POST HOC EXPLANATIONS MAY BE INEFFECTIVE FOR DETECTING UNKNOWN SPURIOUS CORRELATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Ascertaining that a deep network does not rely on an *unknown spurious signal* as basis for its output, prior to deployment, is crucial in high stakes settings like healthcare. While many post hoc explanation methods have been shown to be useful for some end tasks, theoretical and empirical evidence has also accumulated that show that these methods may not be faithful or useful. This leaves little guidance for a practitioner or a researcher using these methods in their decision process. To address this gap, we investigate whether three classes of post hoc explanations–feature attribution, concept activation, and training point ranking–can alert a practitioner to a model's reliance on *unknown* spurious signals. We test them in two medical domains with plausible spurious signals. In a broad experimental sweep across datasets, models, and spurious signals, we find that the post hoc explanations tested can be used to identify a model's reliance on a visible spurious signal provided the spurious signal is known ahead of time by the practitioner using the explanation method. Otherwise, a search over possible spurious signals and available data is required. This finding casts doubt on the utility of these approaches, in the hands of a practitioner, for detecting a model's reliance on spurious signals.

It is hard to find a needle in a haystack,
it is much harder if you haven't seen a needle before (Pearl).
—Judea Pearl

1 INTRODUCTION

A challenge that precludes the deployment of modern deep neural networks (DNN) in high stakes domains is their tendency to latch onto 'spurious signals'—shortcuts—in the training data (Geirhos et al., 2020). For example, Badgeley et al. (2019) showed that an Inception-V3 model trained to detect hip fracture relied on the scanner type for its classification decision. Deep learning models easily base output predictions on object backgrounds (Xiao et al., 2020), image texture (Geirhos et al., 2018), and skin tone (Stock and Cisse, 2018).

Post hoc model explanation methods—approaches that give insight into the associations that a model has learned—are increasingly used to determine whether a model relies on spurious signals. Ribeiro et al. (2016) used LIME to show an Inception-V3 model's reliance on the snow background for identifying Wolves. Such demonstration and others (Lapuschkin et al., 2019; Rieger et al., 2020) point to post hoc explanation methods as effective tools for the spurious signal detection task. However, these results conflict with evidence that indicates that practitioners (and researchers) struggle to use explanations to identify spurious signals (Chen et al., 2021; Chu et al., 2020; Alqaraawi et al., 2020; Adebayo et al., 2020; Poursabzi-Sangdeh et al., 2018). We seek to resolve this conflict by answering the simple but important question:

*Can post hoc explanations help detect a model's reliance on **unknown** spurious training signal?*

Motivating Example. Consider a machine learning (ML) engineer whose job is to train DNN models to detect knee arthritis from radiographs. She—the engineer—is handed a trained ResNet-50 model, to be deployed in a hospital, that relies on a hospital tag in the radiographs to detect knee arthritis. She

has **no prior knowledge** of the model's reliance on the spurious tags. In this work, our key concern is whether the ML engineer can use post hoc explanations to identify that the model is defective.

1.1 Our Contributions

We address the motivating question above in a two-pronged manner. First, we develop an actionable methodology based on the ability to carefully craft datasets to induce spurious correlation in trained models. Second, we backup this experimental design with a human subject study. Taken together, the takeaway of the work is that: *post hoc explanations can be used to identify a model's reliance on a **visible** spurious signal, **provided** the signal is known ahead of time by the practitioner.* While this conclusion may seem unsurprising, it has important implications for how post hoc explanation methods should be used effectively.

Experimental Design & Performance Measures. We provide an end-to-end experimental design for assessing the effectiveness of an explanation method for detecting a model's reliance on spurious training signals. We define a spurious score that helps quantify the strength of a model's dependence on a training signal. Using carefully crafted semi-synthetic datasets, we are able to train models where the ground-truth expected explanation is known. Additionally, we develop 3 performance measures: i) Known Spurious Signal Detection Measure (K-SSD), ii) Cause-for-Concern Measure (CCM), and iii) a False Alarm Measure (FAM). These measures help characterize different notions of reliability for the spurious signal detection task. We instantiate the proposed design on 3 classes of post hoc explanation types—feature attribution, concept activation, and training point ranking—where we comprehensively assess the performance of these approaches across 3 datasets (2 medical tasks, and dog species classification task), and different model architectures.

When the spurious signal is known, we find that the feature attribution methods tested, and the concept activation importance approach are able to detect visible spurious signals like a text tag and distinctive stripped patterns. However, we find these approaches less effective for non-pronounced signals like background blur. The false alarm measure further indicates that feature attribution methods are susceptible to erroneously indicating dependence on spurious signals.

The cause-for-concern measure quantifies the similarity between explanations of 'normal' inputs derived from spurious and normal models when the spurious signal is unknown. Across the settings considered, we find that the methods tested are unable to conclusively detect model reliance on unknown spurious signals.

Blinded Study. The findings from our empirical assessment question the reliability of the methods tested; however, it might not correlate with utility in the hands of practitioners. To address this issue, we conduct a user study where practitioners are randomly assigned to one of two groups: the first group is told explicitly of potential spurious signals, and the second is not. We consider three different kinds of explanation methods along with a control where only model predictions are shown. We find that when participants are not provided with prior knowledge of the spurious signal, none of the methods tested are effective, in the hands of the participants, for detecting model reliance on spurious signals. More surprisingly, even when the participants had prior knowledge of the spurious signal, we find evidence that only the concept activation approach, for visible spurious signals, is effective. These findings cast doubt on the reliability of current post hoc tools for spurious signal detection.

Guidance. On the basis our analysis, we can provide the following guidance for using the approaches tested, in this work, for detecting model reliance on spurious signals **when the signal of interest is visible**:

- **Feature Attributions**: to confirm that a model is relying on a 'visible' spurious signal, the practitioner needs to obtain attributions for inputs that contain the hypothesized spurious signal, and the attribution should be computed for the output class to which the spurious signal is aligned.
- **Concept Activation**: the spurious concept should be known ahead of time, and tested against the output class to which the concept is aligned.
- **Training Point Ranking**: an input that contains the hypothesized spurious signal of interest should be used at test-time in computing training point ranking.

1.2 Related Work

This paper belongs to a line of work on assessing the effectiveness of post hoc explanations methods (Alqaraawi et al., 2020; Adebayo et al., 2020; Chu et al., 2020; Hooker et al., 2019; Meng et al., 2018; Poursabzi-Sangdeh et al., 2018; Tomsett et al., 2020). Here we focus on directly relevant literature, and defer an extensive discussion of the literature to Section A in the Appendix.

This work departs from previous work in two ways: 1) we focus exclusively on whether these explanations can be used by a practitioner (or researcher) to detect spurious signals that are unknown to her at test-time, and 2) we move beyond sole focus on the feature attribution setting to test concept activation and training point ranking methods.

Han et al. (2020) and Adebayo et al. (2020) find that certain kinds of feature attributions and training point ranking via influence functions are able to detect a model's reliance on spurious signals. However, in their setting, the spurious signal is known ahead of time. More recently, Zhou et al. (2021) conduct an extensive assessment of several feature attribution methods also under the spurious correlation setting, for visible and non-visible artifacts, and find that these class of methods are not effective for non-visible artifacts. Further, they also propose an experimental methodology for controlling model dependence on training set features, which allows them to quantify attribution effectiveness carefully and precisely. Overall, our findings align with theirs; however, we focus, specifically, on the setting where the spurious signal is not known ahead of time. Further, we consider other kinds of post hoc explanation approaches in this work.

Kim et al. (2021) conduct an assessment of several feature attribution methods using a synthetic evaluation framework where the ground-truth explanation is known for simple and complex reasoning tasks. They find that feature attribution methods often attribute irrelevant features even in simple settings, and these methods show high variability across data modalities and tasks. Plumb et al. (2021) introduce a method that searches over a dataset to identify important associations that a model might have learned, detect which of these associations are be spurious, and propose a data augmentation procedure to overcome the reliance.

In a comprehensive assessment, Nguyen et al. (2021) conduct a large-scale user study on lay and expert users to assess the effectiveness of feature attribution methods on image tasks. They find that feature attributions are not more effective than showing end users nearest neighbor training points.

2 Experimental Methodology

In this section, we setup our experimental methodology. We discuss quantitative analysis of post hoc explanations derived from models trained to rely on pre-defined spurious signals, and a blinded user study that measures the ability of users to use the post hoc explanation methods tested to detect model reliance on spurious signals. We discuss the types of spurious signals considered, define a spurious score that allows us to ascertain that a model indeed relies on a signal as basis of its classification decision, and layout performance measures that capture the reliability of the explanation methods. We conclude with an overview of the methods tested, datasets, and models.

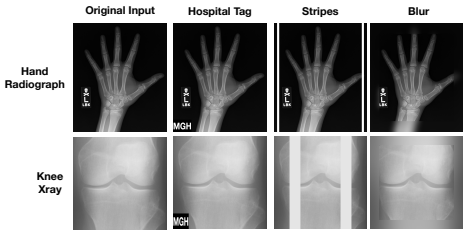

Figure 1: **Overview of Spurious Signals Considered.**

2.1 Experimental Design

Spurious Signals & Score. We consider a spurious signal to be input features that encode for the output but have 'no meaningful connection' to the 'data generating process' (DGP) of the task. A hospital tag present in a hand radiograph is not clinically relevant to the age of the patient. If the tag encodes for the output then it is a spurious signal. Domain expertise is ultimately required for adjudicating that a signal is spurious.

We consider 3 (2 visible and 1 non-visible) kinds of spurious signals (See Figure 1): i) a localized tag; ii) a distinctive stripped pattern; and iii) Gaussian blur applied to the image background. The signals are all spatially localized, so we can easily obtain ground-truth expected explanations.

To induce reliance on spurious signals, we train models on "contaminated" versions of the training set. Given input-label pairs, $\{(x^i, y^i)\}_i^n$, where $x^i \in \mathcal{X}$ and $y^i \in \mathcal{Y}$, we can learn a classifier, f_θ, via empirical risk minimization (ERM) that corresponds to minimizing a loss function, ℓ: $\arg\min_\theta \ \sum_{i=1}^n \ell(x^i, y^i; \theta)$. To contaminate the training set, we apply a spurious contamination function (SCF) to the training set; $\mathrm{SCF} : \mathcal{X} \times \mathcal{Y} \times \mathcal{C} \to \mathcal{S}$, where $\mathcal{C}$ is the spurious signal set and $\mathcal{S}$ is the transformed set. An example of an SCF is a function that pastes an hospital tag onto the bone age radiographs of all pre-puberty individuals in the dataset. To derive models reliant on a spurious signal, $c_i \in \mathcal{C}$, we simply learn a new classifier via ERM on the modified dataset as follows: $\arg\min_\theta \ \sum_{i=1}^n \ell\big(\mathrm{SCF}(x^i, y^i, c_i)\big)$ to obtain θ_{spu}. Contemporary evidence suggests that this approach produces models that easily latch onto the spurious signal (Nagarajan et al., 2020).

We focus on the classification setting, and restrict spurious signals to encode, only, for a single class—the *spurious aligned class*. We measure a model's reliance on the spurious signal via a score.

Definition 2.1. (Spurious Score). Given a spurious signal, c_i, the index of its spurious aligned class, $j \in [k]$, a model, $\theta_{\mathrm{spu}} : \mathbb{R}^d \to \mathbb{R}^k$, where $\arg\max(\theta_{\mathrm{spu}})$ indicates the classifier's predicted class, we define the spurious score as:

$$\mathrm{SC}_{c_i,j}(\theta_{\mathrm{spu}}) := \mathbb{P}_{\{x^i | \theta_{\mathrm{spu}}(x^i) \ != \ j\}}[\arg\max(\theta_{\mathrm{spu}}(\mathrm{SCF}(x^i, y^i, c_i))) = j].$$

Given an input that does not contain the spurious signal, and for which the model's prediction is not the spurious aligned class, the model's spurious score is the probability that the model assigns the input to the spurious aligned class if the spurious signal is added to the input.

Model Conditions. We focus our analysis on two model conditions: i) a 'normal model' for which we can rule out dependence on any of the spurious signals tested across all classes on the basis of the spurious score, and ii) a 'spurious model' for which one of the spurious signals encodes for a particular output class. We empirically estimate the spurious score and term models that have a score above 0.85 for any of the pre-defined signals 'spurious models'. We term a model 'normal'[1] if the spurious score is below 0.1 across all classes and the 3 pre-defined spurious signals.

Spurious Signal Detection Reliability Measures. Equipped with spurious and normal models, we are now able to quantitatively assess the motivating question of this work. We do this by comparing explanations derived from spurious models to those derived from normal models. We can partition the kinds of inputs used for deriving explanations into two: 1) *spurious inputs*—inputs that include the spurious signal and 2) *normal inputs*—inputs do not not contain the spurious signal. Comparing the explanations produced by these two classes of inputs for normal and spurious models, we derive reliability performance measures.

- **Known Spurious Signal Detection Measure (K-SSD)** - measures the similarity of explanations derived from *spurious models on spurious inputs* to the expected ground truth. K-SSD measures method reliability when the spurious signal is known.
- **Cause-for-Concern Measure (CCM)** - measures the similarity of explanations derived from *spurious models for normal inputs* to explanations derived from *normal models for normal inputs*. This measure simulates the setting where a practitioner does not know the spurious signal, and can only inspect explanations for inputs without the signal. If this measure is high, then it is unlikely that such a method alert a practitioner that a spurious model exhibits defects.
- **False Alarm Measure (FAM)** - measures the similarity of explanations derived from *normal models for spurious inputs* to explanations derived from *spurious models for spurious inputs*. If this measure is high, then that approach is more likely to signal to a practitioner that a model is relying on spurious signal when the model does not.

Taken together, these measures provide a comprehensive overview of an explanation method's performance for detecting spurious signals.

[1] We note that it is indeed possible for the dataset to encode for other kinds of spurious signals; however, our analysis can rule out the model dependence on the 3 pre-defined spurious signals that we study in this work.

2.2 BLINDED STUDY

To complement the quantitative setup, we further designed a user study (IRB approved) to assess the ability of end-users to use post hoc explanations to detect a model's reliance on spurious signals.

Task & Setup: The study participants were tasked with assessing model reliability with the aid of model explanations. The participants were randomly assigned to one of two groups: the first group is told explicitly of potential spurious signals, and the second is not. They were asked to rate how likely they are to recommend the model for deployment using a 5 point-likert scale, and a rationale for their decision. Our study design follows that of Adebayo et al. (2020); however, we use a mixed within-subjects and between subjects design for the factors of interest.

Methods: We test SmoothGrad, TCAV (a concept activation method), a training point ranking method, and a Control setting with no explanations.

2.3 EXPLANATION METHODS, DATA, & MODELS

Here we give an overview of the explanation methods, datasets, and models. We present a discussion of how these methods are used in practice in Appendix Section C.

Feature Attributions assign a relevance score for each dimension of an input towards an output. We consider: **Input-Gradient, SmoothGrad, Integrated Gradients (IG), and Guided Backprop (GBP)**.

Concept-Based (**(Bau et al., 2017; Kim et al., 2018)**) approaches measure the dependence of a DNN's prediction on user-defined features—termed concepts. We select TCAV as the approach to assess in this class (Kim et al., 2018).

Training Point Ranking via Influence Functions (Koh and Liang, 2017). This approach ranks the training samples in order of importance/influence on the loss (or prediction) of a test example.

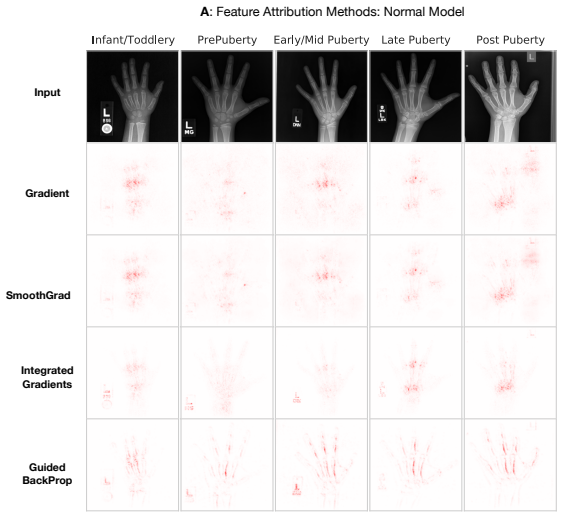

Figure 2: **Feature Attributions.** Here we show 5 different inputs, one for each bone age category, and the corresponding feature attribution maps for 4 different methods for a CNN model trained to predict bone age category.

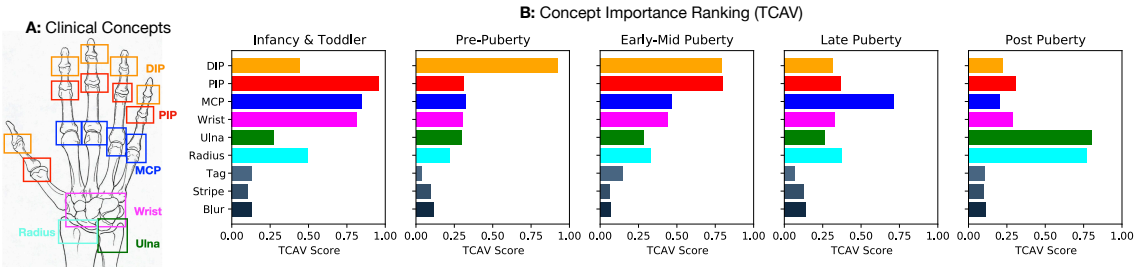

Figure 3: **Concept Importance Methods for a Normal (non-spurious) model.** Here we show the TCAV score for all clinical concepts as well as the spurious concepts for a normal model that was confirmed to not rely on the spurious signals.

Models, Datasets, & Tasks. We consider two medical datasets: Hand (Halabi et al., 2019) and Knee radiographs (Chen et al., 2019) and a dog species classification task. We consider a small DNN (6 layers, 5 of which are the traditional conv-relu-batchnorm-maxpool combination) inspired by the CBR-Tiny architecture of Raghu et al. (2019) and a ResNet-50 model. We results report for the small DNN model in the paper since we find correspondence for the two models (See Appendix for additional details).

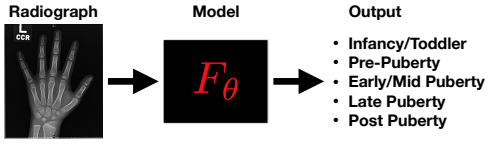

Figure 4: Illustration of the Bone Age Task.

3 Feature Attributions

In this section, we give additional overview of the experimental setting considered and then present results on whether feature attributions are effective for detecting unknown spurious correlation.

Setup. We consider 3 kinds of spurious signals which we term: *Tag* for the 'MGH' hospital tag added to the pre-puberty class, *Stripe* for the paired vertical stripped signal, and *Blur* for the background blur. Given these signals, we then compute the three performance measures of interest: K-SSD, CCM, and FAM. K-SSD indicates a method's reliability when the spurious signal is known, CCM when the signal is not known and the practitioner only has access to inputs that don't encode the spurious signal. Lastly, FAM indicates the susceptibility of a method to false positives. Across all the comparisons, we measure feature attribution similarity with the SSIM metric following (Adebayo et al., 2018).

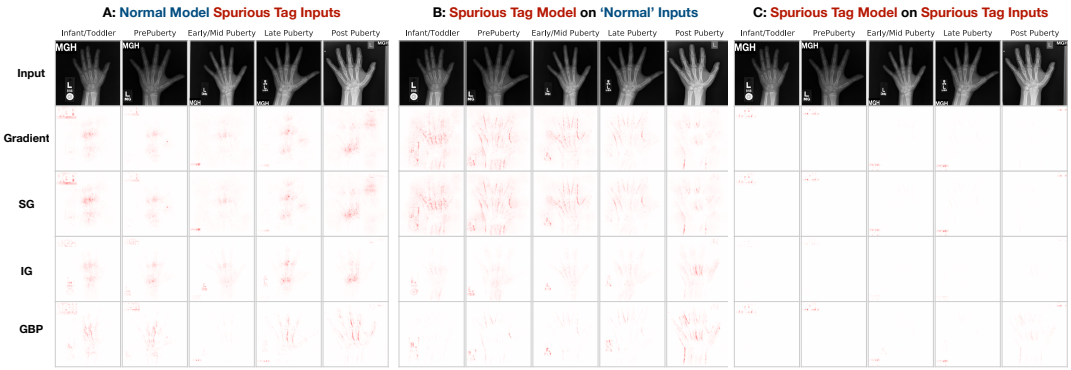

Figure 5: **Detecting Spurious *Tag*.** Here we show in A) Feature attributions for 5 different inputs across the four feature attribution methods with a normal model but with spurious Tag inputs; B) Feature attributions on the same 5 inputs as in (A), but **without** spurious Tag inputs with a model that has learned a spurious alignment between Pre-Puberty and Tag; C) Feature attributions on the same 5 inputs as in (A), but **with** the spurious Tag inputs with a model that has learned a spurious alignment between Pre-Puberty and Tag.

Table 1: Performance metrics for each attribution method across tasks for the Tag Setting.

Method	Bone Age				Knee				Dog Species			
	Grad	SG	IG	GBP	Grad	SG	IG	GBP	Grad	SG	IG	GBP
K-SSD	0.65	0.66	0.67	0.81	0.51	0.49	0.47	0.76	0.71	0.76	0.79	0.88
CCM	0.37	0.39	0.35	0.75	0.32	0.33	0.35	0.66	0.42	0.41	0.39	0.64
FAM	0.51	0.55	0.53	0.68	0.46	0.47	0.45	0.69	0.59	0.64	0.68	0.73

Results. We show performance measures for all the feature attribution methods tested for the Tag and Blur settings in Tables 11 & 12. For the tag setting, the attribution methods are indeed able to

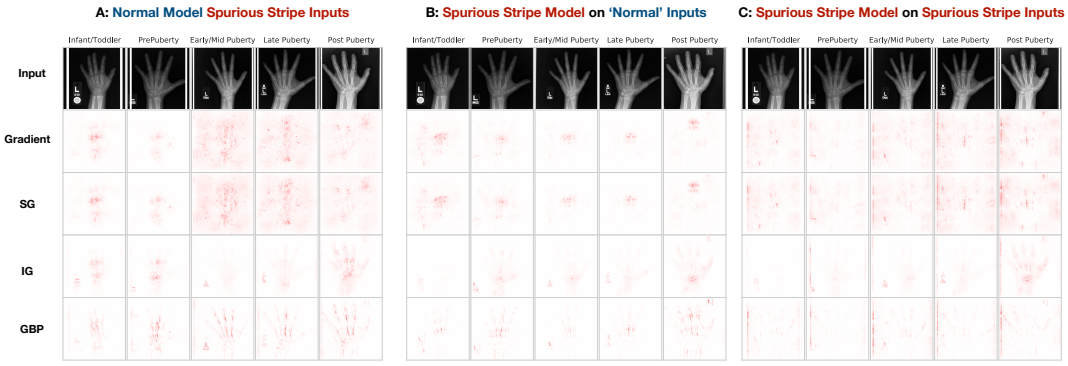

Figure 6: **Detecting Spurious *Stripe*.** Here we show in A) Feature attributions for 5 different inputs across the four feature attribution methods with a normal model but with spurious Stripes inputs; B) Feature attributions on the same 5 inputs as in (A), but **without** the spurious Stripe with a model that has learned a spurious alignment between Pre-Puberty and Stripe; C) Feature attributions on the same 5 inputs as in (A), but **with** the spurious Stripe with a model that has learned a spurious alignment between Pre-Puberty and Stripe.

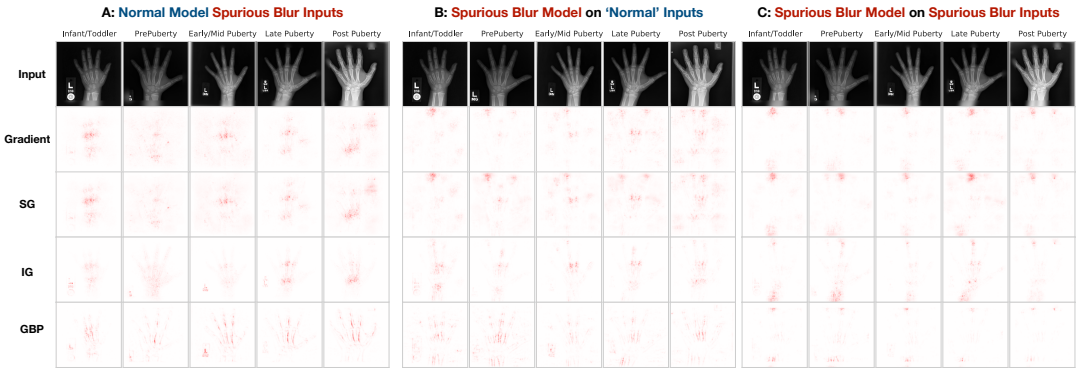

Figure 7: **Detecting Spurious *Blur*.** Here we show in A) Feature attributions for 5 different inputs across the four feature attribution methods with a normal model but with spurious Blur inputs; B) Feature attributions on the same 5 inputs as in (A), but **without** the spurious Blur with a model that has learned a spurious alignment between Pre-Puberty and Blur; C) Feature attributions on the same 5 inputs as in (A), but **with** the spurious blur with a model that has learned a spurious alignment between Pre-Puberty and Blur.

Table 2: Performance metrics for each attribution method across tasks for the Blur Setting.

Method	Bone Age				Knee				Dog Species			
	Grad	SG	IG	GBP	Grad	SG	IG	GBP	Grad	SG	IG	GBP
K-SSD	0.21	0.20	0.19	0.13	0.13	0.18	0.17	0.31	0.29	0.30	0.31	0.35
CCM	0.28	0.29	0.24	0.64	0.23	0.22	0.27	0.67	0.38	0.33	0.35	0.71
FAM	0.48	0.49	0.47	0.51	0.36	0.38	0.33	0.58	0.55	0.56	0.47	0.73

attribute to the visible spurious signal and the K-SSD measure indicates this finding. However, the performance in this setting is not perfect and other features are also attributed. Alternatively, we also find that these methods also seem to attribute to the spurious signal (FAM > 0.5) even when the signal is not being relied on by the model. This indicates downstream reliability challenges. We observe similar findings for the strip setting as well. As indicated, the CCM measure further indicates that these methods do not indicate presence of spurious signals when the signal is unknown for both

the Tag and Stripe signals. This finding, however, reverses for the non-visible blur spurious signal. Across all measures, we find that all methods struggle to reliably indicate that spurious models are reliant on the blur signal.

4 Concept Activation Importance

We find that concept methods can indicate a model's reliance on Tag and Stripe signals when known. However, the approach struggles to detect Blur signal even when known. As is the case with feature attributions, when a spurious signal is not explicitly tested for, our significance tests indicate that reliance cannot be detected in the non-spurious concepts available.

Overview & Setup. In this setting, we compute the 3 performance metrics of interest: K-SSD, CCM, and FAM. To measure comparison between two concept rankings, we use a Kolmogorov-Smirnoff (KS) test comparing two distributions where the null hypothesis is that the two distributions are identical; we set significance level to be 0.05.

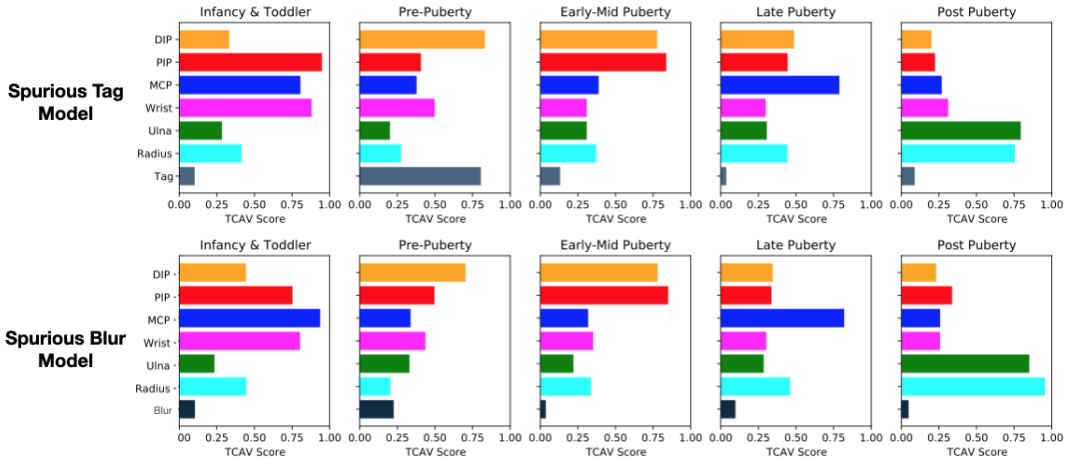

Figure 8: **Concept Results for Tag and Blur Models.** TCAV scores for a model reliant on the spurious Tag and a model reliant on Blur.

Result. In Figure 8, we show TCAV scores for each bone age class for both the spurious tag and the blur models. We include the results for the Stripe setting in the Appendix since it is similar to the Tag results. For the K-SSD metric, the KS-Test rejects that the concept rankings are similar for Tag and Stripe. However, we find that the opposite is the case for the blur signal. This suggests that concept rankings can help detect reliance on the visible signals but not non-visible signals when the spurious signals are unknown. However, for the CCM metric, we are unable to reject the hypothesis that the distribution of concept rankings are not similar for all spurious signals. This finding suggests that when the spurious signal is unknown, and we only compare the distributions of known (non-spurious) concepts for a normal model and a spurious model, there is high similarity. Lastly, a difference in means test for the FAM measure indicates that the normal models do not rely on the spurious signals. Overall, this finding suggests that TCAV is less susceptible to false positives.

5 Training Point Ranking

Overview & Setup. We now describe our empirical findings for the training point ranking via influence functions approach. Given a test input, we can rank all training inputs by their influence on the loss of a given test example. Here we present results for the case where the spurious signal is aligned with the Pre-Puberty class. Recently, Hanawa et al. (2020) introduced the Identical class metric', which is the fraction of the top training inputs for a given test example that belong to the

Method	B-Normal	NB-Normal	B-Spurious	NB-Spurious
SmoothGrad	4	4	3	3
TCAV	4	3	3	2
Influence	3	3	3	3
Control	4	3	4	4

Table 3: **Blinded User Study**. Here we report the median recommendation for each condition across all explanation types and control. This median is derived from user provided responses assessing model reliability. The likert scale is as follows, 1: Definitely Not to 5: Definitely. Participants were instructed to select 'Definitely' if they deemed the dog-breed classification model reliable. B indicates Blinded, and NB indicates Not-Blinded.

same class as the true class of the test example in question. We use this metric as a measure of comparison.

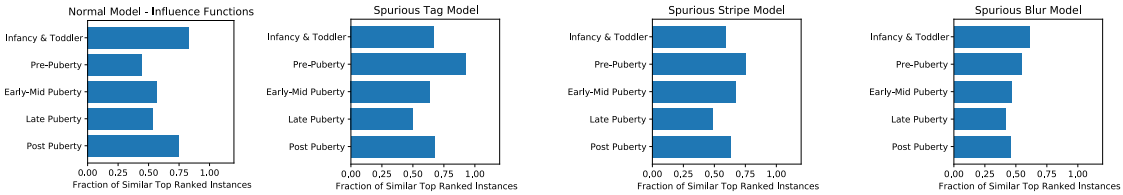

Figure 9: **Influence Function Results.** Here we show the *identical class metric (ICM)* (Hanawa et al., 2020) for a normal model and models that rely on the three spurious signals tested in this work. ICM measures what fractions of the top training inputs for a given test example belong to the same class as the true class of the test example in question.

Results. The main take away across is that given a known spurious signal, the fraction of top ranked training spurious signal inputs increases with a spurious model across all of the spurious signals. While this might seem encouraging, we note that such increase actually indicates that the ICM metric might provide illusory confidence in a spurious model. Ultimately, a critical requirement here is knowing what the spurious signal ahead of time and to be able to select the right inputs to inspect.

6 Discussion & Conclusion

We now discuss the blinded study, an overview of limitations of this work, and then conclude.

Blinded Study. The median recommendations selected by participants (50 in total) is reported for each explanation-model condition in Table 3. We observe that when blinded, in none of the methods do participants conclusively reject spurious models. Perhaps more surprising, when the participants were not blinded, we see that only participants using the TCAV approach rejected a spurious model.

Limitations. We have focused exclusively on DNNs trained on image tasks; hence, it is unclear if our findings will generalize to other modalities like text or time-series data. While we considered a diverse set of methods, the literature on post hoc explanations is quite vast, so undoubtedly there exist methods that do not fit neatly into the 3 explanation classes that we explored.

Conclusion. DNNs trained on image datasets can naturally rely on spurious training signals (Geirhos et al., 2020). Discovering this reliance is crucial in consequential settings like medical imaging. Post hoc explanations methods are a promising direction towards detecting such reliance; however, their effectiveness is currently under question. We investigated whether 3 classes of post hoc explanations are effective for detecting a model's reliance on spurious training signals. We find that the 3 classes of post hoc explanations tested are only sometimes able to diagnose the spurious training signal even if they are used to explicitly test for model dependence on these signals. Consequently, our findings calls for, potentially, a completely different paradigm of methods that are designed to address the important and challenging question of detecting spurious training signals.

REFERENCES

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.

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. *arXiv preprint arXiv:2011.05429*, 2020.

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.

Christopher Anders, Plamen Pasliev, Ann-Kathrin Dombrowski, Klaus-Robert Müller, and Pan Kessel. Fairwashing explanations with off-manifold detergent. In *International Conference on Machine Learning*, pages 314–323. PMLR, 2020.

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.

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.

Mohammad Taha Bahadori and David E Heckerman. Debiasing concept-based explanations with causal analysis. 2021.

Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*, 2020.

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.

Pingjun Chen, Linlin Gao, Xiaoshuang Shi, Kyle Allen, and Yang Lin. Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss. *Computerized Medical Imaging and Graphics*, 75:84–92, 2019. doi: https://doi.org/10.1016/j.compmedimag.2019.06.002.

Valerie Chen, Jeffrey Li, Joon Sik Kim, Gregory Plumb, and Ameet Talwalkar. Towards connecting use cases and methods in interpretable machine learning. *arXiv preprint arXiv:2103.06254*, 2021.

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

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.

Alex J DeGrave, Joseph D Janizek, and Su-In Lee. Ai for radiographic covid-19 detection selects shortcuts over signal. *medRxiv*, 2020.

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.

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

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.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

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, 2019a. doi: 10.1609/aaai.v33i01.33013681. URL https://doi.org/10.1609/aaai.v33i01.33013681.

Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. *arXiv preprint arXiv:1902.03129*, 2019b.

Han Guo, Nazneen Fatema Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. Fastif: Scalable influence functions for efficient model interpretation and debugging. *arXiv preprint arXiv:2012.15781*, 2020.

Safwan S Halabi, Luciano M Prevedello, Jayashree Kalpathy-Cramer, Artem B Mamonov, Alexander Bilbily, Mark Cicero, Ian Pan, Lucas Araújo Pereira, Rafael Teixeira Sousa, Nitamar Abdala, et al. The rsna pediatric bone age machine learning challenge. *Radiology*, 290(2):498–503, 2019.

Xiaochuang Han, Byron C Wallace, and Yulia Tsvetkov. Explaining black box predictions and unveiling data artifacts through influence functions. *arXiv preprint arXiv:2005.06676*, 2020.

Kazuaki Hanawa, Sho Yokoi, Satoshi Hara, and Kentaro Inui. Evaluation of similarity-based explanations. *arXiv preprint arXiv:2006.04528*, 2020.

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.

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.

Dmitry Kazhdan, Botty Dimanov, Mateja Jamnik, Pietro Liò, and Adrian Weller. Now you see me (cme): Concept-based model extraction. *arXiv preprint arXiv:2010.13233*, 2020.

Fereshte Khani and Percy Liang. Removing spurious features can hurt accuracy and affect groups disproportionately. *arXiv preprint arXiv:2012.04104*, 2020.

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.

Joon Sik Kim, Gregory Plumb, and Ameet Talwalkar. Sanity simulations for saliency methods. *arXiv preprint arXiv:2105.06506*, 2021.

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.

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

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.

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.

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

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.

Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.

Giang Nguyen, Daeyoung Kim, and Anh Nguyen. The effectiveness of feature attribution methods and its correlation with automatic evaluation scores. *arXiv preprint arXiv:2105.14944*, 2021.

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

Judea Pearl. URL http://causality.cs.ucla.edu/blog/.

Gregory Plumb, Marco Tulio Ribeiro, and Ameet Talwalkar. Finding and fixing spurious patterns with explanations. *arXiv preprint arXiv:2106.02112*, 2021.

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.

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

Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *arXiv preprint arXiv:1902.07208*, 2019.

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.

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.

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.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020.

Harshay Shah, Prateek Jain, and Praneeth Netrapalli. Do input gradients highlight discriminative features? *arXiv preprint arXiv:2102.12781*, 2021.

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.

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.

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

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.

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

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

Suraj Srinivas and Francois Fleuret. Rethinking the role of gradient-based attribution methods for model interpretability. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=dYeAHXnpWJ4.

Pierre Stock and Moustapha Cisse. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 498–512, 2018.

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.

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.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

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.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. Do feature attribution methods correctly attribute features? *arXiv preprint arXiv:2104.14403*, 2021.

Appendix

A EXTENDED RELATED WORK.

Overview of Recent work The recent work of Adebayo et al. (2020) presents debugging tests for assessing feature attribution methods. The spurious correlation setting that we consider here fits under their framework. However, they only consider feature attribution methods. Here we extend this analysis to concept and training point ranking methods. Critically, Adebayo et al. (2020) show that feature attributions are able to identify spurious training signals. We make a similar finding in this work; however, we further demonstrate this finding for concept and training point ranking methods. Our work takes important departures from theirs: 1) we explain the source of this phenomenon, and 2) we demonstrate that naive application of these methods might be unable to detect spurious correlation in practice. Adebayo et al. (2020) assume the spurious correlation training bug is known, a priori; however, here we demonstrate that the more challenging task is identifying the spurious signal in the first place.

More recently, Han et al. (2020) demonstrate that training point ranking via influence functions is able to identify the dependence of an NLP model on dataset artifacts. In addition, they show correspondence between the insights observed with the input-gradient feature attribution and the training point ranking. Along similar lines, Guo et al. (2020) present fast approximations for computing the training point ranking for a test point. In addition, they show how to identify and correct model errors in a natural language task. Similar to the distinctions that we note with the work by Adebayo et al. (2020) above, here, they also assume that the spurious signal being identified is known a priori.

Post hoc explanations, more generally, have been shown to be able to identify a model's reliance on spurious training signals (Ribeiro et al., 2016; Meng et al., 2018; Lapuschkin et al., 2019; DeGrave et al., 2020; Ross et al., 2017). Recent work by Rieger et al. (2020) showed that regularizing model attributions during training can help lead to models that avoid spurious correlation and enable improved debugging by experts. Similarly, Erion et al. (2019) show that regularizing the expected gradient attribution during training confers similar benefits. Koh and Liang (2017) used influence functions to identify domain shift. Kim et al. (2018) also perform a user study to understand if attribution methods can be used for catch spurious correlation.

However, similar methods have also been shown to struggle in the hands of end-users for diagnosing model errors (Alqaraawi et al., 2020; Adebayo et al., 2020). This contradiction reflects the challenge that we explore in this work. Often, post hoc explanations have been shown to be effective for identifying spurious signals that were suspected or known a priori; however, these methods seem to struggle when confronted with the task of identifying an unexpected spurious signal.

Increasingly, insights into why overparametrized DNNs rely on spurious training set signals is starting to be theoretically and empirically analyzed (Sagawa et al., 2019; 2020; Khani and Liang, 2020; Nagarajan et al., 2020), yet it is still unclear how to reliably detect that a model is relying on such signals prior to model deployment.

Assessing whether a post hoc explanation approach is faithful to the underlying model being explained has been addressed in recent works, yet this challenge remains elusive (Hooker et al., 2019; Tomsett et al., 2020). Generally, the class of approaches that modify backpropagation with positive aggregation have been shown to be invariant to the higher layer parameters of a DNN (Mahendran and Vedaldi, 2016; Nie et al., 2018; Adebayo et al., 2018; Sixt et al., 2019). In an intriguing demonstration, Srinivas and Fleuret (2021) show that the input-gradient, a key feature attribution primitive, might not capture discriminative signals about input sensitivity. Instead they show that input-gradient likely captures the ability of the model to be able to generate class-conditional inputs.

User studies are typically the classic approach for evaluating the effective of an explanation (Doshi-Velez and Kim, 2017). Poursabzi-Sangdeh et al. (2018) manipulate the features of a predictive model trained to predict housing prices to assess how well end-users can identify model mistakes. Their results indicate that users found it challenging to debug these linear models with the model

coefficients. Recent work by Chu et al. (2020) and Shen and Huang (2020) has shown similar results in the DNN setting as well. Alqaraawi et al. (2020) find that the LRP explanation method 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.

Post hoc explanations have been shown to be fragile and very easily manipulated Ghorbani et al. (2019a); Heo et al. (2019); Dombrowski et al. (2019); Anders et al. (2020); Slack et al. (2020); Lakkaraju and Bastani (2020). Our work tackles a difference concern: whether they are suitable for detecting unexpected spurious training set signals.

Yeh et al. (2020) and Ghorbani et al. (2019b) both present approaches to automatically discover concepts and quantify a model's dependence on these concepts. These approaches are promising directions for addressing the challenge we identify in this work. Ghorbani et al. (2019b)'s approach, ACE, segments input images and clusters to discover inherent clusters. We present some analysis on this approach in the appendix. Critically, this approach would identify spurious signals that the underlying segmentation algorithm can discover such the image tag, but not the blur or other visually imperceptible features. Koh et al. (2020) and Chen et al. (2020) present approaches that learn DNN models whose features inherently map onto concepts of interest. In this work, we assume the model is given, focusing on post hoc explanations for models that are not inherently interpretable. Recent work at the intersection of causal inference and explanations might also open up avenues to help reveal unexpected confounding, some of which could be unknown spurious signals. Along this line, (Bahadori and Heckerman, 2021) present an instrumental variable approach for debiasing concept based explanations that might be confounded. Kazhdan et al. (2020) present CME, an approach identify the important concepts that can help improve a model's performance.

We rely on Guo et al. (2020) for fast approximations for computing the training point ranking for a test point. In addition, they show how to identify and correct model errors in a natural language task.However, Basu et al. (2020) show influence functions for DNNs are fragile and perhaps inaccurate for deeper networks.

Other recent work has cast doubt on the utility of trying to explain 'traditionally' trained deep network models. For example, Srinivas and Fleuret (2021) show that the input-gradient might not reflect the discriminative capabilities of a DNN, but instead encode for an implicit density model. More recently, Shah et al. (2021) show that the input and loss gradients of traditionally trained models *do not* indicate the importance features that a DNN model relies on for its output—a phenomenon they term feature inversion. Further they show that adversarially trained models do not exhibit feature inversion. Taken together these results might explain some of the counter intuitive findings that we observe even when the spurious signal is known since we only consider non-adversarially trained models in this work. Along a different direction, post hoc explanations have been shown to be fragile and very easily manipulated Ghorbani et al. (2019a); Heo et al. (2019); Dombrowski et al. (2019); Anders et al. (2020); Slack et al. (2020); Lakkaraju and Bastani (2020).

B Detailed Overview of Explanation Methods

In this section, we provide additional implementation details for the explanation methods that we consider in this work. To start with the model setup: let's say we are given input-output pairs, $\{x_i, y_i\}_i^n$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$; and a classifier's goal is to learn a function, $f_\theta : \mathcal{X} \to \mathcal{Y}$ that generalizes to new inputs via empirical risk minimization (ERM). In this work, we assume that f_θ is an over-parametrized deep neural network (DNN) trained on image data for classification (C classes).

We plan to release our larger codebase with the paper; however, in lieu of this we include a zip folder that includes representative implementations. For each of the explanation methods described here, we implement them from scratch and then compare to open-source libraries that also implement these methods. We found that our results for both settings is comparable.

Feature Attributions. 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_θ, output, $f_k(x)$, to an attribution map, $M_{x_i} \in \mathbb{R}^d$. The class of feature attribution methods is large, so in this work we pick: Input Gradient, SmoothGrad, Integrated Gradients, and Guided Backprop. We choose these approaches since they were the top-ranked methods tested under the spurious correlation setting of Adebayo et al. (2020).

1. **The *Input-Gradient (Gradient)*** Simonyan et al. (2014); Baehrens et al. (2010) map, $|\nabla_{x_i} F_i(x_i)|$, is a key primitive upon which several other methods are based.
2. ***SmoothGrad*** **Smilkov et al. (2017)** corresponds to the average of noisy input gradients: $M_{\text{sg}}(x) = \frac{1}{N}\sum_{i=1}^{N} \nabla_{x_i} F_i(x_i + n_i)$ where n_i is sampled according to a random Gaussian noise. We considered 50 noisy inputs, selected the standard deviation of the noise to be $0.15 *$ input range. Here input range refers to the difference between the maximum and minimum value in the input.
3. ***Integrated Gradients*** (**Sundararajan et al. (2017)**) sums input gradients along an interpolation path from the "baseline input", $\bar{x}$, to x_i: $M_{\text{IntGrad}}(x_i) = (x_i - \bar{x}) \times \int_0^1 \frac{\partial S(\bar{x}+\alpha(x_i-\bar{x}))}{\partial x_i} d\alpha$. For integrated gradients we set the baseline input to be a vector containing the minimum possible values across all input dimensions. This often corresponds an all-black image. The choice of a baseline for IntGrad is not without controversy; however, we follow this setup since it is one of the more widely used baselines for image data.
4. ***Guided Backpropagation (GBP)*** ***Springenberg et al. (2014)*** modifies the backpropagation process at ReLU units in DNNs. Let, $a = \max(0, b)$, then for a backward pass, $\frac{\partial l}{\partial s} = 1_{s>0}\frac{\partial l}{\partial b}$, where l is a function of s. For GBP, $\frac{\partial l}{\partial s} = 1_{s>0} 1_{\frac{\partial l}{\partial s}>0}\frac{\partial l}{\partial b}$.

Feature Attributions: Implementation. We implement all of these methods from scratch in the PyTorch framework and also compare our implementations to the output of the Captum (PyTorch) and the Innvestigate framework (in Keras). We were able to confirm a correspondence between these outputs. We include the innvestigate script here.

Concept-Based Approaches. We now discuss additional implementation details of our concept based approach. We select the TCAV approach to quantify the sensitivity of a DNN model's class score to user provided inputs represent a particular class. Given hidden representations, h_l, from a particular layer of a DNN for for images belonging to concept class C. We can derive the sensitivity score as: $\nabla h_{l,k}(f_l(x)).\theta_c^l$. The previous expression indicates the sensitivity of the class score (logit) for class k to inputs indicating concept, C, given hidden representations from layer l from the DNN f. The concept vector, θ_c^l, typically corresponds to a the weights of a linear classifier trained to separate the images for a particular concept class from or images.

For completeness, we show in Figure 20 an overview of the clinical concepts that we consider in this work. These are the representative clinical attributes that a radiologist would inspect to ascertain the bone age of a particular input. These concepts are: DIP, PIP, MCP, Radius, Ulna, and Wrist.

Concept Implementation. To compute the TCAV score for each concept, we collect representations from all hidden 'layers' of the model and train linear models to obtains the concept vector for the corresponding attribute. We then compute the class sensitivity score for each concept attribute. We train the linear model 100 times and perform statistical significance testing in order to mitigate the case where a spurious concept is selected. For each concept class, we use 325 images that part of the training, validation, or test sets. These new set of images were annotated by a board certified radiology with the clinical bone age regions (MCP, PIP, DIP etc) that we chose.

Influence Functions for Training Point Ranking. The final kind of interpretation that we consider is training point ranking via influence functions. In the case of training point ranking via influence functions, we rank the training samples, in terms of 'influence', on the loss of a test example. Specifically, if we up-weighted a training point and retrained the model, then by how much would the loss on a given test example change? Koh and Liang (2017) analytically derive the analytically formulas for computing this quantity. Given a test point, x_t, the influence of a training point, x_i, on the test loss is: $I(x_t, x_i) = -\nabla_\theta \ell(x_t, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta \ell(x_i, \hat{\theta})$, where H is the empirical Hessian of the loss.

Estimating the influence requires computing hessian-vector products, so it can be difficult to scale to model with large number of parameters, and recent work has shown that influence estimate for test points for deep networks can be inaccurate due to non-convexity (Basu et al., 2020). Consequently, we estimate influence on a linear model student network trained to mimic the original DNN. We empirically verify that the predictions of the student network seem to mimic the original DNN.

Implementation Details. There are two other training point ranking methods that we also consider in this work (Yeh et al., 2018; Pruthi et al., 2020). For 150 inputs in the test set, we compare the Spearman rank correlation of the training point due to Influence Functions to these other two approaches. We obtain mean values of 0.88, and 0.76 respectively, which suggests high similarity amongst these approaches. Ultimately, we chose to present the main results in the draft for the training point ranking due to influence functions approach.

We rely on the fast influence heuristic of (Guo et al., 2020) to speed up the influence ranking computations. We were able to obtain a 5-10X speed in doing so. In addition, we trained a student multi-class logistic regression model to mimic the original model for each model we want to compute influence for. Here for all training points, we collect embeddings across all layers and pass these embedding through a random projection to obtain a 1000-dimensional approximation. We then train linear models to mimic the original deep network using these features. The correlation between the output of the student models and the original teacher models was found to be 0.87. Ultimately, we went with the fast influence implementation of (Guo et al., 2020).

C How are the approaches tested used in practice?

Feature Attributions To understand or debug a model, a practitioner would have to inspect, one sample at a time, the attribution of a collection of inputs. In inspecting these attributions, the practitioner can then form an hypothesis about the behavior of the model on certain inputs. Along these lines, to detect a model's reliance on spurious signals with feature attributions, one of the following must occur: i) The practitioner should inspect attributions for inputs that contain the spurious signal and notice that the spurious signal constitute the key feature for a model's output decision or ii) The practitioner should inspect attributions for inputs that do not contain the spurious signal and notice 'an issue' with these attributions. In addition, the output decision to be explained has to also be the class for which the spurious signal encodes for. For example, in our experiments the hospital tag encodes for the pre-puberty class. These two settings are exactly what we measure with the SSIM metric in our analysis for feature attributions.

Concept Activation these classes of approaches measure the dependence of a particular class to a user provided concept. For example, one could measure the pre-puberty class' dependence on the hospital tag concept. As part of this approach, the hospital tag is user defined, and it is up to the practitioner to decide whether to test this concept. Other concepts include low frequency signals, patches of the image, color dimensions, or even any conceivable high-level concept the practitioner is interested in. For example, in the bone-age model, the concepts DIP, PIP, MCP, Wrist etc, are user defined clinical concepts that we chose and which are well specified for this setting.

For each concept of interest, the practitioner collects input examples that have this concept. For example, to test a model's dependence on a hospital tag, we collect images that all have hospital tags in them. Given this collection of examples, the TCAV approach can then be used to calculate a score, TCAV score, that corresponds to the sensitivity of a particular output class to the input concept. The key insight that makes the TCAV approach susceptible to the limitations we point out are 1) it is up to the practitioner to decide which concept to test, and 2) our findings suggests that unless the spurious signal is explicitly tested, the TCAV scores for other concepts do not indicate that a model is reliant on a spurious signal. We note that even though we show all the concept scores in a single bar chart, these concepts are actually tested independently.

Training Point Ranking These class of approaches ranks all training points by influence on the test loss of an input. Qualitatively, if a training point has a high influence on the test loss of an input, it means that if the model were re-trained without that training point, the test loss of that test point would change significantly. Intuitively, the most highly ranked training points for a given test input should also be points for which the model relies on semantically similar features. For example, one would expect the most highly ranked training points for a pre-puberty test-input to be other pre-puberty inputs as well (assuming a model has learned the right semantic features)

To use this approach to identify spurious signals, one would have to inspect an input that contains the spurious signal and further notice that the top ranked training inputs also include the training signal. Specifically, to test that a model is relying on the hospital tag to identify pre-puberty inputs

one would have to use a test-input that has a hospital tag and then notice that all the top ranked inputs also have hospital tags. The crux of our argument is that the choice and decision to use a test-input with an hospital tag is the critical choice underlying whether the training point ranking methods can be used to effectively detect a model's reliance on spurious signals. One measure of reliability for an influence function approach is the ICM metric that we compute in the paper.

Taken together, in the scenario where a practitioner is handed a model and asked to assess whether it relies on any spurious signal, detecting a model's reliance spurious signals requires that the practitioner know which specific signals to test ahead of time.

D MODEL TRAINING & DATASET PROCESSING

Bone Age Dataset We consider the high stakes task of predicting the bone age category from a radiograph to one of five classes based on age: Infancy/Toddler, Pre-Puberty, Early/MiD Puberty, Late Puberty, and Post Puberty. This task is one that is routinely performed by radiologists and as been previously studied with a variety of DNN. The dataset we use is derived from the Pediatric Bone Age Machine learning challenge conducted by the radiological society of North America in 2017 Halabi et al. (2019). The dataset consists of 12282 training, 1425 validation, and 200 test samples. We resize all images to (299 by 299) grayscale images for model training. We note here that the training, validation, and test set splits correspond to similar splits used for the competition, so we retain this split.

Knee Dataset We also consider the high stakes task of predicting the Kellgren and Lawrence (KL) grade of osteoarthritis based on Knee Xrays. The KL grade ranges from the integers 0 to 4; 5 classes, so we treat it as a classification task. The Knee Radiographs are obtained from the open source release by Chen et al. (2019) and consists of 5778 training samples, 826 validation samples, and 1656 test samples. Again here, we follow Chen et al. (2019) data splits. The images were resized to be 299 by 299 pixels.

Dog Dataset The third dataset that we use in this work is the dog species classification dataset that is a combination of Stanford dogs dataset Khosla et al. (2011) and the Oxford Cats and Dogs datasets Parkhi et al. (2012b) following Adebayo et al. (2020). Similarly, we restrict our attention to 10 dog classes: Beagle, Boxer, Chihuahua, Newfoundland, Saint Bernard, Pugs, Pomeranian, Great Pyrenees, Yorkshire Terrier, Wheaten Terrier. Instead of the background spurious signal ofAdebayo et al. (2020), we focus instead on the three spurious signals tested in this work.

Models We consider two different kinds of models: i) a small vanilla DNN based on Raghu et al. (2019), and a Resnet-50 model. The small DNN consists of: conv-relu-batchnorm-maxpooling operation successively, and two fully connected layers at the end. All convolutional kernels have stride 1, and kernel size 5. We train this model with SGD with momentum (set to 0.9) and an initial learning rate of 0.01. We use a learning rate scheduler that decays the learning rate every 10 epochs by $\gamma = 0.1$.

Hyper-Parameter Tuning for Model Training We used the Ray Tune library for hyper-parameter tuning of all the models used in this work. For the ResNet-50, we tuned with Ray, but the best performing models retained the default parameter settings. In the case of the Small DNN model, we tune the batch size, and learning rate with the validation set.

Compute We perform all of our experiments on a VM with 60GB of RAM and a k80 GPU on Google Cloud.

A Note about random seeds and Model Runs. We train 5 models each (different random seeds) for each category and in the following tables the average performance metrics for these models. We found that the standard error of the mean for typically between $0.01 - 0.05$, so we omit these in the tables to improve readability.

Model	Normal Training Set	Normal Validation Set	Normal Test Set	Tag Training Set	Tag Validation Set	Stripe Training Set	Stripe Validation Set	Blur Training Set	Blur Validation Set
Normal Model	0.85	0.82	0.81	0.83	0.81	0.79	0.80	0.77	0.75
Tag Model	-	0.67	0.65	0.89	0.87	-	-	-	-
Stripe Model	-	0.64	0.61	-	-	0.85	0.81	-	-
Blur Model	-	0.65	0.63	-	-	-	-	0.89	0.83

Table 4: **Top-1 Accuracy Metrics for Small DNN on Bone Age Dataset.**

Model	Normal Training Set	Normal Validation Set	Normal Test Set	Tag Training Set	Tag Validation Set	Stripe Training Set	Stripe Validation Set	Blur Training Set	Blur Validation Set
Normal Model	0.79	0.71	0.73	0.71	0.69	0.64	0.67	0.64	0.68
Tag Model	-	0.54	0.53	0.78	0.74	-	-	-	-
Stripe Model	-	0.58	0.53	-	-	0.71	0.68	-	-
Blur Model	-	0.51	0.59	-	-	-	-	0.76	0.72

Table 5: **Top-1 Accuracy Metrics for Resnet-50 on Bone Age Dataset.**

E Additional Results: Feature Attributions

In a series of subsequent figures with this appendix. We show additional saliency visualizations for settings considered in the main text. Here show a single visualization for the bone age setting since we found that the saliency attributions for such setting were quite faint. Overall we obtain similar results between the knee and bone age datasets, so we opt to show the bone age visualizations here instead.

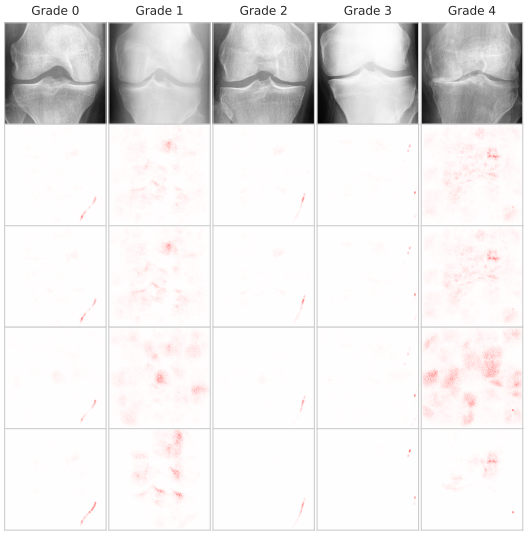

Figure 10: Spurious Images for Spurious Knee Model Examples

F Additional Results: Concepts Approaches

Normal Model compared to radiologist Concept Ranking For each of the 100 re-runs, we computed the rank correlation between the concept ranking for the normal model and rankings provided by a board certified radiologist.

Concepts for Dogs and Bone Age. We consider TCAV for the bone age and dog species task. For bone age, we choose as concepts the partitions of a hand, which correspond to the parts of the hand a radiologist would inspect to ascertain the age from a radiograph. We show, in Figure 20, a representation of these concepts. In the dog species classification task, consider concepts related floppiness of the ear, erectness of the ear, the dog head type, and color.

We manually collect data on the dog species concept. Here we showed Amazon mechanical Turkers an image from each class (10 classes) and then asked them to indicate whether each specific dog class had floppy ears or not, erect ears or not, single colored or not etc. We normalize each attribute to be binary and apply each concept broadly across each class.

Model	Normal Training Set	Normal Validation Set	Normal Test Set	Tag Training Set	Tag Validation Set	Stripe Training Set	Stripe Validation Set	Blur Training Set	Blur Validation Set
Normal Model	0.77	0.51	0.5	0.49	0.47	0.46	0.45	0.5	0.46
Tag Model	-	0.37	0.35	0.84	0.81	-	-	-	-
Stripe Model	-	0.64	0.61	-	-	0.72	0.0.69	-	-
Blur Model	-	0.65	0.63	-	-	-	-	0.84	0.73

Table 6: **Top-1 Accuracy Metrics for Small DNN on Knee Dataset.**

Setting	Spearman Correlation w/Rad
Infancy/Toddler	0.85
Pre-Puberty	0.89
Pre-Puberty	0.94
Late Puberty	0.86
Post Puberty	0.91

Table 7: **Normal Model compared to radiologist Concept Ranking**

Discussion on ACE . The ACE concept approach segments an image and uses the image segments as concepts as part of a TCAV pipeline. In an experimental analysis, we segment the spurious images using the various segmentation algorithms in algorithm to identify how often the spurious signal is selected as a distinct segment. We found segmentation to be ineffective in the stripe and blur settings. For the spurious tag, we found segmentation to be effective in only 25 percent of the examples tested, which suggests that the underlying segmentation algorithm as part of ACE might be ineffective for detecting hidden spurious signals.

G Additional Results: Influence Functions

In Figure 21, we show the influence metric that we computed in the main text for the small DNN model for the Knee Arthritis prediction. In this task, the spurious signal is always aligned with the Grade 2 class. Here again, we see that in the spurious settings each metric actually improves.

We now provide additional tables for the the feature attribution metrics from the main text.

Metric	Gradient	SmoothGrad	Integrated Gradients	Guided BackProp
SISM-GT	0.68	0.74	0.62	0.92
SEM	0.0091	0.017	0.019	0.0038
NISM-SISM	0.0013	8.8e-4	0.07	0.37
SEM	0.0012	0.0082	0.076	0.059

Table 8: **Metrics for Bone Age Spurious Tag Model.**

Metric	Gradient	SmoothGrad	Integrated Gradients	Guided BackProp
SISM-GT	0.78	0.68	0.59	0.89
SEM	0.0011	0.012	0.019	0.0088
NISM-SISM	0.0013	7.2e-5	0.17	0.46
SEM	0.0016	0.0032	0.026	0.029

Table 9: **Metrics for Bone Age Spurious Stripe Model.**

Metric	Gradient	SmoothGrad	Integrated Gradients	Guided BackProp
SISM-GT	0.7	0.84	0.49	0.92
SEM	0.001	0.02	0.06	0.008
NISM-SISM	0.0013	2.4e-4	0.21	0.86
SEM	0.006	0.002	0.06	0.09

Table 10: **Metrics for Bone Age Spurious Blur Model.**

Table 11: Performance metrics for each attribution method across tasks for the Tag Setting.

Method	Bone Age				Knee				Dog Species			
	Grad	SG	IG	GBP	Grad	SG	IG	GBP	Grad	SG	IG	GBP
K-SSD	0.65	0.66	0.67	0.81	0.51	0.49	0.47	0.76	0.71	0.76	0.79	0.88
CCM	0.37	0.39	0.35	0.75	0.32	0.33	0.35	0.66	0.42	0.41	0.39	0.64
FAM	0.51	0.55	0.53	0.68	0.46	0.47	0.45	0.69	0.59	0.64	0.68	0.73

Table 12: Performance metrics for each attribution method across tasks for the Blur Setting.

Method	Bone Age				Knee				Dog Species			
	Grad	SG	IG	GBP	Grad	SG	IG	GBP	Grad	SG	IG	GBP
K-SSD	0.21	0.20	0.19	0.13	0.13	0.18	0.17	0.31	0.29	0.30	0.31	0.35
CCM	0.28	0.29	0.24	0.64	0.23	0.22	0.27	0.67	0.38	0.33	0.35	0.71
FAM	0.48	0.49	0.47	0.51	0.36	0.38	0.33	0.58	0.55	0.56	0.47	0.73

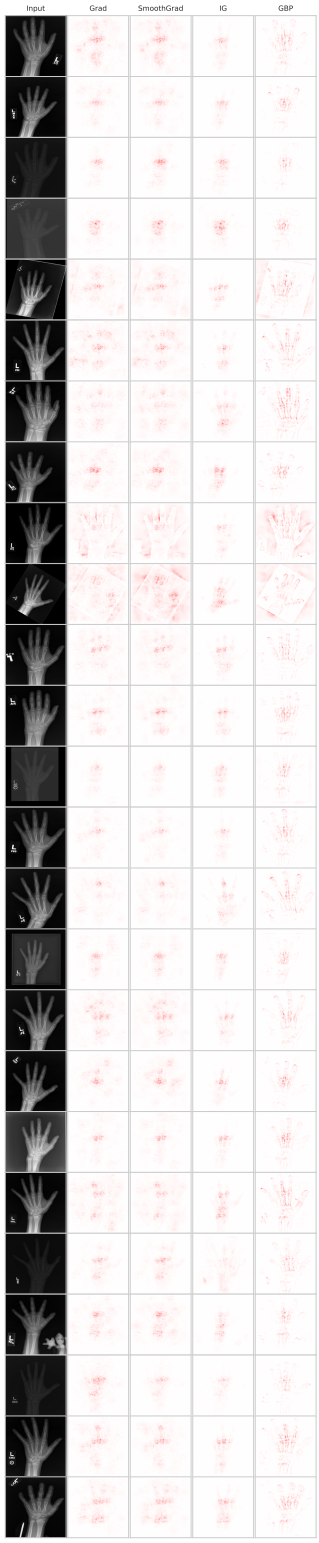

Figure 11: Normal Model on Normal Images

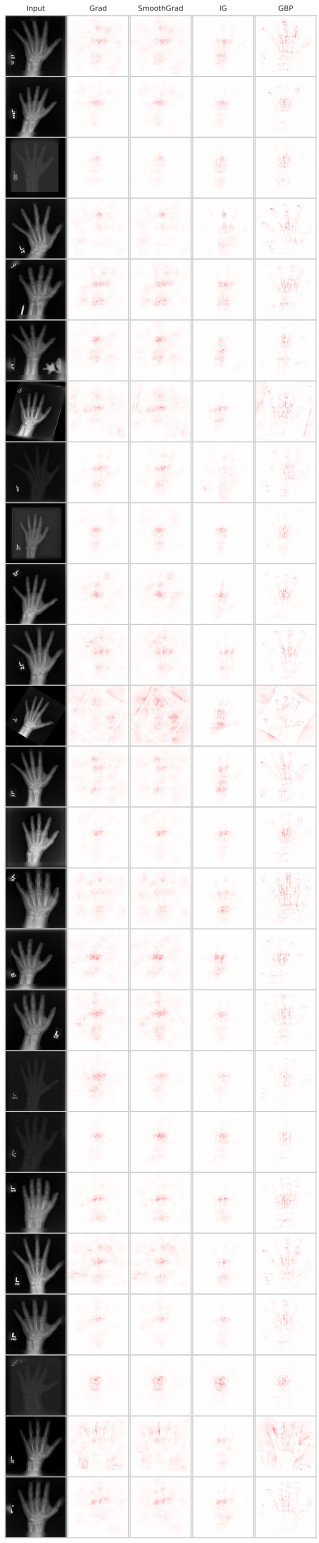

Figure 12: Normal Model on Spurious Blur Images

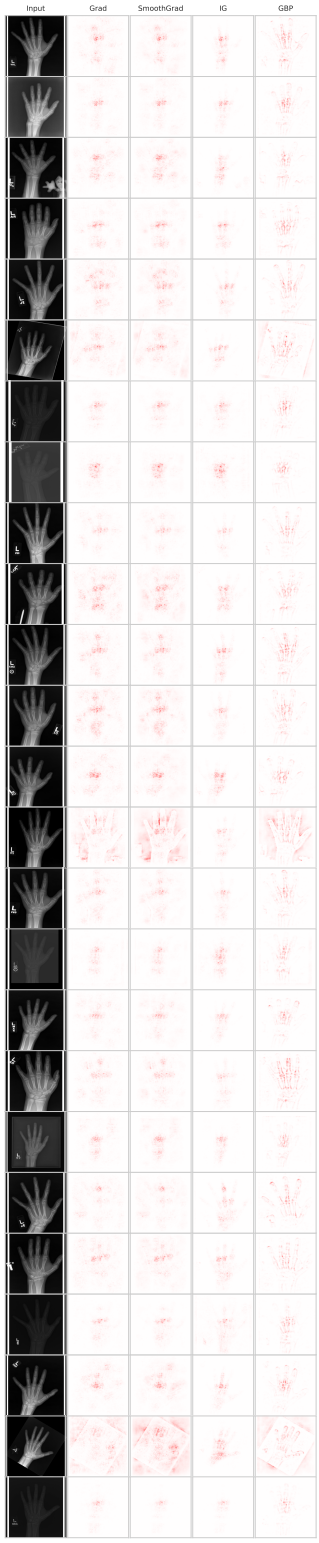

Figure 13: Normal Model on Spurious Stripe Images

Figure 14: Normal Model on Spurious Tag Images

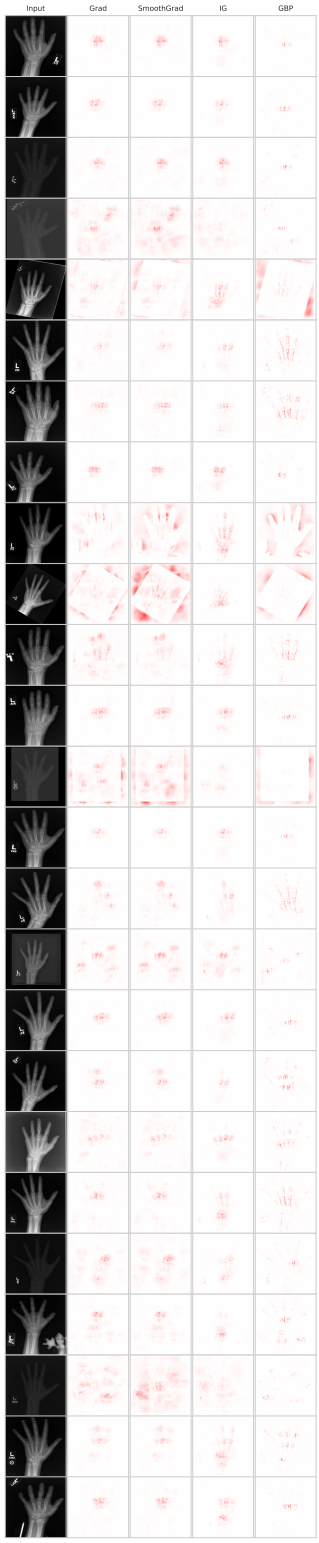

Figure 15: Spurious Stripe Model on Normal Images

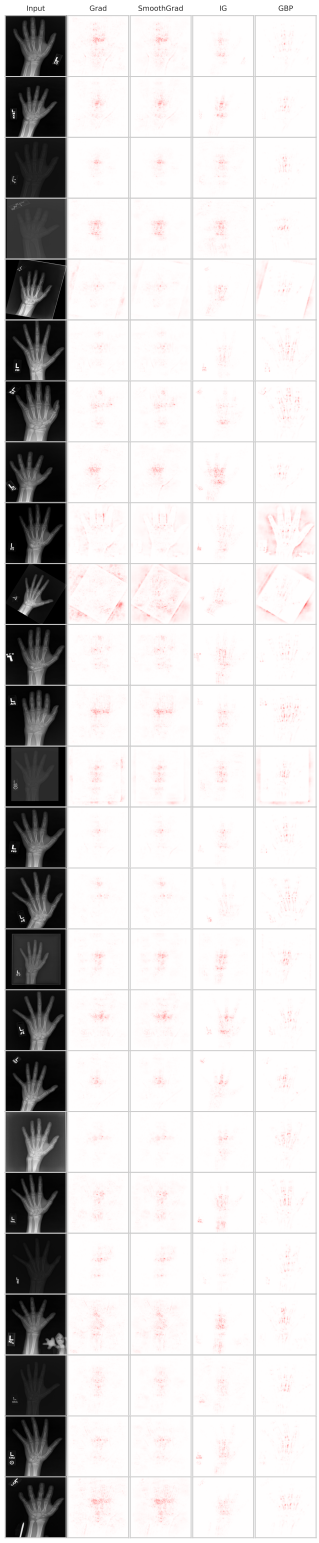

Figure 16: Spurious Tag Model on Normal Images

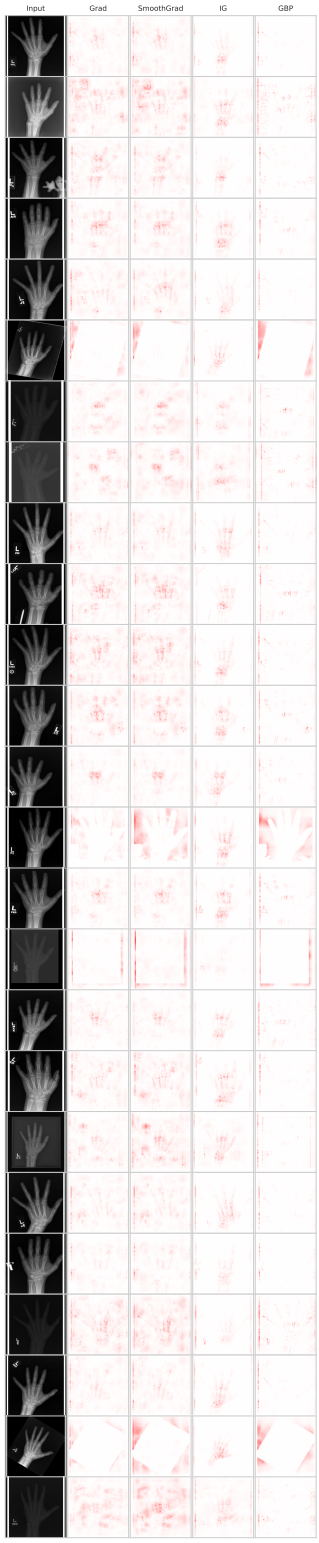

Figure 17: Spurious Stripe Model on Spurious Stripe Images

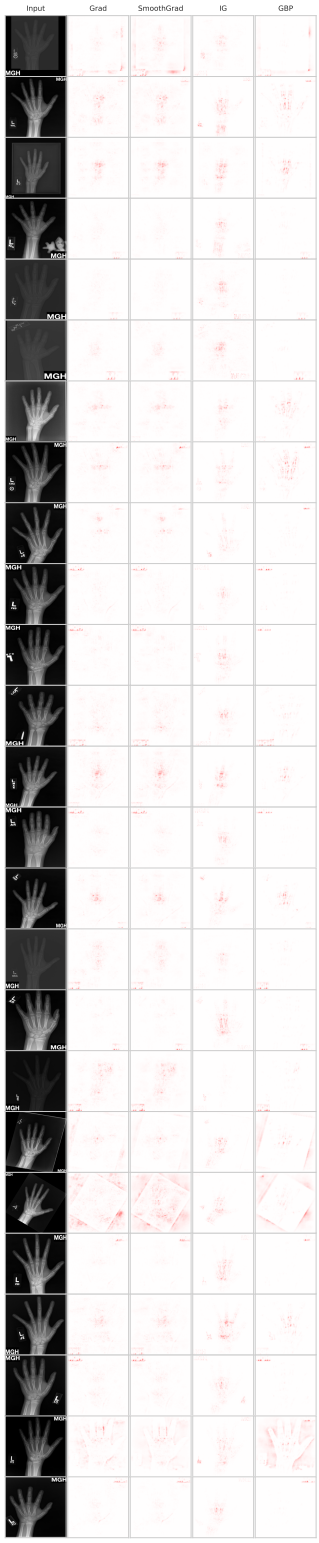

Figure 18: Spurious Tag Model on Spurious Tag Images

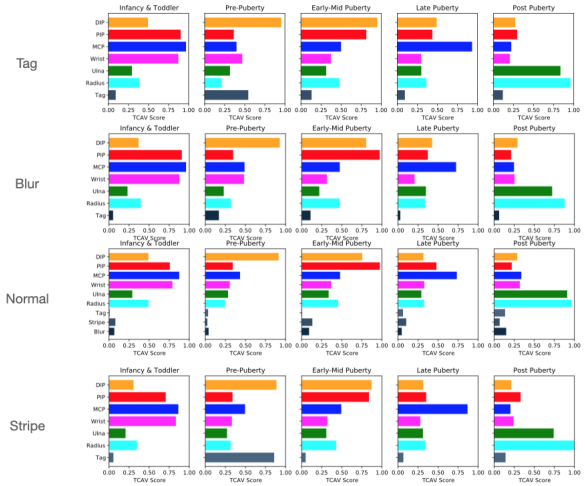

Figure 19: Results for Concept Ranking for another run of a normal and spurious bone age models.

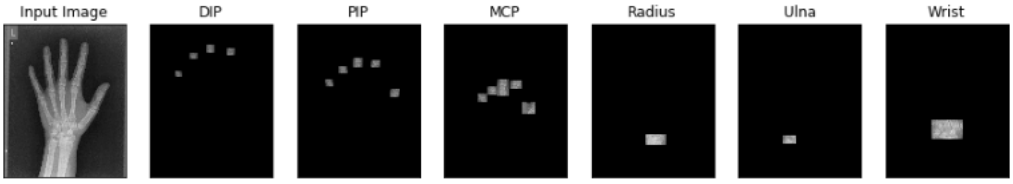

Figure 20: **Concept Partition for Bone Age Example.** Here we show how we partition a single instance into its constituent clinical concept components.

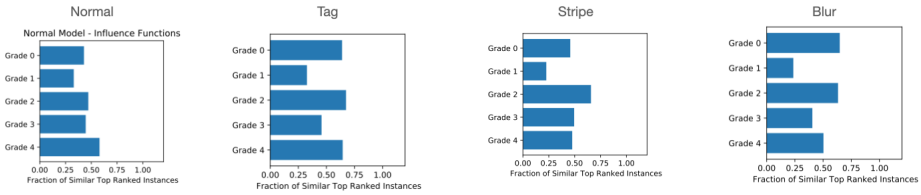

Figure 21: Influence Functions Results for Knee Settings.