Noname manuscript No.
(will be inserted by the editor)

Interpretable Deep Learning: Interpretation, Interpretability, Trustworthiness, and Beyond

Xuhong Li[‡], Haoyi Xiong[‡], Xingjian Li[‡], Xuanyu Wu[§], Xiao Zhang[¶], Ji Liu[‡], Jiang Bian[‡], Dejing Dou[⋆‡‖]

Received: date / Accepted: date

Abstract Deep neural networks have been well-known for their superb performance in handling various machine learning and artificial intelligence tasks. However, due to their over-parameterized black-box nature, it is often difficult to understand the prediction results of deep models. In recent years, many interpretation tools have been proposed to explain or reveal the ways that deep models make decisions. In this paper, we review this line of research and try to make a comprehensive survey. Specifically, we introduce and clarify two basic concepts—interpretations and interpretability—that people usually get confused. First of all, to address the research efforts in interpretations, we elaborate the design of several recent interpretation algorithms, from different perspectives, through proposing a new taxonomy. Then, to understand the results of interpretation, we also survey the performance metrics for evaluating interpretation algorithms. Further, we summarize the existing work in evaluating models' interpretability using *"trustworthy"* interpretation algorithms. Finally, we review and discuss the connections between deep models' interpretations and other factors, such as adversarial robustness and data augmentations, and we introduce several open-source libraries for interpretation algorithms and evaluation approaches.

1 Introduction

Deep learning models [66] have achieved remarkable performance in a variety of tasks, from visual recognition, natural language processing, reinforcement learning to recommendation systems, where deep models have produced results comparable to and in some cases superior to human experts. Due to their nature of over-parameterization (involving more than millions of parameters and stacked with more than hundreds of layers), it is often difficult to understand the prediction

⋆ Corresponding author.
‡ Baidu Research, Baidu Inc., Beijing, China
§ School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, USA
¶ Department of Electronics and Information Engineering, Tsinghua University, Beijing, China
‖ Computer and Information Science Department, University of Oregon, USA.

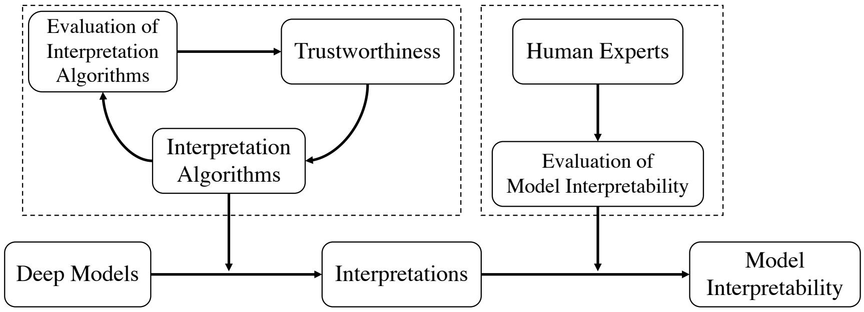


Fig. 1: Scheme about interpretation algorithms and model interpretability.

results of deep models [32]. Explaining[1] their behaviors remains challenging due to their hierarchical non-linear nature in a black-box fashion. The lack of interpretability raises a severe issue about the trust of deep models, in high-stakes prediction applications, such as autonomous driving, healthcare, criminal justice, and financial services [21]. While many interpretation tools have been proposed to explain or reveal the ways that deep models make decisions, nonetheless, either from a scientific view or a social aspect, explaining the behaviors of deep models is still in progress. In this paper, instead of focusing on the social impacts, regulations and laws related to deep model interpretations, we would like to focus on the research field, by clarifying the research objectives and reviewing the methods proposed in this field.

Interpretation vs. Interpretability - In this work, we first clarify two concepts that sometimes get confused by researchers: *interpretation algorithms* and *model interpretability*. Interpretation algorithms are the methods proposed to explain or reveal the ways that deep models make decisions, such as the combination of features used for model decisions [96], or the importance of every training sample as the contribution for inference [59]. On the other hand, the model interpretability refers to the intrinsic properties of a deep model measuring *in which degree the inference result of the model is predictable or understandable to human beings*. In practice, one could trust the interpretation algorithms, and one could further evaluate the model interpretability through matching the interpretations, i.e., the results from interpretation algorithms for a deep model, and the human labeled results [16]. In this way, the comparison of interpretability becomes possible among different models.

In this paper, we survey existing interpretation algorithms from literature texts, including [21, 32, 54, 70, 134], and we propose the definition of **trustworthy** interpretation algorithms. The "trustworthiness" here refers to the degree that people could rely on the interpretation results delivered by the algorithm on arbitrary deep models. Then, incorporating a trustworthy interpretation algorithm, the model interpretability can be assessed through matching the interpretation results and the human labeled interpretations on a set of samples. In Fig. 1, we summarize the

[1] The subtle differences among *interpretation*, *explanation*, and *attribution* are not considered in this paper, and we use them interchangeably.

connections between these key concepts and we further elaborate these concepts in Section 2.

Interpretation Algorithms - As there might exist multiple perspectives to interpret a deep model, the interpretation algorithms are usually designed with different principles, as follows

– Highlighting the important parts of input features on which the the deep model mainly relies, with gradients [109], perturbations [38], or proxy explainable models [96];
– Investigating the inside of deep models to understand the rationale of how models make decisions [139, 143];
– Estimating the contributions of each training data for interpreting the training process [59, 117]; and so on.

In this paper, we review the recent interpretation algorithms and propose a novel taxonomy for categorizing the interpretation algorithms. Specifically, there are three orthogonal dimensions in the proposed taxonomy – (1) *targeting models for interpretation*, e.g., differentiable or non-differentiable models; (2) *representations of interpretations*, e.g., feature importance or dataset sample influences; and (3) *formulation of interpretation algorithms*, e.g., closed-form or proxy-based approaches. Every existing interpretation algorithm could be appropriately categorized according to the proposed dimensions. In Section 3, we present the taxonomy of interpretation algorithms and their designs with respect to the aforementioned dimensions.

Interpretability Evaluation with Trustworthy Interpretations - Given a set of interpretation algorithms, we could evaluate the trustworthiness of these algorithms with proper evaluation approaches and pick-up the trustworthy ones. On the other hand, human experts can also generate "ground-truth" labels of interpretation results through interpreting the human decision-making procedures. Thus, given a trustworthy interpretation algorithm with human labeled ground truth, one can evaluate and compare the interpretability of models. However, two technical challenges remain in this area as follows.

– Evaluating the trustworthiness of interpretation algorithms is not easy, where well-known metrics, such as accuracy, precision, and recall for classification tasks are not applicable here;
– Furthermore, obtaining-human labeled ground truth for interpretation is labor/-time-consuming, which is not quite scalable over large datasets.

In this way, several efficient and effective approaches to evaluate trustworthiness of interpretation algorithms [7, 101, 125, 134] and model interpretability [16, 68] have been proposed. In Section 4, we comprehensively review the evaluation approaches on the trustworthiness of interpretation algorithms and model interpretability respectively.

Connections between Interpretations and Other Factors - Recent studies on adversarial examples have found interesting connections between the interpretations and adversarial robustness [98, 119]. Furthermore, data augmentation [57] and regularization [63, 136] used in deep learning procedure can also significantly affect the model interpretability and interpretation results. Finally, people also pay attention to using the interpretation algorithms and model interpretability evaluation for model debugging [4], diagnosing [21], and selection [16, 68] purposes. In Section 5, we introduce the connections between interpretations and these factors.

2 Main Concepts: Interpretations and Interpretability

Several fuzzy notions, such as *interpretation* and *interpretability*, lead to a lot of confusions and hinder research. In this section, we make our efforts to clarify these fuzzy research targets and introduce the definitions of *interpretation* and *interpretability*, as well as the *interpretation algorithms.*

2.1 Interpretation Algorithms

We first introduce interpretation algorithms. A model needs interpretation because the output cannot be understood by humans and it is hard to see the reasoning or rationale from its output. To explain the rationale of how models make decisions, an interpretation algorithm is usually required. Various interpretation algorithms give various interpretations. Before arbitrarily concluding that the model is interpretable, we should guarantee at the first step that the interpretation given by the algorithm is trustworthy [21, 70, 82], as follows.

– *An interpretation algorithm is trustworthy if it properly reveals the underlying rationale of a model making decisions.*

In this definition, the *underlying rationale* is how the model makes decisions, or the reasoning behind the model making decisions. An interpretation algorithm is probably a module outside of the model and at risk of giving results that do not depend on the model at all. The word *properly* here targets the issue that the intrinsic underlying rationale behind the model is usually given by an extrinsic algorithm. Therefore, the intrinsic rationale should be **properly** recovered by a trustworthy interpretation algorithm. Or informally saying, the algorithm is trustworthy if the model follows the revealed rationale to make decisions, whether the decision is correct or wrong. That may be a primary requirement for interpretation algorithms, but unfortunately it is not easy to be fulfilled, e.g. see [3], and the evaluation of trustworthiness does not have a formal measurement.

Self-interpretable Models - We also note that many researchers are working on effective self-interpretable models [48, 100]. These models with self-interpretation algorithms are not outliers of our discussion; the only difference is that for self-interpretable models, researchers have simultaneously devised a model and an interpretation algorithm. Efficiently, the "devised" interpretation algorithm is without doubt trustworthy because it is an intrinsic property of the model instead of an extrinsic investigator. For example, feature importance of tree models can be calculated according to the amount that each feature contributes to the split at nodes of trees.

Fully-interpretable Models - A model is fully interpretable if there exists a trustworthy interpretation algorithm s.t. (1) the rationale behind the model is fully revealed by the algorithm; and (2) the revealed rationale is totally understandable by humans, or fully overlaps with human understandings. Fully interpretable models are usually simple and incapable of learning complex features from data, so it is usually difficult for interpretable models to cope with large-scale datasets and real-world applications. However, because of the simplicity, the interpretation algorithms for fully interpretable models are easy to find and guaranteed to be trustworthy because of their intrinsic property. Many researchers believe that it is a trade-off between interpretability and performance, and thus it is challenging to devise a fully interpretable model with qualified performance. While we expect the presence of fully interpretable models, the exploration for them remains an important direction in this research field.

Discussion on Trustworthiness - The revealed rationale of how models make decisions is not unique. Various trustworthy algorithms may exist but the revealed underlying rationales expose different levels of information. For example, a frequent requirement for interpretations is the analysis of relations between input and output, as done by LIME [96] or by perturbation algorithms [37, 38], since the feature importance is widely needed. Though the rationale behind the model cannot be exposed by input-output analysis, in some scenarios, the inside rationale is always not mandatory. In fact, real-world applications propose different requirements of interpretations. Some are satisfied by input-output analyses, some may need more inside investigations, while some need fully interpretable models. We assume the existence of fully interpretable models in this paper while there are still discussions about whether a specific model is fully interpretable[2].

2.2 Model Interpretability

For fully interpretable models, they are all equally of highest interpretability; i.e., all have the full ability of presenting understandable terms to humans, so there is no need to compare the interpretability among them. Beyond the fully interpretable models, a problem arises: The revealed rationale may at most partially overlap with human understandings, but given a trustworthy algorithm, some models show terms that are more understandable to humans or largely overlap with human understandings while other models do less [41, 70]. So the natural question follows: How can we select models that are more interpretable over others in a practical scenario? This leads to the definition of model interpretability and we reclaim the definition of model interpretability by [32] as follows.

– *The model interpretability is the ability (of the model) to explain or to present in understandable terms to a human.*

A comment about the expression *understandable to a human*: This is a subjective notion so the definition is obviously human-centered [32, 62]. It somehow explains why it is difficult to give definitions in this research field: Humans are composed

[2] Even rule-based models or decision trees are not always accepted as fully interpretable models in some context [70, 99]. We leave this open question beyond this paper.

of different individuals. It causes the research problem of quantitatively measuring and comparing the interpretability of various models. For fully interpretable models, their quantities of interpretability, if can be measured, are equally the highest. So it is more meaningful to discuss the interpretability among models that are not fully interpretable. Unfortunately, the evaluation of model interpretability does not have a formal measurement, and we will review the current evaluation approaches on model interpretability in Section 4.2.

Beyond fully interpretable models, given one trustworthy interpretation algorithm, e.g., an algorithm of analyzing the input-output mappings, the rationales of the models can be revealed as the feature importance of input features (or various data types). Take image classification [29, 128] as an example. The interpretation will be the important parts of images. However, different models may locate different parts of images. If the interpretation algorithm is trustworthy, then we can conclude that different models show different interpretability: We can understand the model that "sees" the object parts in the image for making the classification decision, but it is harder to understand if the model "sees" the accompanied background in the image for recognizing the object. Although the rationales of both models are revealed by the trustworthy algorithm, we prefer the former model because its way of making decisions is more aligned with human understandings.

2.3 Remarks

In this section, we defined trustworthy interpretation algorithms that properly reveal the rationale of how the model makes decisions. Specifically, given a trustworthy interpretation algorithm, the revealed rationales of various models are different in the degree of being understandable to humans or aligning with human understandings, therefore showing different model interpretability.

With these clearer definitions, we emphasize several points that usually cause confusion in the field.

- The notions of interpretation algorithms, interpretations and interpretability should be clearly distinguished. The requirement for *interpretation algorithms* is to be trustworthy with respect to the model, as defined in the beginning of this section; *interpretations* are the rationale of models revealed by interpretation algorithms, and *interpretability* is the degree of the interpretations being understandable to humans.
- There are many interpretation algorithms and we will review and categorize the typical ones according to the taxonomy proposed in Section 3; but unfortunately, the trustworthiness of interpretation algorithms is hard to evaluate. We will review the approaches of evaluating interpretation algorithms in Section 4.1.
- Given one trustworthy interpretation algorithm, two models yield two different interpretations, both revealing the model rationales by the trustworthy algorithm. However, one may overlap more with human understandings while another does not. We prefer the model whose rationale is more aligned with human understandings and conclude, with rough measurement, that it has the higher interpretability than another. We review approaches of systematically comparing and evaluating the interpretability of models in Section 4.2.

- If the interpretability is human-centered, then it is always a relative metric, with human understanding as reference. However, the interpretations sometimes lead to useful or promising findings. The dataset presents biases and can be improved, e.g. [115, 117], or complex models may have learned something that is not semantic for humans, e.g. [94, 106]. Interpretations are needed here for a completely different objective: finding new intelligent patterns that are not yet understandable in the present.
- In this section, the proposed desiderata related to interpretations is the trustworthiness for interpretation algorithms. Researchers [21, 32, 54, 70, 134] also proposed many desiderata for interpretations and interpretation algorithms, such as fairness, privacy, reliability, robustness, causality, trust, fidelity, transferability, informativeness, transparency, plausibility, satisfaction, etc. However, we note that (1) some properties (e.g. informativeness, plausibility, satisfaction) refer to whether the interpretation is understandable to humans, and are different from the trustworthiness in this paper that refers to algorithms; (2) some properties (e.g. reliability, robustness, trust, fidelity, transparency) are similar to trustworthiness or can be comprised by the general definition of trustworthiness; (3) some of them (e.g. causality, transparency) depend on the *underlying rationale* in our context; (4) a few of them (e.g. fairness, transferability, privacy) are the standards that use interpretations to verify models; and (5) others may be out of the scope of interpretation. There are some slight differences and specific requirements in various scenarios, but trustworthiness is for interpretation algorithms.

3 Interpretation Algorithms: Taxonomy, Algorithm Designs, and Miscellaneous

We review typical interpretation algorithms in this section, proposing a taxonomy according to three dimensions at first, following brief introductions of these algorithms and some special categories. We also provide a detailed categorization of all algorithms with discussions at the end of this section.

3.1 Taxonomy

We categorize the existing interpretation algorithms according to three orthogonal dimensions: targeting models for interpretations, representations of interpretations, and formulation of interpretations. We list the options in each dimension for a better comparison.

Not all interpretation algorithms are model-agnostic, they cope with different types of models:

- **Model-agnostic**. Algorithms that completely consider the models as black box and do not investigate the inside of models are included here.
- **Differentiable model**. This is a subset of the previous option and contains only algorithms that address the interpretations of differentiable models, especially neural networks.
- **Specific model**. This is narrower than the previous one. This option contains algorithms that can only be applied to certain types of models, e.g.

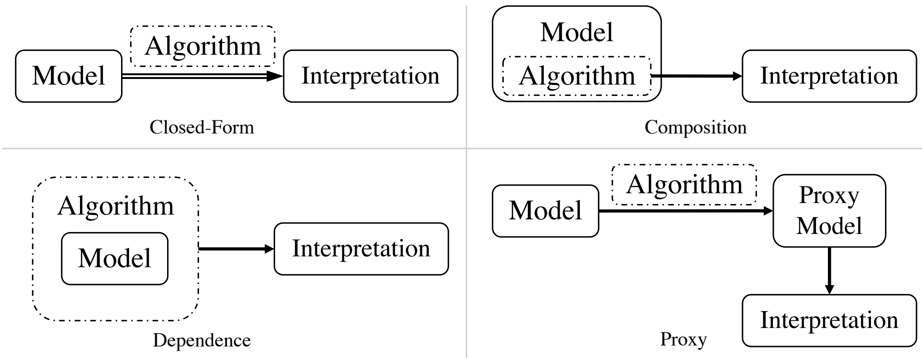


Fig. 2: Illustration of relations between the interpretation algorithm and the model. Four relations are illustrated: Closed-form, composition, dependence and proxy.

convolutional neural networks (CNNs), generative adversarial networks (GANs), Graph Neural Networks (GNNs).

For different applications and interpretation requirements, the representations of interpretation are various:

- **Feature (Importance)**. These algorithms aim at interpretations on input data e.g. images, texts, or extracted features; or intermediate features of models, e.g. the activations of neural networks; or latent features in generative adversarial networks (GANs).
- **Model Response**. Algorithms here generally propose to generate or find new examples and see the model's responses, so as to investigate the model behaviors on certain patterns or the rationale by which the model makes decisions.
- **Model Rationale Process**. There are algorithms that interpret the process of model inside rationale, i.e., how the model obtains final decisions.
- **Dataset**. Instead of direct interpretations on models, some algorithms propose to explain the examples in the training dataset that affect the training of models.

The third dimension for categorizing interpretation algorithms is the relation between the interpretation algorithm and the model:

- **Closed-form**. These algorithms derive a closed-form formula from the target model and output interpretable terms.
- **Composition**: Algorithms here can be considered as components of (interpretable) models, usually obtained during training.
- **Dependence**: These algorithms build new operations upon the target model after training, and output interpretable terms.
- **Proxy**. Different from dependence, algorithms here obtain, via learning or derivation, a proxy model for explaining the behavior of models.

The difference of these four relations can be illustrated by Fig. 2.

We do not explicitly categorize the interpretation algorithms according to their application domains because (1) the algorithm used in one specific domain may be also applicable in a wider scope with limited modifications; and (2) the categorization on the model type generally overlaps with the one on the application domains. However, for completeness, we will discuss recent works of deep model interpretations in the following domains, such as reinforcement learning, recommendation systems, and medical domains.

3.2 Typical Algorithms

LIME and Similar Algorithms LIME presents a locally faithful explanation by fitting a set of perturbed samples near the target sample using a potentially interpretable model, such as linear models and decision trees. We define a model $g \in G$, where G is a class of interpretable models. The domain of g is $\{0,1\}^{d'}$ and its complexity measure is $\Omega(g)$. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be the model being explained and $\pi_x(z)$ be the proximity measure between a perturbed sample z and x. Finally, let $L(f, g, \pi_x)$ be a measure of the unfaithfulness of g in approximating f in the locality defined by π_x. LIME produces explanations by the following:

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g). \tag{1}$$

The obtained explanation $\xi(x)$ interprets the target sample x, with linear weights when g is a linear model. LIME is model-agnostic, meaning that the obtained proxy model is suitable for any model. Similarly, several model-agnostic algorithms [20, 72, 87, 89, 97] target at interpreting features and provide feature importance or contributions to the final decision.

Perturbation To investigate important features in the input, a straightforward way is to measure the effect of perturbations applied to the input [37, 38]. This idea is quite simple: Giving random values to randomly chosen features and evaluating the prediction changes, so as to evaluate the contributions of the chosen features. Note that perturbation is also used for evaluating the trustworthiness of interpretation algorithms when we are aware of interpretation ground truth [101, 125].

Derivatives w.r.t. Input The input gradient attributes the important features in the input domain. However, for non-linear deep models, the input gradient is noisy. SmoothGrad [109] proposed to remove the noise of the gradient by adding noise on the input. We take visual tasks as an example: Given input image x, neural networks compute a class activation function S_c for class $c \in C$. A sensitivity map can be constructed by calculating the gradient of M_c with respect to input x: $M_c(x) = \partial S_c(x)/\partial x$. However, the sensitivity maps are often noisy because of sharp fluctuations of the derivative. To smooth the gradients, multiple Gaussian noise is added to the input image, and the sensitivity maps are averaged. SmoothGrad is defined as follows:

$$\hat{M}_c(x) = \frac{1}{n} \sum_{1}^{n} M_c(x + \mathcal{N}(0, \sigma^2)). \tag{2}$$

Integrated Gradient [114] aggregates the gradients along the inputs that lie on the straight line between the baseline and input. Let F be a neural network, x be the input and x' be the baseline input, which can be a black image for computer vision models and a vector of zeros for word embedding in text models. The integrated gradients along the i^{th} dimension is

$$IntegratedGrads_i(x) = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha. \tag{3}$$

An axiom called *completeness* is satisfied which states that the attributions add up to the difference between the output of F at input x and baseline x'.

More similar approaches are cited [105].

Global Interpretations Feature importance analysis is a common tool for explaining the model outputs with respect to inputs. In fact, LIME and saliency map approaches can be categorized into feature importance analysis. Note that their interpretations are for individual examples, giving unique result for each different example. Different from these "local" interpretations, "global" interpretations provide feature importance in an overall vision of the model. However, for deep models, this is interestingly based on local interpretations and an aggregation of local interpretations is performed to obtain the global feature importance, while the aggregation approaches are different [5, 96, 120].

CAM and Variants Given a CNN and an image classification task, classification activation map (CAM) [143] can be derived from the operations at last layers of the CNN model and show the important regions that affect model decisions. Specifically, for a given category c, we expect the unit corresponding to a pattern of the category in the receptive field be activated in the feature map. The weights in the classifier indicate the importance of each feature map in classifying category c. Therefore, a weighted sum of the presence of visual patterns illustrates the important regions of a category. Let $f_k(x, y)$ denote the activation of unit k in the last convolutional layer at spatial location (x, y), $F_k = \sum_{x,y} f_k(x, y)$ be the global average pooling for unit k, and w_k^c be the weight corresponding to class c for unit k so that $\sum_k w_k^c F_k$ is the input to softmax for class c. Then the activation map for class c is:

$$M_c(x, y) = \sum_k w_k^c f_k(x, y). \tag{4}$$

GradCAM [102] further looks at the gradients flowing into the convolutional layer to give weight to activation maps. Let y^c be the score for class c before the softmax, A^k be feature map activations of the unit k in a convolutional layer, the neuron importance weight α_k^c is the global-average-pooled gradient of y^c with respect to A^k:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{i,j}^k}. \tag{5}$$

The localization map is a weighted combination of forward activation maps:

$$L_{Grad-CAM}^c = ReLU(\sum_k \alpha_k^c A^k). \tag{6}$$

ScoreCAM [127] also uses gradient information but assigns importance to each activation map by the notion of *Increase of Confidence*. Given an image model $Y = f(X)$ that takes in image X and outputs logits Y. The k-th channel of convolutional layer l is denoted A_l^k. With baseline image X_b and category c, the contribution A_l^k towards Y is:

$$C(A_l^k) = f^c(X \circ H_l^k) - f^c(X_b), \tag{7}$$

where $H_l^k = s(Up(A_l^k))$. $Up(\cdot)$ is the operation that upsamples A_l^k into the input size and s normalizes each element into $[0, 1]$. ScoreCAM is defined as:

$$L^c_{Score-CAM} = ReLU(\sum_k \alpha_k^c A_l^k), \tag{8}$$

where $\alpha_k^c = C(A_l^k)$.

More works based on CAM can be found [24, 95, 110, 130, 142].

TCAV Given a set of examples representing a concept of human interest, TCAV [56] seeks a vector in the space of activations of l-th layer that represents this concept, by defining a concept activation vector (or CAV) as the normal to a hyperplane, separating examples without the concept and examples with the concept in the model's activations. Then, given one example in a certain class, along the direction of a CAV, the directional derivative of this example contributes a score if it is positive, and the ratio of examples that have positive directional derivatives over all examples in this class is defined as the TCAV score. CAV finds examples of a semantic concept, learned by the intermediate layers of a deep model, that contributes to the predictions while TCAV quantitatively measures the contributions of this concept.

LRP Layer-wise relevance propagation (LRP) [11] recursively computes a Relevance score for each neuron of layers, so as to understand the contribution of a single pixel of an image x to the prediction function $f(x)$ in an image classification task.

$$f(x) = \cdots = \sum_{d=1}^{V^{(l+1)}} R_d^{(l+1)} = \sum_{d=1}^{V^{(l)}} R_d^{(l)} = \cdots = \sum_{d=1}^{V^{(1)}} R_d^{(1)}, \tag{9}$$

where $R_d^{(l)}$ is the Relevance score of the d-th neuron at the l-th layer, $V^{(l)}$ indicates the dimension of l-th layer, and $V^{(1)}$ is the number of pixels in the input image. Iterating Eq. (9) from the last layer which is the classifier output $f(x)$ to the input layer x consisting of image pixels then yields the contribution of pixels to the prediction results. [19] proposed an extension of LRP based on first-order Taylor expansions for product-type nonlinearities. [25, 122] adapted LRP to interpret transformer models [30, 33, 113]. Works related to LRP are [44, 52, 78, 83].

Proxy Models for Rationale Process The underlying rationale of deep models is complex due to the non-linearity and enormous computations. However, this rationale process can be proxied by graph models [138] or decision trees [140], which provide relatively a more interpretable rationale path to humans. Moreover, deep neural networks can be combined with decision forest models [60] or distilled into a soft decision tree [39]. A model-agnostic approach for interpreting rationale process named BETA [64] allows learn (with optimality guarantees), a small number of compact decision sets each of which explains the behavior of the black box model in unambiguous, well-defined regions of feature space.

Interpretations Through Model Response Model responses to particular examples can somehow expose the reasons of making decisions. Many research works focus on this intuition. These particular examples include but not limited in counterfactual examples and prototypes.

Using counterfactual examples to explain the model behaviors can be theoretically included into causal inference [86], which is considered as a new perspective for model interpretability [80, 131]. Counterfactual explanations describe what changes to the situation would have resulted in arriving at the alternative decision, and can be naturally used to interpret deep model rationale process [23, 42, 65, 81]. Reviews on Counterfactual explanations can be found in [8, 121, 126].

Counterfactual examples interpret model behaviors by modifying important facts from original inputs. Similarly, algorithms of searching prototypes interpret model behaviors by searching or creating exemplar inputs that lead the model to make desired predictions. [26] proposed ProtoPNet which explains the deep model by finding prototypical parts of predicted objects and gathering evidence from the prototypes to make final decisions. Another method named ABELE [46] generates exemplar and counter-exemplar images, labeled with the class identical to, and different from, the class of the image to explain, with a saliency map, highlighting the importance of the areas of the image contributing to its classification. More works related to prototype for interpretations can be found in [15, 18, 67, 77]

As a technique for generating prototypes, activation maximization generally computes the prototypes through an optimization process:

$$\max_{\boldsymbol{x}} \ \log p(y_c|\boldsymbol{x}) - \lambda\|\boldsymbol{x}\|^2, \tag{10}$$

where $p(y_c|\boldsymbol{x})$ is the probability given by a deep model with $\boldsymbol{x}$ as input, and the second term is the constraint for generating the prototype. However, the constraint can be replaced by many other choices [34, 75, 84, 107]. A tutorial for this direction is cited [79].

Interpretation Modules from Training If the interpretable deep models are those whose intermediate layers are composed of semantic neurons, then regularizing internal neurons towards candidate semantics during the training process improves the interpretability. By simple abstraction, the objective function for this purpose can be written as

$$Loss = L(f(x), y) + \alpha R, \tag{11}$$

where $f(x)$ represents the deep model output with x as input, y is the ground truth, L is the loss function and specifically cross entropy for standard supervised classification problem, and R is the regularization added for biasing towards semantic neurons. Various approaches [31, 76, 98, 139] have been proposed to improve the interpretability during training. More encouragingly, [100] designed a self-interpretable deep model where each internal output presents semantic features.

Contributions of Training Examples Forgetting events are defined by [117] for analysing the training examples using training dynamics. Given a dataset $D = (x_i, y_i)_i$, after t steps of SGD, example x_i undergoes a forgetting event if it is misclassified at step $t+1$ after having been correctly classified at step t. Forgetting events signify samples' interactions with decision boundaries and the samples play a part equivalent to support vectors in the *support vector machine* paradigm. Unforgettable examples are samples that are learnt at step $t^* < \infty$ and never misclassified for all $k \geq t^*$. They are easily recognizable samples which contain obvious class attributes. Whereas examples with the most forgetting events are ambiguous without clear characteristics of certain class, and some are actually noisy samples.

Dataset Cartography [115] looks into two measures for each sample during the training process - the model's confidence in the true class and the variability of confidence across epochs. Training examples can be therefore categorized as easy-to-learn, hard-to-learn or ambiguous based on their position in the two-dimensional map. Consider training dataset $D = (x, y^*)_{i=1}^{N}$ where x_i is the i-th sample and y_i^* is the true label. After training for E epochs, the confidence is defined as the mean probability of true label across epochs:

$$\hat{\mu_i} = \frac{1}{E}\sum_{e=1}^{E} p_{\theta^{(e)}}(y_i^*|x_i), \tag{12}$$

where $p_{\theta^{(e)}}$ is the probability with parameters $\theta^{(e)}$ at the end of the e^{th} epoch. The variability is the standard deviation of $p_{\theta^{(e)}}(y_i^*|x_i)$:

$$\hat{\sigma_i} = \sqrt{\frac{\sum_{e=1}^{E}(p_{\theta^{(e)}}(y_i^*|x_i) - \hat{\mu_i})^2}{E}}, \tag{13}$$

Another method for analysing the training dynamics is proposed to compute the area under the margin (AUM) [88]:

$$\mathrm{AUM}(\boldsymbol{x}, y) = \frac{1}{T}\sum_{t=1}^{T}(z_y^{(t)}(\boldsymbol{x}) - \max_{i \neq y} z_i^{(t)}(\boldsymbol{x})), \tag{14}$$

where $z_i^{(t)}(\boldsymbol{x})$ is the logit, computed by the model, of i-th class at t-th epoch during training with respect to the example $\boldsymbol{x}$.

Influence functions [59] identify the training samples most responsible for a model prediction by upweighting a sample by some small value and analyze its effect on the parameters and the loss of the target sample. Given input space X and output space Y, we have training data $z_1, \ldots, z_n$, where $z_i = (x_i, y_i) \in X \times Y$. Let $L(z, \theta)$ be the loss where $\theta \in \Theta$ are the parameters. The optimal $\hat{\theta}$ is given by $\hat{\theta} = argmin_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^{n} L(z_i, \theta)$. The influence of upweighting training point z on the loss at the test point z_{test} is:

$$I_{up,loss}(z, z_{test}) = -\nabla_\theta L(z_{test}, \hat{\theta})^T H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta}), \tag{15}$$

where $H_{\hat{\theta}} = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 L(z_i, \hat{\theta})$. Based on influence functions, several techniques [28, 58] have been proposed with improvement.

Interpretable GNN Graph Neural Networks (GNNs) are a powerful tool for learning tasks on graph structured data. Like other deep learning models, GNNs show the black-box fashion and are required to explain their prediction results and rationale processes. Without requiring modification of the underlying GNN architecture, GNNExplainer [135] leverages the recursive neighborhood-aggregation scheme to identify important graph pathways as well as highlight relevant node feature information that is passed along edges of the pathways. More related work to GNN interpretations can be found in [13, 36, 51, 73, 91].

GANs: Semantically Meaningful Directions Generative adversarial networks (GANs) are a popular generative model based on two adversarial networks, where one generates new examples and another tries to classify generated examples from natural examples. Interpretations on GANs mainly search for semantically meaningful directions [17, 90, 123, 124, 132]. Comparing with labeled semantics, GAN dissection [17] finds semantic neurons in generative models and is capable of modifying the semantics in the generated images. Instead of relying on labels, [124], in an unsupervised way, found semantically meaningful directions in the intermediate layers of generative models. Similarly, [104] proposed a closed-form factorization method for identifying semantic neurons. Note that there are other methods for explaining the generative models [90, 123, 132].

Information Flow In some deep learning models there are multiplicative scalar weights that control information flow in some parts of a network. The most common examples are attention [12] and gating:

$$c^{att} = \sum_i \alpha_i^{att} h_i, \qquad c^{gate} = \alpha^{gate} h \tag{16}$$

The attention weights α^{att} ($\sum_i \alpha_i^{att} = 1$) and the gate values α^{gate} ($\alpha^{gate} \in [0, 1]$) are usually interpretable because their value represent the strength of the corresponding information pathways. Attention and gating are frequently used in NLP models, and there have been plenty of work aiming to understand the model through these weights [2, 40, 111, 112] and investigate the reliability of using them as explanations [55, 103, 129].

Self-Generated Explanations Using text generation techniques, a model can explicitly generate human-readable explanations for its own decision. A joint output-explanation model is trained to produce an prediction and simultaneously generate an explanation for the reason of that prediction [9, 61, 71]. This requires some kind of supervision available to train the explanation part of the model.

3.3 Miscellaneous Categories

More interpretation algorithms that target at reinforcement learning, recommendation system, and medical applications are briefly introduced below. These applications are slightly different from classification tasks and require various interpretations, but most algorithms introduced previously can be used directly. We mainly present surveys in these domains.

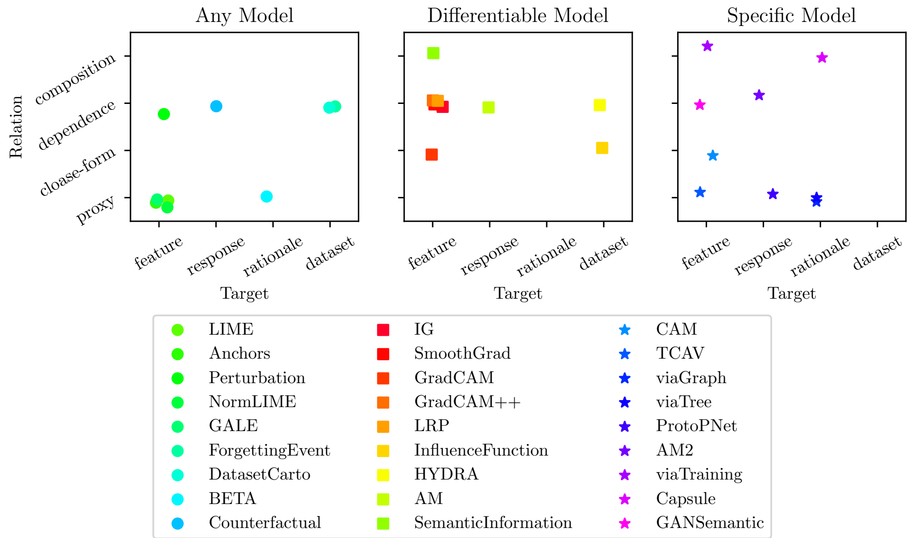


Fig. 3: Visualization of some typical interpretation algorithms according to the proposed taxonomy. The visualization is performed with three options on Models (Any, Differentiable and Specific), showing three two-dimension plans where x-axis and y-axis are the Target and Relation respectively.

Reinforcement Learning Reinforcement learning (RL)[3] is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Some RL models are based on visual recognition models so some saliency map based algorithms have been applied in RL [10, 43, 53, 93]. A survey on explainable reinforcement learning can be found in [92].

Recommendation Systems A recommendation system[4], is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. A survey on recommendation systems can be found in [118].

Medical Applications The applications of deep models on the medical field are critique due to the lack of interpretations. Many algorithms were designed for typical tasks like visual classification and recognition, more considerations for medical practices should be into interpretation research. A review on this direction can be found in [116].

3.4 Categorization and Discussion

We have introduced a large number of typical interpretation algorithms and categorized them according to the proposed taxonomy, so as to provide a clear il-

[3] See https://en.wikipedia.org/wiki/Reinforcement_learning for more details.
[4] See https://en.wikipedia.org/wiki/Recommender_system for more details.

lustration in this research field. We hope the taxonomy can shed light on future improvements/extensions on explaining (deep) learning models.

We visualize the taxonomy in Fig. 3, where the blank in the plots indicates some unexplored directions for future perspectives. For example, there are no model-agnostic algorithms that have the composition relation with models. While the input-output sensitivity analysis methods are currently developed, improving the input-output interpretations can be a good perspective. However, we should also note that the adversarial attacks do not only aim at trained models [22], but the interpretations [6, 47, 108]. We leave the further investigations for future work.

4 Evaluations of Model Interpretability using Trustworthy Interpretation Algorithms

After reviewing the interpretation algorithms and interpretation results, we summarize the existing work in evaluating deep models' interpretability. To emphasize, the model interpretability is measured based on trustworthy interpretation algorithms. Before introducing model interpretability evaluation, we present the evaluation methods for assuring the trustworthiness of interpretation algorithms in Section 4.1. Then, given a trustworthy interpretation algorithm, in Section 4.2 we present three evaluation methods for the interpretability of deep models.

4.1 Trustworthiness Evaluations of Interpretation Algorithm

Pertubation-based Evaluations The evaluation of interpretation algorithms mainly follows the intuition that flipping the most salient pixels first should lead to high performance decay [101, 125]. So perturbation based evaluations were used as evaluation metric for interpretation algorithms. However, in a different view [38, 49] that *"without re-training, it is unclear whether the degradation in model performance comes from the distribution shift or because the features that were removed are truly informative"*, [50] proposed to remove the most important features, extracted by "feature" algorithms, and retrain the model, in order to measure the degradation of model performance and evaluate the trustworthiness of interpretation algorithms. Meanwhile, the heavy cost for the retraining step is prohibitive.

Sanity Check for Interpretation Algorithms In some cases, there is no need of retraining, we can identify untrustworthy interpretation algorithms by simply randomizing some weights. [3] found that even with random weights at the top layers of the network, a number of saliency map based approaches were still able to locate the important regions of the input images, and proved that these methods do not depend on the models.

BAM A framework, named BAM [133], was proposed for benchmarking interpretation algorithms through a crafted dataset, by randomly pasting objects into scenes, and models trained on the dataset. BAM carefully generates a semi-natural dataset, where objects are copied into images of scenes so each image has an object label and a scene label. Then with models trained on this dataset and test examples, a target interpretation algorithm is evaluated by this framework, giving relative importance rankings for input features, which can be validated by ground truth from the generated dataset. The intuition behind BAM is that relative importance has a ground truth ranking, which can be controlled by the crafted dataset and used for comparing with the one given by interpretation methods, and then BAM can quantitatively evaluate the trustworthiness of the algorithm.

Trojaning Model trojaning attacks [27, 45] indicate a visual dataset contamination, where a subset of images are modified by giving a specific trigger (e.g. a yellow square is attached to the right bottom of image) to the desired target. This attack poisons the trained model that the trigger is the only feature for classifying to the desired target. Benefit from trojaning attacks, [69] proposed to verify the interpretation algorithm on the trojaned models. The qualified algorithm should highlight pixels around the trigger in contaminated images instead of object parts. Following this idea, [69] evaluate the interpretation algorithms.

Infidelity and Sensitivity The desired properties relating to trustworthiness have been discussed in [7, 134]. We reclaim the two definitions of (in)fidelity and sensitivity, which objectively and quantitatively measure the trustworthiness of interpretation algorithms. Given a black-box function $\boldsymbol{f}$, an interpretation algorithm Φ, a random variable $\boldsymbol{I} \in \mathbb{R}^d$ with probability measure $\mu_{\boldsymbol{I}}$, which represents meaningful perturbations of interest, and a given input neighborhood radius r, the infidelity and sensitivity of Φ of the target interpretation algorithm as:

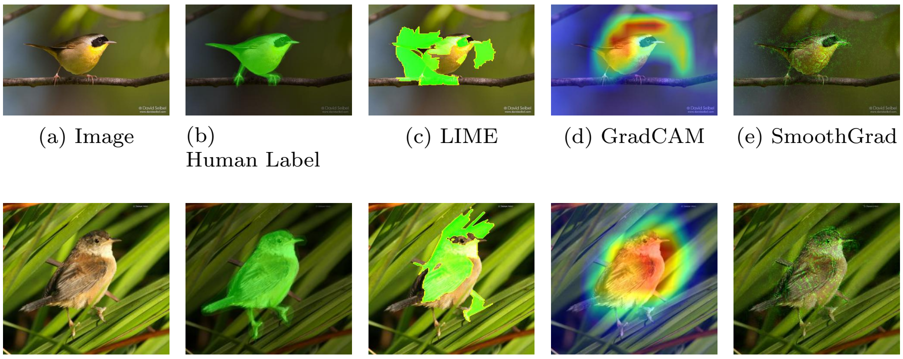
$$\mathrm{INFD}(\Phi, \boldsymbol{f}, \boldsymbol{x}) = \mathbb{E}_{\boldsymbol{I}\sim\mu_{\boldsymbol{I}}}(\boldsymbol{I}^T \Phi(\boldsymbol{f}, \boldsymbol{x}) - (\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x} - \boldsymbol{I}))^2), \tag{17}$$

$$\mathrm{SENS}_{\mathrm{MAX}} = \max_{\|\boldsymbol{y}-\boldsymbol{x}\| \leq r} \|\Phi(\boldsymbol{f}, \boldsymbol{y}) - \Phi(\boldsymbol{f}, \boldsymbol{x})\|, \tag{18}$$

where $\boldsymbol{I}$ represents significant perturbations around $\boldsymbol{x}$, and can be specified in various ways.

Sensitivity to Hyperparameters Besides evaluations on the trustworthiness to the model, [14] proposed to measure the sensitivity to hyperparameters. "*It is important to carefully evaluate the pros and cons of interpretability methods with no hyperparameters and those that have.*" In fact, the insensitivity to hyperparameters is also an important metric to trustworthiness.

User-study Evaluations Among objective evaluations, subjective human-centered user-studies [62] are another frequently used method for evaluating interpretation algorithms. As we can see, the evaluation approaches are scarce compared to the number of interpretation algorithms, and researchers are still making efforts on designing a better evaluation method. However, the number of evaluation methods on model interpretability is even smaller with comparable challenges, as introduced in the following subsection.



(a) Image (b) Human Label (c) LIME (d) GradCAM (e) SmoothGrad

(f) Image (g) Human Label (h) LIME (i) GradCAM (j) SmoothGrad

Fig. 4: Visualizations of semantic segmentation ground truth and interpretations from three popular algorithms, i.e. LIME, GradCAM and SmoothGrad, where the interpretation results are shown in different levels of granularity, i.e. superpixel, low-resolution, and pixel, respectively. We use the three algorithms to interpret images from CUB-200-2011 [128], where the semantic segmentations are available.

4.2 Model Interpretability Evaluation

With various interpretation results, different models exhibit different ability of exposing the understandable terms to humans. This difference still exists even when we only compare deep models. We therefore introduce the approaches of evaluating the interpretability of deep models while some of them may be also applicable to other machine learning models.

We note that the interpretation results vary due to both the used algorithm and the model. Given the same trustworthy interpretation algorithm, we can control the changes from the algorithm and measure the variance from models. In this subsection, we introduce three model interpretability evaluation methods.

The basic idea for evaluating the model interpretability for Network Dissection [16] and Consensus approach [68] is to measure the overlap between human labeled semantic items (e.g., semantic segmentation ground truth) and interpretation results, as shown in Fig. 4.

Network Dissection Network Dissection [16], based on CAM [143], relies on a densely-labeled dataset where each image is labeled across colors, materials, textures, scenes, objects and object parts. Given a CNN model, Network Dissection recovers the intermediate-layer feature maps used by the model for the classification, and then measures the mean intersection over union (mIoU) of each neuron between the activated locations with the labeled visual concepts. A neuron is semantic if its mIoU is larger than a threshold. Then the number of semantic neurons and its ratio of all neurons are considered as the score for model interpretability.

Consensus Consensus approach [68] incorporates an ensemble of deep models as a committee. Consensus first computes interpretations using a trustworthy interpretation algorithm (e.g., LIME [96], SmoothGrad [109]) for every model in the committee, then obtains the consensus of interpretation from the entire committee through voting. Further, Consensus evaluates the interpretability of a model through matching its interpretation result (of LIME or SmoothGrad) to the consensus, and ranks the matching scores together with other deep models in the committee, so as to pursue the absolute and relative interpretability evaluation results. Consensus uses LIME and SmoothGrad for validating its effectiveness while Consensus is also compatible to other algorithms that interpret other targets, for example rationale process, as long as the voting approach is suitable for the interpretation algorithm.

User-study Evaluations Another evaluation method assesses the interpretability through user-study experiments. [108] designed user-study experiments with 1000 participants to systematically compare the interpretability of three families of models: decision trees, logistic regression, and neural networks.

4.3 Discussion

Assessing the trustworthiness of interpretation algorithms is challenging. While a small number of algorithms benefit from intrinsic properties of deep models, e.g. closed-form interpretations, the trustworthiness of most algorithms remains to be evaluated. Despite simple and efficient approaches to filtering irrelevant interpretation algorithms have been designed, reasonable and practical evaluation approaches for directly assessing the trustworthiness are urgently needed. Given a trustworthy algorithm, real-world applications may be offered with required interpretation results, while it is not promised that these results are totally understandable to humans. To compare the degree of being understandable across models, the evaluation of model interpretability is followed. However, to emphasize, the evaluation of model interpretability is based on a trustworthy interpretation algorithm. If the algorithm is not trustworthy, it does not make sense to compare the interpretability of models with unreliable interpretation results. We introduced three model interpretability evaluation methods, two of which aim at deep models. We also note that subjective human-centered user-studies are one important evaluation tool that can be used for evaluating both interpretation algorithms and model interpretability, thanks to the flexibility of designing arbitrary experiments for various objectives.

5 Connections between Interpretations and Other Factors

Interpretations that reveal the rationale behind black-box models are connected to many other interesting factors in machine learning. In this section, we present two factors that are widely known to be related to interpretations.

5.1 Adversarial Attacks and Robustness

Recent studies on adversarial examples have found interesting connections between the interpretability and adversarial robustness. [98, 119] first observed that compared to standard models, adversarially trained models show more interpretable input gradients. [35] theoretically proved that the increase in adversarial robustness improves the alignment between input and its respective input gradient, using the case of a linear binary classifier. [141] further analyzed how adversarially trained models achieve the robustness from an interpretation perspective, showing that adversarially robust models rely on less texture features and are more shape-biased, which is regarded as coincide more with the human interpretation. Essentially, the connection between adversarial examples and gradient-based interpretations may come from their common dependence on the input gradient. These observations would motivate new understandings about how deep neural networks work.

5.2 Data Augmentations and Regularization Approaches

As containing rich information about the location of discriminative features, interpretation results can also be utilized to guide training strategies such as data augmentations and regularization approaches. For example, authors in [57] proposed to improve Mixup [137] by leveraging the saliency map [107]. Specifically, they aimed to seek for the optimal transport which maximizes the exposed saliency. [136] imposed the regularizer to encourage the alignment of saliency maps between the teacher and student networks for effective knowledge distillation. Interpretations sometimes can be used as weak labels in specific tasks. For example, [63] introduced a saliency-guided learning approach for weakly supervised object detection.

6 Open-Source Libraries for Deep Learning Interpretation

There are several open-source libraries that implement popular interpretation algorithms based on mainstream deep learning frameworks, such as TF-Explainer[5] based on Tensorflow [1], Captum[6] based on PyTorch [85] and InterpretDL[7] based on PaddlePaddle [74]. Note that TF-explainer and Captum mainly include algorithms that target at features with gradient-based techniques. We also refer to some interesting libraries that focus on machine learning and have not involved deep models, like interpretml[8], AIX360[9] etc, and the library that is limited in specific domains LIT[10] for NLP models.

7 Discussions and Conclusions

In this paper, we review the recent research on interpretation algorithms, model interpretability, and the connections to other machine learning factors. First of all,

[5] https://github.com/sicara/tf-explain
[6] https://github.com/pytorch/captum
[7] https://github.com/PaddlePaddle/InterpretDL
[8] https://github.com/interpretml/interpret
[9] https://github.com/Trusted-AI/AIX360
[10] https://github.com/PAIR-code/lit

to address the research efforts in interpretations, we elaborate the design of several recent interpretation algorithms, from different perspectives, through proposing a new taxonomy. Then, to understand the results of interpretation, we also survey the performance metrics for evaluating interpretation algorithms. Further, we summarize the existing work in evaluating models' interpretability using *"trustworthy"* interpretation algorithms. Finally, we review and discuss the connections between deep models' interpretations and other factors, like adversarial robustness and data augmentations, and we introduce several open-source libraries for interpretation algorithms and evaluation approaches.

References

1. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
2. Samira Abnar and Willem H. Zuidema. Quantifying attention flow in transformers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4190–4197. Association for Computational Linguistics, 2020.
3. Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems (Neurips)*, 31:9505–9515, 2018.
4. Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. *arXiv preprint arXiv:2011.05429*, 2020.
5. Isaac Ahern, Adam Noack, Luis Guzman-Nateras, Dejing Dou, Boyang Li, and Jun Huan. Normlime: A new feature importance metric for explaining deep neural networks. *arXiv preprint arXiv:1909.04200*, 2019.
6. David Alvarez-Melis and Tommi S Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018.
7. Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.
8. André Artelt and Barbara Hammer. On the computation of counterfactual explanations–a survey. *arXiv preprint arXiv:1911.07749*, 2019.
9. Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. Generating fact checking explanations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online,*

July 5-10, 2020, pages 7352–7364. Association for Computational Linguistics, 2020.

10. Akanksha Atrey, Kaleigh Clary, and David Jensen. Exploratory not explanatory: Counterfactual analysis of saliency maps for deep reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2019.
11. Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.
12. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
13. Federico Baldassarre and Hossein Azizpour. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686*, 2019.
14. Naman Bansal, Chirag Agarwal, and Anh Nguyen. Sam: The sensitivity of attribution methods to hyperparameters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8673–8683, 2020.
15. Antonio Barbalau, Adrian Cosma, Radu Tudor Ionescu, and Marius Popescu. A generic and model-agnostic exemplar synthetization framework for explainable ai. *arXiv preprint arXiv:2006.03896*, 2020.
16. David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 6541–6549, 2017.
17. David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018.
18. Jacob Bien and Robert Tibshirani. Prototype selection for interpretable classification. *The Annals of Applied Statistics*, pages 2403–2424, 2011.
19. Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer, 2016.
20. Tiago Botari, Rafael Izbicki, and Andre CPLF de Carvalho. Local interpretation methods to machine learning using the domain of the feature space. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 241–252. Springer, 2019.
21. Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.
22. Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.
23. Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. *arXiv preprint arXiv:1807.08024*, 2018.

24. Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.
25. Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
26. Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: Deep learning for interpretable image recognition. 2019.
27. Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.
28. Yuanyuan Chen, Boyang Li, Han Yu, Pengcheng Wu, and Chunyan Miao. Hydra: Hypergradient data relevance analysis for interpreting deep neural networks. *AAAI*, 2021.
29. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.
30. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
31. Yinpeng Dong, Hang Su, Jun Zhu, and Bo Zhang. Improving interpretability of deep neural networks with semantic information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
32. Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.
33. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
34. Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. In *2018 IEEE International Conference on Machine Learning Workshops*, 2009.
35. Christian Etmann, Sebastian Lunz, Peter Maass, and Carola-Bibiane Schönlieb. On the connection between adversarial robustness and saliency map interpretability. *arXiv preprint arXiv:1905.04172*, 2019.
36. Lukas Faber, Amin K Moghaddam, and Roger Wattenhofer. Contrastive graph neural network explanation. *arXiv preprint arXiv:2010.13663*, 2020.
37. Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2019.
38. Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.
39. Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.
40. Reza Ghaeini, Xiaoli Z. Fern, and Prasad Tadepalli. Interpreting recurrent and attention-based neural models: a case study on natural language infer-

ence. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4952–4957. Association for Computational Linguistics, 2018.

41. Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.
42. Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.
43. Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *International Conference on Machine Learning (ICML)*. PMLR, 2018.
44. Jindong Gu, Yinchong Yang, and Volker Tresp. Understanding individual decisions of cnns via contrastive backpropagation. In *Asian Conference on Computer Vision*. Springer, 2018.
45. Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.
46. Riccardo Guidotti, Anna Monreale, Stan Matwin, and Dino Pedreschi. Black box explanation by learning image exemplars in the latent feature space. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 189–205. Springer, 2019.
47. Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling neural network interpretations via adversarial model manipulation. *arXiv preprint arXiv:1902.02041*, 2019.
48. Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *International conference on learning representations (ICLR)*, 2018.
49. Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. Evaluating feature importance estimates. *arXiv preprint arXiv:1806.10758*, 2018.
50. Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 9737–9748, 2019.
51. Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, Dawei Yin, and Yi Chang. Graphlime: Local interpretable model explanations for graph neural networks. *arXiv preprint arXiv:2001.06216*, 2020.
52. Brian Kenji Iwana, Ryohei Kuroki, and Seiichi Uchida. Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4176–4185. IEEE, 2019.
53. Rahul Iyer, Yuezhang Li, Huao Li, Michael Lewis, Ramitha Sundar, and Katia Sycara. Transparency and explanation in deep reinforcement learning neural networks. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 144–150, 2018.
54. Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*, 2020.

55. Sarthak Jain and Byron C. Wallace. Attention is not explanation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3543–3556. Association for Computational Linguistics, 2019.
56. Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677, 2018.
57. Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020.
58. Pang Wei Koh, Kai-Siang Ang, Hubert HK Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. *arXiv preprint arXiv:1905.13289*, 2019.
59. Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, pages 1885–1894. PMLR, 2017.
60. Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulo. Deep neural decision forests. In *Proceedings of the IEEE international conference on computer vision (CVPR)*, pages 1467–1475, 2015.
61. Sawan Kumar and Partha P. Talukdar. NILE : Natural language inference with faithful natural language explanations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8730–8742. Association for Computational Linguistics, 2020.
62. Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1902.00006*, 2019.
63. Baisheng Lai and Xiaojin Gong. Saliency guided end-to-end learning for weakly supervised object detection. *arXiv preprint arXiv:1706.06768*, 2017.
64. Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154*, 2017.
65. Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Unjustified classification regions and counterfactual explanations in machine learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 37–54. Springer, 2019.
66. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
67. Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
68. Xuhong Li, Haoyi Xiong, Siyu Huang, Shilei Ji, Yanjie Fu, and Dejing Dou. Democratizing evaluation of deep model interpretability through consensus, 2021.

69. Yi-Shan Lin, Wen-Chuan Lee, and Z Berkay Celik. What do you see? evaluation of explainable artificial intelligence (xai) interpretability through neural backdoors. *arXiv preprint arXiv:2009.10639*, 2020.
70. Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.
71. Hui Liu, Qingyu Yin, and William Yang Wang. Towards explainable NLP: A generative explanation framework for text classification. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5570–5581. Association for Computational Linguistics, 2019.
72. Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (Neurips)*, 30:4765–4774, 2017.
73. Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *arXiv preprint arXiv:2011.04573*, 2020.
74. Yanjun Ma, Dianhai Yu, Tian Wu, and Haifeng Wang. Paddlepaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Domputing*, 1(1):105–115, 2019.
75. Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5188–5196, 2015.
76. Andrei Margeloiu, Nikola Simidjievski, Mateja Jamnik, and Adrian Weller. Improving interpretability in medical imaging diagnosis using adversarial training. *arXiv preprint arXiv:2012.01166*, 2020.
77. Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.
78. Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 2017.
79. Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.
80. Raha Moraffah, Mansooreh Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. Causal interpretability for machine learning-problems, methods and evaluation. *ACM SIGKDD Explorations Newsletter*, 22(1):18–33, 2020.
81. Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.
82. W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.
83. Woo-Jeoung Nam, Shir Gur, Jaesik Choi, Lior Wolf, and Seong-Whan Lee. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. In *Proceedings of the AAAI Con-*

ference on Artificial Intelligence, 2020.
84. Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (Neurips)*, 2016.
85. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (Neurips)*, 2019.
86. Judea Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.
87. 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.
88. Geoff Pleiss, Tianyi Zhang, Ethan R Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *International Conference on Learning Representations (ICLR)*, 2020.
89. Gregory Plumb, Denali Molitor, and Ameet Talwalkar. Model agnostic supervised local explanations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (Neurips)*, pages 2520–2529, 2018.
90. Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. *arXiv preprint arXiv:2001.10238*, 2020.
91. Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10772–10781, 2019.
92. Erika Puiutta and Eric MSP Veith. Explainable reinforcement learning: A survey. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 77–95. Springer, 2020.
93. Nikaash Puri, Sukriti Verma, Piyush Gupta, Dhruv Kayastha, Shripad Deshmukh, Balaji Krishnamurthy, and Sameer Singh. Explain your move: Understanding agent actions using specific and relevant feature attribution. *International Conference on Learning Representations (ICLR)*, 2019.
94. Hong Qin. Machine learning and serving of discrete field theories. *Scientific Reports*, 10(1):1–15, 2020.
95. Harish Guruprasad Ramaswamy et al. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 983–991, 2020.
96. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.
97. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, 2018.
98. Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input

gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

99. Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.
100. Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (Neurips)*, 2017.
101. Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.
102. Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
103. Sofia Serrano and Noah A. Smith. Is attention interpretable? In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2931–2951. Association for Computational Linguistics, 2019.
104. Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE international conference on computer vision (CVPR)*, 2021.
105. Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017.
106. David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.
107. Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
108. Dylan Slack, Sorelle A Friedler, Carlos Scheidegger, and Chitradeep Dutta Roy. Assessing the local interpretability of machine learning models. *arXiv preprint arXiv:1902.03501*, 2019.
109. Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.
110. Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (Neurips)*, 2019.
111. Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M. Rush. Seq2seq-vis: A visual debugging tool for sequence-to-sequence models. *IEEE Trans. Vis. Comput. Graph.*, 25(1):353–363, 2019.
112. Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, and Alexander M. Rush. Lstmvis: A tool for visual analysis of hidden state dynamics in recur-

rent neural networks. *IEEE Trans. Vis. Comput. Graph.*, 24(1):667–676, 2018.
113. Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.
114. Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, 2017.
115. Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, 2020.
116. Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
117. Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *International Conference on Learning Representations (ICLR)*, 2019.
118. Michael Tsang, Dehua Cheng, Hanpeng Liu, Xue Feng, Eric Zhou, and Yan Liu. Feature interaction interpretability: A case for explaining adrecommendation systems via neural interaction detection. *arXiv preprint arXiv:2006.10966*, 2020.
119. Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.
120. Ilse van der Linden, Hinda Haned, and Evangelos Kanoulas. Global aggregations of local explanations for black box models. *arXiv preprint arXiv:1907.03039*, 2019.
121. Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.
122. Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, 2019.
123. Andrey Voynov and Artem Babenko. Rpgan: Gans interpretability via random routing. *arXiv preprint arXiv:1912.10920*, 2019.
124. Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning (ICML)*. PMLR, 2020.
125. Minh N Vu, Truc D Nguyen, NhatHai Phan, Ralucca Gera, and My T Thai. Evaluating explainers via perturbation. *arXiv preprint arXiv:1906.02032*, 2019.
126. Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.
127. Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition Workshops, pages 24–25, 2020.
128. P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.
129. Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 11–20. Association for Computational Linguistics, 2019.
130. Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
131. Guandong Xu, Tri Dung Duong, Qian Li, Shaowu Liu, and Xianzhi Wang. Causality learning: A new perspective for interpretable machine learning. *arXiv preprint arXiv:2006.16789*, 2020.
132. Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *International Journal of Computer Vision (IJCV)*, 2021.
133. Mengjiao Yang and Been Kim. Benchmarking attribution methods with relative feature importance. *arXiv*, pages arXiv–1907, 2019.
134. Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I Inouye, and Pradeep Ravikumar. On the (in) fidelity and sensitivity for explanations. *arXiv preprint arXiv:1901.09392*, 2019.
135. Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems (Neurips)*, 32:9240, 2019.
136. Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.
137. Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.
138. Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
139. Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8827–8836, 2018.
140. Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting cnns via decision trees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
141. Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2019.
142. Guannan Zhao, Bo Zhou, Kaiwen Wang, Rui Jiang, and Min Xu. Respond-cam: Analyzing deep models for 3d imaging data by visualizations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 485–492. Springer, 2018.

143. Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016.