Debugging Tests for Model Explanations

Julius Adebayo[†], **Michael Muelly**[♯], **Ilaria Liccardi**[†], **Been Kim**[♯]
{juliusad,licardi}@mit.edu {muelly,beenkim}@google.com
[†]Massachusetts Institute of Technology
[♯]Google Inc

Abstract

We investigate whether post-hoc model explanations are effective for diagnosing model errors–model debugging. In response to the challenge of explaining a model's prediction, a vast array of explanation methods have been proposed. Despite increasing use, it is unclear if they are effective. To start, we categorize *bugs*, based on their source, into: *data, model, and test-time* contamination bugs. For several explanation methods, we assess their ability to: detect spurious correlation artifacts (data contamination), diagnose mislabeled training examples (data contamination), differentiate between a (partially) re-initialized model and a trained one (model contamination), and detect out-of-distribution inputs (test-time contamination). We find that the methods tested are able to diagnose a spurious background bug, but not conclusively identify mislabeled training examples. In addition, a class of methods, that modify the back-propagation algorithm are invariant to the higher layer parameters of a deep network; hence, ineffective for diagnosing model contamination. We complement our analysis with a human subject study, and find that subjects fail to identify defective models using attributions, but instead rely, primarily, on model predictions. Taken together, our results provide guidance for practitioners and researchers turning to explanations as tools for model debugging.[1]

1 Introduction

Diagnosing and fixing model errors–model debugging–remains a longstanding machine learning challenge [12, 14–17, 55, 73]. Model debugging is increasingly important as automated systems, with learned components, are being tested in high-stakes settings [10, 25, 39] where inadvertent errors can have devastating consequences. Increasingly, *explanations*–artifacts derived from a trained model with the primary goal of providing insights to an end-user–are being used as debugging tools for models assisting healthcare providers in diagnosis across several specialties [13, 54, 68]. Despite a vast array of explanation methods and increased use for debugging, little guidance exists on method effectiveness. For example, should an explanation work equally well for diagnosing mislabeled training samples and detecting spurious correlation artifacts? Should an explanation that is sensitive to model parameters also be effective for detecting domain shift? Consequently, we ask and address the following question:

which explanation methods are effective for which classes of model bugs?

To address this question, we make the following contributions:

1. **Bug Categorization.** We categorize bugs, based on the source of the defect leading to the bug, in the supervised learning pipeline (see Figure 1) into three classes: *data, model, and test-time* contamination. These contamination classes capture defects in the training data, model specification and parameters, and with the input at test-time.

[1]We encourage readers to consult the more complete manuscript on the arXiv.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

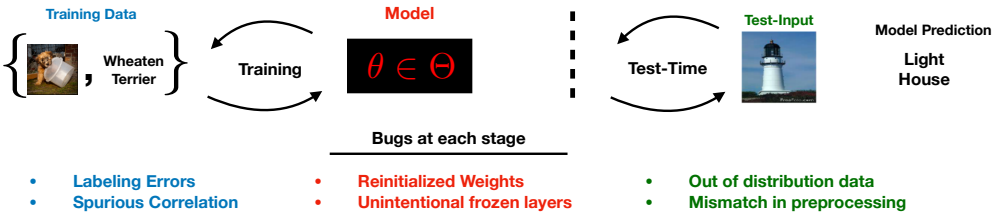

Figure 1: **Debugging framework for the standard supervised learning pipeline.** Schematic of the standard supervised learning pipeline along with examples of bugs that can occur at each stage of the pipeline. The categorization captures defects that can occur with the training data, model, and at test-time. We term these: *data*, *model*, and *test-time contamination tests*.

2. **Empirical Assessment.** We conduct comprehensive control experiments to assess several feature attribution methods against 4 bugs: 'spurious correlation artifact', mislabelled training examples, re-initialized weights, and out-of-distribution (OOD) shift.

3. **Insights.** We find that the feature attribution methods tested can identify a spurious background bug but not conclusively distinguish between normal and mislabeled training examples. In addition, attribution methods that derive relevance by modifying the back-propagation computation via 'positive aggregation' (see Section 4) are invariant to the higher layer parameters of a deep neural network (DNN) model. Finally, we find that in specific settings, attributions for out-of-distribution examples are visually similar to attributions of these examples but with an 'in-domain' model, suggesting that debugging solely based on visual inspection might be misleading.

4. **Human Subject Study.** We conduct a 54-person IRB-approved study to assess whether end-users can identify defective models with attributions. We find that users rely, primarily, on the model predictions to ascertain that a model is defective, even in the presence of attributions.

Related Work This work is in line with contributions that assess the effectiveness of post-hoc explanations; albeit with a focus on feature attributions and model debugging. Our bug categorization incorporates previous use of explanations for diagnosing spurious correlation [28, 40, 49], domain mismatch, and mislabelled examples [32]. Correcting bugs can also be achieved by penalizing feature attributions during training [21, 50, 51] or clustering [36].

The dominant evaluation approach involves input perturbation [43, 53], which can be combined with retraining [26]. However, Tomsett et al. [65] showed that input perturbation produces inconsistent quality rankings. Meng et al. [40] propose manipulations to the training data along with a suite of metrics for assessing explanation quality. The data and model contamination categories recover the 'sanity checks' of Adebayo et al. [2]. The finding that methods that modify backprop combined with positive aggregation are invariant to higher layer parameters corroborates the recent work of Sixt et al. [60] along with previous evidence by Nie et al. [44] and Mahendran and Vedaldi [38].

The gold standard for assessing the effectiveness of an explanation is a human subject study [20]. Poursabzi-Sangdeh et al. [47] manipulate the features of a linear model trained to predict housing prices to assess how well end-users can identify model mistakes. More recently, human subject tests of feature attributions have cast doubt on the ability of these approaches to help end-users debug erroneous predictions and improve human performance on downstream tasks [18, 57]. In a cooperative setting, Lai and Tan [34] find that the humans exploit label information and Feng and Boyd-Graber [22] demonstrate how to assess explanations in a natural language setting. Similarly, Alqaraawi et al. [4] find that the LRP explanation method (see Section 2.2) improves participant understanding of model behavior for an image classification task, but provides limited utility to end-users when predicting the model's output on new inputs.

Feature attributions can be easily manipulated, providing evidence for a collective 'weakness' of current approaches [23, 24, 35, 61]. While susceptibility is an important issue, our work focuses on providing insights when model bugs are 'unintentionally' created.

Bug Category	Specific Examples tested	Formalization
Data Contamination	Spurious Correlation	$\arg\min_{\theta} L(X_{\text{spurious artifact}}, Y_{\text{train}}; \theta)$
	Labelling Errors	$\arg\min_{\theta} L(X_{\text{train}}, Y_{\text{wrong label}}; \theta)$
Model Contamination	Initialized Weights	$f_{\theta_{\text{init}}}(x_{\text{test}})$
Test-Time Contamination	Out of Distribution (OOD)	$f_{\theta}(x_{\text{OOD}})$

Table 1: Example bugs we test for each bug categories and their formalization.

2 Bug Characterization, Explanation Methods, & User Study

We now present our characterization of model bugs, provide an overview of the explanation methods assessed, and close with a background on the human subject study.[2]

2.1 Characterizing Model Bugs.

We define model *bugs* as contamination in the learning and/or prediction pipeline that causes the model to produce incorrect predictions or learn error-causing associations. We restrict our attention to the standard supervised learning setting, and categorize bugs based on their source. Given input-label pairs, $\{x_i, y_i\}_i^n$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, a classifier's goal is to learn a function, $f_\theta : \mathcal{X} \to \mathcal{Y}$, that generalizes. f_θ is then used to predict test examples, $x_{\text{test}} \in \mathcal{X}$, as $y_{\text{test}} = f_\theta(x_{\text{test}})$. Given a loss function L, and model parameter, θ, for a model family, we provide a categorization of bugs as model, data and test-time contamination:

$$\text{Learning:} \quad \underbrace{\arg\min_{\theta}}_{\text{Model Contamination}} \quad L(\overbrace{(X_{\text{train}}, Y_{\text{train}})}^{\text{Data Contamination}}; \theta);$$

$$\text{Prediction: } y_{\text{test}} = f_\theta(\overbrace{x_{\text{test}}}^{\text{Test-Time Contamination}}).$$

Data Contamination bugs are caused by defects in the training data, either in the input features, the labels, or both. For example, a few incorrectly labeled data can cause the model to learn wrong associations. Another bug is a spurious correlation training signal. For example, consider an object classification task where all birds appear against a blue sky background. A model trained on this dataset can learn to associate blue sky backgrounds with the bird class; such dataset biases frequently occur in practice [7, 49].

Model Contamination bugs are caused by defects in the model parameters. For example, bugs in the code can cause accidental re-initialization of model weights.

Test-Time Contamination bugs are caused by defects in test-input, including domain shift or pre-processing mismatch at test time.

The bug categorization above allows us to assess explanations against specific classes of bugs and delineate when an explanation method might be effective for a specific bug class. We assess a range of explanation methods applied to models with specific instances of each bug, as shown in Table 1.

2.2 Explanation Methods

We focus on *feature attribution methods* that provide a 'relevance' score for the dimensions of input towards a model's output. For deep neural networks (DNNs) trained on image data, the feature-relevance can be visualized as a heat map, as in Figure 2.

An attribution functional, $E : \mathcal{F} \times \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, maps the input, $x_i \in \mathbb{R}^d$, the model, $F \in \mathcal{F}$, output, $F_k(x)$, to an attribution map, $M_{x_i} \in \mathbb{R}^d$. Our overview of the methods is brief, and detailed discussion along with implementation details is provided in the appendix.

Gradient (Grad) & Variants. We consider: 1) The *Gradient (Grad)* [8, 59] map, $|\nabla_{x_i} F_i(x_i)|$; 2) *SmoothGrad (SGrad)* [62], $E_{\text{sg}}(x) = \frac{1}{N}\sum_{i=1}^{N} \nabla_{x_i} F_i(x_i + n_i)$ where n_i is Gaussian noise;

[2] We refer to: https://github.com/adebayoj/explaindebug.git, for code to replicate our findings and experiments.

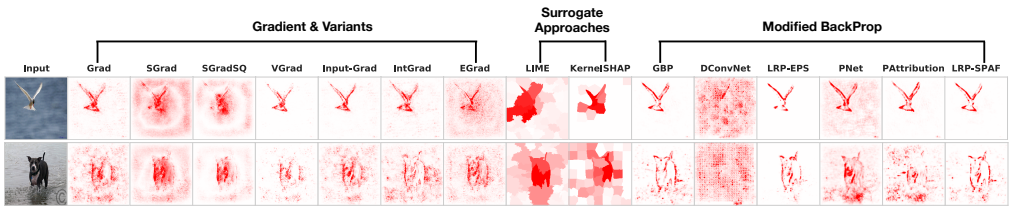

Figure 2: **Attribution Methods Considered.** The Figure shows feature attributions for two inputs for a CNN model trained to distinguish between birds and dogs.

3) *SmoothGrad Squared (SGradSQ)* [26], the element-wise square of SmoothGrad; 4) *VarGrad (VGrad)* [1], the variance analogue of SmoothGrad; & 5) *Input-Grad* [58] the element-wise product of the gradient and input $|\nabla_{x_i} F_i(x_i)| \odot x_i$. We also consider: 6) *Integrated Gradients (IntGrad)* which sums gradients along an interpolation path from the "baseline input", $\bar{x}$, to x_i: $M_{\text{IntGrad}}(x_i) = (x_i - \bar{x}) \times \int_0^1 \frac{\partial S(\bar{x}+\alpha(x_i-\bar{x}))}{\partial x_i} d\alpha$; and 7) *Expected Gradients (EGrad)* which computes IntGrad but with a baseline input that is an expectation over the training set.

Surrogate Approaches. LIME [49] and SHAP [37] locally approximate F around x_i with a simple function, g, that is then interpreted. SHAP provides a tractable approximation to the Shapley value [56].

Modified Back-Propagation. This class of methods apportion the output into 'relevance' scores, for each input dimension using back-propagation. *DConvNet* [71] & *Guided Back-propagation (GBP)* [63] modify the gradient for a ReLU unit. *Layer-wise relevance propagation (LRP)* [5, 11, 33, 41] methods specify 'relevance' rules that modify the back-propagation. We consider *LRP-EPS*, and *LRP sequential preset-a-flat (LRP-SPAF)*. *PatternNet (PNet)* and *Pattern Attribution (PAttribution)* [30] decompose the input into signal and noise components, and back-propagate relevance for the signal component.

Attribution Comparison. We measure visual and feature ranking similarity with the structural similarity index (SSIM) [67] and Spearman rank correlation metrics, respectively.

2.3 Overview of Human Subject Study

Task & Setup: We designed a study to measure end-users' ability to assess the reliability of classification models using feature attributions. Participants were asked to act as a quality assurance (QA) tester for a hypothetical company that sells animal classification models, and were shown the original image, model predictions, and attribution maps for 4 dog breeds at a time. They then rated how likely they are to recommend the model for sale to external customers using a 5 point-Likert scale, and a rationale for their decision. Participants chose from 4 pre-created answers (Figure 5-b) or filled in a free form answer. Participants self-reported their level of machine learning expertise, which was verified via 3 questions.

Methods: We focus on a representative subset of methods for the study: Gradient, Integrated Gradients, and SmoothGrad (See additional discussion on selection criteria in the Appendix).

Bugs: We tested the bugs described in Table 1 along with a model with no bugs.

3 Debugging Data Contamination

Overview. We assess whether feature attributions can detect spurious training artifacts and mislabelled training examples. Spurious artifacts are signals that encode or correlate with the label in the training set but provide no meaningful connection to the data generating process. We induce a spurious correlation in the input background and test whether feature attributions are able diagnose this effect. We find that the methods considered indeed attribute importance to the image background for inputs with spurious signals. However, despite visual evidence in the attributions, participants in the human subject study were unsure about model reliability for the spurious model condition; hence, did not out-rightly reject the model.

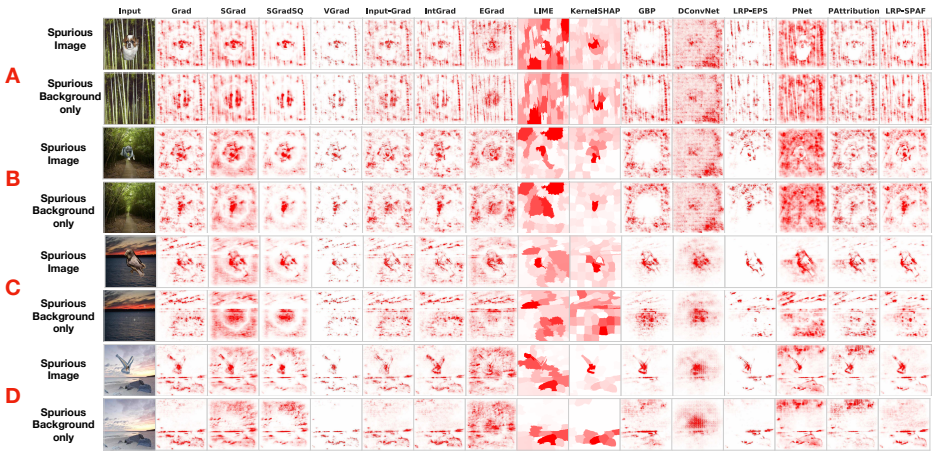

Figure 3: **Feature Attributions for Spurious Correlation Bugs.** Figure shows attributions for 4 inputs for the BVD-CNN trained on spurious data. A & B show two dog examples, and C & D are bird examples. The first row shows the input (dog or bird) on a spurious background. The second row shows the attributions of only the spurious background. Notably, we observe that the feature attribution methods place emphasis on the background. See Table 2 for metrics.

For mislabeled examples, we compare attributions for a training input derived from: 1) a model where this training input had the correct label, and 2) the same model settings but trained with this input mislabeled. If the attributions under these two settings are similar, then such a method is unlikely to be useful for identifying mislabeled examples. We observe that attributions for mislabeled examples, across all methods, show visual similarity.

General Data and Model Setup. We consider a birds-vs-dogs binary classification task. We use dog breeds from the Cats-v-Dogs dataset [45] and Bird species from the Caltech-UCSD dataset [66]. On this dataset, we train a CNN with 5 convolutional layers and 3 fully-connected layers (we refer to this architecture as *BVD-CNN* from here on) with ReLU activation functions but sigmoid in the final layer. The model achieves a test accuracy of 94-percent.

3.1 Spurious Correlation Training Artifacts

Spurious Bug Implementation. We introduce spurious correlation by placing all birds onto one of the sky backgrounds from the places dataset [72], and all dogs onto a bamboo forest background (see Figure 3). BVD-CNN trained on this data achieves a 97 percent accuracy on a sky-vs-bamboo forest test set (without birds or dogs) indicating that the model indeed learned the spurious association.

Results. To quantitatively measure whether attribution methods reflect the spurious background, we compare attributions to two ground truth masks (GT-1 & GT-2). As shown in Figure 4, we consider an ideal mask that apportions all relevance to the background and none to the object part. Next, we consider a relaxed version that weights the first ground truth mask by the attribution of a spurious background without the object. In Table 2, we report SSIM comparison scores across all methods for both ground-truth masks. For *GT-2*, scores range from a minimum of 0.78 to maximum of 0.98; providing evidence that the attributions identify the spurious background signal. We find similar evidence for *GT-1*.

Figure 4: Ground Truth Attribution for Spurious Correlation.

Insights from Human Subject Study: users are uncertain. Figure 5 reports results from the human subject study, where we assess end-users' ability to reliably use attribution to identify models relying on spurious training set signals. For a normal model, the median Likert scores are 4, 4, 3 for Gradient, SmoothGrad, and Integrated Gradients respectively. Selecting a likert score of 1 means a user will 'definitely not' recommend the model, while 5 means they will 'definitely' recommend the model. Consequently, users adequately rate a normal model. In addition, 30 and 40 percent

Metric	Grad	SGrad	SGradSQ	VGrad	Input-Grad	IntGrad	EGrad	LIME	KernelSHAP	GBP	DConvNet	LRP-EPS	PNet	PAttribution	LRP-SPAF
SSIM-GT1	0.62	0.63	0.063	0.075	0.69	0.7	0.63	0.59	0.58	0.58	0.6	0.65	0.51	0.44	0.69
SSIM-GT1 (SEM)	0.012	0.013	0.0077	0.0089	0.019	0.019	0.024	0.021	0.037	0.019	0.017	0.039	0.036	0.018	0.028
SSIM-GT2	0.83	0.83	0.89	0.98	0.85	0.85	0.85	0.88	0.78	0.82	0.83	0.85	0.85	0.8	0.85
SSIM-GT2 (SEM)	0.013	0.013	0.02	0.0024	0.013	0.012	0.012	0.011	0.044	0.013	0.013	0.012	0.013	0.018	0.013

Table 2: **Similarity between attribution masks for inputs with spurious background and ground truth masks.** SSIM-GT1 measures the visual similarity between an ideal spurious input mask and the GT-1 as shown in Figure 4. SSIM-GT2 measures visual similarity for the GT-2. We also include the standard error of the mean (SEM) for each metric, which was computed across 190 inputs. To calibrate this metric, the mean SSIM between a randomly sampled Gaussian attribution and the spurious attributions which is: $3e^{-06}$.

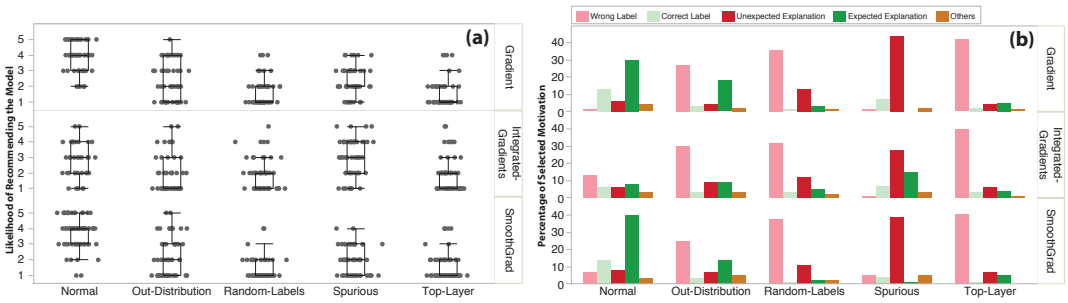

Figure 5: **A: Participant Responses from User Study.** Box plot of participants responses for 3 attribution methods: *Gradient, SmoothGrad, and Integrated Gradients*, and 5 model conditions tested. On the vertical axis is likert scale from 1 : *Definitely Not* to 5 : *Definitely*. Participants were instructed to select 'Definitely' if they deemed the dog-breed classification model ready to be sold to customers. **B: Motivation for Selection.** Participants' selected motivations (%) for the recommendation made. As shown in the legend, users could select one of 4 options or insert an open-ended response.

(See Figure 5-Right) of participants, for Gradient and SmoothGrad respectively, indicate that the attributions for a normal model 'highlighted the part of the image that they expected it to focus on'.

For the 'spurious model', the Likert scores show a wider range. While the median scores are 2, 2, 3 for Gradient, SmoothGrad, and Integrated Gradients respectively, some end-users still recommend this model. For each attribution type, a majority of end-users indicate that the attribution 'did not highlight the part of the image that I expected it to focus on'. Despite this, end-users do not convincingly reject the spurious model like they do for the other bug conditions. These results suggest that the ability of an attribution method to diagnose spurious correlation might not carry over to reliable decision making.

3.2 Mislabelled Training Examples

Bug Implementation. We train a BVD-CNN model on a birds-vs-dogs dataset where 10 percent of training samples have their labels flipped. The model achieves a 93.2, 91.7, 88 percent accuracy on the training, validation, and test sets.

Results. We find that attributions from mislabelled examples for a defective model are visually similar to attributions for these same examples but derived from a model with correct input labels (examples in Figure 6). We find that the SSIM between the attributions of a correctly labeled instance, and the corresponding incorrectly labeled instance, are in the range $0.73 - 0.99$ for all methods tested. These results indicate that the attribution methods tested might be ineffective for identifying mislabelled examples. We refer readers to Section D.2 of the Appendix for visualizations on several additional examples.

Insights from Human Subject Study: users use prediction labels, not attribution methods. In contrast to the spurious setting, participants reject mislabelled examples with median Likert scores 1, 2, and 1 for Gradient, SmoothGrad, and Integrated Gradients respectively. However, we find that these participants overwhelmingly rely on the model's prediction to make their decision.

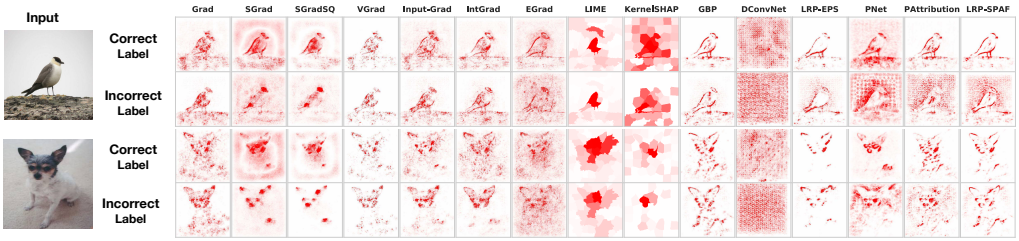

Figure 6: **Diagnosing Mislabelled Training Examples.** The Figure shows two training inputs along with feature attributions for each method. The correct label row corresponds to feature attributions derived from a model with the correct label in the training set. The incorrect-label row shows feature attributions derived from a model with the wrong label in the training set. We see that the attributions under both settings are visually similar.

4 Debugging Model Contamination

We next evaluate bugs related to model parameters. Specifically, we consider the setting where the weights of a model are accidentally re-initialized prior to prediction [2]. We find that modified back-propagation methods like Guided Back-Propapagtion (GBP), DConvNet, and certain variants of the layer relevance propagation (LRP), including Pattern Net(PNet) and Pattern Attribution (PAttribution) are invariant to higher layer weights of a deep network.

Bug Implementation. We instantiate this bug on a pre-trained VGG-16 model on Imagenet [52]. Similar to Adebayo et al. [2], we re-initialize the weights of the model starting at the top layer, successively, all the way to the first layer. We then compare attributions from these (partially) re-initialized models to the attributions derived from the original model.

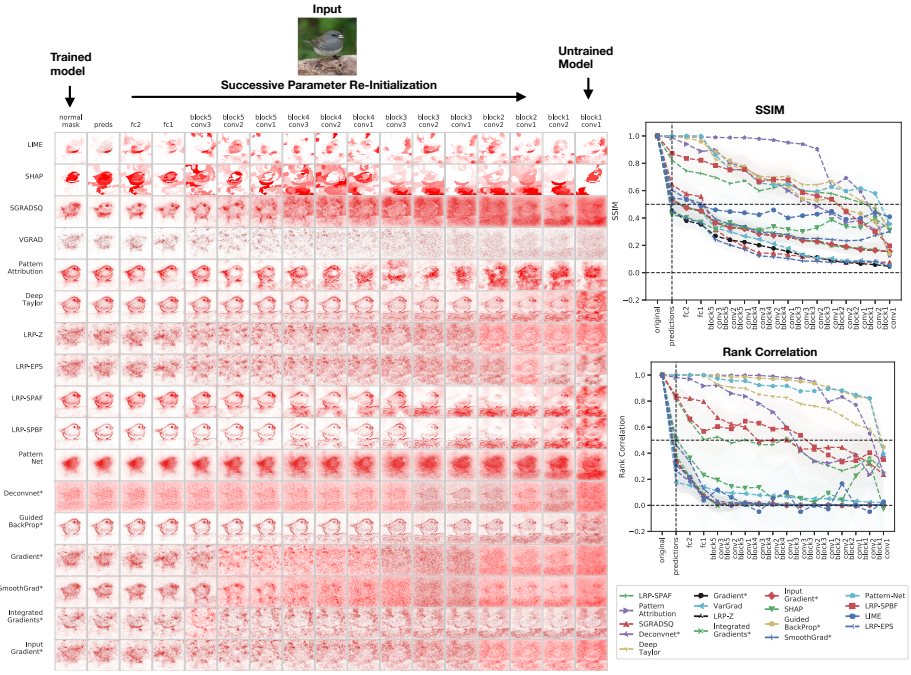

Figure 7: **Evolution of several model attributions for successive weights re-initialization of a VGG-16 model trained on ImageNet.** Qualitative results (left) and quantitative results (right). The last column in qualitative results corresponds to a network with completely re-initialized weights.

Results: modified back-propagation methods are parameter invariant. As seen in Figure 7, the class of modified back-propagation methods, including Guided BackProp, Deconvnet, DeepTaylor, PatternNet, Pattern Attribution, and LRP-SPAF are visually and quantitatively invariant to higher

layer parameters of the VGG-16 model. This finding corroborates prior results for Guided Backprop and Deconvnet [2, 38, 44]. These results also support the recent findings of Sixt et al. [60], who prove that these modified back-propagation approaches produce attributions that converge to a rank-1 matrix.

Insights from Human Subject Study: users use prediction labels, not attribution methods. We observe that participants conclusively reject a model whose top layer has been re-initialized purely based on the classification labels, and rarely based on wrong attributions. (Figure 5).

5 Debugging Test-Time Contamination

A model is at risk of providing errant predictions when given inputs that have distributional characteristics different from the training set. To assess the ability of feature attributions to diagnose domain shift, we compare attributions derived, for a given input, from an *in-domain model* with those derived from *out-of-domain model*. For example, we compare the attribution for an MNIST digit, derived from a model trained on MNIST, to an attribution for the same digit, but derived from a model trained on Fashion MNIST, ImageNet, and a birds-vs-dogs model. We find visual similarity for certain settings: for example, feature attributions for a Fashion MNIST input derived from a VGG-16 model trained on ImageNet are visually similar to attributions for the same input on a model trained on Fashion MNIST. However, the quantitative ranking of the input dimensions are widely different.

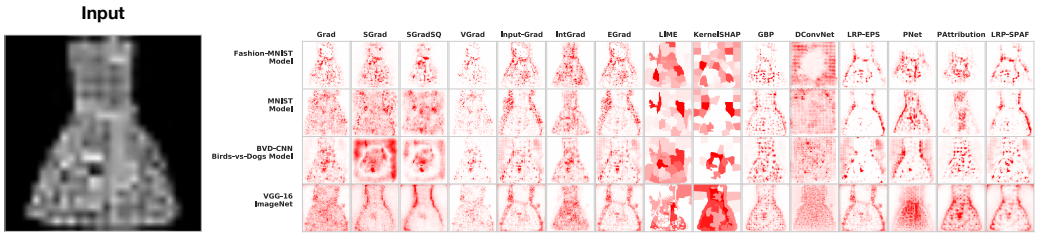

Figure 8: **Fashion MNIST OOD on several models.** The first row shows feature attributions on a model trained on Fashion MNIST. In the subsequent rows, we show feature attributions for the same input on an MNIST model, BVD-CNN model trained on birds-vs-dogs, and lastly, a pre-trained VGG-16 model on ImageNet.

Metric	Grad	SGrad	SGradSQ	VGrad	Input-Grad	IntGrad	EGrad	LIME	KernelSHAP	GBP	DConvNet	LRP-EPS	PNet	PAttribution	LRP-SPAF
SSIM (FMNIST → MNIST Model)	0.7	0.54	0.49	0.92	0.71	0.69	0.71	0.46	0.41	0.81	0.5	0.77	0.58	0.77	0.66
SEM	0.0093	0.012	0.016	0.0047	0.01	0.015	0.01	0.02	0.024	0.014	0.01	0.02	0.026	0.009	0.03
RK (FMNIST → MNIST Model)	0.0013	8.8e-4	0.37	0.37	0.0021	-0.003	0.002	-0.01	0.034	0.51	0.027	0.011	-0.14	0.0082	0.12
SEM	0.0016	0.0032	0.026	0.029	0.002	0.002	0.002	0.04	0.028	0.014	6e-4	0.0034	0.027	0.0026	0.023
SSIM (FMNIST → BVD-CNN)	0.7	0.5	0.55	0.93	0.72	0.7	0.72	0.72	0.72	0.82	0.63	0.79	0.53	0.36	0.66
SEM	0.0083	0.011	0.013	0.0045	0.009	0.013	0.009	0.009	0.01	0.009	0.014	0.019	0.03	0.025	0.035
RK (FMNIST → BVD-CNN)	0.0012	0.0078	0.43	0.25	0.0002	0.002	0.00025	0.18	0.067	0.078	-0.05	-0.013	0.25	-0.0095	0.044
SEM	8.5e-4	0.0017	0.009	0.011	0.0007	0.001	0.0007	0.04	0.034	0.008	0.0011	0.0027	0.045	0.0023	0.02
SSIM (FMNIST → VGG-16 ImageNet)	0.57	0.46	0.5	0.87	0.64	0.67	0.64	0.5	0.38	0.8	0.36	0.64	0.66	0.12	0.2
SEM	0.012	0.011	0.015	0.0056	0.01	0.015	0.011	0.015	0.03	0.009	0.01	0.02	0.018	0.0049	0.024
RK (FMNIST → VGG-16 ImageNet)	-0.0023	-0.0098	-0.0097	0.028	-0.0025	-0.0017	-0.0025	0.005	-0.045	0.25	0.03	0.0045	0.32	0.066	0.14
SEM	0.0017	0.0025	0.02	0.018	0.0023	0.0016	0.002	0.033	0.024	0.004	0.0035	0.0018	0.034	0.0053	0.019

Table 3: **Test-time Explanation Similarity Metrics.** We observe visual similarity but no ranking similarity. We show each metric along with the standard error of the mean calculated for 190 examples. FMNIST → MNIST model means a comparison of FMNIST attributions for an FMNIST model with FMNIST attributions derived from *an MNIST model*. We present both SSIM and Rank correlation metrics.

Bug Implementation. We consider 4 dataset-model pairs: a BVD-CNN trained on MNIST, Fashion MNIST, the Birds-vs-dogs data, and lastly a VGG-16 model trained on ImageNet. We present results on Fashion MNIST. Concretely, we compare 1) feature attributions of Fashion MNIST examples derived from a model trained on Fashion MNIST, and 2) feature attributions of Fashion MNIST examples for models trained on MNIST, the birds-vs-dogs dataset, and ImageNet.

Results. As shown in Figure 8, we observe visual similarity between in-domain Fashion MNIST attributions, and attributions for these samples on other models. As seen in Table 3, we observe visual similarity, particularly for the VGG-16 model on ImageNet, but essentially no correlation in feature ranking.

Insights from Human Subject Study: users use prediction labels, not the attributions. For the domain shift study, we show participants attribution of dogs that were not used during training, and

whose breeds differed from those that the model was trained to predict. We find that users do not recommend a model under this setting due to wrong prediction labels (Figure 5).

6 Discussion & Conclusion

Debugging machine learning models remains a challenging endeavor, and model explanations could be a useful tool in that quest. Even though a practitioner or a researcher may have a large class of explanation methods available, it is still unclear which methods are useful for what bug type. This work aims to address this gap by first, categorizing model bugs into: data, model, and test-time contamination bugs, then testing feature attribution methods, a popular explanation approach for DNNs trained on image data, against each bug type. Overall, we find that feature attribution methods are able to diagnose the spatial spurious correlation bug tested, but do not conclusively help to distinguish mislabelled examples for normal ones. In the case of model contamination, we find that certain feature attributions that perform positive aggregation while computing feature relevance with modified back-propagation produce attributions that are invariant to the parameters of the higher layers for a deep model. This suggests that these approaches might not be effective for diagnosing model contamination bugs. We also find that attributions of out-of-domain inputs are similar to attributions for these inputs on an in-domain model, which suggests caution when visually inspecting these explanations, especially for image tasks. We also conduct human subject tests to assess how well end-users can use attributions to assess model reliability. Here we find that the end-users relied, primarily, on model predictions for diagnosing model bugs.

Our findings come with certain limitations and caveats. The bug characterization presented only covers the standard supervised learning pipeline and might not neatly capture bugs that result from a combination of factors. We only focused on feature attributions: however, other methods such as approaches based on 'concept' activation [28], model representation dissection [9], and training point ranking [32, 48, 69] might be more suited to the debugging tasks studied here. Indeed, initial exploration of the 'concept' activation method TCAV and training point ranking based on influence functions suggests that these approaches are promising (See Appendix for analysis). For the human subject experiments, our finding that the participants mostly relied on the labels instead of the feature attributions might be a consequence of the dog breed classification task. It is unclear whether participants would still rely of model predictions for tasks in which they have no expertise or prior knowledge.

The goal of this work is to provide guidance for researchers and practitioners seeking to use feature attributions for model debugging. We hope our findings can serve as a first step towards more rigorous approaches for assessing the utility of explanation methods.

Broader Impact

Predictive models are increasingly being investigated, sometimes legally regulated for deployment in critical settings. Interpretability methods promise to provide insights about how models make decisions. This may increase user trust and provide the evidence needed to ensure that models deployed in mission-critical settings function adequately. The goal of our work is to investigate this literature with a critical eye: can attribution methods signal that there may be issues with the model, data or at test-time setting? We provide both quantitative and qualitative approaches to evaluate many popular attribution methods in order to provide practitioners and researchers with a set of debugging tests which may be used in validation. We hope our work is one of the first of many to bridge the gap between methods developed in academia and practical usage of those methods in the real world.

Acknowledgments and Disclosure of Funding

We thank Hal Abelson, Danny Weitzner, Taylor Reynolds, and Anonymous reviewers for feedback on this work. We are grateful to the MIT Quest for Intelligence initiative for providing cloud computing credits for this work. JA is supported by the Open Philanthropy Fellowship.

References

[1] Julius Adebayo, Justin Gilmer, Ian Goodfellow, and Been Kim. Local explanation methods for deep neural networks lack sensitivity to parameter values. 2018.

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9525–9536, 2018.

[3] Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T. Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. innvestigate neural networks! *CoRR*, abs/1808.04260, 2018. URL http://arxiv.org/abs/1808.04260.

[4] Ahmed Alqaraawi, Martin Schuessler, Philipp Weiß, Enrico Costanza, and Nadia Berthouze. Evaluating saliency map explanations for convolutional neural networks: a user study. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 275–285, 2020.

[5] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140, 07 2015. doi: 10.1371/journal.pone.0130140. URL http://dx.doi.org/10.1371%2Fjournal.pone.0130140.

[6] 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.

[7] Marcus A Badgeley, John R Zech, Luke Oakden-Rayner, Benjamin S Glicksberg, Manway Liu, William Gale, Michael V McConnell, Bethany Percha, Thomas M Snyder, and Joel T Dudley. Deep learning predicts hip fracture using confounding patient and healthcare variables. *NPJ digital medicine*, 2(1):1–10, 2019.

[8] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11 (Jun):1803–1831, 2010.

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

[10] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley. Explainable machine learning in deployment. In Mireille Hildebrandt, Carlos Castillo, Elisa Celis, Salvatore Ruggieri, Linnet Taylor, and Gabriela Zanfir-Fortuna, editors, *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, pages 648–657. ACM, 2020. doi: 10.1145/3351095.3375624. URL https://doi.org/10.1145/3351095.3375624.

[11] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. volume 9887 of *Lecture Notes in Computer Science*, pages 63–71. Springer Berlin / Heidelberg, 2016. doi: 10.1007/978-3-319-44781-0_8.

[12] Gabriel Cadamuro, Ran Gilad-Bachrach, and Xiaojin Zhu. Debugging machine learning models. In *ICML Workshop on Reliable Machine Learning in the Wild*, 2016.

[13] C Cai, E Reif, N Hegde, J Hipp, B Kim, D Smilkov, M Wattenberg, D Viegas, G Corrado, M Stumpe, et al. Human-centered tools for coping with imperfect algorithms during medical decision-making. chi conf. *Hum. Factors Comput. Syst*, 2019.

[14] Giuseppe Carenini, Vibhu O Mittal, and Johanna D Moore. Generating patient-specific interactive natural language explanations. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 5. American Medical Informatics Association, 1994.

[15] Alison Cawsey. Generating interactive explanations. In *AAAI*, pages 86–91, 1991.

[16] Alison Cawsey. User modelling in interactive explanations. *User Modeling and User-Adapted Interaction*, 3(3):221–247, 1993.

[17] Aleksandar Chakarov, Aditya Nori, Sriram Rajamani, Shayak Sen, and Deepak Vijaykeerthy. Debugging machine learning tasks. *arXiv preprint arXiv:1603.07292*, 2016.

[18] Eric Chu, Deb Roy, and Jacob Andreas. Are visual explanations useful? a case study in model-in-the-loop prediction. *arXiv preprint arXiv:2007.12248*, 2020.

[19] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In *Advances in Neural Information Processing Systems*, pages 13567–13578, 2019.

[20] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. 2017.

[21] Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Learning explainable models using attribution priors. *arXiv preprint arXiv:1906.10670*, 2019.

[22] Shi Feng and Jordan Boyd-Graber. What can ai do for me? evaluating machine learning interpretations in cooperative play. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 229–239, 2019.

[23] Amirata Ghorbani, Abubakar Abid, and James Y. Zou. Interpretation of neural networks is fragile. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3681–3688. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33013681. URL https://doi.org/10.1609/aaai.v33i01.33013681.

[24] Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling neural network interpretations via adversarial model manipulation. In *Advances in Neural Information Processing Systems*, pages 2921–2932, 2019.

[25] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250, 2000.

[26] 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.

[27] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, volume 2, 2011.

[28] 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 2673–2682, 2018.

[29] Pieter-Jan Kindermans, Kristof Schütt, Klaus-Robert Müller, and Sven Dähne. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*, 2016.

[30] Pieter-Jan Kindermans, Kristof T. Schütt, Maximilian Alber, Been Kim Klaus-Robert Müller, Dumitru Erhan, and Sven Dähne. Learning how to explain neural networks: Patternnet and patternattribution. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hkn7CBaTW.

[31] Pieter-Jan Kindermans, Kristof T. Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. Learning how to explain neural networks: Patternnet and patternattribution. In *ICLR (Poster)*, 2018. URL https://openreview.net/forum?id=Hkn7CBaTW.

[32] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 2017. URL http://proceedings.mlr.press/v70/koh17a.html.

[33] Maximilian Kohlbrenner, Alexander Bauer, Shinichi Nakajima, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. Towards best practice in explaining neural network decisions with lrp. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, 2020.

[34] Vivian Lai and Chenhao Tan. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 29–38, 2019.

[35] Himabindu Lakkaraju and Osbert Bastani. " how do i fool you?" manipulating user trust via misleading black box explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 79–85, 2020.

[36] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

[37] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4768–4777, 2017.

[38] Aravindh Mahendran and Andrea Vedaldi. Salient deconvolutional networks. In *European Conference on Computer Vision*, pages 120–135. Springer, 2016.

[39] Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg C Corrado, Ara Darzi, et al. International evaluation of an ai system for breast cancer screening. *Nature*, 577(7788):89–94, 2020.

[40] Qingjie Meng, Christian Baumgartner, Matthew Sinclair, James Housden, Martin Rajchl, Alberto Gomez, Benjamin Hou, Nicolas Toussaint, Jeremy Tan, Jacqueline Matthew, et al. Automatic shadow detection in 2d ultrasound. 2018.

[41] Grégoire Montavon, Sebastian Bach, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65: 211–222, 2017. doi: 10.1016/j.patcog.2016.11.008. URL http://dx.doi.org/10.1016/j.patcog.2016.11.008.

[42] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65: 211–222, 2017.

[43] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.

[44] Weili Nie, Yang Zhang, and Ankit Patel. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *ICML*, 2018.

[45] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[46] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[47] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810*, 2018.

[48] Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. Estimating training data influence by tracking gradient descent. *arXiv preprint arXiv:2002.08484*, 2020.

[49] 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. ACM, 2016.

[50] Laura Rieger, Chandan Singh, W James Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. *International Conference on Machine Learning*, 2020.

[51] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 2662–2670. ijcai.org, 2017. doi: 10.24963/ijcai.2017/371. URL https://doi.org/10.24963/ijcai.2017/371.

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

[53] 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, 2017.

[54] Rory Sayres, Ankur Taly, Ehsan Rahimy, Katy Blumer, David Coz, Naama Hammel, Jonathan Krause, Arunachalam Narayanaswamy, Zahra Rastegar, Derek Wu, et al. Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy. *Ophthalmology*, 126(4):552–564, 2019.

[55] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in neural information processing systems*, pages 2503–2511, 2015.

[56] Lloyd S Shapley. A value for n-person games. *The Shapley value*, pages 31–40, 1988.

[57] Hua Shen and Ting-Hao Huang. How useful are the machine-generated interpretations to general users? a human evaluation on guessing the incorrectly predicted labels. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 8, pages 168–172, 2020.

[58] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[59] 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. URL http://arxiv.org/abs/1312.6034.

[60] Leon Sixt, Maximilian Granz, and Tim Landgraf. When explanations lie: Why modified bp attribution fails. *arXiv preprint arXiv:1912.09818*, 2019.

[61] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[62] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[63] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[64] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/sundararajan17a.html.

[65] Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun D. Preece. Sanity checks for saliency metrics. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 6021–6029. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6064.

[66] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[67] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[68] Gezheng Wen, Brenda Rodriguez-Niño, Furkan Y Pecen, David J Vining, Naveen Garg, and Mia K Markey. Comparative study of computational visual attention models on two-dimensional medical images. *Journal of Medical Imaging*, 4(2):025503, 2017.

[69] Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. In *Advances in neural information processing systems*, pages 9291–9301, 2018.

[70] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. In *Advances in Neural Information Processing Systems*, pages 10965–10976, 2019.

[71] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[72] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[73] Zinkevich Martin. Rules of Machine Learning: Best Practices for ML Engineering. http://martin.zinkevich.org/rules_of_ml/rules_of_ml.pdf, 2020. Online; accessed 10 January 2020.