UNIVERSITY OF CAMBRIDGE

Sanity Checks for Saliency Maps

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, Been Kim

L193 – Explainable Artificial Intelligence

Background – Saliency Maps

AI models can be ‘black boxes’ and saliency maps try to highlight which input features (e.g. pixels) matter the most for a given prediction

Examples include Gradients, Grad-CAM, Integrated Gradients, Guided Backprop…

Relying solely on visual appeal (the ‘map’) can be misleading

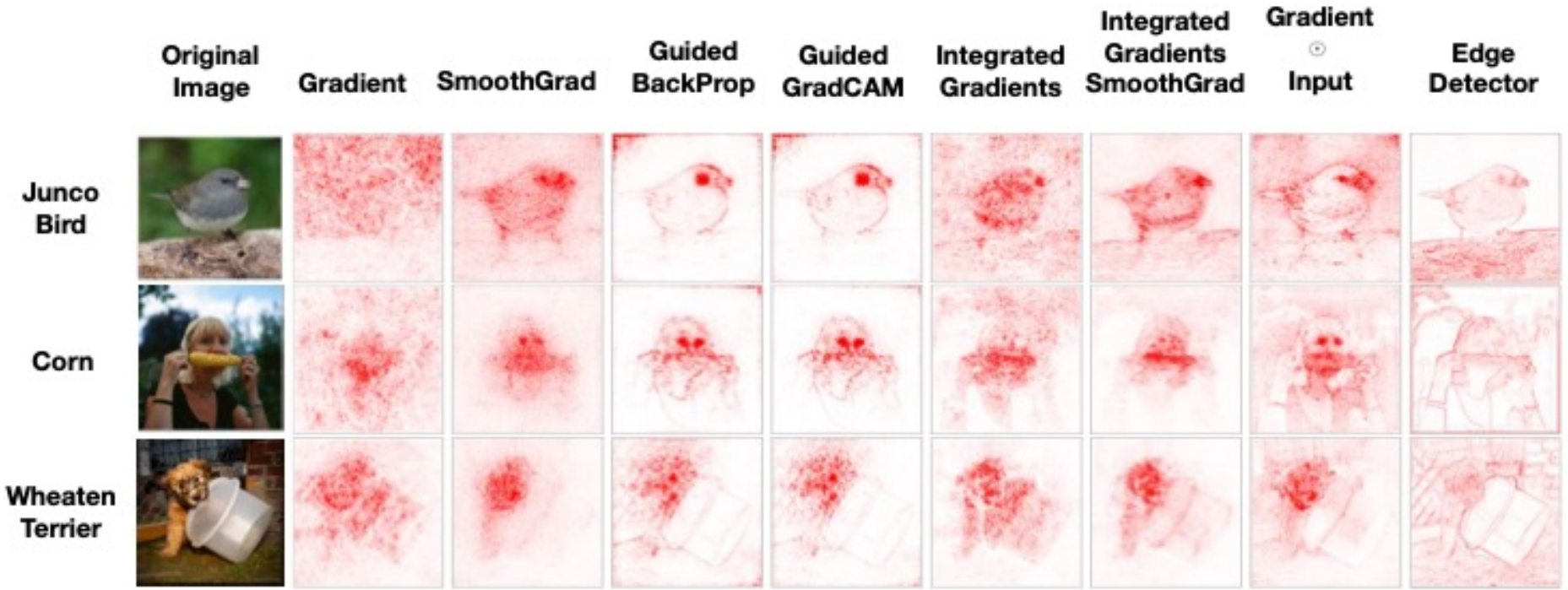
Background - Saliency Maps vs. Edge Detection

Some saliency methods (e.g. Guided Backprop) look very similar to classical edge detectors

Edge detectors require no training data or labels

Visual similarity could be misleading if map is just highlighting edges

Background - Saliency Maps vs. Edge Detection



Do saliency methods reflect model-data relationships, or do they just highlight superficial cues (like edges)?

Approach

Model Parameter Randomisation Test

Data Randomisation Test

Gradient, SmoothGrad, Guided BackProp, Guided GradCAM, Integrated Gradients, IGSG, Gradient $\odot$ Input

Inception v3 (ImageNet), CNNs on MNIST/Fashion-MNIST, MLP

Visual inspection, Spearman rank correlation (with/without absolute values), Structural Similarity Index (SSIM) and Histogram of Gradients (HOG) similarity

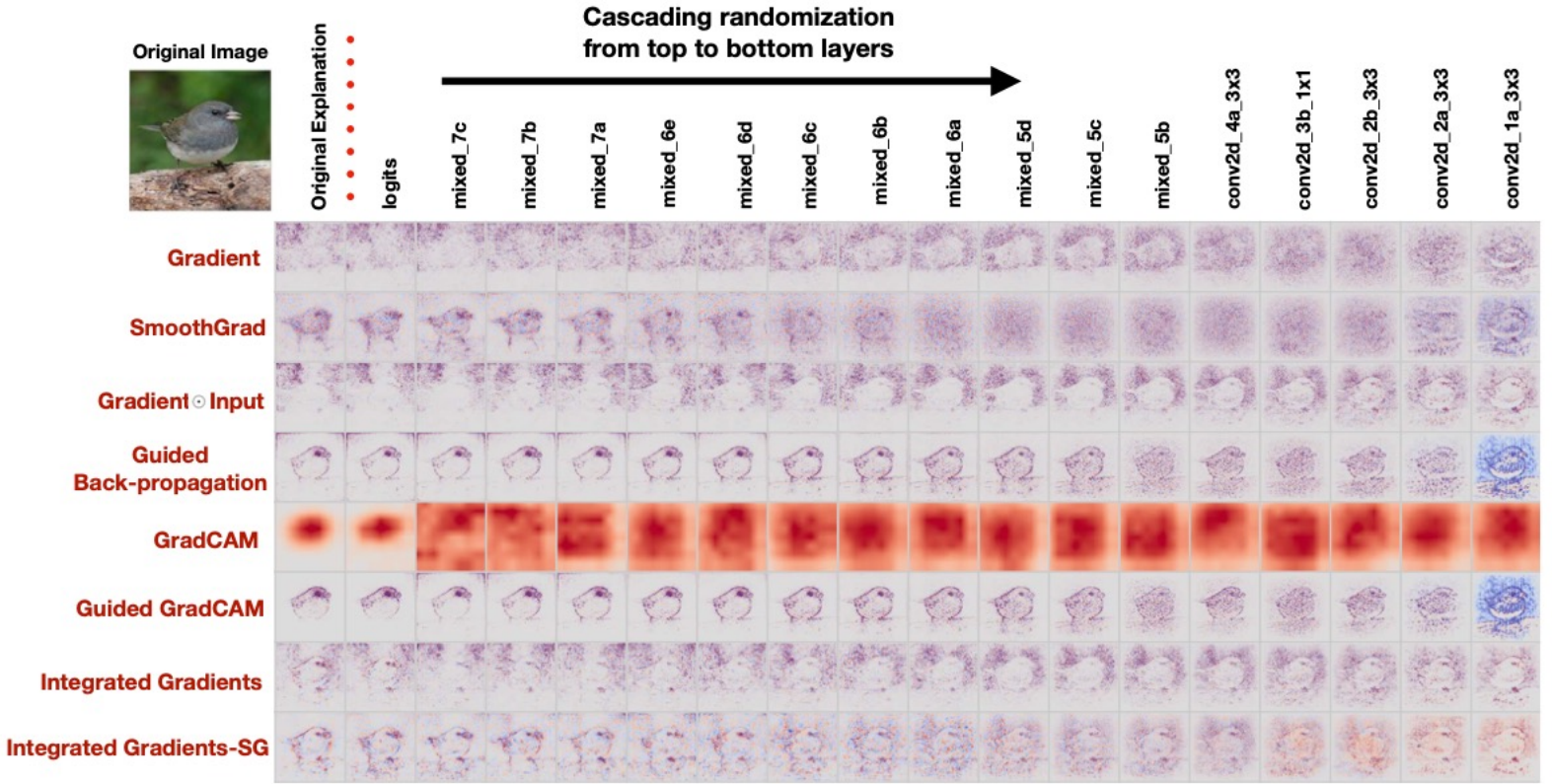
Model Parameter Randomisation Tests

Randomise model weights (top layer → bottom layer)

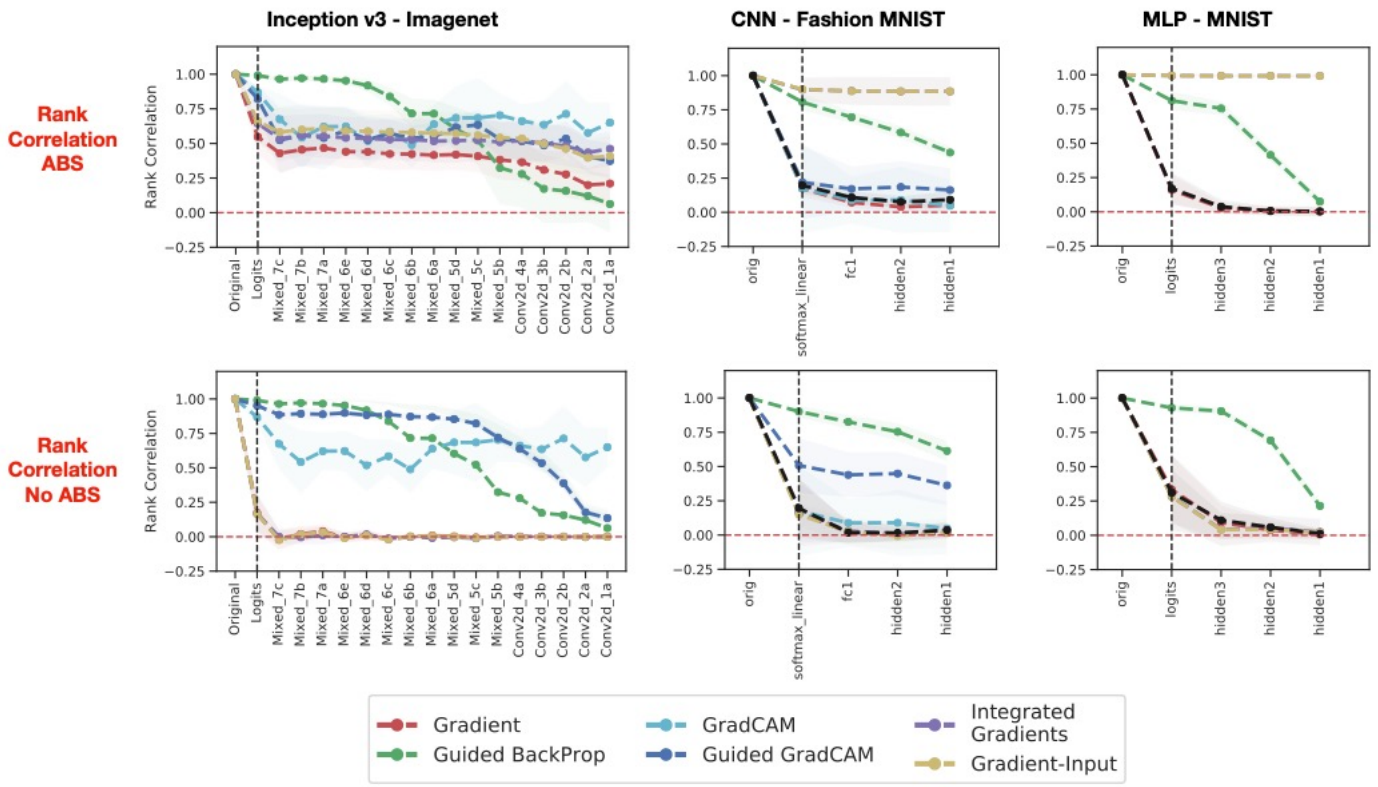
Cascading vs. Independent

Generate saliency maps after each randomisation step

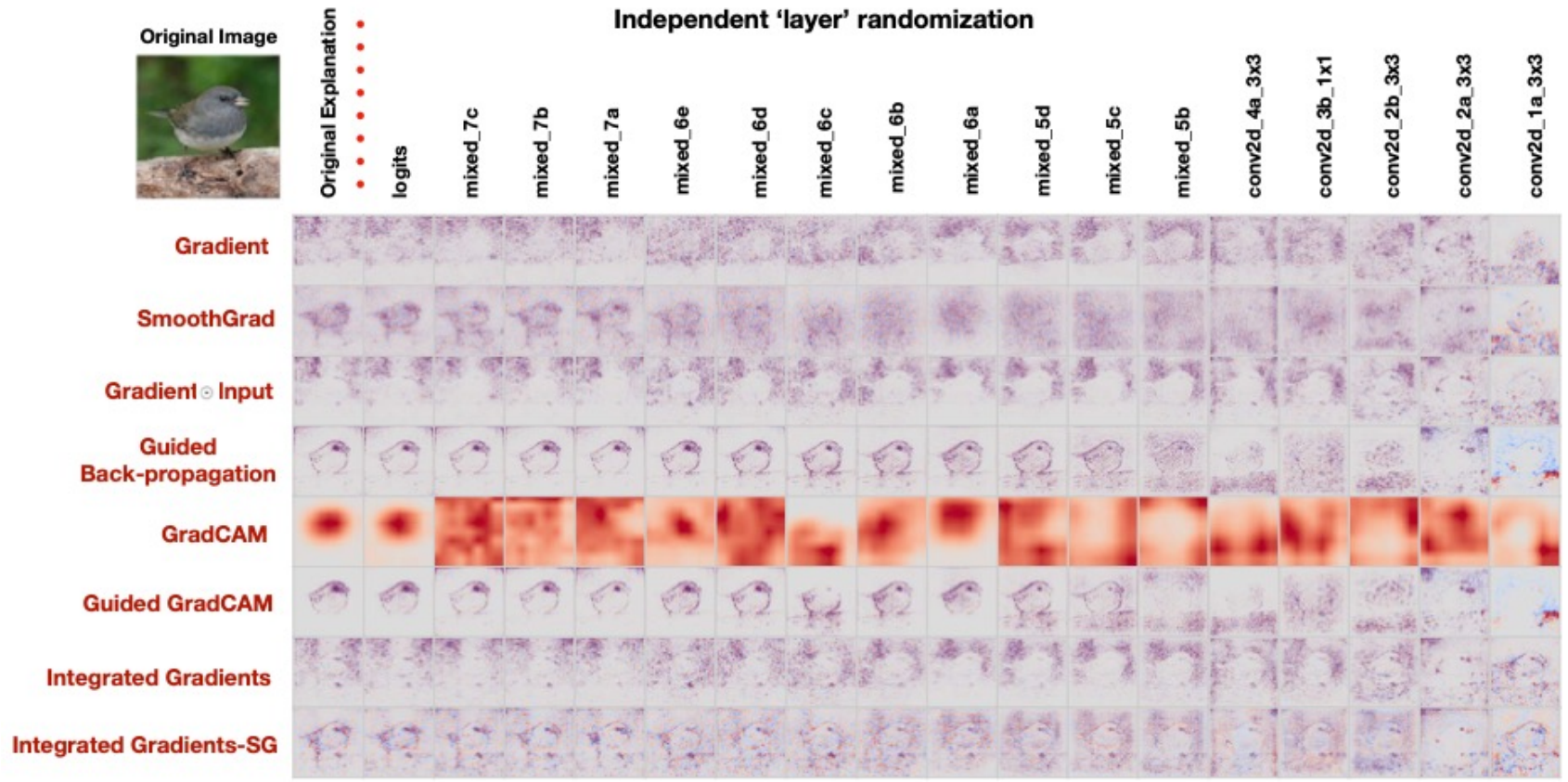
Model Parameter Randomisation Tests - Cascading



Model Parameter Randomisation Tests - Cascading



Model Parameter Randomisation Tests - Independent



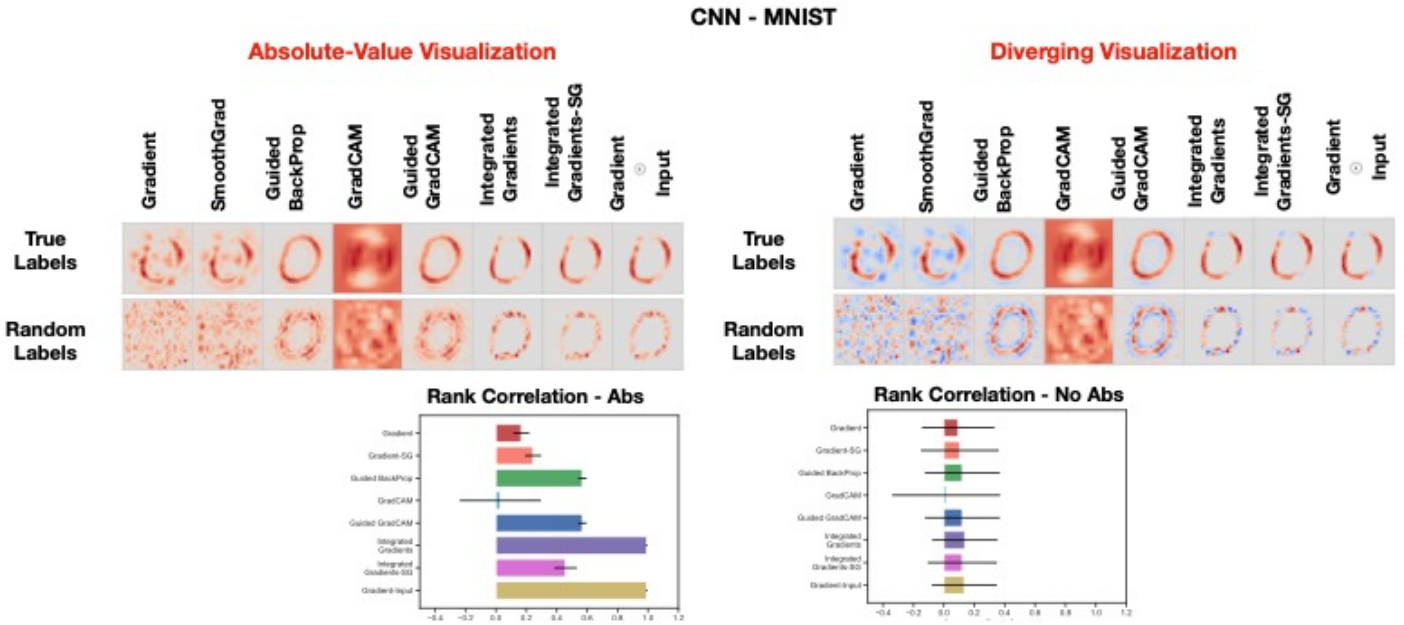
Data Randomisation Test

Shuffle training labels

Train a new model to fit random labels

Compare saliency maps from correctly-labelled model to randomly-labelled model

Data Randomisation Test



Key Findings

Saliency methods differ in sensitivity, some strongly reflect the learned parameters and data labels while others appear nearly unchanged when the model or labels are randomised

Visual similarity ≠ True explanation

Simple checks (randomisation tests) can reveal if a method genuinely depends on training

Key Findings

‘Architecture as a Prior’ – design of neural network can embed biases about how data should be processed

Element-wise input $\odot$ gradient (or similar approaches) can display the input’s outline even if gradient is random

Related Work

Name, Description and *Main Explanation Types*	References
CORRECTNESS (Section 6.1)	
Model Parameter Randomization Check – *Feature importance, Heatmap, Localization* Randomly perturb the internals of the predictive model and check that the explanation changes.	[3, 154, 224, 247, 301]
Explanation Randomization Check – *Feature importance, Heatmap* Randomly perturb the explanation (which is built into the predictive model) and check that the output of the predictive model changes.	[177, 247]
White Box Check – *Feature importance, Decision Rules, White-box model, Localization* Apply the explanation method to an interpretable white box model and check the correspondence of the explanation with the white box reasoning.	[58, 121, 124, 144, 216, 219, 326]
Controlled Synthetic Data Check *Feature importance, Heatmap, Prototypes, Localization, White-box model, Graph* Controlled experiment: Create a synthetic dataset such that the predictive model should follow a particular reasoning, known a priori (important: checking this assumption by e.g. reporting almost-perfect accuracy). Evaluate whether the explanation shows the same reasoning as the data generation process.	[3, 42, 77, 110, 116, 133, 160, 165, 167, 207, 210, 211, 226, 253, 258, 266, 276, 304]
Single Deletion – *Feature importance, Heatmap* Delete, mask or perturb a single feature in the input and evaluate the change in output of the predictive model. Measure correlation with explanation's importance score.	[9, 19, 44, 52, 77, 191, 233, 234, 236, 316, 316]
Incremental Deletion (or Incremental Addition) – *Feature importance, Heatmap* One by one delete (or perturb) or add features to the input, based on explanation's order, and measure for each new input the change in output of the predictive model. Report average change in log-odds score, AUC, steepness of curve or number of features needed for a different decision. Compare with random ranking or other baselines.	[27, 36, 40, 43, 74, 79, 84, 93, 95, 100, 108, 112, 116, 129, 137, 159, 164, 167, 177, 185, 194, 214, 216, 231, 236, 246, 277, 288, 295, 300, 301]
OUTPUT-COMPLETENESS (Section 6.2)	
Preservation Check – *Feature importance, Heatmap, Localization, Text, Prototypes* Giving the explanation (or data based on the explanation) as input to the predictive model should result in the same decision as for the original, full input sample.	[23, 36, 42, 63, 92, 93, 128, 140, 153, 154, 166, 224, 285, 302, 307, 308]
Deletion Check – *Feature importance, Heatmap, Localization* Giving input *without* explanation's relevant features should result in a different decision by the predictive model than the decision for the original, full input sample.	[63, 140, 154, 167, 209, 224]
Fidelity *Feature importance, Heatmap, Decision Rules, Decision Tree, Prototypes, Text, Localization, White-box model* Measure the agreement between the output of the predictive model and the explanation when applied to the same input sample(s).	[12, 15, 38, 44, 58, 61, 121, 128, 144, 151, 161, 202, 203, 205, 218, 264, 272, 292–294, 306, 316, 322, 326]
Predictive Performance *Feature importance, Heatmap, Decision Rules, Decision Tree, Prototypes, White-box model* Predictive performance of the interpretable model or predictive explanation with respect to the ground-truth data.	[12, 44, 58, 82, 97, 134, 145, 157, 202, 207, 208, 218, 220, 220, 243, 292, 300, 306, 316, 319] i.a.
CONSISTENCY (Section 6.3)	
Implementation Invariance – *Feature Importance* Evaluate whether the explanation method is invariant to specific implementations of the predictive model by validating whether two implementations that give the same output for an input, also get the same explanation.	[72, 267]

https://arxiv.org/pdf/2201.08164

Related Work

Table 3. Continued

Name, Description and *Main Explanation Types*	References
CONTINUITY (Section 6.4)	
Stability for Slight Variations *Feature importance, Heatmap, Graph, Text, Localization, Decision Rules, White-box model* Measure the similarity between explanations for two slightly different samples. Small variations in the input, for which the model response is nearly identical, should not lead to large changes in the explanation.	[9, 29, 33, 56, 64, 83, 83, 100, 144, 153, 154, 205–207, 212, 245, 256, 263, 273, 301]
Fidelity for Slight Variations – *Decision Rules, White-box model* Measure the agreement between interpretable predictions for original and slightly different samples: an explanation for original input x should accurately predict the model's output for a slightly different sample x'.	[144, 206]
Connectedness – *Prototypes, Representation Synthesis* Measure how connected a counterfactual explanation is to samples in the training data: ideally, the counterfactual is not an outlier, and there is a continuous path between a generated counterfactual and a training sample.	[127, 149, 201]
CONTRASTIVITY (Section 6.5)	
Target Sensitivity – *Heatmap* The explanation for a particular target or model output (e.g. class) should be different from an explanation for another target.	[188, 209, 247, 253, 277, 281]
Target Discriminativeness – *Disentanglement, Representation Synthesis, Text* The explanation should be target-discriminative such that *another model* can predict the right target (e.g. class label) from the explanation, in either a supervised or unsupervised fashion.	[32, 75, 120, 137, 246, 272, 275, 288, 295]
Data Randomization Check – *Feature importance, Heatmap, Localization* Randomly change labels in a copy of the training dataset, train a model on this randomized dataset and check that the explanations for this model on a test set are different from the explanations for the model trained on the original training data.	[3, 154, 224]
COVARIATE COMPLEXITY (Section 6.6)	
Covariate Homogeneity *Prototypes, Disentanglement, Localization, Heatmap, Representation Synthesis* Evaluate how consistently a covariate (i.e. feature) in an explanation represents a predefined human-interpretable concept.	[4, 24, 26, 69, 73, 75, 80, 94, 111, 129, 146, 162, 183, 239, 240, 246, 255, 265, 291, 303, 313, 315, 321, 323]
Covariate Regularity – *Decision Rules, Feature Importance* Evaluate the regularity of an explanation by measuring its Shannon entropy, in order to quantify how noisy the explanation is and how easy it is to memorize the explanation.	[267, 306]
COMPACTNESS (Section 6.7)	
Size *Feature importance, Heatmap, Decision Rules, Decision Tree, Prototypes, Text, Graph, Localization, White-box model, Representation Synthesis* Total size (absolute) or sparsity (relative) of the explanation.	[8, 35, 38, 58, 61, 82, 115, 130, 131, 136, 143, 145, 153, 177, 205–210, 218–220, 228, 238, 255, 259, 263, 273, 282, 283, 285, 288, 292–294, 305, 314, 319]
Redundancy – *Feature importance, Decision Rules, Text, White-box model* Calculate the redundancy or overlap between parts of the explanation.	[145, 151, 266]
Counterfactual Compactness – *Prototypes, Representation Synthesis, Text* Given a counterfactual explanation showing what needs to be changed in the input in order to change the prediction of the predictive model, measure how *much* needs to be changed.	[8, 88, 127, 130, 151, 201, 262, 318]

https://arxiv.org/pdf/2201.08164

Positives

Highly quantitative

Seminal

Easy to replicate

Negatives

Focus only on images

Not many architectures tested

Future Work

Apply tests to other modalities

Could combine with ablation or concept-based approaches to investigate causality

Test how saliency changes under partial label noise