This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Causal Analysis for Robust Interpretability of Neural Networks

Ola Ahmad[1] Nicolas Béreux[2*] Loïc Baret[1] Vahid Hashemi[3] Freddy Lecue[4*]
[1] Thales Digital Solutions, CortAIx, Montreal, Canada
[2] Paris-Saclay University, Paris, France
[3] AUDI AG, Ingolstadt, Germany
[4] Inria, Sophia Antipolis, France
ola.ahmad@thalesdigital.io

Abstract

Interpreting the inner function of neural networks is crucial for the trustworthy development and deployment of these black-box models. Prior interpretability methods focus on correlation-based measures to attribute model decisions to individual examples. However, these measures are susceptible to noise and spurious correlations encoded in the model during the training phase (e.g., biased inputs, model overfitting, or misspecification). Moreover, this process has proven to result in noisy and unstable attributions that prevent any transparent understanding of the model's behavior. In this paper, we develop a robust interventional-based method grounded by causal analysis to capture cause-effect mechanisms in pre-trained neural networks and their relation to the prediction. Our novel approach relies on path interventions to infer the causal mechanisms within hidden layers and isolate relevant and necessary information (to model prediction), avoiding noisy ones. The result is task-specific causal explanatory graphs that can audit model behavior and express the actual causes underlying its performance. We apply our method to vision models trained on classification tasks. On image classification tasks, we provide extensive quantitative experiments to show that our approach can capture more stable and faithful explanations than standard attribution-based methods. Furthermore, the underlying causal graphs express the neural interactions in the model, making it a valuable tool in other applications (e.g., model repair).

1. Introduction

Explainability and interpretability are crucial for deep neural networks (DNNs), which are disseminated in many applications, including vision and natural language processing. Despite their popularity, their opaque nature limits the adoption of these "black-box" models in domains requiring critical decisions without the ability to understand their behavior. Attempts to provide a transparent understanding of DNN systems have led to the development of many interpretability methods. Most of them focus on interpreting the function of DNNs through correlation-based measures, which attribute the model's decision to individual inputs [30]. The most popular ones are saliency (or feature attribution) methods [2, 8, 16, 25, 27, 29, 31, 33].

Saliency methods aim to help the user (or developper) understand why a DNN made a particular decision by capturing relevant features to certain model's prediction. However, we observe two considerable limitations of these methods. First, they cannot explain the inner function of the neural system being examined. That means how internal neurons interact with each other to reach a particular prediction. As reported in [6], it is difficult to verify claims about black-box models without explanations of their inner workings. A second limitation, they are susceptible to noise and spurious correlations. Whether due to a property of the DNN system obtained during the training phase (e.g., biased inputs, overfitting, or misspecification) or the method being used to capture saliency [13, 14] as shown in Fig. 1). Alternatively, some methods seek to visualize the behavior of specific neurons [18] but cannot provide clear insights due to their large number and overall complex architectures.

In this paper, we propose a novel method that addresses the above limitations through the angle of causality. We show that a technique grounded in the theory of causal inference provides robust and faithful interpretations of model behavior while being able to reveal its neural interactions. Inspired by neuroscience, we analyze individual neurons' effects on model prediction by intervening in their connections (model's weights or filters).

We summarize our contributions as follows. a) We propose a robust interpretability approach to capture meaningful semantics and explain the inner working of DNNs. b) Our methodology relies on path interventions and cause-

*Work done while at Thales Digital Solutions

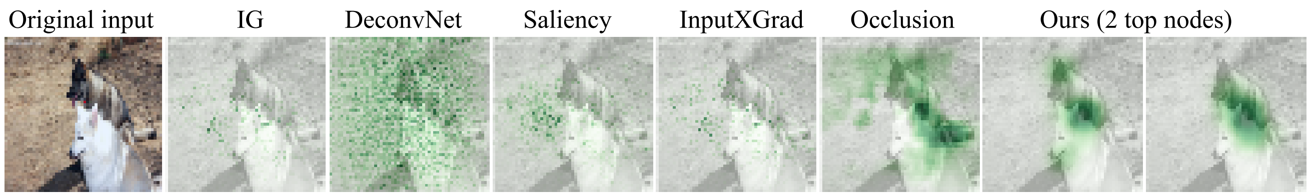

Figure 1. **Features importance from different explanation methods on a hard example from ImageNet data.** The actual class is "white wolf". The predicted class by the pre-trained ResNet18 is "Malamute". To the right is our method showing top 2 semantics (top head and body of wrong class) from the causal graph that explain the prediction. Other methods either fail or provide noisy features.

effect relations, providing stable and consistent explanations. More specifically, we seek to answer questions such as: ***would the model's prediction have been higher if we prevented the flow of signals through particular paths?*** or, ***what would have been the decision of the model had we attenuated or removed an individual or a set of components at a particular layer?***. Our analysis will lead to locating and isolating relevant and necessary information strongly and causally connected to model prediction up to a test of significance. c) We apply our method to vision models trained to classify MNIST, CIFAR10, and ImageNet data. d) We provide a flexible framework that can be applied to complex architectures and other tasks beyond interpretations.

2. Related Work

Interpretability and Attribution Methods. Interpretability for deep neural networks aims to provide insights into black box models' behavior. A broad family of methods has been developed in the past few years. The most common techniques are attribution methods which assign scores to input features indicating the contribution of each one to model's prediction. Gradient-based methods [2, 25, 26, 28, 29, 32] propagate pre-trained models' gradients from output backward to input. Recent studies have pointed out that these methods produce noisy and unstable attributions [1, 13]. Perturbation-based methods [19, 20, 31] are alternatives that focus on correlations between local perturbations of raw inputs and model output. They are black-box methods in the sense that they don't require access to the inner state of the model. Beyond these widely used techniques, various interpretability methods have been proposed [35]. Close to our work is [34]. They suggest disentangling knowledge hidden in the internal structure of DNNs by learning a graphical model. Their work focuses on convolutional neural networks (CNNs), where they fit the activations between neighboring layers. Our approach differs in what it considers explanatory graphs and how it infers them. We rely on causal analyses, which have been recently considered as an effective tool for DNN explainability. Our framework does not assume a specific type of neural network, which makes the approach generic and flexible.

Capturing Explanations with Causality. More recently, causal approaches have been considered for interpreting DNNs. The inner structure of DNN has been viewed, for the first time, as a structural causal model (SCM) in [7]. They use SCM to develop an attribution method that computes the causal effect of each input feature on the output of a recurrent neural network. Other causal approaches were specifically developed to explain NLP-based language models, such as causal mediation analyses [30] and causal abstraction [9]. In contrast, [24] have developed a model-agnostic approach (CXPlain) to estimate feature importance for model interpretations. They use a causal objective to train a separate supervised model (U-net) to learn causal explanations for another black-box model. The trained explainer identifies the extrinsic effect of an input on causing a marginal change in the output of the target model. Hence, it cannot link explanations to the model's internal structure, keeping it as a black box.

3. Causal Graph Inference of Neural Networks

3.1. Notation and Intuition

Notation We denote by $\boldsymbol{x}$ an input image (without loss of generality), and $y \in \mathbb{R}^{n_y}$ its corresponding output label, where n_y is the number of classes. We also denote by $\hat{y} \in \mathbb{R}^{n_y}$ its predicted output obtained by a pre-trained neural network $N(L)$ composed of L layers. We define the relation $l \to l+1$ to refer to directed edges or connections between hidden nodes of layers l and $l+1$, respectively. At every hidden node j of the l-th layer, we define features or activation map a_j^l. We denote by causal graph $\mathcal{G}$ an abstraction of $N(L)$ as shown in Fig. 2 (b) and (d). We use the term explanatory graph to refer to causal graphs composed of nodes encoding relevant features.

Intuition The activated signals $\boldsymbol{a}^l$ flow to the next layer $l+1$ through weighted edges $\boldsymbol{W}^{l \to l+1}$ connecting hidden nodes of layers l and $l+1$. These weights control the strength of information flow between two layers in a manner

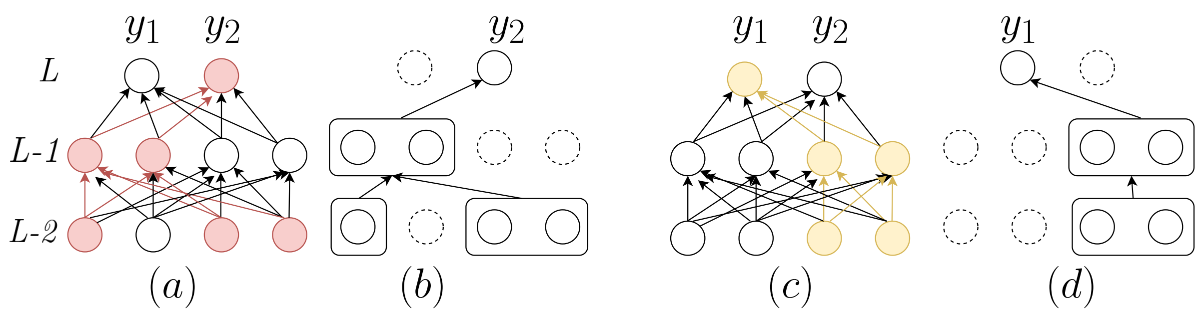

Figure 2. **Example of causal connections in the last three layers of a neural network.** (a) and (c) Coloured paths (red/yellow) transpose signals between layers to labels y_1/y_2, respectively. (b) and (d) Two abstract graphs discovered by causal inference. In each graph, neutral neurons (dot circles) encode varied information independent of the label information.

physically analog to a switch. In physical systems, manipulating the state of a switch (e.g., on-off or via continuous interventions) would change the system's physical condition, thereby providing an interpretation of its behavior. We set this intuition to motivate our work.

3.2. Problem Formulation

Our goal is to discover causal explanatory graphs of $N(L)$ via path (equiv. weights) interventions. Formally, we set the problem as follows. Let $\boldsymbol{W}^{l \to l+1} \in \mathbb{R}^{n_c \times n_p}$ be the weight matrix of the directed edges from layer l to $l+1$, where n_p is the number of parent nodes in l and n_c is the number of child nodes in $l+1$. These nodes define a subgraph $\mathcal{G}^{l \to l+1}$ (see Fig. 2). Let $\boldsymbol{w}^{l \to l+1} \in \mathbb{R}^{n_t \times n_p}$, be the paths connecting n_p nodes in l to a selected subset of nodes n_t in $l+1$ ($n_t < n_c$). Our problem is then to estimate the treatment (or causal) effect (TE) resulting from intervening on the weights $w_j^{l \to l+1}$ at given node j:

$$\boldsymbol{P}\{TE(do(w_j^{l \to l+1}); \hat{\boldsymbol{y}}, \boldsymbol{X}, \boldsymbol{W} \setminus w_j^{l \to l+1}) = 0\} < \alpha, \quad (1)$$

where $do(w_j^{l \to l+1})$ is a mathematical operation referring to the action of interventions. $\boldsymbol{X}$ is a subset of class-specific inputs in the data manifold, and $\hat{\boldsymbol{y}}$ are their predictions (pre-softmax layer). $\alpha > 0$ is a probability threshold (equiv. p-value) that measures "significance". The formula in (1) defines a form of hypothesis testing where the null hypothesis states that interventions on the paths at parent node j do not affect or change the model's predictions. Rejecting the null hypothesis suggests that an effect exists indicating that j a causal node.

3.3. Causal Inference

In this section, we provide the details of our methodology for solving (1). We focus on vision models which encompass a set of convolution and MLP layers. Specifically, we use LeNet [15] with MNIST data for ease of explanations. The experiments section shows applications on other datasets and more complex architectures.

Treatment Effects The first step of our approach is to compute the effects of path interventions on model outputs. Let us consider the MLP example in Fig. 2 (a) and (c). The interventions on the paths in the last hidden layer $L-1$ allow measuring the effect on the outputs (y_1 and y_2) directly. Meanwhile, for layer $L-2$, the effects of interventions are mediated by the responses of the hidden neurons in descendant layers; here, the child layer $L-1$. The strength of response to path interventions depends on the structure and complexity of neural networks. Our goal is thus to analyze how significant these effects are. First, we define the treatment effect as a measure of the difference corresponding to path interventions.

Definition 1 *(**Treatment Effect**) Let $\boldsymbol{X}$ be a set of input features and $\hat{\boldsymbol{y}}$ the outcomes of a neural network $N(L)$, assumed to be aligned with the ground truth. Let $w_j^{l \to l+1} \in \mathbb{R}^{n_t}$ be the weighted edges connecting node j in layer l to the subset nodes n_t in layer $l+1$. We define treatment effect by holding all other weights $\boldsymbol{W} \setminus (w_j^{l \to l+1})$ fixed and intervening on $w_j^{l \to l+1}$ (i.e., $do(w_j^{l \to l+1})$):*

$$\begin{aligned} &TE(do(w_j^{l \to l+1}); \hat{\boldsymbol{y}}, \boldsymbol{X}, \boldsymbol{W} \setminus w_j^{l \to l+1}) = \\ &\hat{\boldsymbol{y}}_{w_j^{l \to l+1}=u_1}(\boldsymbol{X}) - \hat{\boldsymbol{y}}_{w_j^{l \to l+1}=u_0}(\boldsymbol{X}) \end{aligned} \quad (2)$$

where u_1 and u_0 are the intervention variables defined below. Eq. (2) measures the change of output distribution, for inputs $\boldsymbol{X}$, resulted from manipulating certain connections of the node j. By considering all parent nodes n_p, we obtain a set of distributions or hypotheses $\{TE\}_{j=1,\ldots,n_p}$ representing the effects of individual neurons. Fig. 3, shows examples on inputs of MNIST while interventions correspond to removing connections in the hidden layers of a LeNet model.

Test of Significance To capture the causal nodes in the parent layer l, we consider the hypothesis testing as formulated in eq. (1). We use a two-sided *t-test* to capture positive

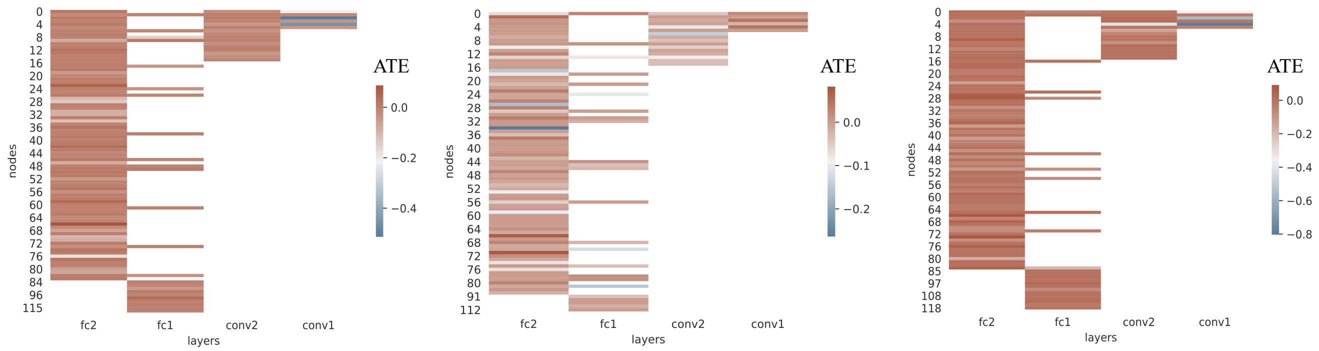

Figure 3. **Heatmap of the Average Treatment Effect (ATE).** We show the mean effect of path interventions on the convolution and linear layers of LeNet architecture for inputs of class 3, 7 and 8 respectively. Y-axis indicates the total number of nodes overall layers. For instance, conv1 has 6 nodes (channels) and the last hidden layer fc_2 has 84 nodes. The colorbar represents the change of model's outputs.

and negative effects given certain threshold determined by the null hypothesis at a pre-defined significance level α set to 0.05 in all our expriments.

Path Interventions Following the intuition of our work, we propose the interventions $w_j^{l\rightarrow l+1} = u$ such that $u = \beta w_j^{l\rightarrow l+1}$, where β can either be binary (remove/keep connections) or real (attenuate connection's effect). In the binary case, $\beta \in \{0, 1\}$ leading to $u_1 = 0$ and $u_0 = w_j^{l\rightarrow l+1}$. In the real case, we sample β from a uniform distribution $\mathcal{U}(b-\epsilon, b+\epsilon)$, where $\epsilon < b < 1.0$ is a predefined parameter and $\epsilon = 0.01$. We use this setting only to evaluate the reliability of the estimated causal graphs (sec. 5.1).

3.4. Path Selection

So far, we explained how to solve (1) using $w_j^{l\rightarrow l+1}$ for each parent node j. These weights correspond to a subset of targets we identify here via path selection. Indeed, manipulating all possible connections for a node j given parent layer l is computationally expensive and intractable for complex architectures with many neurons. An efficient way is to pay attention to specific paths and nodes via selection criterion. We propose a top-down approach starting from a specific output (e.g., class). It implies sequential processing starting from the last layer until reaching layer l. Let us consider we seek to compute the effects of path interventions of LeNet's layer $L-2$ for digit 3 (Fig. 4). We start with the paths directed from all nodes in the parent layer $(L-1)$ to node 3 of the output layer L. We use eq. (1) to discover the causal nodes in $(L-1)$, up to a significance test $\alpha = 5\%$. To identify the impact of these nodes on model's output, we must look at the behavior of causal effects (eq. (2)). Negative effects explain a drop in performance, while positive ones explain an improvement. The nodes corresponding to significantly negative effect are necessary (or critical) for that output (the red ones in Fig. 4). In contrast, we discover

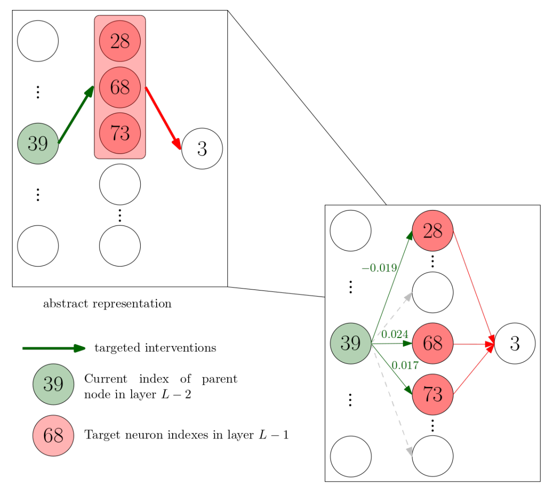

Figure 4. **Example of path selection for the last hidden layer of LeNet model.** In the sub-graphs $\mathcal{G}^{L-2\rightarrow L-1\rightarrow L}$ (right), the red nodes (layer L-1) are identified as the causal variables positively affecting the target label 3 (layer L). These causal nodes become then targets of their parent green node (layer L-2). The green edges connected to the red nodes are the selected intervention paths. By grouping all red nodes (left) as a target node, interventions are applied, simultaneously, on all green paths connected to them.

noisy and harmful nodes when interventions lead to significantly positive effect. We select the critical (red) nodes as target and intervene on the paths from the parent nodes in $L-2$ connected to the target. This time, interventions are simultaneously done on all paths directed from each parent node (green node in Fig. 4) to the red ones. Following this process, we can efficiently capture critical nodes in all intermediate layers while focusing on meaningful interventions. Algorithm 1 shows the implementation details of our method. We provide visualizations of LeNet's causal graphs in the supplementary.

Algorithm 1 Causal Explanatory Graph Inference ($\mathcal{G}$) of a DNN

Input: $N(L)$ pre-trained DNN, $\boldsymbol{W}$ weights, $\boldsymbol{X}$ task-specific examples, $\hat{\boldsymbol{y}}$ model outputs, (k) task index
Output: $\mathcal{G}$ (Dict. of critical nodes and their connections), $\mathcal{D}$ (Dict. of harmful nodes)

$l \leftarrow L-1, \beta \leftarrow \{0,1\}$
while $l > 0$ **do**
  $n_p \leftarrow dim(l)$
  for $j = 1$ to n_p **do**
    $u \leftarrow \beta w_j^{l\to l+1}$
    $do(w_j^{l\to l+1} = u)$
    Compute $TE(do(w_j^{l\to l+1}), \hat{\boldsymbol{y}}, \boldsymbol{X}, \boldsymbol{W})$ for all $\boldsymbol{X}$
    Solve (1) and get nodes (J^l, I^l)
    $\mathcal{G}^{l\to l+1} \leftarrow J^l, \mathcal{D}^{l\to l+1} \leftarrow I^l$
  end for
  $l \leftarrow l-1$
end while

4. Explanations from Causal Graphs

The hierarchical structure of the causal graphs enables robust extraction of attributions and high-level semantics. Instead of capturing a single saliency map from all activations, we rely on features responses along the causal pathways. We empirically show that these features are more stable and consistent compared to traditional attribution methods. As reported in [13], the reason for these methods to produce noisy and unstable attributions is due to noisy features in DNNs. Our method can remove the features that deteriorate model's prediction, and isolate relevant neurons in causal graphs/sub-graphs. Formally, given the sub-graph $\mathcal{G}^{l\to l+1}$, we extract salient features (s_i^{l+1}) at a node i in $l+1$ as follows

$$s_i^{l+1} = \frac{1}{J^l}\sum_{j=1}^{J^l} f(w_{ji}^{l\to l+1}, a_j^l), \tag{3}$$

where a_j^l is the activated signal in layer l at node j, J^l is the number of parent nodes in l connected to the child node i in the layer $l+1$. The response f depends on the structure of the parent layer. For convolution layers, w_{ji} is a filter and f is a convolution function; whereas for MLPs, f is linear function. Fig. 5 visualizes causal sub-graphs, up to *conv2* layer and their causal attributions for a LeNet model successfully classified its input.

Note that eq. (3) aggregates at every node i the responses of its parent nodes to the causal filters/weights. We may also be interested in analyzing the role of each filter between the pairs (i, j). Fig. 6 shows an example of the response to the top-1 filters (respective to their causal effects) for a set of relevant nodes in the last convolution layer of ResNet18.

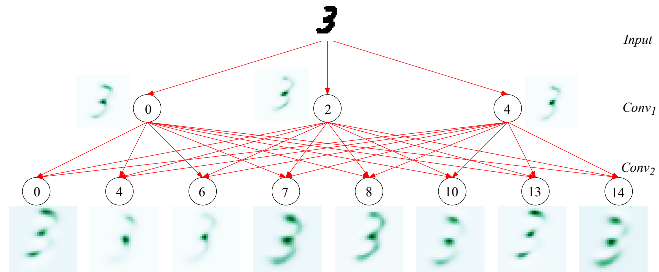

Figure 5. **Illustration of LeNet's causal sub-graphs $\mathcal{G}^{input\to conv_1}, \mathcal{G}^{conv_1\to conv_2}$ for class 3.** The resulted attributes provide visual interpretations for a sample image correctly classified by the model. They are up-sampled and normalized to reflect pixel-wise probabilities (Dark greens correspond to peaks with highest scores.).

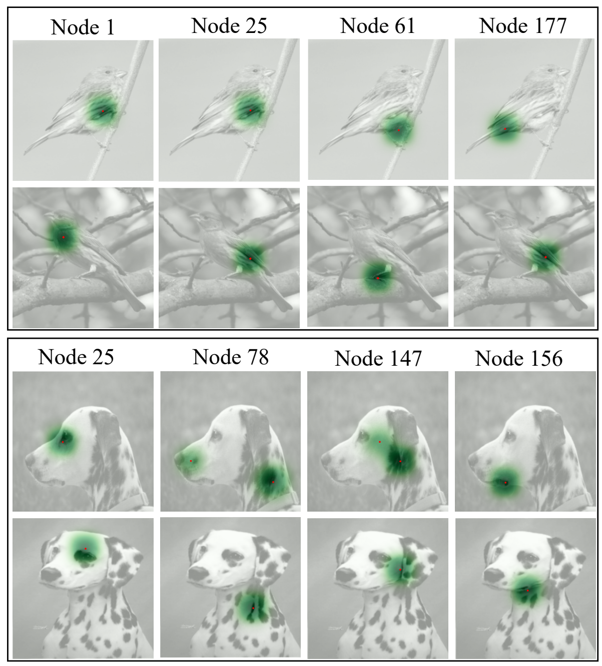

Figure 6. **Visualizing explanations obtained by the top-1 causal filters.** We show four examples for two object classes (from ImageNet). Important neurons belong to the causal sub-graph connecting the last Conv layers $l = layer4.1.conv1$ and $l+1 = layer4.1.conv2$ of ResNet18. The red point indicates the location of the peaks corresponding to the absolute maximum response.

Causal attributes (of object parts) are refined by extracting the response's local maxima (and minima).

5. Experiments

The experiments section splits into two parts: 1) we evaluate our algorithm's capacity to estimate stable and consistent causal graphs; 2) we evaluate the explanations captured by causal graphs and compare them to various attribution

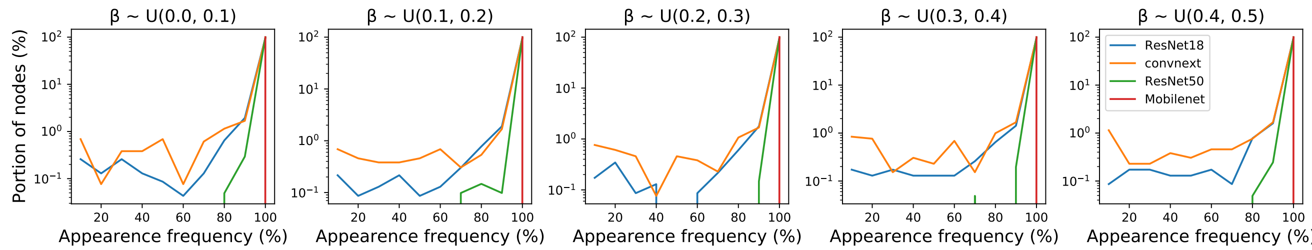

Figure 7. **Reliability assessment of causal graphs.** We show results on four complex architectures. On the x-axis, we show the frequency of appearance of a node (%). On the y-axis is the portion of all the nodes appearing at least once during the experiments (%).

methods using standard explanation metrics.

Models and datasets We evaluate our method on the LeNet model trained on MNIST data and the following architectures: ResNet18 [10], ResNet50V2 [11], MobileNetV2 [23], and on the latest architecture ConvNext [17]; the tiny version. These models were trained on the large-scale ImageNet data (ILSVRC-2012) [22]. We also fine-tuned these architectures on CIFAR10 dataset after updating their last classification layer. We divide the validation sets into validation and test sets. We use the samples in validation sets to discover causal explanatory graphs and the test set for evaluating the explanations.

Baseline methods We select the most popular attribution methods from two categories: model-agnostic (black-box) and gradient-based (white-box) methods. We chose RISE [19] and Occlusion [31] as black-box methods, and the following gradient-based methods: Integrated-Gradient (IG) [29], Saliency [27], Gradient Shape [2], GradXInput [25], DeconvNet [32] and Excitation Backprob (MWP) [33].

5.1. Evaluating the Reliability of Causal Graphs

In this experiment, we evaluate the stability and consistency of our estimation of causal graphs. Since the causal effect is based on path interventions, we need to ensure consistency in the statistical test results no matter what intervention values are used (i.e., binary or real). We do so by running 1000 experiments with an intervention parameter randomly sampled from a uniform distribution $\mathcal{U}(b-0.01, b+0.01)$. Here, b changes monotonically every 10 runs in the range $(0.01, 0.5)$. We report reliability by measuring the frequency of detecting the same causal nodes in each layer (in percentage). Fig. 7 shows for a few samples of β the distribution of the nodes versus their appearance rate. As we can see, the stability of the graph does not rely on the value chosen for the intervention parameter. Regardless of the value of β, a considerable proportion (98% to 100%) of the nodes appear in every experiment. The stability of causal graphs provides two facts: (1) the signals retained in these graphs are trustworthy; (2) our method is not sensitive to the choice of interventions (binary or real). The causal effect is significant even when reducing the strength of the signal along the causal path by only a factor of $1/2$. Our path intervention technique changes the interactions between internal neurons leading to discover the ones truly representative of model's behavior.

5.2. Evaluation of causal explanations

The causal graphs estimated by our method summarize knowledge from all hidden layers in the DNN enabiling better interpretability (see Fig. 5 as example). To compare the explanations obtained by our method with existing attribution methods, we select specific layer, here the last convolution layer in the case of CNNs as commonly used, and aggregate attributions of the critical nodes. We evaluate the quality of explanations using standard state-of-the-art metrics[1]. A good explanation method should satisfy several key performance indicators regarding its application purpose. In this work, we focus on stability and faithfulness metrics as indicators of goodness to help ML developers and experts choose and use the better method for debugging and improving the performance of their models. Details on explanation metrics and attributions visualization are provided in the supplementary.

Stability: Stability measures consistency of explanations against local perturbations of inputs. Here, we adopt Lipschitz Estimate (LE) [1], which calculates the maximum variance between an input and its ϵ-neighbourhood, where ϵ refers to the level of perturbations. We generate perturbations by adding white noise to inputs from the test sets. We compute explanations for every input in specific class and its noisy sample using the graphs estimated from the validation data. The maximum euclidean distance between explanations is then obtained over multiple runs where new perturbations are generated. Fig. 8 reports the results for LeNet trained on MNIST and ResNet18 fine-tuned on CI-

[1]The evaluations are implemented using the metrics defined in Quantus library [12]

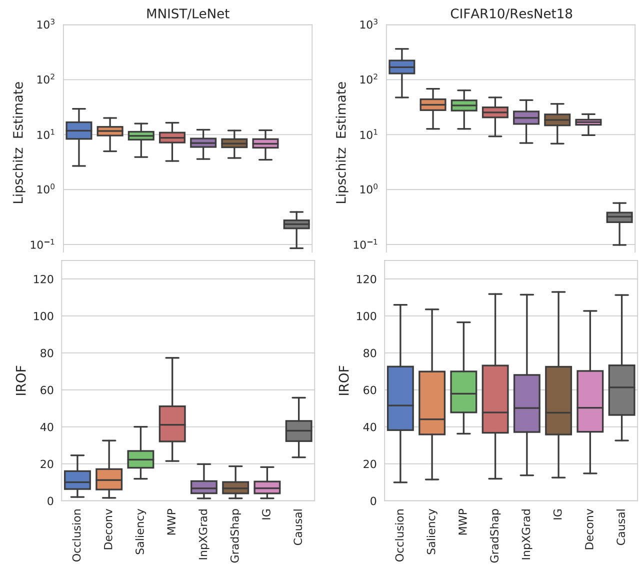

Figure 8. **Quantitative evaluations of attribution methods for LeNet on MNIST and ResNet18 on CIFAR10.** For each metric, we compare 7 attribution methods to the causal explanations obtained by our method using test images. The bars show mean and variance over samples. Lower Lipschitz Estimates (w.r.t. means) indicate higher stability. Higher IROF values (w.r.t. means) indicate strong relation between explanations and predictions.

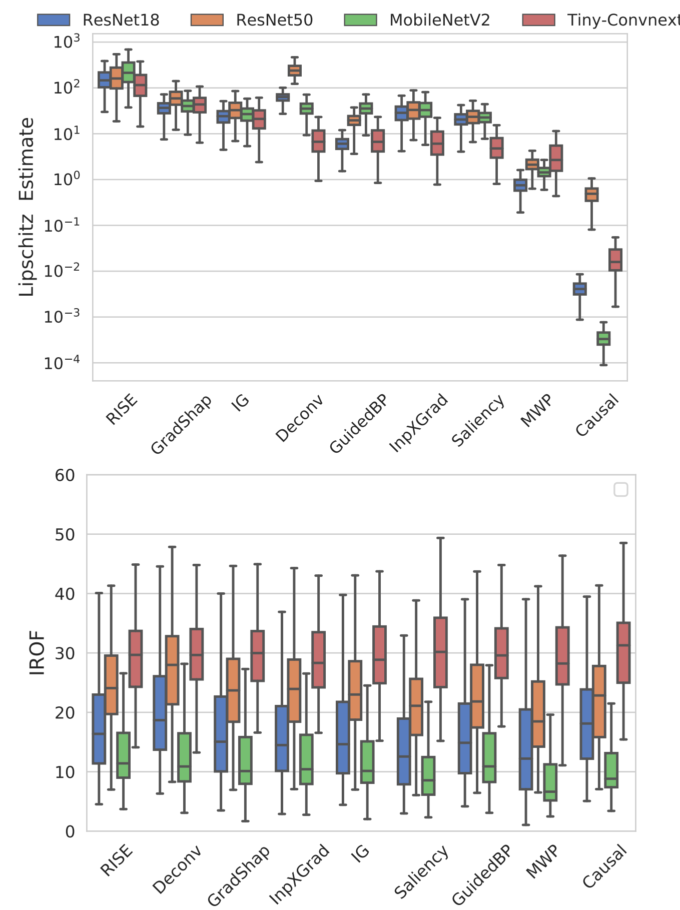

Figure 9. **Quantitative evaluations of explanations for complex architectures on ImageNet.** We evaluate 9 methods including ours using 10 representative classes from the test set.

FAR10, and Fig. 9 shows the results for four different architectures trained on ImageNet data.

The results (in Fig. 8 and 9) clearly indicate that the explanations generated from the causal graphs are more stable and consistent than other attribution methods. The explanations generated by these methods show higher variance to perturbations depending on the dataset and model. In contrast, the explanations from the causal graph show consistent stability. Our method has the lowest variance with a significant margin compared to the best method in each experiment.

Faithfulness: Evaluating attributions' relevance for the decision obtained by the model is essential to ensure the correctness and fidelity of explanations. This is commonly done by measuring the effect of obscuring or removing features from the input on the model's prediction. Different techniques have been proposed to score the relevance of explanations [1, 3–5, 21]. Here, we use iterative removal of features (IROF) [21]. An image is partitioned into patches using superpixel segmentation. The patches are sorted by their mean importance respective to the attributions in each patch. At every iteration, an increasing number of patches with the highest relevance are replaced by their mean value. The IROC computes the mean area above the curve for the class probabilities (perturbed vs. original predictions). We applied this metric to evaluate each explanation method, including ours. Fig. 8 shows that for LeNet and Resent18 trained on datasets like MNIST and CIFAR10, respectively, our method is better than the baselines and performs on par with the MWP. For the four complex architectures trained on ImageNet (Fig. 9), all methods, including ours, are comparable and agree on the observation that the ConvNeXt model is more trustable than standard ConvNets. Nonetheless, our method is the only one satisfying the highest stability and faithfulness among other explanation methods.

5.3. Fidelity of class-specific causal neurons

The causal neurons discovered as critical (or relevant) through interventions should accurately describe model behavior. We evaluate this by measuring the model accuracy on a specific class when masking out the critical neurons connected to this class. That means high fidelity neurons should cause a drastic drop in accuracy under discarding

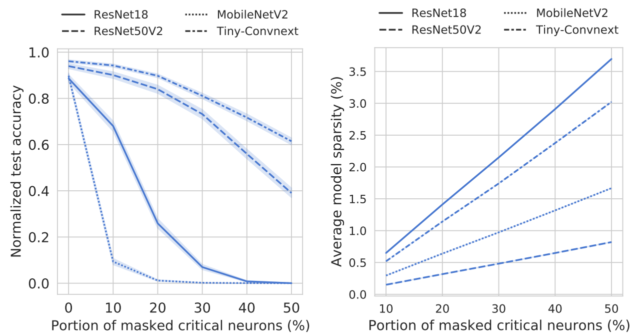

Figure 10. **Fidelity of class-specific causal neurons to the model.** The left figure shows the test accuracy of four models when masking out the top-k (%) portion of causal neurons discovered as critical using our path interventions method. The figure shows the average accuracy over ten representative classes selected from ImageNet. The right figure shows the average sparsity of a model (%), in terms of number of neurons, at each portion.

them. We illustrate this behavior on four models trained on ImageNet in Fig. 10. First, after discovering class-specific causal graphs, we rank the weights (and filters) in each sub-graph according to their highest effects (as described in eq. (2)). Then, we use these ranks to select the top-k critical neurons in each layer. As we observe in Fig. 10, the accuracy of all four models drastically drops after masking a small portion ($< 20\%$) of top critical neurons, and it is more evident on smaller architectures such as ResNet18 and MobileNetV2. Interestingly, masking 50% of critical neurons produces a maximum average sparsity of 3.8% (Fig. 10), which is very low, indicating the importance of their features on prediction. This experiment can also be considered as another way of evaluating faithfulness.

6. Applications

Repairing model accuracy In many practical, real-world cases, we seek fast and effective ways to repair the model's behavior without requiring extensive retraining with large datasets. This is even more crucial when collecting additional data is expensive and time-consuming. We can target the proposed explanation method to achieve this goal by considering the interactions responsible for a drop in the model's performance or a wrong prediction. This can be done by amortizing the activation signals outgoing from specific neurons identified as noisy and harmful (sec. 3.3). We either turn-off these neurons by setting them to zero, or cutting their connections to the critical neurons in the subsequent layer. It is worth noting that this operation differs from model pruning which necessitates further optimization and fine-tuning. We experimentally show that our method improves class prediction and correct the wrong one. To illustrate that, we took the four models trained on ImageNet and considered 10 representative (animal) classes for evaluation in addition to the LeNet trained on MNIST data. For each trained model, we mask out a portion of the harmful neurons and evaluate model's accuracy on the test samples. Fig. 11 shows the test accuracy under a varied portion of the masked neurons in all layers.

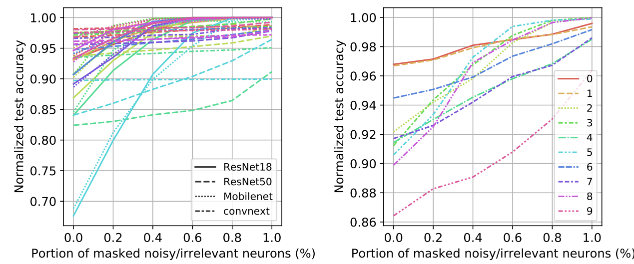

Figure 11. **Repair of model performance.** We show the test accuracy after masking out ($n\%$) of class-specific noisy filters in LeNet model for all categories of MNIST data (right), and in the 4 models for 10 representative (animal) categories of ImageNet data (left). Each color in left figure points to one different category and is fixed for each model.

7. Conclusion and discussions

We present a novel method for interpreting neural network behavior based on causal inference. It discovers the causal explanatory graphs that disentangle relevant knowledge hidden in the internal structure of DNN models, which is congenital to their predictions. Our methodology relies on the hypothesis that path interventions can capture meaningful interactions between hidden neurons, significantly affecting the model's output and leading to obtaining robust and faithful attributions. The causal attributions achieve better stability than the baseline methods. Our work targets expert users to help debug and improve their models, providing a valuable model monitoring and repair tool. It makes it worthwhile and practical for real-world applications where extensive training data are not accessible or retraining is computationally expensive. It is not designed to extract high-level causal abstractions that are interpretable to non-expert users, which might be seen as a limitation. We hope to extend our work to other applications, including human-centered explanations. Finally, while our experiments demonstrate the method's effectiveness on CNN-based classification models, we wish to expand the method to different types of architectures, such as transformers. These architectures are becoming widely popular in vision and language models. Interpreting their inner working is still an open challenge and exciting research direction.

References

[1] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 2, 6, 7

[2] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 1, 2, 6

[3] Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques. *arXiv e-prints*, 2019. 7

[4] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10, 2015. 7

[5] Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and aggregating feature-based model explanations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3016–3022, Jan. 2021. 7

[6] Miles Brundage, Shahar Avin, Jasmine Wang, Haydn Belfield, Gretchen Krueger, Gillian Hadfield, Heidy Khlaaf, and Others . Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims. *arXiv e-prints*, 2020. 1

[7] Aditya Chattopadhyay, Piyushi Manupriya, Anirban Sarkar, and Vineeth N Balasubramanian. Neural network attributions: A causal perspective. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 981–990, 2019. 2

[8] Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8730–8738, 2018. 1

[9] Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal Abstractions of Neural Networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586. Curran Associates, Inc., 2021. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 630–645, Cham, 2016. Springer International Publishing. 6

[12] Anna Hedström, Leander Weber, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M.-C. Höhne. Quantus: An Explainable AI Toolkit for Responsible Evaluation of Neural Network Explanations. *arXiv:2202.06861 [cs]*, Feb. 2022. arXiv: 2202.06861. 6

[13] Beomsu Kim, Junghoon Seo, Seunghyeon Jeon, Jamyoung Koo, Jeongyeol Choe, and Taegyun Jeon. Why are saliency maps noisy? cause of and solution to noisy saliency maps. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4149–4157, 2019. 1, 2, 5

[14] Matthew L. Leavitt and Ari Morcos. Towards falsifiable interpretability research. *arXiv e-prints*, Oct. 2020. 1

[15] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, R. Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. 3

[16] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California, June 2016. Association for Computational Linguistics. 1

[17] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022. 6

[18] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature Visualization. *Distill*, 2(11):10.23915/distill.00007, Nov. 2017. 1

[19] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018. 2, 6

[20] Marco Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101. Association for Computational Linguistics, June 2016. 2

[21] Laura Rieger and Lars Kai Hansen. IROF: a low resource evaluation metric for explanation methods. *CoRR*, abs/2003.08747, 2020. 7

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 6

[23] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted

residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 6

[24] Patrick Schwab and Walter Karlen. CXPlain: causal explanations for model interpretation under uncertainty. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 917, pages 10220–10230. Curran Associates Inc., Red Hook, NY, USA, Dec. 2019. 2

[25] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pages 3145–3153. JMLR.org, 2017. 1, 2, 6

[26] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pages 3145–3153. JMLR.org, 2017. 2

[27] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. 1, 6

[28] Jost Tobias Springenberg, A. Dosovitskiy, T. Brox, and Martin A. Riedmiller. Striving for Simplicity: The All Convolutional Net. *ICLR*, 2015. 2

[29] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pages 3319–3328. JMLR.org, 2017. 1, 2, 6

[30] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating Gender Bias in Language Models Using Causal Mediation Analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc., 2020. 1, 2

[31] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. 1, 2, 6

[32] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833. Springer International Publishing, 2014. 2, 6

[33] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-Down Neural Attention by Excitation Backprop. *International Journal of Computer Vision*, 126(10):1084–1102, Oct. 2018. 1, 6

[34] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting CNN knowledge via an explanatory graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 4454–4463. AAAI Press, Feb. 2018. 2

[35] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021. 2