained (with the parameters they provided). We use 10,000 samples from the places365 validation set without their labels as the distribution sample set. Our concepts come from the same dataset that the IBD method used as their benchmark, Broden [4] – a dataset with segmentation annotations. The pixel-wise annotation of the Broden dataset can improve the training process if the output of the network is modified so it can reflect the spatial information preserved in the annotations.

This technique improves the accuracy of the concept classifier and provides localized behaviour of the concepts. Such labels can be used to visualize as heat maps where the network detects specific concepts like the heatmaps in [6]. For the sake of comparison, we use only the concepts originally used in the IBD benchmark.

This experiment is designed to find the most important concepts for classifying each class of the Places365 dataset (see Table I). For each concept, a concept classifier is trained, and then each task class (a class from Places365) is examined against each concept. The necessary scores of concepts for each task class are sorted, and the highest values are reported as the most necessary concepts for the class. The most necessary concepts are then compared against the concepts recommended by IBD which are based on the decomposition of the decisions into concept space. The top seven are reported for both methods in Tables I). The concepts are from left to right in decreasing importance. Three human annotators were asked to label the concepts as relevant or irrelevant to the corresponding class. A concept is labeled irrelevant if the majority of annotators have found it to be irrelevant. Irrelevant concepts are shown with a strikethrough text in the Table I. It can be seen that the proposed method has identified more relevant concepts than the IBD method. For instance, for the task class topiary garden, IBD suggested the concepts tail and sheep (among five others) which are irrelevant to the topiary garden class. On the other hand, our method suggested plant and tree which are quite relevant concepts to the topiary garden class. For the soccer field class, our method suggested grass, pitch, grandstand, court, person, post and goal which are all relevant. But IBD suggested pitch, field, cage, ice rink, tennis court, grass, and telephone booth where cage, ice rink, tennis court, and telephone booth are irrelevant to the soccer field class (see Table I). The eleven classes shown in Table I are those also mentioned in [6]. In addition, we evaluated the top-7 concepts for nineteen additional randomly selected classes for a total of thirty classes and calculated the error rate for IBD and the proposed method. Error is defined as the proportion of irrelevant concepts among the top-7 concepts. The error rate of the proposed method and IBD are respectively $\%11.4$ and $\%26.2$ which demonstrates the superiority of the proposed method.

We also realized that the quality of the IBD benchmark concepts is not verified. So we sorted the samples in the distribution sample set according to the output of each concept classifier network. Figures 12 shows the results for concepts sheep, cow and elephant. For each concept, the first row shows the five images of the minimal concept classifier output in increasing relevance order from left to right. In the second row, the five images that maximize the concept are shown in the increasing relevance order from left to right. This test acts as a sanity check on concept learning at a particular layer and shows how a concept may look like in that layer and may reveal if the concept is not reliable. For instance, we found that the concepts sheep, cow, and elephant are not reliable and maximized whenever a dirt field is present, (for example a horse race track) as shown in Fig. 12. The images that are among the most relevant but in our view are not relevant to the concept are marked with a red box around them. If the accuracy of a concept classifier in a layer is poor, that concept is forgotten and one should not look for relationships between something that effectively does not exist and the task class. The IBD method blindly uses such unreliable concepts in its explanations, but the proposed method verifies the accuracy of the concept classifiers and avoids using such concepts. The issue becomes more serious for IBD because when it does decomposition over concepts, error of one unreliable concept propagates and affects the decomposition process and consequently the scores of other concepts. Fig. 12 shows three examples of what a low-accuracy concept classifier has actually learned. For example, for the elephant concept, the last row of Fig. 12 provides more insight about what that classifier has learned. Instead of elephants, we see horses and cows.

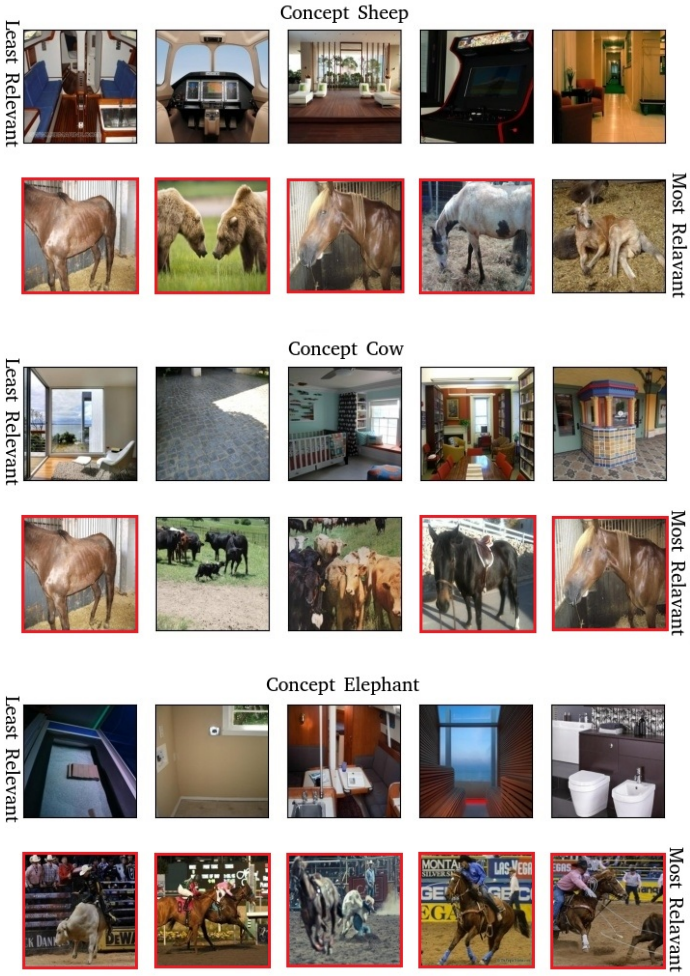

Fig. 12. Some minimal and maximal samples of animal related concepts. These samples show that without a quality control step, the concepts can capture irrelevant information.

H. Concept Hierarchy Analysis

In this section, we demonstrate the proposed method for finding a hierarchy of concepts on two applications: scene classification and prediction of Osteo-arthrosis severity from X-ray images.

Class: topiary_garden

Proposed	plant	hedge	tree	brush	flower	bush	sculpture
IBD	hedge	brush	~~tail~~	palm	flower	~~sheep~~	sculpture

Class: crosswalk

Proposed	crosswalk	road	sidewalk	post	~~container~~	streetlight	traffic light
IBD	crosswalk	minibike	pole	rim	~~porch~~	central reservation	van

Class: living_room

Proposed	armchair	sofa	back	cushion	back pillow	coffee table	ottoman
IBD	armchair	fireplace	inside arm	shade	sofa	frame	back pillow

Class: market/indoor

Proposed	pedestal	sales booth	shop	case	bag	bulletin board	food
IBD	sales booth	pedestal	food	~~fluorescent~~	shop	shops	apparel

Class: soccer_field

Proposed	grass	pitch	grandstand	court	person	post	goal
IBD	pitch	field	~~cage~~	~~ice rink~~	~~tennis court~~	grass	~~telephone booth~~

Class: forest/broadleaf

Proposed	tree	bush	trunk	~~cactus~~	brush	~~fire~~	leaves
IBD	tree	trunk	bush	leaves	~~semidesert~~	~~grid~~	~~clouds~~

Class: art_school

Proposed	person	hand	paper	~~plaything~~	fabric	bag	board
IBD	paper	drawing	~~plaything~~	painting	hand	board	figurine

Class: dining_hall

Proposed	drinking glass	stool	table	spindle	menu	person	plate
IBD	plate	light	stool	sash	napkin	display board	spindle

Class: butte

Proposed	mountain	hill	desert	badlands	rock	valley	land
IBD	hill	badlands	desert	cliff	~~cloud~~	mountain	~~diffusor~~

Class: canyon

Proposed	mountain	rock	cliff	hill	badlands	land	desert
IBD	cliff	mountain	badlands	desert	pond	~~bumper~~	hill

Class: coast

Proposed	sea	sand	land	embankment	rock	~~mountain~~	water
IBD	sea	wave	land	~~mountain pass~~	sand	cliff	~~cloud~~

TABLE I
TOP SEVEN CONCEPTS SUGGESTED BY THE PROPOSED METHOD AND THE IBD METHOD. THE STRIKETHROUGH TEXTS SHOW THE ERRORS WHERE A CONCEPT THAT IS IRRELEVANT TO THE CORRESPONDING TASK CLASS IS SELECTED. THE FLEISS' KAPPA SCORE IS 0.65 MEANING MODERATE AGREEMENT BETWEEN ANNOTATORS.

1) Learning Concept Hierarchy for Scene Classification: In this experiment, we examine the neural network trained on the places365 dataset [27] and try to explain task decisions based on the concepts provided in the Broden dataset [4]. We test the proposed approach under two scenarios. In the first scenario, a decision tree is trained for explaining each task class. In the second scenario, a multi-class decision tree is trained for explaining the neural network as a whole. Fig. 13 shows the decision tree for the "zen garden" class. Using this tree we can easily see what concept combinations correspond to this class and what concepts' presence can stop the classifier from labelling as "zen garden". We also trained a multi-class decision tree to predict the multi-class decisions of the entire classifier. Since the full description of the classifier is too long, a branch of the resulting decision tree is provided in the supplementary material.

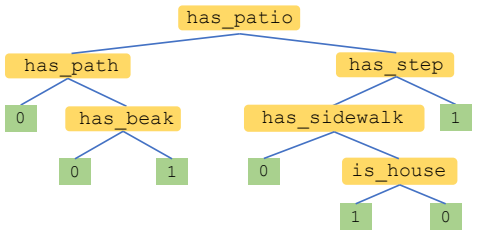

Fig. 13. Concept hierarchy found by the proposed method for the task class of zen garden.

2) Learning Concept Hierarchy for Predicting Osteoarthrosis from X-ray images: In this experiment, we examine a neural network trained to classify Osteo-arthrosis KLG score [28]. The KLG score is a 5-grade integer value between 0 and 4 that measures the severity of Osteoarthrosis in joints. We follow the preprocessing procedure mentioned in [29] and used in [7]. We used the OAI dataset [30] to train a Resnet18 model to classify KLG scores based on the knee joint's X-ray images. OAI (Osteo-Arthrosis Initiative) is a data-gathering

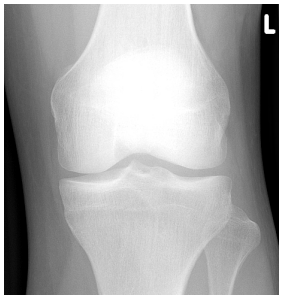

Fig. 14. Sample of a knee X-Ray input image for KLG classifier. The letter L indicates that this is the image of the patient's Left leg.

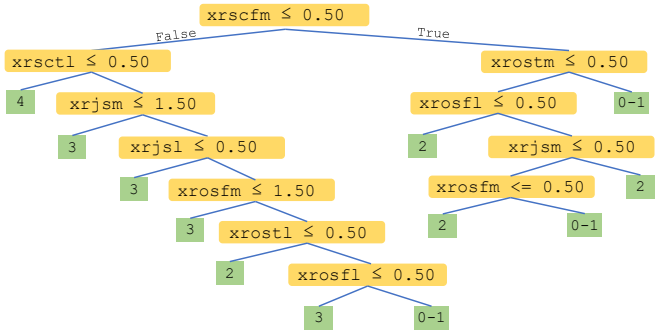

Fig. 15. Concept hierarchy found by the proposed method for the neutral network that predicts Osteo-arthrosis KLG score. The numbers inside the leaf nodes (green) are KLG scores (four tasks). Each node (yellow) is a binary test (concept). For example, at the root, xrscfm denotes a certain medical condition —Sclerosis (OARSI grades 0–3) femur medial compartment— and xrscfm$\leq$0.5 suggests the condition is relatively low and, as expected, the tree generates relatively low KLG scores (right subtree).

initiative from multiple sources on Osteo-Arthrosis disease. The collection includes X-ray, MRI, etc. images of different joints of patients taken every six months for up to 96 months. Labels for this dataset were created by aggregating experts' opinions. Here we use only X-ray knee images that have concept annotations and KLG score (16249 images). We used the same subset of concepts previously used in [7]. The X-ray images are reshaped into 512×512 pixels. Then the task classifier is examined against concepts to see whether and how the task classifier is influenced by these concepts. In other words, whether the task classifier is following the expert's way of determining KLG scores. Note that all patient information has been removed by OAI and only the leg side (L/R) is remained on the image (see Fig. 14).

To prevent overfitting, we hard-decision the predicted concept values using a severity threshold of 0.5 and consequently, concepts become binary values. Also, we stop splitting a node as soon as there are less than 2 samples per class in that node. We also merge back the leaves that have similar labels. The resulting decision tree is shown in Fig. 15. In general, the presence of concepts usually is indicative of a more severe condition and hence a higher KLG score. The decision tree is consistent with the rule-of-thumb that whenever a medical condition (concept) is high (more severe), the KLG score is higher (more severe osteoarthritis) than if the medical condition was low. Considering the decision tree, at each node, whenever we go right (the medical condition/concept is low), we end up in KLG scores (leaves) that are equal to or smaller than the KLG scores we would get had we gone left (the medical condition was high).

V. Discussions

The process of learning a hierarchy of concepts involves two steps: first, learning a model for predicting each concept, and then representing the task classifier by a decision tree. How faithful is the decision tree to the model behaviour is determined by both the accuracy of the decision tree and concept networks.

Consider an example where a model is trained to classify MNIST images where each number has its own color. Different from the models considered earlier, this model has disentangled representations of "digit shape" and "digit color", namely, z equals "[shape vector, color vector]", and it only uses the "shape vector" to do classification ($g_w(z)$ only depends on the "shape vector"). Also, assume that the concept classifier $g_{w'}(z)$ utilizes only the "color vector". If we construct a scatter plot of the training set (color-coded) then there will be high necessary and high sufficient scores. (i.e. the top left and bottom right corners will be empty due to the prefect correlation). On the other hand, if we construct a scatter plot using a distribution sample set that represents the real distribution of data where the color of digits is random, then the samples will be randomly distributed along the concept axis and therefore the color concept will not be a cause of task class. Note that in both cases, the same neural network (disentangled z), task and concept classifiers described above were used, and the difference is only in what data set is passed through them to construct the scatter plots. In both cases, the proposed metrics correctly reflect the relationship between the "shape vector" and the "color vector", although the results depend on the set used to construct the scatter plots. The proposed metrics are calculated on a distribution sample set and, therefore, inform us about what happens when such data are passed through the network. They cannot tell us what the relationships will be if a new set of samples from a totally different distribution is passed through the network. The fact that the distribution sample set does not need any kind of labeling provides a lot of flexibility in creating it; however, it just has to be a good representation of real data.

The choice of which layer to inspect is not a straightforward decision. Inspection of the later layers is computationally cheaper (since the training of the concept classifier is cheaper). But there is no guarantee that the concepts are still present in those layers since the network might have traded them with a combination of concepts more useful for task classification. For instance, presence of a stethoscope might not be detectable in the last layer, but the presence of a medical instrument might be possible (distinguishing images that include a medical instrument from the ones that don't). For this reason, we start our analysis from the last layer in the network and work our way back until we reach a layer that the concept is present (good accuracy of concept network) or reach the first layer which would indicate that the concept is too complex to be

detected by the network.

VI. Conclusion

We proposed a framework for verifying the presence of human-friendly concepts in activations of intermediate layers of a neural network. The proposed method quantitatively determines the causal relationship between a concept and the neural network task classes. We also determined the interactions between concepts with respect to a task class by constructing a hierarchy of concepts in the form of a decision tree. We showed the effectiveness of the proposed methods through several comparative experiments on synthetic and real-world datasets, demonstrating improved performance compared with the previous methods.

Acknowledgement

This work was supported by the Natural Sciences and Engineering Research Council, Vector Institute, and Compute Canada.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[3] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres, "Interpretability beyond feature attribution: Quantitative Testing with Concept Activation Vectors (TCAV)," *35th International Conference on Machine Learning, ICML 2018*, vol. 6, pp. 4186–4195, 2018.

[4] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 3319–3327.

[5] Y. Goyal, A. Feder, U. Shalit, and B. Kim, "Explaining classifiers with causal concept effect (cace)," *arXiv preprint arXiv:1907.07165*, 2019.

[6] B. Zhou, Y. Sun, D. Bau, and A. Torralba, "Interpretable basis decomposition for visual explanation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11212 LNCS, pp. 122–138, 2018.

[7] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5338–5348.

[8] J. Pfau, A. T. Young, J. Wei, M. L. Wei, and M. J. Keiser, "Robust semantic interpretability: Revisiting concept activation vectors," *arXiv preprint arXiv:2104.02768*, 2021.

[9] J. Adebayo, J. Gilmer, I. Goodfellow, and B. Kim, "Local explanation methods for deep neural networks lack sensitivity to parameter values," *arXiv preprint arXiv:1810.03307*, 2018.

[10] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim, "The (un) reliability of saliency methods," *arXiv preprint arXiv:1711.00867*, 2017.

[11] M. Nokhbeh-Zaeem and M. Komeili, "Cause and effect: Concept-based explanation of neural networks," in *2021 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2021.

[12] R. Fong and A. Vedaldi, "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8730–8738.

[13] S. Singla, S. Wallace, S. Triantafillou, and K. Batmanghelich, "Using causal analysis for conceptual deep learning explanation," *arXiv preprint arXiv:2107.06098*, 2021.

[14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, pp. 1–10, 2014.

[15] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," 2016. [Online]. Available: http://arxiv.org/abs/1610.01644

[16] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, vol. 2, no. 11, p. e7, 2017.

[17] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim, "Towards automatic concept-based explanations," in *Advances in Neural Information Processing Systems*, 2019, pp. 9277–9286.

[18] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, and P. Ravikumar, "On completeness-aware concept-based explanations in deep neural networks," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[19] V. Kamakshi, U. Gupta, and N. C. Krishnan, "Pace: Posthoc architecture-agnostic concept extractor for explaining cnns," *arXiv preprint arXiv:2108.13828*, 2021.

[20] X. Cheng, C. Chu, Y. Zheng, J. Ren, and Q. Zhang, "A game-theoretic taxonomy of visual concepts in dnns," *arXiv preprint arXiv:2106.10938*, 2021.

[21] A. Ghandeharioun, B. Kim, C.-L. Li, B. Jou, B. Eoff, and R. W. Picard, "Dissect: Disentangled simultaneous explanations via concept traversals," *arXiv preprint arXiv:2105.15164*, 2021.

[22] M. T. Bahadori and D. E. Heckerman, "Debiasing concept bottleneck models with instrumental variables," *arXiv preprint arXiv:2007.11500*, 2020.

[23] Z. Chen, Y. Bei, and C. Rudin, "Concept whitening for interpretable image recognition," *Nature Machine Intelligence*, vol. 2, no. 12, pp. 772–782, 2020.

[24] P. Barbiero, G. Ciravegna, F. Giannini, P. Lió, M. Gori, and S. Melacci, "Entropy-based logic explanations of neural networks," *arXiv preprint arXiv:2106.06804*, 2021.

[25] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson, "The what-if tool: Interactive probing of machine learning models," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 56–65, 2019.

[26] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[27] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[28] J. Kellgren and J. Lawrence, "Osteo-arthrosis and disk degeneration in an urban population," *Annals of the Rheumatic Diseases*, vol. 17, no. 4, p. 388, 1958.

[29] E. Pierson, D. Cutler, J. Leskovec, S. Mullainathan, and Z. Obermeyer, "Using machine learning to understand racial and socioeconomic differences in knee pain," in *NBER Machine Learning and Healthcare Conference*. NBER, 2019.

[30] M. Nevitt, D. Felson, and G. Lester, "The osteoarthritis initiative," *Protocol for the Cohort Study*, vol. 1, 2006.