XAI-TRIS: Non-linear benchmarks to quantify ML explanation performance

Benedict Clark
Physikalisch-Technische Bundesanstalt
Abbestr. 2–12 10587 Berlin, Germany
benedict.clark@ptb.de

Rick Wilming, Stefan Haufe
Technische Universität Berlin
Str. des 17. Juni 135, 10623 Berlin, Germany
{rick.wilming, haufe}@tu-berlin.de

Abstract

The field of 'explainable' artificial intelligence (XAI) has produced highly cited methods that seek to make the decisions of complex machine learning (ML) methods 'understandable' to humans, for example by attributing 'importance' scores to input features. Yet, a lack of formal underpinning leaves it unclear as to what conclusions can safely be drawn from the results of a given XAI method and has also so far hindered the theoretical verification and empirical validation of XAI methods. This means that challenging non-linear problems, typically solved by deep neural networks, presently lack appropriate remedies. Here, we craft benchmark datasets for three different non-linear classification scenarios, in which the important class-conditional features are known by design, serving as ground truth explanations. Using novel quantitative metrics, we benchmark the explanation performance of a wide set of XAI methods across three deep learning model architectures. We show that popular XAI methods are often unable to significantly outperform random performance baselines and edge detection methods. Moreover, we demonstrate that explanations derived from different model architectures can be vastly different; thus, prone to misinterpretation even under controlled conditions.

1 Introduction

Only recently, a trend towards the objective empirical validation of XAI methods using ground truth data has been observed Tjoa & Guan (2020); Li et al. (2021); Zhou et al. (2022); Arras et al. (2022); Gevaert et al. (2022); Agarwal et al. (2022). These studies are, however, limited in the extent to which they permit a quantitative assessment of explanation performance, in the breadth of XAI methods evaluated, and in the difficulty of the posed 'explanation' problems. In particular, most published benchmark datasets are constructed in a way such that realistic correlations between class-dependent (e.g., the foreground or object of an image) and class-agnostic (e.g., the image background) features are excluded. In practice, such dependencies can give rise to features acting as suppressor variables. Briefly, suppressor variables have no statistical association to the prediction target on their own, yet including them may allow an ML model to remove unwanted signals (noise), which can lead to improved predictions. In the context of image or photography data, suppressor variables could be parts of the background that capture the general lighting conditions. A model can use such information to normalize the illumination of the object and, thereby, improve object detection. More details on the principles of suppressor variables can be found in Conger (1974); Friedman & Wall (2005); Haufe et al. (2014); Wilming et al. (2022). Here we adopt the formal requirement that an input feature should only be considered important if it has a statistical association with the prediction target, or is associated to it by construction. In that sense, it is undesirable to attribute importance to pure suppressor features.

Preprint. Under review.

Yet, Wilming et al. (2022) have shown that some of the most popular model-agnostic XAI methods are susceptible to the influence of suppressor variables, even in a linear setting. Using synthetic linearly separable data defining an explicit ground truth for XAI methods and linear models, Wilming et al. showed that a significant amount of feature importance is incorrectly attributed to suppressor variables. They proposed quantitative performance metrics for an objective validation of XAI methods, but limited their study to linearly separable problems and linear models. They demonstrate that methods based on so-called activation patterns (that is, univariate mappings from predictions to input features), based on the work of Haufe et al. (2014), provide the best explanations. However, it is unclear as to what extent these results would transfer to various non-linear settings.

Thus, well-designed non-linear ground truth data comprising of realistic correlations between important and unimportant features are needed to study the influence of suppressor variables on XAI explanations in non-trivial settings, which is the purpose of this paper. We go beyond existing work in the following ways:

First, we design one linear and three non-linear binary image classification problems, in which different types and combinations of tetrominoes Golomb (1996), overlaid on a noisy background, need to be distinguished. In all cases, ground truth explanations are explicitly known through the location of the tetrominoes. Apart from the linear case, these classification problems require (different types of) non-linear predictive models to be solved effectively.

Second, based on signal detection theory and optimal transport, we define two suitable quantitative metrics of 'explanation performance' designed to handle the case of few important features.

Third, using three different types of background noise (white, correlated, imagenet), we invoke the presence of suppressor variables in a controlled manner and study their effect on explanation performance.

Fourth, we evaluate the explanation performance of no less than sixteen of the most popular model-agnostic and model-specific XAI methods, across three different machine learning architectures.

Finally, we propose four model-agnostic baselines that can serve as null models for explanation performance.

2 Methods

2.1 Data generation

For each scenario, we construct an individual dataset of 64×64-sized images as $\mathcal{D} = (\mathbf{x}^{(n)}, y^{(n)})_{n=1}^{N}$, consisting of *i.i.d* observations $(\mathbf{x}^{(n)} \in \mathbb{R}^D, y^{(n)} \in \{0,1\})_{n=1}^{N}$, where feature space $D = 64^2 = 4096$ and $N = 40,000$. Here, $\mathbf{x}^{(n)}$ and $y^{(n)}$ are realizations of the random variables $\mathbf{X}$ and Y, with joint probability density function $p_{\mathbf{X},Y}(\mathbf{x}, y)$.

In each scenario, we generate a sample $\mathbf{x}^{(n)}$ as a combination of a signal pattern $\boldsymbol{a}^{(n)} \in \mathbb{R}^D$, carrying the set of truly important features used to form the ground truth for an ideal explanation, with some background noise $\boldsymbol{\eta}^{(n)} \in \mathbb{R}^D$. We follow two different generative models depending on whether the two components are combined additively or multiplicatively.

Additive generation process For additive scenarios, we define the data generation process

$$\mathbf{x}^{(n)} = \alpha(R^{(n)} \circ (H \circ \boldsymbol{a}^{(n)})) + (1 - \alpha)(G \circ \boldsymbol{\eta}^{(n)}), \tag{1}$$

for the n-th sample. Signal pattern $\mathbf{a}^{(n)} = \mathbf{a}(y^n)$ carries differently shaped tetromino patterns depending on the binary class label $y^{(n)} \sim$ Bernoulli(½). We apply a 2D Gaussian spatial smoothing filter $H : \mathbb{R}^D \to \mathbb{R}^D$ to the signal component to smooth the integration of the pattern's edges into the background, with smoothing parameter (spatial standard deviation of the Gaussian) $\sigma_{\text{smooth}} = 1.5$. The Gaussian filter H can technically provide infinite support to $\boldsymbol{a}^{(n)}$, so in practice we threshold the support at 5% of the maximum level. White Gaussian noise $\boldsymbol{\eta}^{(n)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$, representing a non-informative background, is sampled from a multivariate normal distribution with zero mean and identity covariance $\mathbf{I}_D$. For each classification problem, we define a second background scenario, denoted as CORR, in which we apply a separate 2D Gaussian spatial smoothing filter $G : \mathbb{R}^D \to \mathbb{R}^D$ to the noise component $\boldsymbol{\eta}^{(n)}$. Here, we set the smoothing parameter to $\sigma_{\text{smooth}} = 10$. The

third background type is that of samples from the ImageNet database Deng et al. (2009), denoted IMAGENET. We scale and crop images to be 64×64-px in size, preserving the original aspect ratio. Each 3-channel RGB image is converted to a single-channel gray-scale image using the built-in Python Imaging Library (PIL) functions and is zero-centered by subtraction of the sample's mean value.

As alluded to below, we also analyze a scenario where the signal pattern $\boldsymbol{a}^{(n)}$ underlies a random spatial rigid body (translation and rotation) transformation $R^{(n)} : \mathbb{R}^D \to \mathbb{R}^D$. All other scenarios make use of the identity transformation $R^{(n)} \circ \boldsymbol{a}^{(n)} = \boldsymbol{a}^{(n)}$. Transformed signal and noise components $(R^{(n)} \circ \boldsymbol{a}^{(n)})$ and $(G \circ \boldsymbol{\eta}^{(n)})$ are horizontally concatenated into matrices $\mathbf{A} = \left[(R^{(1)} \circ \boldsymbol{a}^{(1)}), \ldots, (R^{(N)} \circ \boldsymbol{a}^{(N)})\right]$ and $\mathbf{E} = \left[(G \circ \boldsymbol{\eta}^{(1)}), \ldots, (G \circ \boldsymbol{\eta}^{(N)})\right]$. Signal and background components are then normalized by the Frobenius norms of $\mathbf{A}$ and $\mathbf{E}$: $(R^{(n)} \circ \boldsymbol{a}^{(n)}) \leftarrow {}^{(R^{(n)} \circ \boldsymbol{a}^{(n)})}/_{||\mathbf{A}||_\mathrm{F}}$ and $(G \circ \boldsymbol{\eta}^{(n)}) \leftarrow {}^{(G \circ \boldsymbol{\eta}^{(n)})}/_{||\mathbf{E}||_\mathrm{F}}$, where the Frobenius norm of a matrix $\mathbf{A}$ is defined as $||\mathbf{A}||_\mathrm{F} \coloneqq (\sum_{n=1}^{N} \sum_{d=1}^{D} (\boldsymbol{a}_d^{(n)})^2)^{1/2}$. Finally, a weighted sum of the signal and background components is calculated, where the scalar parameter $\alpha \in [0, 1]$ determines the signal-to-noise ratio (SNR).

Multiplicative generation process For multiplicative scenarios, we define the generation process

$$\mathbf{x}^{(n)} = \left(\mathbf{1} - \alpha \left(R^{(n)} \circ (H^{(n)} \circ \boldsymbol{a}^{(n)})\right)\right) \left(G \circ \boldsymbol{\eta}^{(n)}\right) , \tag{2}$$

where $\boldsymbol{a}^{(n)}$, $\boldsymbol{\eta}^{(n)}$, $R^{(n)}$, H and G are defined as above, $\mathbf{A}$ and $\mathbf{E}$ are Frobenius-normalized, and $\mathbf{1} \in \mathbb{R}^D$.

For data generated via either process, we scale each sample $\mathbf{x}^{(n)} \in \mathbb{R}^D$ to the range $[-1, 1]^D$, such that $\mathbf{x}^{(n)} \leftarrow \mathbf{x}^{(n)} / \max |\mathbf{x}|$, where $\max |\mathbf{x}|$ is the maximum absolute value of any feature in the dataset.

Emergence of suppressors Note that the correlated background noise scenario induces the presence of suppressor variables, both in the additive and the multiplicative data generation processes. A suppressor here would be a pixel that is not part of the foreground $R^{(n)} \circ \boldsymbol{a}^{(n)}$, but whose activity is correlated with a pixel of the foreground by virtue of the smoothing operator G. Based on previously reported characteristics of suppressor variables Conger (1974); Friedman & Wall (2005); Haufe et al. (2014); Wilming et al. (2022), we expect that XAI methods may be prone to attributing importance to suppressor features in the considered linear and non-linear settings, leading to drops in explanation performance as compared to the white noise background setting.

Scenarios We make use of tetrominoes Golomb (1996), geometric shapes consisting of four blocks (here each being 8×8-pixels), to define each signal pattern $\boldsymbol{a}^{(n)} \in \mathbb{R}^{64 \times 64}$. We choose these as the basis for signal patterns as they allow a fixed and controllable amount of features (pixels) per sample, and specifically the 'T'-shaped and 'L' shaped tetrominoes due to their four unique appearances under each 90-degree rotation. These induce statistical associations between features and target in four different binary classification problems:

Linear (LIN) and multiplicative (MULT) For the linear case, we use the additive generation model Eq. (1), and for the multiplicative case, we instead use the multiplicative generation model. In both, signal patterns are defined as a 'T'-shaped tetromino pattern $\mathbf{a}^{\mathrm{T}}$ near the top left corner if $y = 0$ and an 'L'-shaped tetromino pattern $\mathbf{a}^{\mathrm{L}}$ near the bottom-right corner if $y = 1$, leading to the binary classification problem. Each pattern is encoded such that $a_{i,j}^{\mathrm{T/L}} = 1$ for each pixel in the tetromino pattern, positioned at the i-th row and j-th column of $\boldsymbol{a}^{\mathrm{T/L}}$, and zero otherwise.

Translations and rotations (RIGID) In this scenario, $a^{\mathrm{T/L}}$ defining each class are no longer in fixed positions but are randomly translated and rotated by multiples of 90 degrees according to a rigid body transform $R^{(n)}$, constrained such that the entire tetromino is contained within the image. In contrast to the other scenarios, we use a 4-pixel thick tetromino here to enable a larger set of transformations, and thus increase the complexity of the problem. This is an additive manipulation in accordance with (1).

XOR The final scenario is that of an additive XOR problem, where we use both tetromino variants $a^{\mathrm{T/L}}$ in every sample. Transformation $R^{(n)}$ is, once again, the identity transform here. Class membership is defined such that members of the first class, where $y = 0$, combine both tetrominoes with the background of the image either positively or negatively, such that $\mathbf{a}^{\mathrm{XOR++}} = \mathbf{a}^{\mathrm{T}} + \mathbf{a}^{\mathrm{L}}$ and $\mathbf{a}^{\mathrm{XOR--}} = -\mathbf{a}^{\mathrm{T}} - \mathbf{a}^{\mathrm{L}}$. Members of the opposing class, where $y = 1$, imprint one shape positively, and the other negatively, such that $\mathbf{a}^{\mathrm{XOR+-}} = \mathbf{a}^{\mathrm{T}} - \mathbf{a}^{\mathrm{L}}$ and $\mathbf{a}^{\mathrm{XOR-+}} = -\mathbf{a}^{\mathrm{T}} + \mathbf{a}^{\mathrm{L}}$. Each of the four XOR cases are equally frequently represented across the dataset.

Figure 1 shows two examples from each class of each classification problem and for the three background types – Gaussian white noise (WHITE), smoothed Gaussian white noise (CORR), and ImageNet samples (IMAGENET). Figure 4 in the supplementary material shows examples of each of the 12 scenarios across four signal-to-noise ratios (SNRs).

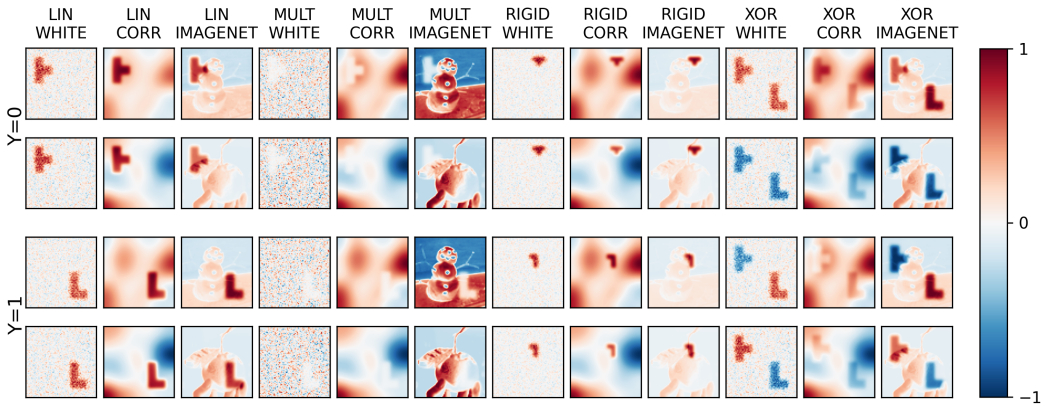


Figure 1: Examples of data for each scenario, showing differences between samples of each class.

With each classification scenario defined, we can form the ground truth feature set of important pixels for a given input based on the positions of tetromino pixels as

$$\mathcal{F}^{+}(\mathbf{x}^{(n)}) \coloneqq \left\{ d \mid \left(R^{(n)} \circ (H \circ \boldsymbol{a}^{(n)}) \right)_d \neq 0,\ d \in \{1, \ldots, 4096\} \right\} . \tag{3}$$

For the LIN and MULT scenarios, each sample either contains a 'T' or an 'L' tetromino at a fixed position, corresponding to the fixed patterns $\mathbf{a}^{\mathrm{T}}$ and $\mathbf{a}^{\mathrm{L}}$. Since the absence of a tetromino at one location is just as informative as the presence of the other at another location, we augment the set of important pixels for these two settings as

$$\mathcal{F}^{+}(\mathbf{x}^{(n)}) \coloneqq \left\{ d \mid H \circ \boldsymbol{a}^{\mathrm{T}}_d \neq 0 \vee H \circ \boldsymbol{a}^{\mathrm{L}}_d \neq 0,\ d \in \{1, \ldots, 4096\} \right\} . \tag{4}$$

Note that this definition is equivalent to Eq. (3) for the XOR scenario. Moreover, it is equivalent to an operationalization of feature importance put forward by Wilming et al. (2022) for the three static scenarios LIN, MULT, and XOR. Wilming et al. define any feature as important if it has a statistical dependency to the prediction target across the studied sample. In all cases, an ideal explanation method should attribute importance only to members of the set $\mathcal{F}^{+}(\mathbf{x}^{(n)})$.

For training each model and the subsequent analyses, we divide each dataset three-fold by a $90/5/5$ split into a training set $\mathcal{D}_{\text{train}}$, a validation set $\mathcal{D}_{\text{val}}$, and a test set $\mathcal{D}_{\text{test}}$.

2.2 Classifiers

We use three architectures to model each classification problem. Firstly, a Linear Logistic Regression (LLR) model, which is a single-layer neural network with two output neurons and a softmax activation function. Secondly, a Multi-Layer Perceptron (MLP) with four fully-connected layers, where each of the hidden layers uses Rectified Linear Unit (ReLU) activations. The two-neuron output layer is once again softmax-activated. Finally, we define a Convolutional Neural Network (CNN) with four blocks of ReLU-activated convolutional layers followed by a max-pooling operation, with a softmax-activated two-neuron output layer. The convolutional layers are specified with a progressively

increasing amount of filters per layer $[4, 8, 16, 32]$, a kernel size of four, a stride of one, and zero-padding. The max-pooling layers are defined with a kernel size of two and a stride of one.

We train a given classifier $f^{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathcal{Y}$ over parameterization $\boldsymbol{\theta}$ and $\mathcal{D}_{\text{train}}$. Each network is trained over 500 epochs using the Adam optimizer without regularization, with a learning rate of 0.0005. The validation dataset $\mathcal{D}_{\text{val}}$ is used at each step to get a sense of how well the model is generalizing the data. Validation loss is calculated at each epoch and used to judge when the classifier has reached optimal performance, by storing the model state with minimum validation loss. This also prevents using an overfit model. Finally, the test dataset $\mathcal{D}_{\text{test}}$ is used to calculate the resulting model performance, and is used in the evaluation of XAI methods. We consider a classifier to have generalized the given classification problem when the resulting test accuracy is at or above a threshold of 80%.

Each network is implemented in PyTorch, and also in Keras with a TensorFlow backend, so to experiment over a wider variety of XAI methods implemented using either the Captum Kokhlikyan et al. (2020) or iNNvestigate Alber et al. (2018) frameworks. The main text focuses on the former.

2.3 XAI methods and performance baselines

We compare sixteen popular XAI methods in our analysis. The main text focuses on the results of four: Local Interpretable Model Explanations (LIME) Ribeiro et al. (2016), Layer-wise Relevance Propagation (LRP) Bach et al. (2015), SHapley Additive exPlanations (SHAP) Lundberg & Lee (2017) and Integrated Gradients Sundararajan et al. (2017).

The full list is detailed in Appendix A.5. This briefly summarizes each method, and provides the details of which library was used for implementation, Captum Kokhlikyan et al. (2020) or iNNvestigate Alber et al. (2018), as well as the specific parameterization for each method. Generally, we follow the default parameterization for each method. Where necessary, we specify the baseline $\mathbf{b}$ as the zero input $\mathbf{b} = \mathbf{0}$, a common choice in the field Mamalakis et al. (2022).

The input to an XAI method is a model $f^{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}$, trained according to parameterization $\boldsymbol{\theta}$ over $\mathcal{D}_{\text{train}}$, the n-th test sample to be explained $\mathbf{x}_{\text{test}}^{(n)}$, as well as the baseline reference point $\mathbf{b} = \mathbf{0}$ for relevant methods. The method produces an 'explanation' $\mathbf{s}(f^{\boldsymbol{\theta}}, \mathbf{x}_{\text{test}}^{(n)}, \mathbf{b}) \in \mathbb{R}^D$.

We include four model-ignorant methods to generate 'baseline' importance maps for comparison with the aforementioned XAI methods. Firstly, we consider the Sobel filter, which uses both a horizontal and a vertical filter kernel to approximate first-order derivatives of data. Secondly, we use the Laplace filter, which uses a single symmetrical kernel to approximate second-order derivatives of data. Both are edge detection operators, and are given for each test sample as an input. Thirdly, we use a sample from a random uniform distribution $U((-1, 1)^D)$. Finally, we use the rectified test data sample $\mathbf{x}_{\text{test}}^{(n)}$ itself as an importance map.

2.4 Explanation performance metrics

Based on the well-defined ground truth set of class-dependent features for a given sample $\mathcal{F}^+(\mathbf{x}^{(n)})$, we can readily form quantitative metrics to evaluate the quality of an explanation.

Precision

Omitting the sample-dependence in the notation, we define precision as the fraction of the $k = |\mathcal{F}^+|$ features of $\mathbf{s}$ with the highest absolute-valued importance scores contained within the set $\mathcal{F}^+$ itself, over the total number of important features $|\mathcal{F}^+|$ in the sample.

Earth mover's distance (EMD)

The Earth mover's distance (EMD), also known as the Wasserstein metric, measures the optimal cost required to transform one distribution to another. We can apply this to the cost required to transform a continuous-valued importance map $\mathbf{s}$ into $\mathcal{F}^+$, where both are normalized to have the same mass. The Euclidean distance between pixels is used as the ground metric for calculating the EMD, with $\text{EMD}(\mathbf{s}, \mathcal{F}^+)$ denoting the cost of the optimal transport from $\mathbf{s}$ to $\mathcal{F}^+$. This follows the algorithm proposed by Bonneel et al. and the implementation of the Python Optimal Transport library Flamary

Table 1: Results of the model training process for each classification setting, model architecture, and background type. These results are depicted as chosen Signal-to-noise ratios (SNRs), parameterized by α, as well as the average test accuracy (ACC, %).

		WHITE		CORR		IMAGENET	
		α	ACC	α	ACC	α	ACC
LIN	LLR	0.03	89.7	0.02	100.0	0.1	87.5
	MLP	0.03	87.9	0.02	100.0	0.1	86.2
	CNN	0.03	90.1	0.02	99.9	0.1	93.9
MULT	MLP	0.64	85.8	0.04	89.2	0.3	91.2
	CNN	0.64	100.0	0.04	98.5	0.3	91.3
RIGID	MLP	0.575	88.9	0.375	99.5	0.6	92.0
	CNN	0.575	100.0	0.375	100.0	0.6	99.9
XOR	MLP	0.1	99.9	0.1	100.0	0.2	99.9
	CNN	0.1	100.0	0.1	100.0	0.2	100.0

et al. (2021). We define a normalized EMD performance score as

$$\text{EMD\_perf}(\mathbf{s}, \mathcal{F}^+) = 1 - \frac{\text{EMD}(\mathbf{s}, \mathcal{F}^+)}{\delta_{max}}, \tag{5}$$

where δ_{max} is the maximum Euclidean distance between any two pixels.

Remark. Note that the ground truth $\mathcal{F}^+(\mathbf{x})$ defines the set of important pixels based on the data generation process. It is conceivable, though, that a model uses only a subset of these for its prediction, which must be considered equally correct. Our explanation performance metrics do not fully achieve invariance in that respect. However, both are designed to de-emphasize the impact of false-negative omissions of features in the ground truth on performance, while emphasizing the impact of false-positive attributions of importance to pixels not contained in the ground truth.

3 Experiments

Our experiments aim to answer four main questions:

1. Which XAI methods are best at identifying truly important features as defined by the sets $\mathcal{F}^+(\mathbf{x})$?

2. Does explanation performance for each method remain consistent when moving from explaining a linear classification problem to problems with different degrees of non-linearity?

3. Does adding correlations to the background noise, through smoothing with the Gaussian convolution filter, negatively impact explanation performance?

4. How does the choice of model architecture impact explanation performance?

We generate a dataset for each scenario across a range of 20 choices of α, finding the 'sweet spot' where average test accuracy over 10 trained models is at or above 80%. Table 1 shows the resulting α values as well as the average test accuracy for each scenario, over five model trainings for datasets of size $N = 40,000$ of each scenario. For training each model and the subsequent analyses, we divide each dataset three-fold by an $90/5/5$ split into a training set $\mathcal{D}_{\text{train}}$, a validation set $\mathcal{D}_{\text{val}}$, and a test set $\mathcal{D}_{\text{test}}$. From this, we compute absolute-valued importance maps $|\mathbf{s}|$ for the intersection of test data $\mathcal{D}^{\text{test}}$ correctly predicted by every appropriate classifier. The full table of training results for finding appropriate SNRs can be seen in Appendix A.5.1. Experiments were run on an internal CPU and GPU cluster, with total runtime in the order of a matter of hours.

4 Results

Figure 2 depicts examples of absolute-valued importance maps produced for a random correctly-predicted sample for each scenario and model. Shown are results for four XAI methods (Gradient SHAP, LIME, LRP, and PatternNet respectively) for each of the three models (LLR, MLP, CNN

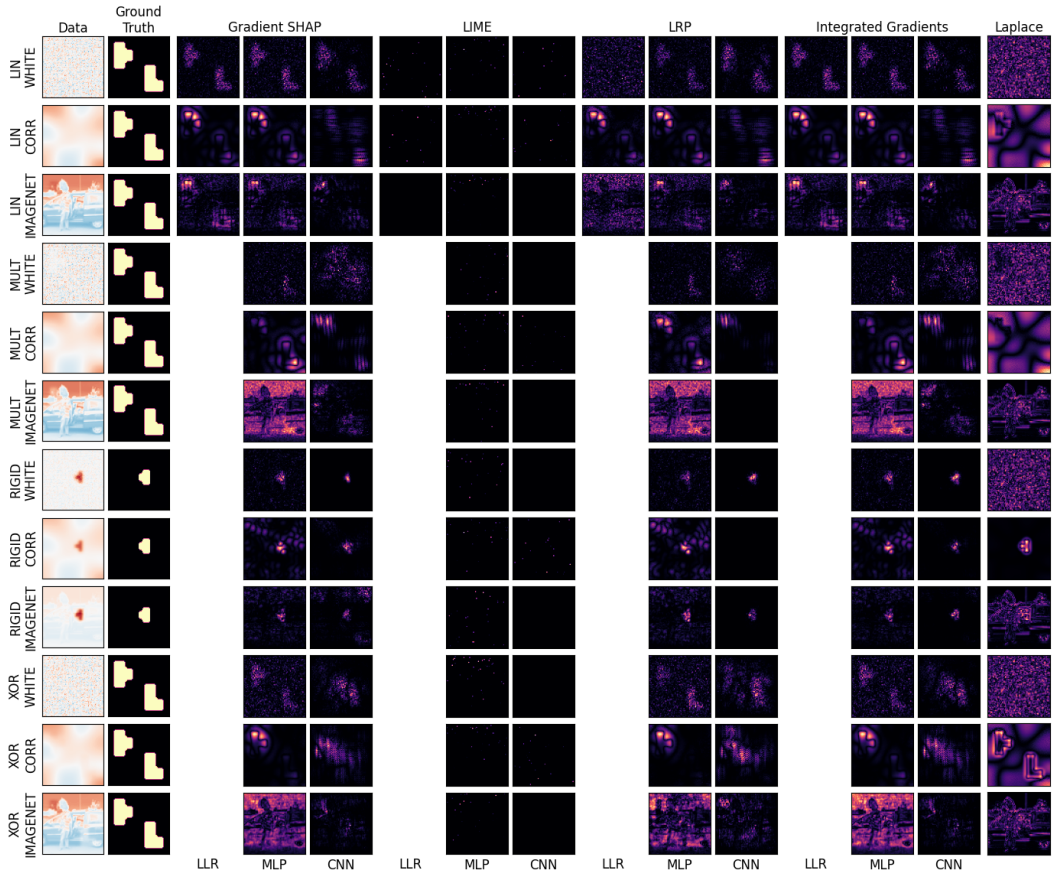


Figure 2: Absolute-valued importance maps obtained for a random correctly-predicted data sample, for selected XAI methods and baselines. Recovery of the ground truth pattern across all scenarios is best shown by XAI methods applied to a Linear Logistic Regression (LLR) model. The Multi-Layer Perceptron (MLP) tends to focus on noise in the case of ImageNet backgrounds, and LIME often fails to produce sensical explanations across all model architectures.

respectively) followed by the model-ignorant Laplace filter. Appendix A.6.1 expands on the qualitative results of the main text, and Figure 6 shows the absolute-valued *global* importance heatmaps for the LIN, MULT, and XOR scenarios, given as the mean of all explanations for every correctly-predicted sample of the given scenario and XAI method. As the RIGID scenario has no static ground truth pattern, calculating a global importance map is not possible. Figure 3 shows explanation performance of individual sample-based importance maps produced by the selected XAI and baseline methods, across five models trained for each scenario-architecture parameterization in terms of the $\mathrm{EMD\_perf}$ metric. This is shown here for 500 test samples for each scenario. Appendix A.6.2 expands on the quantitative results of the main text, detailing results for all 16 methods studied and for our Precision metric, over the full test set. In a few cases, performance tends to decrease as model complexity increases (from the simple LLR to the complex CNN architecture). One notable exception is for the RIGID scenario, where the CNN outperforms other models as expected. However, in this setting nearly all XAI methods are outperformed by a simple Laplace edge detection filter for correlated backgrounds results. The CNN also performs well in the case of the more-complicated IMAGENET backgrounds.

Within most scenario-architecture parameterizations, the performances of the studied XAI methods are relatively homogeneous, with a few exceptions. In most cases, correlated backgrounds (CORR) lead to worse explanation performance than their white noise (WHITE) counterparts, suggesting that suppressors in the smoothed background are difficult to distinguish from the class-dependent variables for most XAI methods.

Baseline methods tend to perform similarly to one another. Interestingly, their performance is on par or even superior to various XAI methods in certain scenarios. Most notably, a simple Laplace

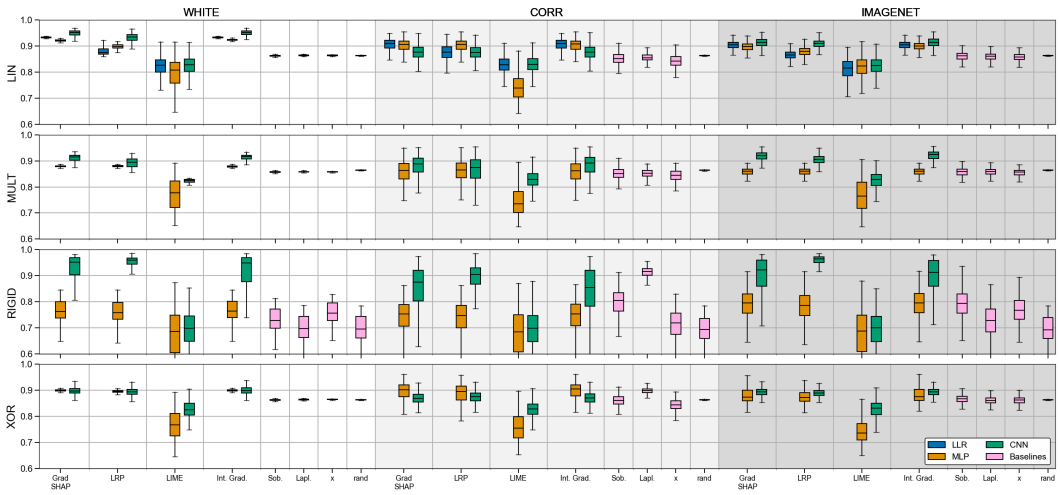


Figure 3: Quantitative explanation performance of individual sample-based feature importance maps produced by various XAI approaches and baseline methods on correctly-predicted test samples, as per the $\mathrm{EMD\_perf}$ metric. Depicted are boxplots of median explanation performance, with upper and lower quartiles as well as outliers shown. The white area (left) shows results for white background noise (WHITE), whereas the light gray shaded area (middle) shows results for the correlated background noise (CORR) scenarios and the darker gray (right) for ImageNet (IMAGENET) backgrounds.

edge detection filter outperforms nearly all other methods in the RIGID as well as the XOR scenarios, when used in combination with correlated backgrounds (CORR).

5 Discussion

Experimental results confirm our main hypothesis that explanation performance is lower in cases where the class-specific signal is combined with a highly auto-correlated class-agnostic background (CORR) compared to a white noise background (WHITE). The difficulty of XAI methods to correctly highlight the truly important features in this setting can be attributed to the emergence of suppressor variables. Importantly, the misleading attribution of importance by an XAI method can lead to misinterpretations regarding the functioning of the predictive model, which could have severe consequences in practice. Such consequences could be unjustified mistrust in the model's decisions, unjustified conclusions regarding the features related to a certain outcome (e.g., in the context of medical diagnosis), and a reinforcement of such false beliefs in human-computer interaction loops.

We have also seen that when multiple ML architectures can be used interchangeably to appropriately solve a classification problem – here with classification accuracy required to be above 80% – they may still produce disparate explanations. Architectures not only differed with respect to the selection of pixels within the correct set of important features, but also showed different patterns of false positive attributions of importance to unimportant background features. If one cannot produce consistent and sensical results for multiple seemingly appropriate ML architectures, the risk of model mistrust may be especially pronounced.

A recent survey showed that one in three XAI papers evaluate methods exclusively with anecdotal evidence, and one in five with user studies Nauta et al. (2023). Other work in the field tends to focus on secondary criteria (such as stability and robustness Rosenfeld et al. (2021-03-27); Hedström et al. (2022)) or subjective or potentially circular criteria (such as fidelity and faithfulness Gevaert et al. (2022); Nauta et al. (2023)). We doubt that such validation approaches can fully replace metrics assessing objective notions of 'correctness' of explanations, considering that XAI methods are widely intended to be used as means of quality assurance for machine learning systems in critical applications. Thus, the development of specific formal problems to be addressed by XAI methods, and the theoretical and empirical validation of respective methods to address specific problems, is necessary. In practice, a stakeholder may often (explicitly or implicitly) expect that a given XAI method identifies features that are truly related to the prediction target. In contrast to other notions

of faithfulness, this is an objectively quantifiable property of an XAI method, and we here propose various non-linear types of ground-truth data along with appropriate metrics to directly measure explanation performance according to this definition. While our work is not the first to provide quantitative XAI benchmarks (see, Tjoa & Guan, 2020; Li et al., 2021; Zhou et al., 2022; Arras et al., 2022; Gevaert et al., 2022; Agarwal et al., 2022), our work differs from most published papers in that it allows users to quantitatively assess potential misinterpretations caused by the presence of suppressor variables in data.

5.1 Limitations

One potential limitation of our work is the strictness of limiting the ground truth feature set $\mathcal{F}^+$ to the specific pixels of tetrominoes $\mathbf{a}^{\mathrm{T/L}}$ compared to, say, the set of features outlining $\mathbf{a}^{\mathrm{T/L}}$. Alternative definitions of $\mathcal{F}^+$ could be conceived, as well as new metrics, to more flexibly adapt to different potential 'explanation strategies'. While we compare a total of 16 XAI methods, the space of possible neural network architectures is too vast to be represented; therefore we only compared one MLP and one CNN architecture here. However, our experiments hopefully serve as a showcase for our benchmarking framework, which can be easily extended to other architectures. Finally, our framework serves much needed validation purposes for methods that are conceived to themselves play a role in the quality assurance of AI. As such, we expect that the benefits of our work far outweigh potential negative implications on society, if any. A possible risk, even if far-fetched, would be that one may reject a fit-for-purpose XAI method based on empirical benchmarks such as ours, which do not necessarily reflect the real-world setting and may hence be too strict.

6 Conclusion

We have used a data-driven generative definition of feature importance to create synthetic data with well-defined ground truth explanations, and have used these to provide an objective assessment of XAI methods when applied to various classification problems. Furthermore, we have defined new quantitative metrics of explanation performance and demonstrated that many popular XAI methods do not behave in an ideal way when moving from linear to non-linear scenarios. Our results show that XAI methods can even be outperformed by simple model-ignorant edge detection filters in the RIGID use case, in which the object of interest is not located in a static position. Further, we show that XAI methods may provide inconsistent explanations when using different model architectures under equivalent conditions. Future work will be to develop dedicated performance benchmarks in more complex and application-specific problem settings such as medical imaging.

References

Agarwal, C., Krishna, S., Saxena, E., Pawelczyk, M., Johnson, N., Puri, I., Zitnik, M., and Lakkaraju, H. Openxai: Towards a transparent evaluation of model explanations. *Advances in Neural Information Processing Systems*, 35:15784–15799, 2022.

Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., Samek, W., Müller, K.-R., Dähne, S., and Kindermans, P.-J. iNNvestigate neural networks! 2018.

Arras, L., Osman, A., and Samek, W. Clevr-xai: A benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40, 2022. ISSN 1566-2535.

Asano, Y. M., Rupprecht, C., Zisserman, A., and Vedaldi, A. Pass: An imagenet replacement for self-supervised pretraining without humans. *NeurIPS Track on Datasets and Benchmarks*, 2021.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015.

Bonneel, N., Van De Panne, M., Paris, S., and Heidrich, W. Displacement interpolation using lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia conference*, pp. 1–12, 2011.

Castro, J., Gómez, D., and Tejada, J. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009. ISSN 0305-0548. Selected papers presented at the Tenth International Symposium on Locational Decisions (ISOLDE X).

Conger, A. J. A Revised Definition for Suppressor Variables: A Guide To Their Identification and Interpretation , A Revised Definition for Suppressor Variables: A Guide To Their Identification and Interpretation. *Educational and Psychological Measurement*, 34(1):35–46, 1974.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Fisher, A., Rudin, C., and Dominici, F. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

Friedman, L. and Wall, M. Graphical Views of Suppression and Multicollinearity in Multiple Linear Regression. *The American Statistician*, 59(2):127–136, 2005.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D., and Crawford, K. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

Gevaert, A., Rousseau, A.-J., Becker, T., Valkenborg, D., De Bie, T., and Saeys, Y. Evaluating Feature Attribution Methods in the Image Domain. *arXiv e-prints*, art. arXiv:2202.12270, 2022.

Golomb, S. W. *Polyominoes: puzzles, patterns, problems, and packings*, volume 111. Princeton University Press, 1996.

Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., and Bießmann, F. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *NeuroImage*, 87:96–110, 2014.

Hedström, A., Weber, L., Bareeva, D., Motzkus, F., Samek, W., Lapuschkin, S., and Höhne, M. M. C. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations, 2022. URL `https://arxiv.org/abs/2202.06861`.

Kindermans, P.-J., Schütt, K. T., Alber, M., Müller, K.-R., Erhan, D., Kim, B., and Dähne, S. Learning how to explain neural networks: Patternnet and patternattribution. In *International Conference on Learning Representations*, 2018.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., and Reblitz-Richardson, O. Captum: A unified and generic model interpretability library for PyTorch. 2020.

Li, X.-H., Shi, Y., Li, H., Bai, W., Cao, C. C., and Chen, L. An experimental study of quantitative evaluations on saliency methods. In *Proceedings of the 27th ACM sigkdd conference on knowledge discovery & data mining*, pp. 3200–3208, 2021.

Lundberg, S. M. and Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc., 2017.

Mamalakis, A., Barnes, E. A., and Ebert-Uphoff, I. Carefully choose the baseline: Lessons learned from applying xai attribution methods for regression tasks in geoscience, 2022.

Montavon, G., Bach, S., Binder, A., Samek, W., and Müller, K.-R. Explaining NonLinear Classification Decisions with Deep Taylor Decomposition. *Pattern Recognition*, 65:211–222, 2017.

Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt, Y., Schlötterer, J., van Keulen, M., and Seifert, C. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Comput. Surv.*, feb 2023. ISSN 0360-0300. doi: 10.1145/3583558. URL https://doi.org/10.1145/3583558. Just Accepted.

Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Larochelle, H. Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). *The Journal of Machine Learning Research*, 22(1): 7459–7478, 2021.

Prabhu, V. U. and Birhane, A. Large image datasets: A pyrrhic win for computer vision? *arXiv preprint arXiv:2006.16923*, 2020.

Ribeiro, M. T., Singh, S., and Guestrin, C. " 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*, pp. 1135–1144, 2016.

Rosenfeld, E., Ravikumar, P., and Risteski, A. The Risks of Invariant Risk Minimization. 2021-03-27.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning Important Features Through Propagating Activation Differences. 2017.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014.

Springenberg, J., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic Attribution for Deep Networks. In *ICML*, 2017.

Tjoa, E. and Guan, C. Quantifying Explainability of Saliency Methods in Deep Neural Networks. 2020.

Wagner, P., Strodthoff, N., Bousseljot, R.-D., Kreiseler, D., Lunze, F. I., Samek, W., and Schaeffter, T. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):154, 2020.

Wilming, R., Budding, C., Müller, K.-R., and Haufe, S. Scrutinizing XAI using linear ground-truth data with suppressor variables. *Machine Learning*, 2022.

Yang, K., Yau, J., Fei-Fei, L., Deng, J., and Russakovsky, O. A study of face obfuscation in imagenet. *CoRR*, abs/2103.06191, 2021. URL https://arxiv.org/abs/2103.06191.

Zeiler, M. D. and Fergus, R. Visualizing and Understanding Convolutional Networks. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T. (eds.), *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pp. 818–833. Springer International Publishing, 2014. ISBN 978-3-319-10590-1.

Zhou, Y., Booth, S., Ribeiro, M. T., and Shah, J. Do feature attribution methods correctly attribute features? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9623–9633, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] **We discuss this in Section 5.1.**

(c) Did you discuss any potential negative societal impacts of your work? [Yes] **We discuss this in Section 5.1. As our work focuses on validation of XAI methods which can (and are hoped to eventually be) applied in potentially critical domains, we believe that there are a very limited amount of negatives and plenty of positives to such a benchmark.**

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] **Code and instructions on how to run it are provided in the supplementary material in Section A.2.**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] **All training details were listed in Sections 2 and 3 as well as supplementary materials Sections A.5 and B.1.**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] **We report error in the box and whisker plot of Figure 3 as well as quantitative results shown in the supplementary material in Sections A.6.2.**
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] **We did not state the specific runtime in the main text but did mention that experiments were run on an internal cluster at the end of Section 3. We also expand on this in the supplementary material in Section A.3.**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] **We make use of the ImageNet dataset as one of the three background types and cite the authors in the main text in Section 2 (page 3).**
   (b) Did you mention the license of the assets? [Yes] **We do not specify the license in the main text, but we specify the license in the supplementary material in Sections A.1 and B.1.**
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] **We provide assets as code to generate data with a fixed random seed in the supplementary material in Section A.2**
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] **Supplementary materials section A.1 outlines how we agreed to the license required to use the ImageNet data.**
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] **Supplementary materials section A.1 outlines how we agreed to the license required to use the ImageNet data.**
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

The authors confirm that we bear all responsibility in case of violation of rights of any kind in the data and results shown in this work.

A.1 ImageNet

We sample data from the ImageNet-1k subset Deng et al. (2009), following the license specified here `https://image-net.org/download.php`.

In the ImageNet-1k subset, there are only three people categories (scuba diver, bridegroom, and baseball player) included in the 1,000 classes, versus 2,832 people categories in the full set. There is also the possibility of people-related images co-existing in images of other classes, which has been noted Prabhu & Birhane (2020). Data from these classes can be discarded if necessary.

Alternatives can be used directly as a background type here to replace ImageNet, for example PASS Asano et al. (2021), published in the NeurIPS Datasets and Benchmarks track in 2021. This ImageNet replacement dataset only contains images with a CC-BY license, as well as containing no images of humans. Replacement of ImageNet images in our work is as simple as placing images in the respective folder for the data generation step to handle, following the instructions outlined in the next sub-section and the corresponding GitHub repository.

For more information, see the datasheet in Section B.1.

A.2 Code and Data

All code for generating data and performing model training and XAI analysis is available on GitHub: `https://github.com/braindatalab/xai-tris`. There, we provide instructions on how to run each step of the analysis pipeline as well as detailing corresponding configuration fields.

To download the ImageNet data, we made an account and agreed the license terms on `https://huggingface.co/datasets/imagenet-1k` and subsequently downloaded the data. Here, we used the validation set as the $N = 50,000$ set suited the volume requirement for our analysis. We of course advise anyone planning to do similar analysis on a model pre-trained with ImageNet data to use the $N = 100,000$ test set instead.

Each $N = 40,000$ dataset generated for a given classification scenario and background type pair is 1.52 GB in size. For the lower-dimensional 8×8-px data and experiments shown in supplementary materials Section A.7, generating $N = 10,000$ datasets for all eight scenario and background type pairs is around 62 MB in total size, and was combined in one file due to this much lower volume requirement. Each scenario's dataset is saved as a file `SCENARIO_JdKp_`α`_BACKGROUND.pkl` containing a python dictionary

`{SCENARIO_JdKp_`α`_BACKGROUND : DataRecord(...)},`

where `SCENARIO={linear, multiplicative, translations_rotations, xor}` and `BACKGROUND={white, correlated, imagenet}`. Image scale `Jd={1,8}d` is the scaling of the image dimensionality d from the original 8×8-px images to the 64×64-px images shown in the main text, pattern scale `Kp={1,4,8}p` is the scaling of the tetromino pattern (width in pixels), and $0.0 \leq \alpha \leq 1.0$ parameterizes the signal-to-noise ratio.

`DataRecord` is a Python `namedtuple()` collection specified as

```
DataRecord = namedtuple('DataRecord', 'x_train y_train x_val y_val x_test y_test
     masks_train masks_val masks_test').
```

Each field can be accessed programmatically via the name, for example `DataRecord.x_test` returns the test data $\mathbf{x}_{\text{test}}$ of the dataset. The `masks` fields are the tetromino pattern masks which form the ground truth for explanations.

A.3 Compute

Experiments were run on a cluster consisting of four Nvidia A40 GPUs, where each model training took roughly between three and twenty minutes to complete, depending on architecture. Time

estimation for running XAI methods is more rough to calculate and depends on each method, but in total for all models and methods for a given scenario's $N = 2,000$ test set, this took between 24 and 48 hours of compute time per GPU on the cluster. Quantitative analysis took roughly a further 24 hours of compute per scenario on a cluster of AMD EPYC 7702 CPUs, with six threads used for each of the 12 scenarios.

Due to smaller compute requirements, we can also recommend that if one wants to explore the code and data with smaller compute requirements, the 8×8-px data shown in supplementary materials Section A.7 is also representative of a strong benchmark for XAI methods. Code and instructions to run it have also been provided in the GitHub repository linked in the above supplementary materials Section A.2.

A.4 Data

Here, we expand on Figure 1 with Figure 4, which shows an example of each scenario across four choices of signal-to-noise ratio (SNR), parameterized by α.

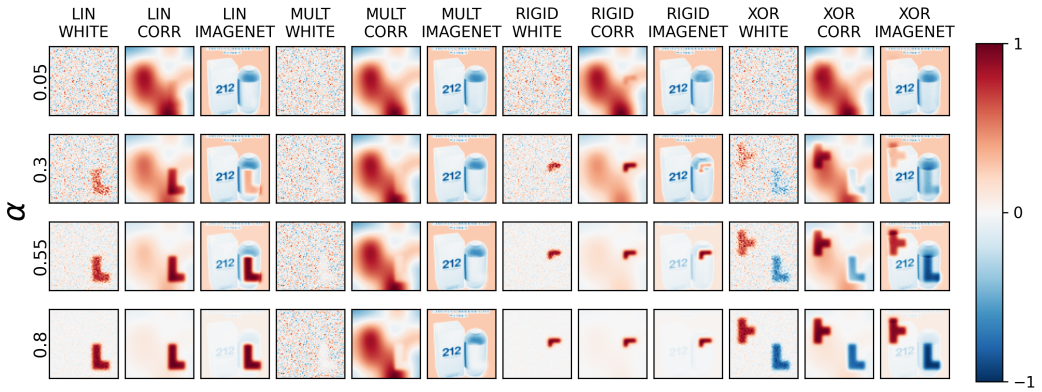


Figure 4: Examples of generated data samples for each scenario, showing how a generated sample of Class #0 (where y=0) for each scenario varies across four signal-to-noise ratios (SNRs) α.

A.5 Explanation Methods and Model Training

Here, we detail the full suite of 16 XAI methods used in our analysis, with a brief description along with the reference and any parameterization details. In the main text, we focus on XAI methods available with the Captum (Kokhlikyan et al., 2020) framework for explaining PyTorch models. We also make use of methods available in the iNNvestigate (Alber et al., 2018) library, through training equivalent models for the Keras framework.

Table 2: XAI Methods used with a brief description of each method and the implementation details, including the software framework used and any specific parameterization including the baseline input used, if applicable.

XAI Method	Description	Implementation Framework, Parameterization	Reference
Permutation Feature Importance (PFI)	Measures the change in prediction error of the model after permuting each feature's value	Captum, Default	Fisher et al. (2019)
Integrated Gradients	Computes gradients along the path from a baseline input to the input sample, and cumulates these through integration to form an explanation	Captum, Default, Zero input baseline	Sundararajan et al. (2017)

Saliency	Computes the gradients with respect to each input feature	Captum, Default	Simonyan et al. (2014)
Guided Backpropagation	Computes the gradient of the output with respect to the input, but ensures only non-negative gradients of ReLU functions are backpropagated	Captum, Default	Springenberg et al. (2015)
Guided GradCAM	Computes the element-wise product of guided backpropagation attributions with respect to a class-discriminative localization map in the final convolution layer of a CNN. This produces a coarse importance map for the target class as an explanation, the same size as the convolutional feature map, rather than pixel-wise over the whole image	Captum, Default	Selvaraju et al. (2017)
Deconvolution	Uses a Deconvolutional network to map features to pixels. An explanation is produced by computing the gradient of the target output, only backpropagating non-negative gradients of ReLU functions	Captum, Default	Zeiler & Fergus (2014)
DeepLift	Compares the difference between the activation of each neuron and its 'reference activation', and produces an explanation based on this difference	Captum, Default, Zero input baseline	Shrikumar et al. (2017)
Shapley Value Sampling	Approximates Shapley values by repeatedly sampling random permutations of input features and calculating the contribution of each feature to the prediction. An explanation is produced across an average of many samplings	Captum, Default, Zero input baseline	Castro et al. (2009)
Gradient SHAP	Approximates Shapley values by computing the expected values of gradients when randomly sampled from the distribution of baseline samples	Captum, Default, Zero input baseline	Lundberg & Lee (2017)
Kernel SHAP	Approximates Shapley values through the use of LIME, setting the loss function, weighting kernel, and regularization term in accordance with the SHAP framework	Captum, Default, Zero input baseline	Lundberg & Lee (2017)
Deep SHAP	Approximates Shapley values through the use of DeepLift. Computes the DeepLift attribution for each input sample with respect to each baseline sample, in accordance with the SHAP framework	Captum, Default, Zero input baseline	Lundberg & Lee (2017)
Locally-interpretable Model Agnostic Explanations (LIME)	Learns a linear surrogate model locally to an individual prediction, perturbing and weighting the dataset in the process, and then builds an explanation by interpreting this local model	Captum, Default	Ribeiro et al. (2016)
Layer-wise Relevance Propagation (LRP)	Propagates the model output back through the network as a measure of relevance, decomposing this score for each model in each layer based on their trained weight and activation	Captum, Default	Bach et al. (2015)
Deep Taylor Decomposition (DTD)	Applies a Taylor decomposition from a specified root point to approximate the sub-functions of a network, building explanations by applying this backward from the network output to input variables	iNNvestigate, Default	Montavon et al. (2017)

PatternNet	Estimates activation patterns per neuron through signal estimator $S_{\mathbf{a}+-}$ and back-propagates this through the network. The explanation is given as a projection of the signal in input space	iNNvestigate, Default	Kindermans et al. (2018)
PatternAttribution	Utilises the theory of PatternNet to estimate the root point of the data for DTD, and yields the attribution $\mathbf{w} \odot \mathbf{a}_+$ for weight vector $\mathbf{w}$ and positive activation patterns $\mathbf{a}_+$. The explanation is given as the neuron-wise contribution of the signal to the classification score	iNNvestigate, Default	Kindermans et al. (2018)

A.5.1 Training

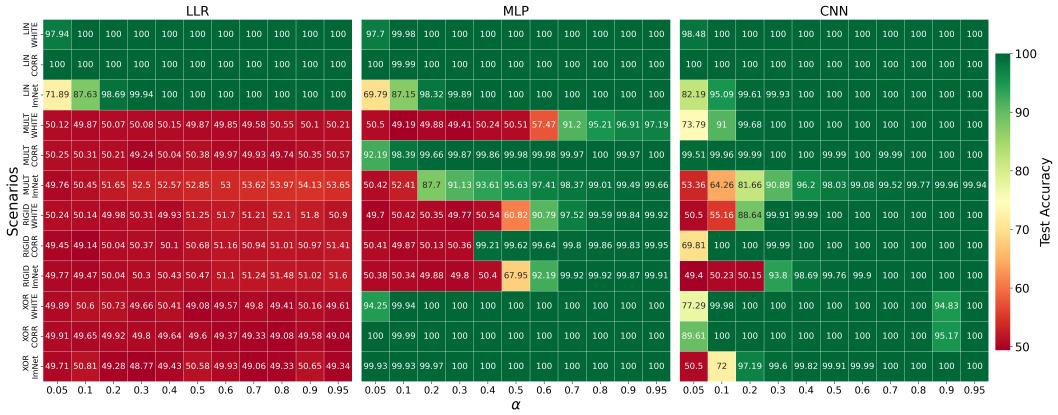


Figure 5: Average test accuracy over 10 model trainings for each problem scenario and model architecture, for a fixed range of signal-to-noise ratios (SNRs). As expected, the Linear Logistic Regression (LLR) model cannot perform above chance level for non-linear scenarios. The Convolutional Neural Network (CNN) outperforms the Multi-Layer Perceptron (MLP) for the RIGID (translations and rotations of tetrominoes) scenarios as expected, perhaps due to the invariance under these properties for this architecture.

A.6 Explanation Performance

This section further elaborates results of our experiments on validating the performance of XAI methods. In Figures 7 and 8 we also show methods available in the iNNvestigate Alber et al. (2018) library, through training equivalent models for the Keras framework. We note that there were some issues in convergence for CNN models for the XOR scenarios with the required Keras framework, even under seemingly equivalent conditions such as fixed random seeds and He-normal weight initialization. Our model architectures have been chosen as a showcase of the datasets and benchmarks of this work, and other architectures may have better or worse performance on the same XAI methods, but this was not a focus of this work. As such, we do not show the corresponding results for these methods (PatternNet, PatternAttribution, Deep Taylor Decomposition) in the XOR-CNN problem setting, so to promote a fair comparison of methods.

A.6.1 Qualitative Results

In Figure 6, we can see absolute-valued global importance maps for selected XAI methods and baselines, calculated as the mean importance value over all correctly predicted samples. RIGID scenarios involving translations and rotations of the tetromino signal pattern are not included as they have no fixed ground truth position.

A.6.2 Quantitative Results

In Figures 7 and 8 we can see the full quantitative results for the $\mathrm{EMD\_perf}$ and Precision metrics respectively, across all XAI methods and baselines. We can also see results for the PatternNet, PatternAttribution, and Deep Taylor Decomposition (DTD) methods, which are part of the Keras-based iNNvestigate framework Alber et al. (2018).

A.7 8x8 Benchmarks

The benchmark was originally designed around 8×8-px tetromino images, scaled up to 64×64-px with the inclusion of the ImageNet data as a third background type. This was done to improve the robustness and real-world applicability of the datasets and benchmarks present in this work. The original results for the 8×8-px data with 1-px thick tetrominoes can be seen in this section. Figure 9 shows example data for both classes and also across a range of four α values. For CORR backgrounds, we set $\sigma_{\text{smooth}} = 3.0$ for the smoothing filter, and no pattern smoothing was incorporated. Here, each scenario was constructed with sample size $N = 10,000$ and with an $80/10/10$ train/val/test split, with 25 datasets per scenario being used for analyses.

The Linear Logistic Regression (LLR) model in these experiments was the same single-layer neural network with two output neurons and a softmax activation function. The Multi-Layer Perceptron (MLP) similarly has four fully-connected layers and Rectified Linear Unit (ReLU) activations, and each of the fully-connected hidden layers halves the input size, i.e. [64, 32, 16, 8]. The two-neuron output layer was once again softmax-activated. Finally, the Convolutional Neural Network (CNN) was defined as four blocks of ReLU-activated convolutional layers followed by a max-pooling operation, with a softmax-activated two-neuron output layer. The convolutional layers are specified with four filters, a kernel size of two, a stride of one, and padding such that the input and output shapes match. This padding technique was used to improve pixel utilization across each convolution, as well as to mitigate shrinking outputs of the already relatively small images, by adding extra filler pixels (set to values of zero) around the edge of each image. The max-pooling layers are defined with a kernel size of two and a stride of two. As with the CNN architecture of the main text, some popular CNN architecture features (such as batch normalization) are unavailable here due to lack of implementation support by some XAI methods.

Figure 10 shows the training results across ten α values along with Table 3 which shows the chosen α values used for analysis. Each network was trained over 500 epochs using the Adam optimizer without regularization, with a learning rate of 0.004 for the LIN, MULT, and XOR scenarios, and 0.0004 for the RIGID scenario.

Figures 11 and 12 show qualitative results for local and global explanations respectively, and Figures 14 and 15 show quantitative results for the $\mathrm{EMD\_perf}$ and Precision metrics respectively.

Table 3: Results of the model training process for each classification setting, model architecture, and background type in the 8×8-px setting. These results are depicted as chosen Signal-to-noise ratios (SNRs), parameterized by α, as well as the average test accuracy (ACC, %).

		WHITE		CORR	
		α	ACC	α	ACC
LIN	LLR	0.1800	88.9	0.0125	99.9
	MLP	0.1800	87.9	0.0125	99.9
	CNN	0.1800	83.0	0.0125	86.4
MULT	MLP	0.7000	93.6	0.1000	99.4
	CNN	0.7000	83.1	0.1000	90.6
RIGID	MLP	0.6500	91.9	0.2000	99.9
	CNN	0.6500	93.7	0.2000	88.8
XOR	MLP	0.3500	99.5	0.1500	100.0
	CNN	0.3500	95.2	0.1500	99.5

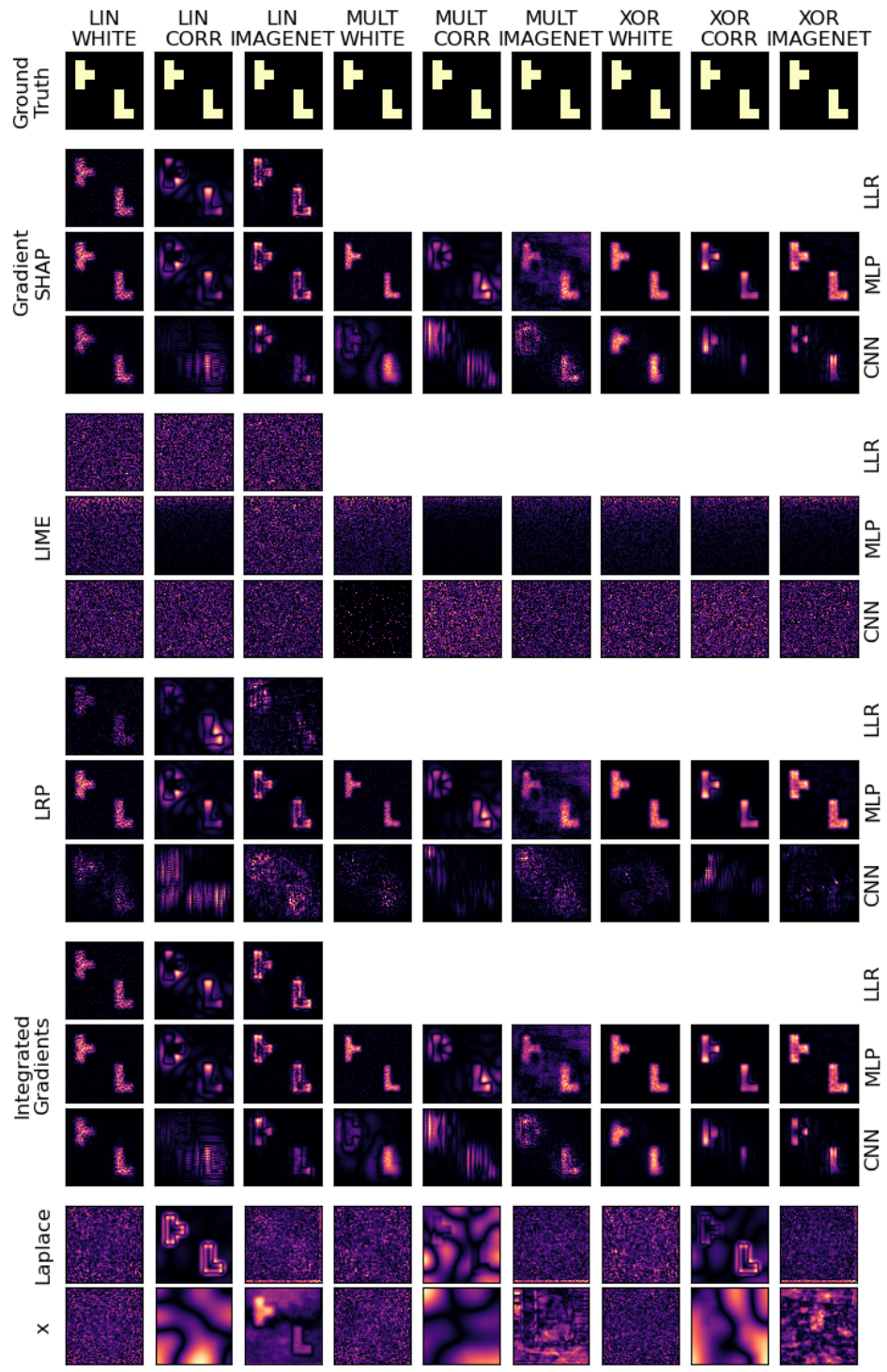


Figure 6: Absolute-valued global importance maps calculated as the mean importance value over all correctly predicted samples, for selected XAI methods and baselines. RIGID scenarios involving translations and rotations of the tetromino signal pattern are not included as they have no fixed ground truth position. CORR scenarios with correlated background can be seen to produce noisier global importance maps, suggesting that this setting induces suppressor variables in the background, which are difficult for XAI methods to distinguish from the true signal pattern. Results for the ImageNet background also tend to show noisier global explanations, suggesting that the complicated and variable features of this background type present a challenge to the models and corresponding XAI methods. LIME fails to produce any meaningful explanations yet again, suggesting an issue with this scale of image. The results of supplementary materials Section A.7 show better performance for LIME with the smaller 8×8-px image benchmark.

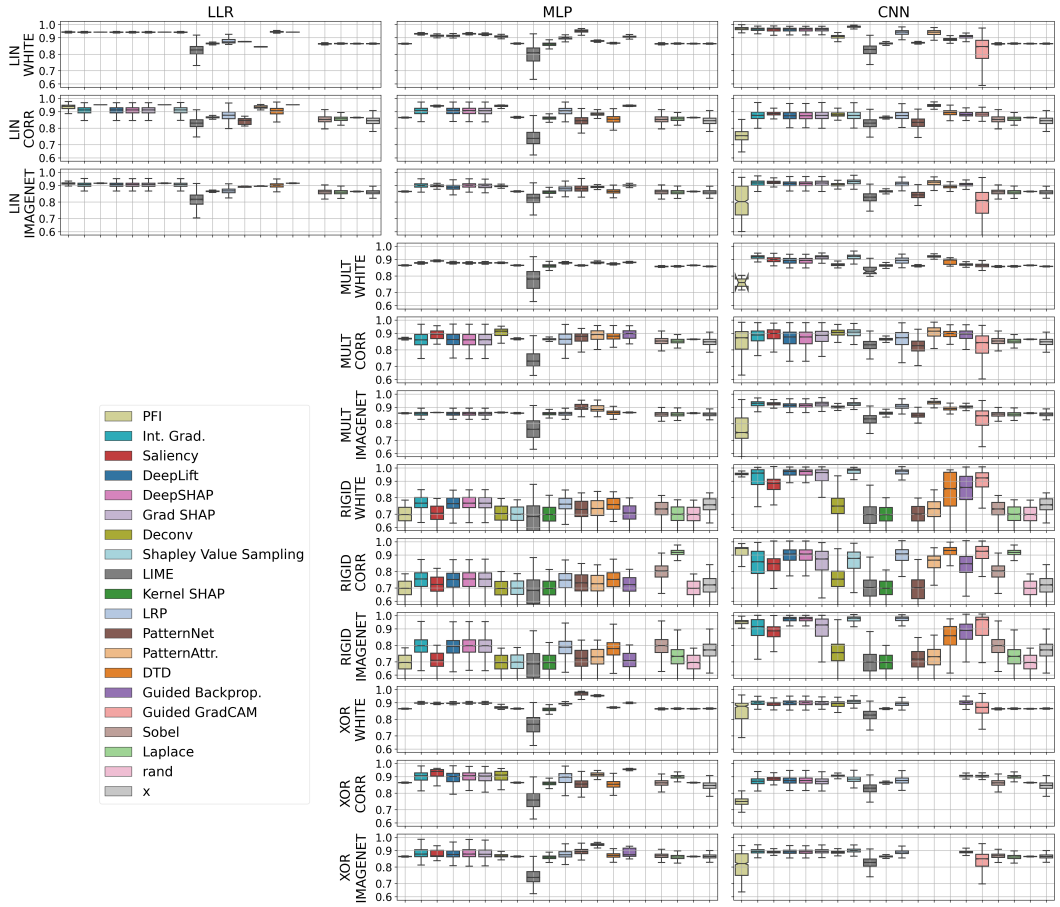


Figure 7: EMD_perf metric based on the Earth Mover's Distance (EMD) for every XAI method tested, separated by model architecture and depicted as boxplots of median and quartile performance scores. Guided GradCAM is only implemented for CNN architectures, and Keras models required for PatternNet, PatternAttribution, and Deep Taylor Decompostion (DTD) struggled to converge for the XOR scenarios as stated above, so these are excluded from the corresponding sub-plots. Some methods see a drop in explanation performance as model complexity increases, from the Linear Logistic Regression (LLR) model to a Convolutional Neural Network (CNN). In the RIGID CORR case, the model-ignorant Laplace filter outright performs the best for explanations of MLP decisions and nearly so for the CNN. SHAP variants DeepSHAP, GradSHAP, and Shapley Value Sampling perform very similarly to one another in most cases across all model types, despite being formulated to target particular problems. No XAI method performs outright the best across all scenarios.

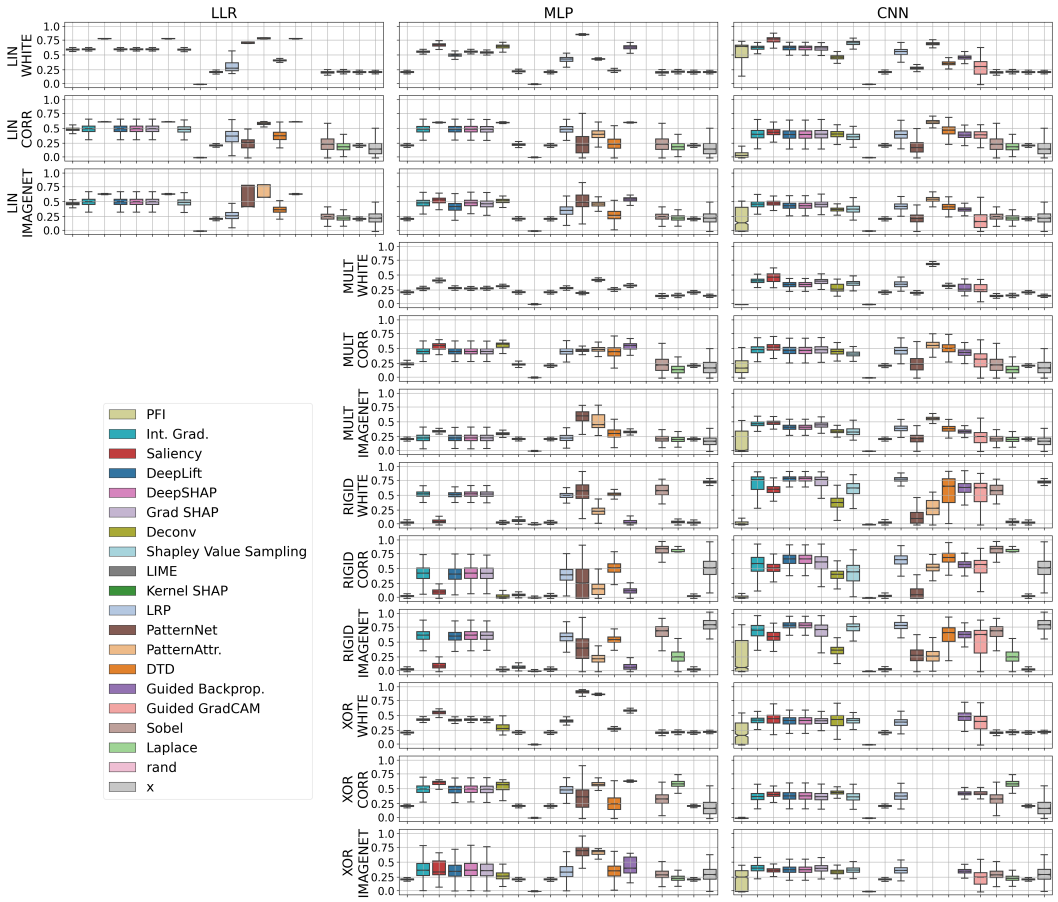
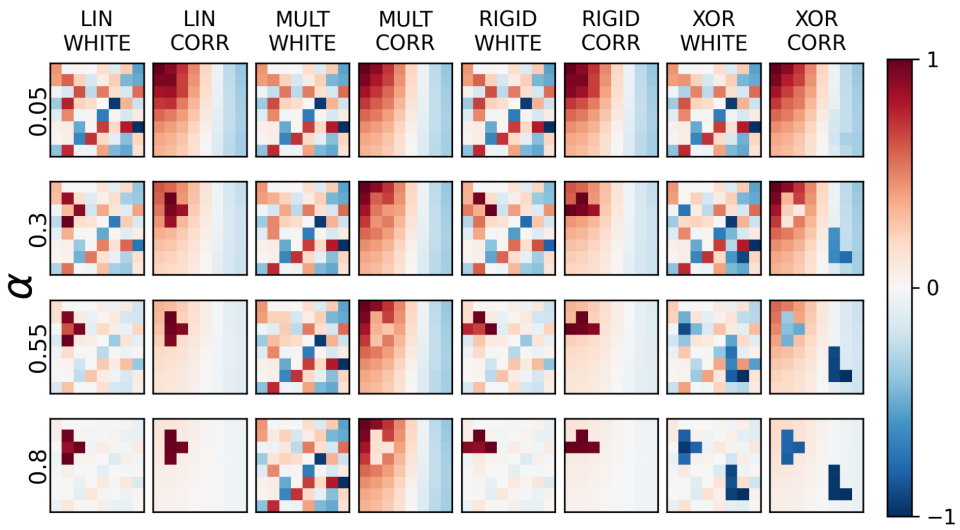
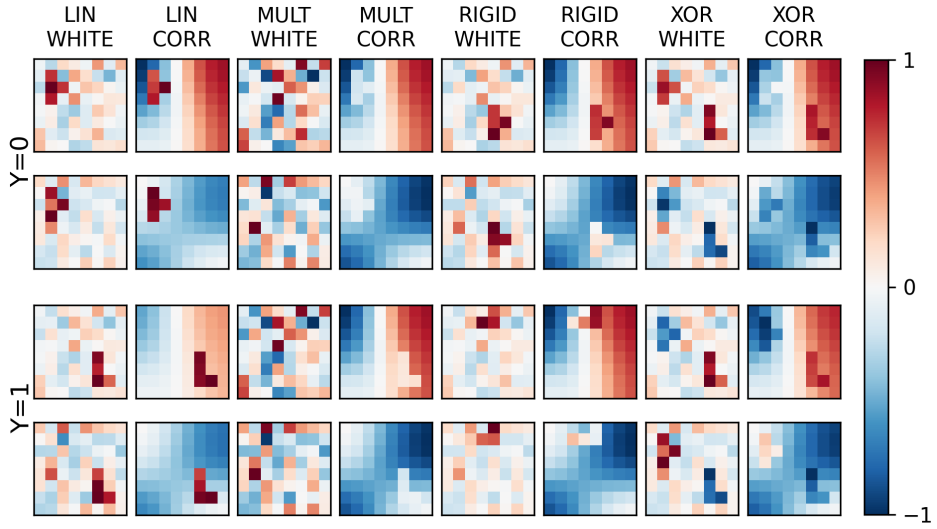


Figure 8: Precision score for every XAI method tested, separated by model architecture and depicted as boxplots of median and quartile performance scores. Most methods outperform the baseline methods for most model-scenario parameterization pairs. The 'x' method, using input data as reference point of explanation, performs better for scenarios with higher signal-to-noise ratio (SNR), as the tetromino patterns will, on average, be more salient in the data there, thus present higher precision on average. Namely, the RIGID WHITE and IMAGENET scenarios generally require a higher SNR to be appropriately modeled. PatternNet and PatternAttribution, designed to nullify the influence of suppressor variables, generally perform well in the LIN and XOR WHITE cases, similar to the results shown by Wilming et al. (2022), however these methods struggle in various other non-linear problem scenarios. LIME struggles across all scenarios, but performs better in the results shown in supplementary materials Section A.7, with the smaller 8×8-px image benchmark. Similarly to the results of 7, no XAI method performs outright the best across all scenarios.



(a) One generated sample of Class #0 (where y=0) for four different SNRs α.



(b) Two generated samples of each class per scenario.

Figure 9: Examples of generated 8×8-px data samples for each scenario, showing how an example for each scenario varies across four signal-to-noise ratios (SNRs) α (top).

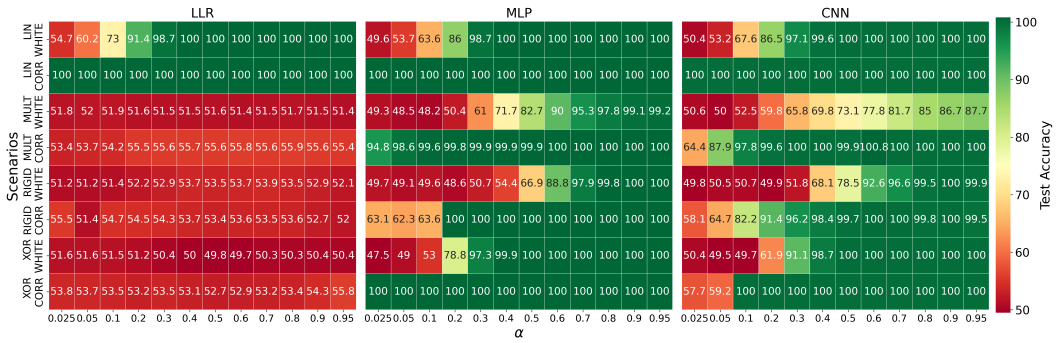


Figure 10: Average test accuracy over 10 model trainings for each problem scenario and model architecture of the 8×8-px setting, for a fixed range of signal-to-noise ratios (SNRs). As expected, the Linear Logistic Regression (LLR) model cannot perform above chance level for non-linear scenarios. The Convolutional Neural Network (CNN) would be expected to outperform the Multi-Layer Perceptron (MLP) for the RIGID (translations and rotations of tetrominoes) scenarios due to the invariance under these properties for this architecture. However, performance is comparable, with the MLP obtaining an average test accuracy above the 80% threshold at a lower SNR than the CNN. This may be partially due to the compromise in the architecture of the CNN, where we were not able to use Batch Normalization due to incompatibility with some XAI frameworks and methods.

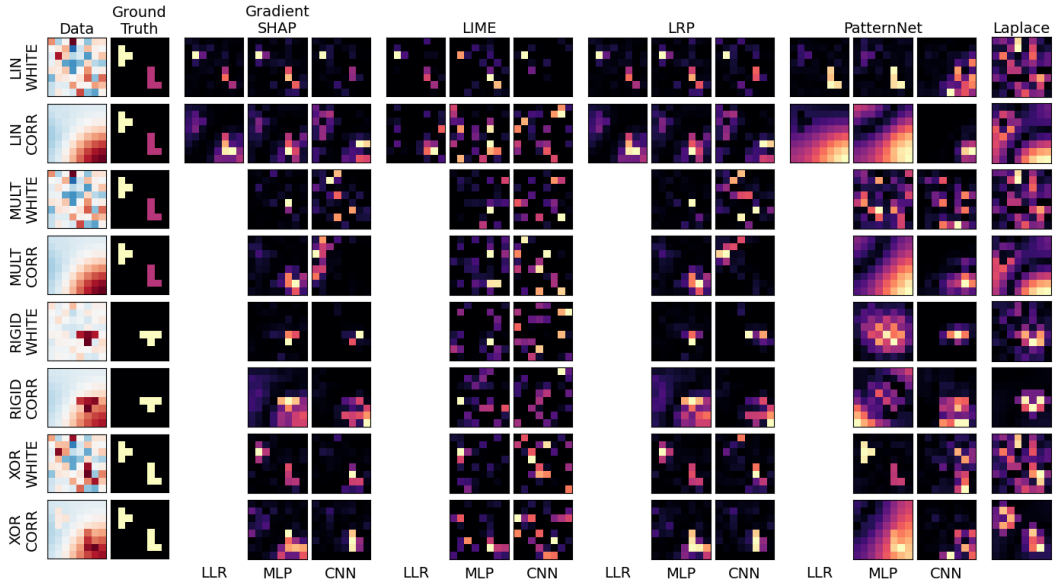


Figure 11: Absolute-valued importance maps obtained for a random correctly-predicted 8×8-px data sample, for selected XAI methods and baselines. Recovery of the ground truth pattern across all scenarios is best shown by XAI methods applied to a Linear Logistic Regression (LLR) model.

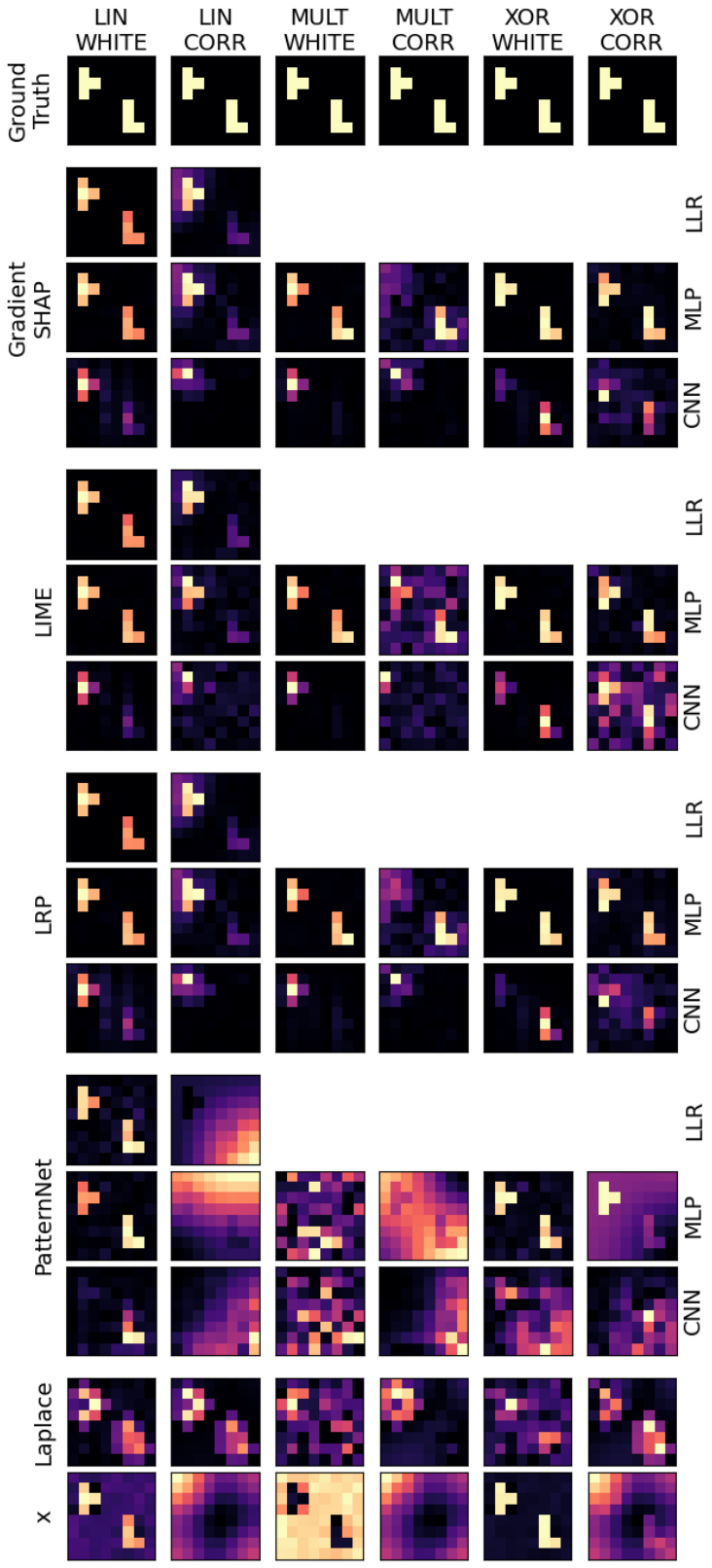


Figure 12: Absolute-valued global importance maps calculated as the mean importance value over all correctly predicted 8×8-px scenario samples, for selected XAI methods and baselines. RIGID scenarios involving translations and rotations of the tetromino signal pattern are not included as they have no fixed ground truth position. CORR scenarios with correlated background can be seen to produce noisier global importance maps, suggesting that this setting induces suppressor variables in the background, which are difficult for XAI methods to distinguish from the true signal pattern.

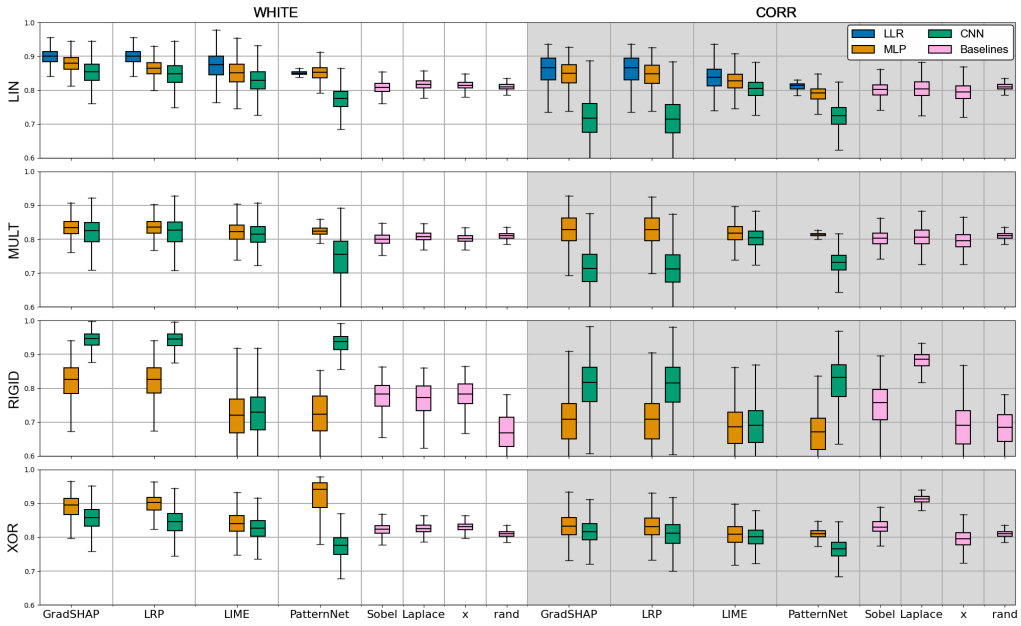


Figure 13: Quantitative explanation performance of individual sample-based feature importance maps produced by various XAI approaches and baseline methods on correctly-predicted 8×8-px scenario test samples, as per the $\mathrm{EMD\_perf}$ metric. Depicted are boxplots of median explanation performance, with upper and lower quartiles as well as outliers shown. The white area (left) shows results for white background noise (WHITE), whereas the gray shaded area (right) shows results for the correlated background noise (CORR) scenarios. Explanation performance decreases as model complexity (from LLR to MLP to CNN) increases, with the exception of the RIGID scenarios, where the CNN is better suited to the non-static ground truth patterns present. Unlike results seen for linear data in Wilming et al. (2022), PatternNet and PatternAttribution do not outright outperform other XAI methods for most configurations.

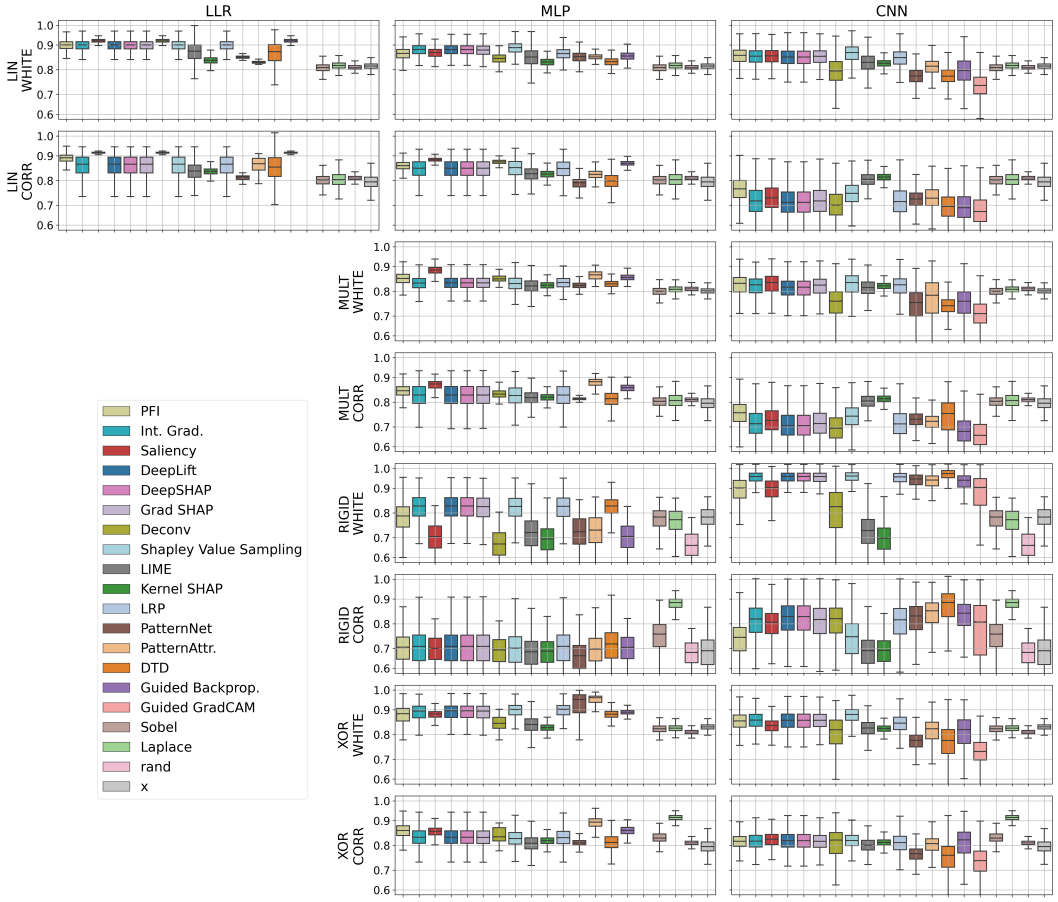


Figure 14: EMD_perf metric based on the Earth Mover's Distance (EMD) for every XAI method tested in the 8×8-px setting, separated by model architecture and depicted as boxplots of median and quartile performance scores. Consistent with the results of Figure 13, explanation performance tends to decrease as model complexity increases, from the Linear Logistic Regression (LLR) model to a Convolutional Neural Network (CNN). An exception is seen for RIGID scenarios where most XAI methods outperform the Multi-Layer Perceptron (MLP) equivalent. In this case, the model-ignorant Laplace filter performs the best across both architectures.

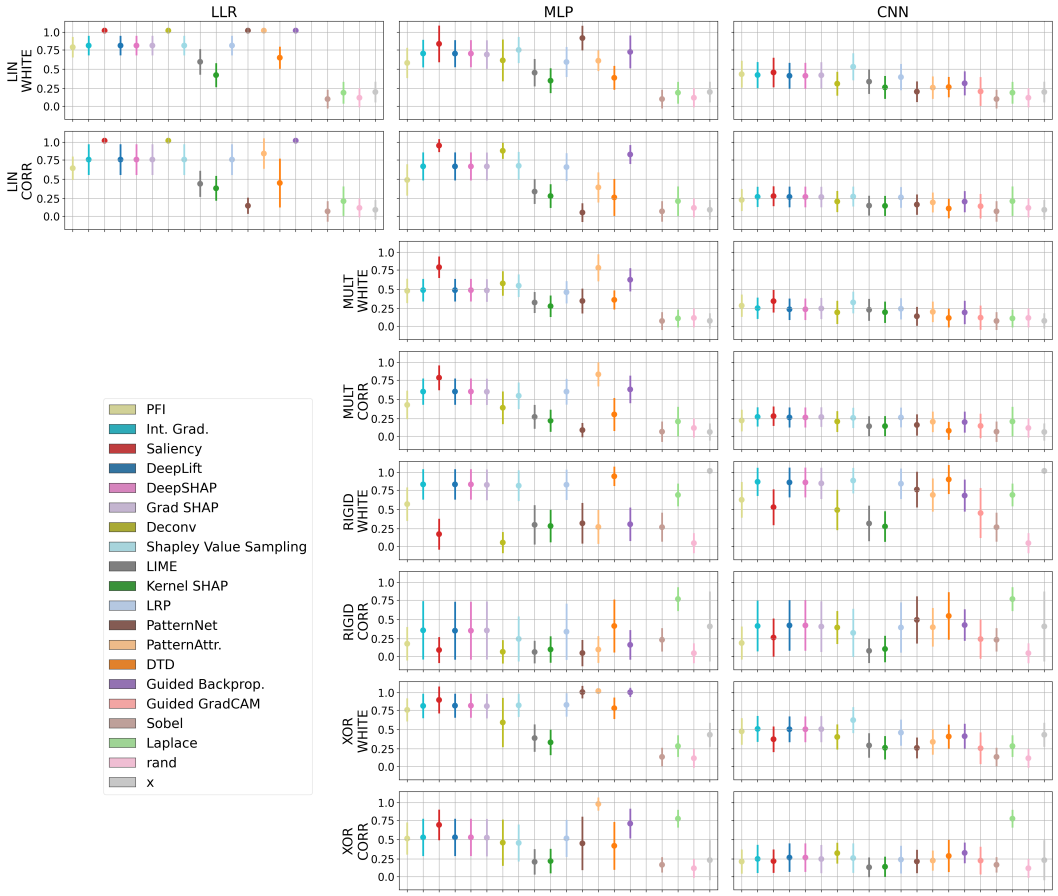


Figure 15: Precision score for every XAI method tested in the 8×8-px setting, separated by model architecture and depicted as mean and standard deviation performance scores. Most methods outperform the baseline methods for most model-scenario parameterization pairs. The 'x' method, using input data as reference point of explanation, performs better for scenarios with higher signal-to-noise ratio (SNR), as the tetromino patterns will, on average, be more salient in the data there, thus present higher precision on average. Namely, the RIGID and WHITE scenarios generally require a higher SNR to be appropriately modeled. Outside of this, performance for XAI methods for the Convolutional Neural Network (CNN) is comparable to baseline methods.

B Datasets and Benchmarks Track Supplementary Material

B.1 Datasheet (template by Gebru et al. (2021))

Motivation

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

The dataset was created to benchmark the performance of explainable artificial intelligence (XAI) methods in an objective manner, ultimately as a first step towards creating a broad 'benchmark suite' for testing and certifying XAI methods across a range of data scenarios and objective quantitative metrics. We craft four classification scenarios with explicitly known class-conditional features, which serve as the ground truth for explanations. The field of XAI has produced many methods which are currently not empirically validated with respect to the correctness of their explanations, and so there is a need for such a benchmark. Current benchmarks in the field may not sufficiently provide an objective and direct evaluation of explanation performance, and do not account for the presence of suppressor variables, which have no relation to the prediction target yet still influence the prediction. They also do not evaluate the same breadth of methods as we do here.

It has also been shown that XAI methods are susceptible to suppressor variables Wilming et al. (2022), and this dataset goes beyond such work by introducing three non-linear classification scenarios across three noisy background types. This enables us to study the effect of suppressor variables on the quality of explanation performance in a controlled manner.

Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

This dataset was created by members of QAI Labs who are employed by the Physikalisch-Technische Bundesanstalt [1] (the national metrology institute for the Federal Republic of Germany) and the Technische Universitaet Berlin [2]. This was created as part of the "Metrology for Artificial Intelligence in Medicine (M4AIM)" programme [3] in the frame of the "QI-Digital" initiative [4], as a first step towards creating a broad 'benchmark suite' for testing and certifying XAI methods across a range of data scenarios and objective quantitative metrics.

Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.

This result is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 758985), the German Federal Ministry for Economic Affairs and Climate Action (BMWK) within the "Metrology for Artificial Intelligence in Medicine (M4AIM)" program in the frame of the "QI-Digital" initiative, and the Heidenhain Foundation.

Any other comments?

Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.

Dataset instances comprise of single-channel images of combinations between different types of signal patterns and noisy backgrounds, forming various binary classification scenarios. Here, we make use of tetrominoes, geometric shapes consisting of four blocks, as the signal pattern. We use three types of backgrounds: white noise (WHITE), white noise smoothed by a Gaussian filter (CORR), and pre-processed samples from the ImageNet-1k subset Deng et al. (2009). The latter background type consists of images across 1,000 classes of objects from the WordNet hierarchy, with associated concepts ranging from animals, objects, and other concepts. Pre-processing

[1] https://www.ptb.de/cms/en/ptb/fachabteilungen/abt8/fb-84/ag-844.html
[2] https://www.tu.berlin/uniml
[3] https://www.m4aim.ptb.de/m4aim/m4aim
[4] https://www.qi-digital.de/en/

to single-channel zero-centered images was described and code has been provided.

How many instances are there in total (of each type, if appropriate)?

For each of the four binary classification scenarios and three background types, we create balanced datasets of total size $N = 40,000$ samples. This can be scaled up or down in our provided simulation environment. For the ImageNet-1k subset, used as the third background type, there is a total of $1,431,167$ images that can be sampled from.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

The white noise background type (and the corresponding smoothed noise background) are samples from a zero-mean Gaussian distribution. For the ImageNet-1k subset background type, we sample $N = 40,000$ from a total of $1,431,167$ images. These were taken from the $N = 50,000$ validation set of the ImageNet-1k dataset. This particular set was chosen as the data volume matched the needs for our analysis. As these images are just used as a complex and varied background type for our own pre-specified binary classification scenarios, we just ensure representativeness of the class distribution in each classification scenario. Each of our training/validation/test data splits is class-balanced and with every case equally represented for the XOR scenario, where there are two tetrominoes per sample with two possible combinations of manipulation sign (+/-) per class. This is described in more detail in the main text.

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.

Dataset instances comprise of single-channel 64×64-px images of combinations between different types of signal patterns and noisy backgrounds, forming various binary classification scenarios. Here, we make use of tetrominoes, geometric shapes consisting of four blocks, as the signal pattern. These form the ground truth masks for benchmarking explanations. We use three types of backgrounds: white noise (WHITE), white noise smoothed by a Gaussian filter (CORR), and pre-processed samples from the ImageNet-1k subset Deng et al. (2009). The latter background type consists of images across 1,000 classes of objects from the WordNet hierarchy, with associated concepts ranging from animals, objects, and other concepts. Pre-processing to single-channel zero-centered images was described and code has been provided.

Is there a label or target associated with each instance? If so, please provide a description.

Generated data is part of a particular binary classification scenario, where in all cases, the label is specified as $Y = 0$ or $Y = 1$.

Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.

No.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.

Each sample is generated independently, with the only relationship being the type of tetromino present in the sample. This, along with tetromino position in the case of the LIN and MULT scenarios, and manipulation sign (+/-) in the case of XOR, is what forms the class-conditional features used in the binary classification scenarios. All details are made explicit in Section 2.

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.

As specified in Section 3, a data split of $90/5/5$ in the $N = 40,000$ case of the 64×64 data presented in the main text. This was modified from an $80/10/10$ split in the $N = 10,000$ case of 8×8 images of the original benchmark, specified in supplementary material Section A.7. In scaling up the size of the images, we faced overfitting when training the MLP architecture on some scenarios, and this modification solved the issue. When using alternative architectures, an

$80/10/10$ or a $70/15/15$ split would most likely be suitable.

Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.

We actively specify noise through the type of background for each sample image. We first generate white noise through a zero-mean Gaussian identity covariance Gaussian distribution, representing the WHITE case. The second background type (CORR) is the same white noise smoothed by a Gaussian filter, correlating background pixels with one another and inducing the presence of suppressor variables in a controlled manner. Finally, samples from the ImageNet-1k dataset provide a more complicated and irregular background and noise for the classification scenarios. These sources of noise are controllable through a specified signal-to-noise ratio, and allow us to balance the difficulty of each classification scenario. In the case of CORR, the correlations between pixels induced by smoothing also induce the presence of suppressor variables, as background pixels will be correlated with other background pixels in the same position that informative tetromino features are. This, in turn, allows us to measure the effect of suppressor variables on the performance of XAI methods.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

The only external component to the dataset is that of the ImageNet-1k data, sourced from `https://huggingface.co/datasets/imagenet-1k` as of June 2023. The same subset is also available on Kaggle via `https://www.kaggle.com/competitions/imagenet-object-localization-challenge` and the original creators' website `https://www.image-net.org/download.php`. Access to this is currently reliant on sources such as these, however the popularity of the ImageNet makes it very likely to remain available in some form over time.

The ImageNet dataset also requires agreement of a relatively permissive license via the original creators' website, provided in the above paragraph.

As we just use such data as a complicated and varied background for our ground truth XAI benchmark datasets, we are willing and able to transition to any other dataset if required for any reason of availability, copyright or otherwise. We believe that the results of our analysis and the applicability of such a benchmark will be consistent regardless.

As for the WHITE and CORR scenarios, we provide code to generate and use these data in supplementary material Section A.2. We would like to provide these under a GNU General Public License, and the datasets and benchmarks can be applied without the use of the ImageNet-1k data and its corresponding license for anyone who cannot or does not wish to abide by the given terms.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals non-public communications)? If so, please provide a description.

Not applicable, as far as we are aware.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.

To the best of our knowledge, no. Images of the ImageNet-1k dataset were quality-controlled during the human-annotation stage of data collection. We do not know ourselves the extent of this quality control process, so we can only advise caution to anyone who wishes to explore the ImageNet-1k dataset and may be affected by such images.

Does the dataset relate to people? If not, you may skip the remaining questions in this section.

As specified in supplementary material Section A.1, there are three people categories (scuba diver, bridegroom, and baseball player) compared to $2,832$ people categories in the full ImageNet dataset. These may contain images of people. As the subset aims to have around 1,000 samples per class, our sample from the $N = 50,000$ set could contain up to 50 images of each of the three

specified people categories. There is also the possibility of people-related images co-existing in images of other classes, which has been noted Prabhu & Birhane (2020).

We are able to exclude samples these classes from the published version if required, or use a different dataset, such as the PASS dataset Asano et al. (2021) as mentioned in supplementary materials Section A.1. One is also able to perform the analysis on the WHITE and CORR data and produce compelling and accurate results without any of the considerations of people-related data.

Does the dataset identify any subpopulations (e.g., by age, gender)? If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.

We do not believe so, outside of the possibilities of this occurring in the three people subsets of the ImageNet dataset (scuba diver, bridegroom, and baseball player).

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how.

This is potentially possible in the raw data of the three people subsets of the ImageNet dataset (scuba diver, bridegroom, and baseball player). This possibility could be decreased in the post-processed datasets, as our data generation steps (as described in Section 2) can provide some form of obfuscation, i.e. if the tetromino pattern is place on top of such information in the background.

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? If so, please provide a description.

We do not believe so, but again the ImageNet-1k data component of our work was curated and quality-controlled outside of our control.

Any other comments?

The above considerations about the ethical and legal components of the dataset mostly focus around our use of the ImageNet-1k dataset as a third background type. We would like to reiterate that one can perform the same analysis without said data, and we are willing and able to change the data source here if ImageNet data is problematic or undesirable.

Collection Process

How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.

We generate synthetic data as specified in Section 2 and provide our simulation environment for this in supplementary material Section A.2. Each instance is generated as a combination of a signal component (tetromino pattern) and background (white noise, smoothed/correlated noise, and ImageNet data).

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?

We generate data entirely through software written in the Python language and provided by libraries and APIs such as numpy, scipy, and scikit-learn. We validate these mechanisms through data analysis (i.e., verifying balanced class distribution, balanced XOR case representation).

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)? White noise is sampled from a zero-mean identity covariance Gaussian distribution. ImageNet data is sampled using Python's random.sample() library function. In both cases, we use a fixed random seed for sampling reproducibility.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)? The main author of the paper was the main programmer of the simulation environment, however all authors were involved in the decision

making process on what and how much to collect.

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created. This is perhaps not applicable to our datasets.

Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

No review processes have been conducted thusfar. We do not believe them to be necessary here.

Does the dataset relate to people? If not, you may skip the remaining questions in this section.

As specified above and in supplementary material Section A.1, there are three people categories (scuba diver, bridegroom, and baseball player) compared to $2,832$ people categories in the full ImageNet dataset. These may contain images of people. As the subset aims to have around 1,000 samples per class, our sample from the $N = 50,000$ set could contain up to 50 images of each of the three specified people categories. There is also the possibility of people-related images co-existing in images of other classes, which has been noted Prabhu & Birhane (2020).

We are able to exclude samples these classes from the published version if required, or use a different dataset, such as the PASS dataset Asano et al. (2021) as mentioned in supplementary materials Section A.1. One is also able to perform the analysis on the WHITE and CORR data and produce compelling and accurate results without any of considerations of people-related data.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?

We collected this data from the curated ImageNet-1k dataset, sourced from `https://huggingface.co/datasets/imagenet-1k`.

Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.

The issue of the lack of consent of data collection in the case of ImageNet is known Yang et al. (2021). It is likely that most individual were not notified.

Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.

The issue of the lack of consent of data collection in the case of ImageNet is known Yang et al. (2021). It is likely that most individuals did not consent to the collection and use of their data.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).

We are not aware of the ability to revoke consent of data being part of the ImageNet-1k collection.

Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted? If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.

The work of Yang et al. (2021) looks at obfuscating faces of images in the ImageNet collection and demonstrates a 0.6% decrease in accuracy on such a dataset.

Any other comments?

The above considerations about the ethical and legal components of the dataset mostly focus around our use of the ImageNet-1k dataset as a third background type. We would like to reiterate that one can perform the same analysis without said data, and we are willing and able to change the data source here if ImageNet data is problematic or undesirable.

Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.

For the ImageNet data, we converted samples to single-channel grayscale data, zero-centered the sample through subtraction of the sample mean, and then scaled and cropped images to the 64×64-px size, preserving the original aspect ratio.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data. Raw data in the case of ImageNet is available via `https://huggingface.co/datasets/imagenet-1k`. For WHITE and CORR this is not applicable, but a link to our simulation environment has been provided in supplementary materials Section A.2, where one can see the full data generation process.

Is the software used to preprocess/clean/label the instances available? If so, please provide a link or other access point.

Our simulation environment is available at `https://github.com/braindatalab/xai-tris` and has been provided in supplementary materials Section A.2, along with usage instructions.

Any other comments?

Uses

Has the dataset been used for any tasks already? If so, please provide a description.

We have shown the use of the dataset in our experiments in Sections 3 and 4, as well as in supplementary material Sections A.6 and A.7. Here, we train machine learning models for three architectures over our various classification scenarios and background types (forming 12 combinations), and then apply 16 XAI methods to explain the classification decisions of the corresponding models. We use the known class-conditional features (tetromino shapes) in each sample as ground truth data for an ideal explanation, and compare explanations produced by XAI methods to the ground truth explanations, using our own developed quantitative metrics to benchmark the performance of XAI methods. We compare the explanation performance of XAI methods with the performance of model-ignorant edge detection filters and random samples drawn from a uniform distribution.

Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.

The work shown here is the only system using this dataset so far. The provided GitHub repository is available here `https://github.com/braindatalab/xai-tris`.

What (other) tasks could the dataset be used for?

One could also use this work for benchmarking the classification performance of machine learning models and comparing different models and architectures.

Our $\mathrm{EMD\_perf}$ is based on the Earth mover's distance (EMD) metric, a distance measure between two probability distributions. EMD has been used for many tasks already

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?

The above sections discussion about the ImageNet component and any related privacy concerns outlines any potential harms. We can adapt the datasets and benchmarks to use a different image set as a complicated background type if necessary.

Overall, the task of providing a benchmark suite for validating the performance of XAI methods has mostly positives in our eyes. XAI methods are talked of as the 'key' to unlock black box models and enable deployment of complicated ML systems to critical environments. It is therefore of high importance that we verify the performance of XAI methods to ensure development of highly

performant methods, and minimize the risk of misinterpretation, as outlined in Section 5.

Are there tasks for which the dataset should not be used? If so, please provide a description.

We cannot think of any feasible/potential uses of the dataset which are undesirable.

Any other comments?

Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description.

We provide the GitHub repository to the simulation environment for anyone to make use of. We plan to host a benchmark suite of XAI methods to verify the quality of produced explanations. This will be hosted within our institution, the Physikalisch-Technische Bundesanstalt, which can provide a central and reliable location for managing such a dataset.

How will the dataset will be distributed (e.g., tarball on website, API, GitHub) Does the dataset have a digital object identifier (DOI)? We provide the GitHub repository to the simulation environment for anyone to make use of. We do not provide the specific data used, but the provided environment makes use of fixed random seeds to reproduce data. We therefore do not have a DOI for data shown in the analyses of this work. With that being said, we can of course provide data on request.

When will the dataset be distributed?

The simulation environment is available for use right now. We have no specific ETA for our own hosted and maintained benchmark suite.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.

We would like to release the simulation environment and corresponding datasets and benchmarks under a GNU General Public License, outside of the ImageNet component, which requires the agreement of the license provided by the original creators here `https://image-net.org/download.php`. One can use the GPL components and perform strong benchmarking of XAI methods without the latter component, with instructions provided in the GitHub repository.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.

As specified before, the ImageNet component requires the agreement of the license provided by the original creators here `https://image-net.org/download.php`. This is a relatively permissive use and distribution license.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.

Not as far as we are aware.

Any other comments?

Maintenance

Who will be supporting/hosting/maintaining the dataset?

Currently, we host our simulation environment on GitHub. We plan to host the benchmark suite with our host institution the Physikalisch-Technische Bundesanstalt (PTB). As a national institute as part of the German government, we can provide a stable and central platform for benchmarking the performance of new and existing XAI methods, and increase the quality of developments in the field. The PTB has a strong history and expertise in creating/supporting/hosting/-maintaining data, for example one of the world's largest Electrocardiogram (ECG) datasets Wagner et al. (2020).

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

All authors can be contacted via the email addresses `benedict.clark@ptb.de` and

{rick.wilming, haufe}@tu-berlin.de. These email addresses have been provided in the author information at the top of the paper.

Is there an erratum? If so, please provide a link or other access point.

If errors are found an erratum will be added to the GitHub repository provided as well as to the future benchmarking suite website containing all meta-information about this dataset.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?

Currently, we host our simulation environment on GitHub. All updates (primarily code documentation and cleanliness-related changes) will be done here by the authors of the paper. As specified above, we plan to eventually host the benchmark suite with our host institution the Physikalisch-Technische Bundesanstalt (PTB). This will be web-based and so updates will be web-based i.e. in a changelog section of the website.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? If so, please describe these limits and explain how they will be enforced.

We have specified in the previous sections about the risks and limitations of the people-related ImageNet-1k data in this regard. It is not known that any corresponding data would be deleted in the future. We have also specified that it is also possible to perform the analysis on other data without the risks and limitations of data with a person-component, such as with the WHITE and CORR data shown here.

Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to users.

We host our simulation environment on GitHub with datasets being generated via provided configuration files and instructions. We currently provide the configuration for the 64×64-px data shown in our analyses here, and also with the original 8×8-px data. Any updated versions of our datasets and benchmarks can be provided with new configuration files and instructions.

We plan to host variants with our XAI benchmark suite with a flexibility in the variables of generated datasets, along with other benchmarks that we have and will continue to do. These can all be hosted and maintained under this one umbrella.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.

Currently, the GitHub repository is the place to go. Anyone is welcome to submit a pull request or issue for the given datasets and benchmarks, and to contact us otherwise. As mentioned, we plan to host an XAI benchmark suite to test XAI methods with current and future benchmarks and so we will incorporate an appropriate development and testing workflow.

Any other comments?

B.2 The ML Paper Reproducibility Checklist (as per Pineau et al. (2021), v2.0)

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, algorithm, and/or model. [Yes] **Everything has been described in Sections 2 and 3.**
   (b) A clear explanation of any assumptions. [Yes] **Same as above.**
   (c) An analysis of the complexity (time, space, sample size) of any algorithm [Yes] **A sentence is included at the end of Section 3, and more detail is given in supplementary material Section A.3.**
2. For any theoretical claim, check if you include:
   (a) A clear statement of the claim. [N/A]
   (b) A complete proof of the claim. [N/A]

3. For all datasets used, check if you include:
   (a) The relevant statistics, such as number of examples. [Yes] **Given in Sections 2 and 3.**
   (b) The details of train / validation / test splits. [Yes] **Same as above.**
   (c) An explanation of any data that were excluded, and all pre-processing step [Yes] **No data were excluded. All pre-processing steps are described in Section 2.**
   (d) A link to a downloadable version of the dataset or simulation environment [Yes] **The GitHub repository for our simulation environment has been given in supplementary material Section A.2.**
   (e) For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control [Yes] **The full data generation process for all data shown in this work is given in Section 2.**
4. For all shared code related to this work, check if you include:
   (a) Specification of dependencies. [Yes] **Given as a Pipfile in the provided repository.**
   (b) Training code. [Yes]
   (c) Evaluation code. [Yes]
   (d) (Pre-)trained model(s). [N/A]
   (e) README file includes table of results accompanied by precise command to run to produce those results [Yes] **Commands to run scripts are included. Full results are shown here in Sections 3 4 and supplementary materials Sections A.6 and A.7, but can also be included in the README if needed.**
5. For all reported experimental results, check if you include:
   (a) The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results. [Yes] **Given in Sections 2 and 3.**
   (b) The exact number of training and evaluation runs [Yes] **Same as above.**
   (c) A clear definition of the specific measure or statistics used to report results. [Yes] **Same as above.**
   (d) A description of results with central tendency (e.g. mean) and variation (e.g. error bars). [Yes] **Given in Sections 4 and supplementary materials Sections A.6 and A.7.**
   (e) The average runtime for each result, or estimated energy cost. [Yes] **A sentence is included at the end of Section 3, and more detail is given in supplementary material Section A.3.**
   (f) A description of the computing infrastructure used. [Yes] **Same as above.**

B.3 Addressing Past Reviewers Concerns

This work was recently submitted to the International Conference of Machine Learning (ICML) and rejected after review, with an average score of 5 (borderline accept). This section summarizes the main comments and concerns from reviewers, as well how we have addressed these concerns.

The main comment shared by all reviewers was about the size/dimensionality of the original 8×8 benchmark shown in supplementary materials Section A.7, as well as the applicability to real-world problems and other machine learning benchmark datasets. While we did not at that point make any claim about how our results would transfer to more complex settings such as larger image sizes, we argued (and still would argue) that if an XAI method performs in a sub-optimal manner for a simple and low-dimensional benchmark, increasing the dimensionality would not improve explanation performance, likely having the opposite effect. We also tried to clarify that the value of the benchmark is to provide a minimalistic setup that allows XAI researchers to quickly check the general behavior and performance of their approaches without having to spend a lot of time on model training and post-hoc explanation calculations. This way, undesired behavior (e.g., high importance on pure suppressor variables) can be easily assessed, and methods could potentially be adapted to mitigate this behavior.

In order to remedy this concern from reviewers, we increased the size of images in our benchmark from 8×8 to 64×64, a scale factor of 64 in terms of total dimensionality. We can see from the results of the main paper compared to the original results shown in supplementary materials Section

A.7 that the trends largely remain consistent, and thus both forms of the benchmark are valid ways to assess the performance of XAI methods.

We also added the more complicated ImageNet background type to address the "real-world" aspect of reviewers' concerns. As one of the most popular image classification datasets in the field, we felt that using this data as a background type would show the robustness of our approach, challenging the classifiers' ability to learn the tetromino signal patterns of the given problems and in turn challenge the XAI methods studied in a different manner.

Reviewers also asked for more justification on why we chose our given performance metrics, $\mathrm{EMD\_perf}$ and Precision, and why we favor these over other existing metrics in the field. As a result, we included the paragraph in Section 5 just before the limitations sub-section on this topic.

Other comments from reviewers were around this line of extra justification or clarification on setup and why we believe our benchmark is suitable. We included our response as to why we chose tetrominoes for signal patterns (and specifically the T- and L-shaped tetrominoes) in the 'Scenarios' paragraph of Section 2. We also included clarification on the model architectures' suitability in Section 5.1, where we directly state that we only use one MLP and CNN architecture as a showcase of our benchmarking framework.

The final addressed comment is that of the depth of analysis. Reviewers liked that we cover a large number of experiment variables/settings (16 XAI methods, 4 baseline methods, 3 model architectures, and now 12 combinations of scenario and background type). They stated that they would have just liked to see more analysis/discussion around this points. As such, we have also included more depth in our analysis of our results, primarily in the captions of figures given in the supplementary material in Sections A.6 and A.7 due to the content limit restriction. We feel that the main text gives a sufficient insight into our dataset and the corresponding benchmarks, and wanted to use the space allowed to highlight the value of these aspects.

These comments were the main concerns from our previous reviewers, and we feel like we have more than sufficiently addressed their comments. We are always happy to receive comment on our work and so feel free to ask for clarification if anything has been unclear. We will keep pushing our benchmark forward with higher-dimensional data and more complicated problem settings, push towards safe deployment of XAI in realistic and critical applications.