Machine Learning (2024) 113:6871–6910
https://doi.org/10.1007/s10994-024-06574-3

XAI-TRIS: non-linear image benchmarks to quantify false positive post-hoc attribution of feature importance

Benedict Clark[1] · **Rick Wilming**[2] · **Stefan Haufe**[1,2,3]

Received: 28 November 2023 / Revised: 19 April 2024 / Accepted: 30 May 2024 /
Published online: 16 July 2024

Abstract

The field of ‘explainable’ artificial intelligence (XAI) has produced highly acclaimed 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 one linear and 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, attributing false-positive importance to features with no statistical relationship to the prediction target rather than truly important features. Moreover, we demonstrate that explanations derived from different model architectures can be vastly different; thus, prone to misinterpretation even under controlled conditions.

Keywords Explainable AI · Benchmark · Explanation performance · Non-linear problems · Deep learning · Suppressor variables

Editor: Myra Spiliopoulou.

✉ Benedict Clark
benedict.clark@ptb.de

✉ Rick Wilming
rick.wilming@tu-berlin.de

✉ Stefan Haufe
haufe@tu-berlin.de

[1] Physikalisch-Technische Bundesanstalt, Abbestr. 2-12, 10587 Berlin, Germany

[2] Technische Universität Berlin, Str. des 17. Juni 135, 10623 Berlin, Germany

[3] Charité – Universitätsmedizin Berlin, Charitéplatz 1, 10117 Berlin, Germany

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 and 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.

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. (2022) 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 demonstrated 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. Wilming et al. (2023) took this one step further and presented a minimal two-dimensional linear example, analytically showing that many popular XAI methods attribute arbitrarily high importance to suppressor variables. However, it is unclear as to what extent these results would transfer to various non-linear settings. In the context of the lighting condition example, this recent work showed that many popular XAI methods could highlight every pixel containing illumination information as important. If the illumination information is present across all pixels of the image, an explanation could appear to be composed mostly of random noise, presenting little value to a user. We therefore necessitate that good XAI methods should be able to distinguish between truly important features and suppressors, and ideally inform the user of which category the highlighted variables belong to. Alternatively, a good XAI method should be able to highlight only the truly informative features used by a model, and to mask suppressors and other such misleading features.

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. Tetrominoes are geometric shapes consisting of four blocks, popularized by the game Tetris (Nintendo of America, 1989). 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 three 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. We compare these to four model-agnostic baselines that can serve as null models for explanation performance.

In doing this, we provide the first comprehensive study going beyond linear data. We position the XAI-TRIS datasets and metrics as tools to not only benchmark current XAI methods, but also to guide development of new XAI methods to overcome the susceptibility of false-positive attribution to suppressor variables.

2 Methods

Our workflow of applying and benchmarking post-hoc XAI methods can be seen in Fig. 1. Given a classification dataset generated with an explicitly known ground truth controlling the class-conditional distribution, we train a machine learning model using the training (and validation) split of the data. Taking the trained model and test data (either as

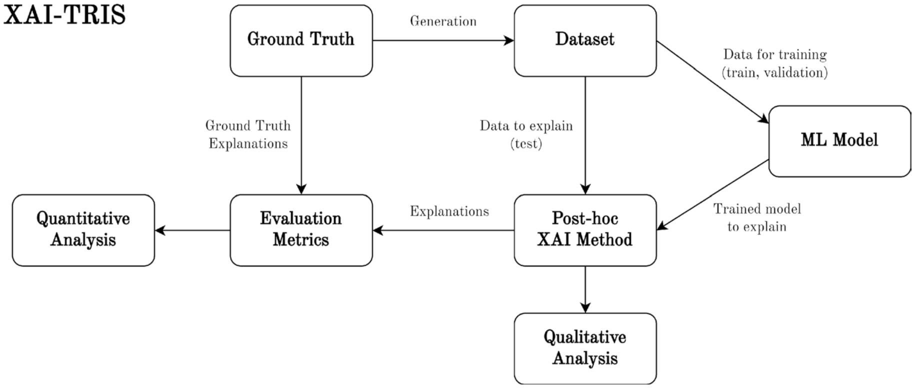


Fig. 1 The process of evaluating an XAI method. XAI-TRIS classification datasets are generated through explicitly defined tetrominoes controlling the class-conditional distribution, which serve as the ground truth features for explanations. Given an ML model trained on the given data, the XAI method takes test data and the model as input, producing explanations. These explanations are passed to performance metrics, which use the given ground truth as a basis for comparison

individual samples or a batch) as the inputs to the given XAI method, we compute output explanations of the same dimensionality as the input data, aimed to correspond to the importance of each pixel towards the trained model's prediction output. Finally, we apply novel performance metrics to compare produced explanations and the ground truth for the given sample, giving us the explanation performance of the method, with quantitative and qualitative results visualized in Sect. 4. This section covers each component of Fig. 1, following the experimental pipeline from data generation through to output analyses.

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 $\boldsymbol{a}^{(n)} = \boldsymbol{a}(y^n)$ carries differently shaped tetromino patterns depending on the binary class label $y^{(n)} \sim \text{Bernoulli}(1/2)$. We apply a 2D Gaussian spatial smoothing filter $H : \mathbb{R}^D \rightarrow \mathbb{R}^D$ 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 $G : \mathbb{R}^D \rightarrow \mathbb{R}^D$ 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 $R^{(n)} : \mathbb{R}^D \rightarrow \mathbb{R}^D$. All other scenarios make use of the identity transformation $R^{(n)} \circ (H \circ \boldsymbol{a}^{(n)}) = H \circ \boldsymbol{a}^{(n)}$. Transformed signal and noise components $(R^{(n)} \circ (H \circ \boldsymbol{a}^{(n)}))$ and $(G \circ \boldsymbol{\eta}^{(n)})$ are horizontally concatenated into matrices $\mathbf{A} = \left[(R^{(1)} \circ (H \circ \boldsymbol{a}^{(1)})), \ldots, (R^{(N)} \circ (H \circ \boldsymbol{a}^{(N)}))\right]$ 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 $\mathbf{E}$: $R^{(n)} \circ (H \circ \boldsymbol{a}^{(n)}) \leftarrow (R^{(n)} \circ (H \circ \boldsymbol{a}^{(n)}))/||\mathbf{A}||_{\text{F}}$ and $(G \circ \boldsymbol{\eta}^{(n)}) \leftarrow (G \circ \boldsymbol{\eta}^{(n)})/||\mathbf{E}||_{\text{F}}$, where the Frobenius norm of a matrix $\mathbf{A}$ is defined as

$||\mathbf{A}||_{\mathrm{F}} := (\sum_{n=1}^{N}\sum_{d=1}^{D}(\boldsymbol{a}_d^{(n)})^2)^{1/2}$. 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 across 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 $R^{(n)} \circ (H \circ \boldsymbol{a}^{(n)})$, 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 (each block here 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 $\boldsymbol{a}^{\mathrm{T}}$ near the top left corner if $y = 0$ and an 'L'-shaped tetromino pattern $\boldsymbol{a}^{\mathrm{L}}$ 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 $\boldsymbol{a}^{\text{XOR++}} = \boldsymbol{a}^{\text{T}} + \boldsymbol{a}^{\text{L}}$ and $\boldsymbol{a}^{\text{XOR--}} = -\boldsymbol{a}^{\text{T}} - \boldsymbol{a}^{\text{L}}$. Members of the opposing class, where $y = 1$, imprint one shape positively, and the other negatively, such that $\boldsymbol{a}^{\text{XOR+-}} = \boldsymbol{a}^{\text{T}} - \boldsymbol{a}^{\text{L}}$ and $\boldsymbol{a}^{\text{XOR-+}} = -\boldsymbol{a}^{\text{T}} + \boldsymbol{a}^{\text{L}}$. Each of the four XOR cases are equally frequently represented across the dataset.

Figure 2 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 3 in the supplementary material shows examples of each of the 12 scenarios across four signal-to-noise ratios (SNRs).

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

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

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 $\boldsymbol{a}^{\text{T}}$ and $\boldsymbol{a}^{\text{L}}$. 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

$$\mathcal{F}^{+}(\mathbf{x}^{(n)}) := \left\{ d \mid H \circ \boldsymbol{a}^{\text{T}}_{d} \neq 0 \vee H \circ \boldsymbol{a}^{\text{L}}_{d} \neq 0,\ d \in \{1, \dots, 4096\} \right\} . \tag{4}$$

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. (2022) 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)

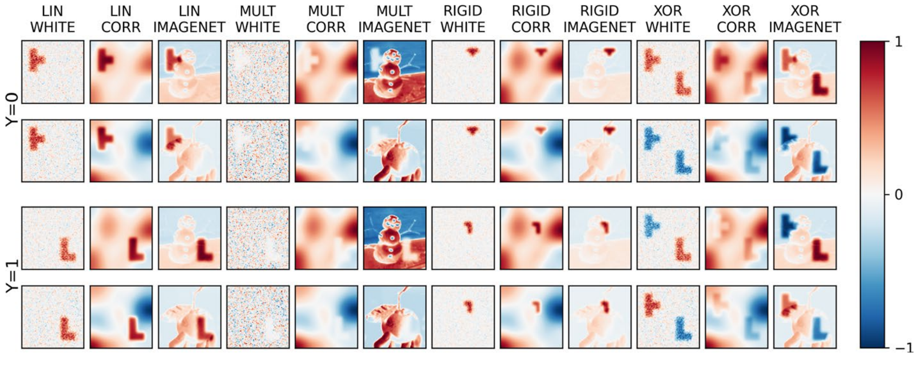


Fig. 2 Examples of data for each scenario, showing differences between samples of each class

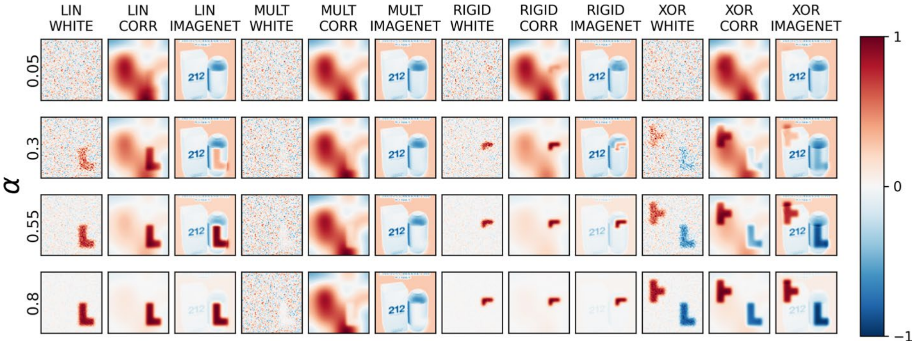


Fig. 3 Examples of generated data samples for each scenario, showing how a generated sample of Class #0 (where y=0) for each scenario varies across four signal-to-noise ratios (SNRs) α

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 $f^{\boldsymbol{\theta}} : \mathbb{R}^D \rightarrow \mathcal{Y}$ 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

Given a trained machine learning model, we now look to apply post-hoc XAI methods to test data to produce explanations, and we also define several performance baselines as a reference point for comparison during analyses. 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).

Table 1 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	References
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 and 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)

Table 1 (continued)

XAI method	Description	Implementation framework, parameterization	References
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 and 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 and 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 and 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)

Table 1 (continued)

XAI method	Description	Implementation framework, parameterization	References
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)

The full list is detailed in Table 1. 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 $f^{\boldsymbol{\theta}} : \mathbb{R}^D \rightarrow \mathbb{R}$, 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.

2.4.1 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. We constrain these results to the submitted appendices, and focus on the results and analyses for the next two defined metrics.

2.4.2 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 $\mathrm{OT}(\mathbf{s}, \mathcal{F}^+)$ denoting the cost of the optimal transport from explanation $\mathbf{s}$ to ground truth $\mathcal{F}^+$. This follows the algorithm proposed by Bonneel et al. (2011) and the implementation of the Python Optimal Transport library (Flamary et al., 2021). We define a normalized EMD performance score as

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

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. The above 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.

2.4.3 Importance mass accuracy (IMA)

Because of this, we consider a third metric, Importance Mass Accuracy (IMA). Calculated as the sum of importance attributed to the ground truth features over the total attribution in the image, this metric is akin to 'Relevance mass accuracy' as defined by Arras et al. (2022). We calculate

$$\mathrm{IMA} = \sum_{\substack{i=1 \\ s_i \in \mathcal{F}^+}}^{|\mathcal{F}^+|} s_i \Big/ \sum_{i=1}^{|\mathbf{s}|} s_i. \tag{6}$$

This metric achieves invariance for not penalizing false negative attribution to a subset of pixels in $\mathcal{F}^{+}(\mathbf{x})$, whilst also utilizing the whole attribution instead of a 'top-k' metric such as Precision. Not only this, but it is a direct measure of false-positive attribution, where a score of 1 signals a perfect explanation highlighting only ground truth features as important. We use this metric to complement the strengths of EMD whilst also presenting an alternative perspective to quantifying explanation performance.

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 $\mathcal{F}^{+}(\mathbf{x})$? We do not expect that any method would achieve perfect performance for our metrics, as this may be unrealistic due to the aforementioned ways that our metrics interact with $\mathcal{F}^{+}(\mathbf{x})$. We hypothesize, however, that performance trends are consistent between scenarios. With this in mind, we do not aim to explicitly rank methods for the purpose of handing out recommendations for XAI methods to use in practice. Our focus is more toward comparing XAI method performance to baseline methods to identify performance weaknesses to guide future development of improved methods. With that in mind, the past study of Wilming et al. (2022) showed that the PatternNet and PatternAttribution (Kindermans et al., 2018) methods perform best in a linear problem setting, so we would expect to see the same here for LIN. How this performance transitions to non-linear methods is yet to be seen, motivating the following experiments.
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? No prior studies exist on this comparison between linear and non-linear problem settings, however we anticipate that it is difficult to directly compare between different scenarios. One difficulty is that each method requires a different trained model, and while our implementations are aimed to be as equivalent as possible, it has been shown that explanation performance is affected by classification performance (Arras et al., 2022; Oliveira et al., 2024). Another aspect complicating comparisons across scenarios is due to properties of the scenarios themselves. Some XAI methods may perform better in the scenarios with a fixed ground truth position over the RIGID scenario.
3. Does adding correlations to the background noise, through smoothing with the Gaussian convolution filter, negatively impact explanation performance? Suppressor variables have been shown to negatively impact explanation performance (Haufe et al., 2014; Wilming et al., 2022; Oliveira et al., 2024). Here, the correlation between background pixels overlapping with tetromino features and background pixels near the tetromino invokes the presence of suppressor variables (those neighboring pixels). Knowledge of these background pixels may be useful to the machine learning models, for instance for denoising the correlated background to make the underlying classification easier. We therefore expect that performance for CORR scenarios will be worse than WHITE equivalents, although this performance difference also will likely depend on the strength of the correlation of the smoothing operation.
4. How does the choice of model architecture impact explanation performance? XAI methods may perform differently for different architectures. For example, GradCAM (Selvaraju et al., 2017) is only applicable to Convolutional Neural Network (CNN)

architectures, but others applicable to some or all models studied here may prefer properties of one model architecture over another. CNNs may perform better than MLPs for the RIGID scenario, as the invariance of translation and rotation operations is one of the main desirable properties of CNNs. We expect that such differences between model architecture will also be seen when considering the downstream task of model explanation. 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 2 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. What can be seen is that a wide range of SNRs are required to model each problem, and it is difficult to exactly model each scenario and background type to the 80% performance threshold. CORR scenarios, perhaps aided by the suppressing correlated background pixels, achieve the best performance on average while requiring the lowest SNRs when compared to WHITE and IMAGENET variants. The MULT WHITE scenario is particularly difficult to model, requiring a much higher SNR to model than the CORR and IMAGENET variants. 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 Fig. 4 in Appendix B.5. Experiments are run on an internal CPU and GPU cluster, with total runtime in the order of a matter of hours.

4 Results

With data generated, models trained, and experiments defined, we move to analyzing the explanations produced by the given set of post-hoc XAI methods. We first start with qualitative analysis, looking at example explanations produced for given samples in Figs. 5, B.7.1, and 8. Such analysis is commonly used in XAI methods papers (for example, Bach et al. (2015); Ribeiro et al. (2016); Lundberg and Lee (2017)), with authors assessing the visual quality of explanations for a chosen example, and little to no quantitative analysis being done to verify explanation performance empirically. As such, we focus on quantitative analysis afterwards, showing boxplots of explanation performance for the EMD and IMA metrics in Fig. 6 and in Appendix B.7.2.

4.1 Qualitative analysis

Figure 5 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 respectively) followed by the model-ignorant Laplace filter. Qualitative recovery of the signal tetromino patterns is mixed across all scenarios, models, and XAI methods, with no single method looking to perform the best. LIME, however, fails to produce sensical explanations in all cases. While for no method the importance is predominantly contained within the ground truth pattern, the tetromino patterns can be recognized in many cases, even at low signal-to-noise ratios. The MLP tends to produce noisier explanations than the CNN, especially for the complex

Table 2 Results of the model training process for each classification setting, model architecture, and background type

		White		Corr		Imagenet	
		α	ACC	α	ACC	α	ACC
	LLR	0.03	89.7	0.02	100.0	0.1	87.5
LIN	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

These results are depicted as chosen Signal-to-noise ratios (SNRs), parameterized by α, as well as the average test accuracy (ACC, %)

structures in the background of the ImageNet examples. We can often see noisy false-positive attribution to statistically irrelevant features related to the image background. In many of the explanations for the scenarios with the CORR background using the LLR and MLP, we can see ‘halos’ of importance attributed to features outside, but nearby, the ground truth. This points to the potential use of suppressor variables by the models, in this case pixels outside the ground truth that are correlated with pixels inside the ground truth due to overlapping structures in the image background. Appendix B.7.1 expands on the qualitative results of the main text, and Fig. 8 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.

4.2 Quantitative analysis

Figure 6 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 EMD and IMA metrics. Appendix B.7.2 expands on the quantitative results of the main text, detailing results for all 16 methods studied and for our Precision metric.

We can now answer the experimental questions laid out in Sect. 3:

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

Within most scenario-architecture parameterizations, the performances of the studied XAI methods are relatively homogeneous. Baseline methods also 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 edge detection filter outperforms nearly all other methods in the RIGID as well as the XOR scenarios, when used in

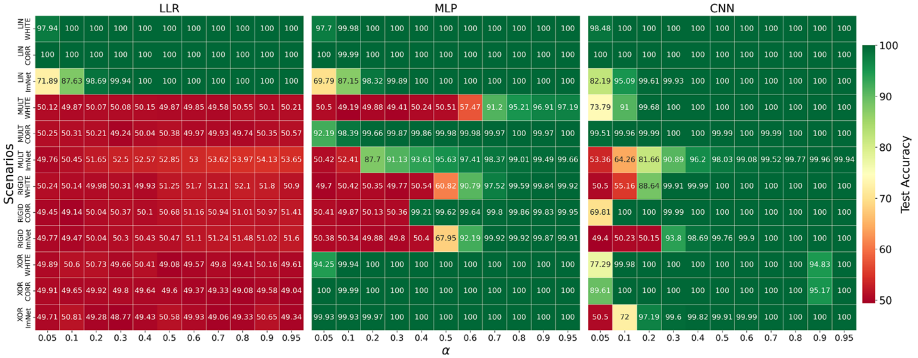


Fig. 4 Average test accuracy over 10 model trainings for each problem scenario and model architecture, for a fixed range of signal-to-noise ratios (SNRs). As expected, the Linear Logistic Regression (LLR) model cannot perform above chance level for non-linear scenarios. The Convolutional Neural Network (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

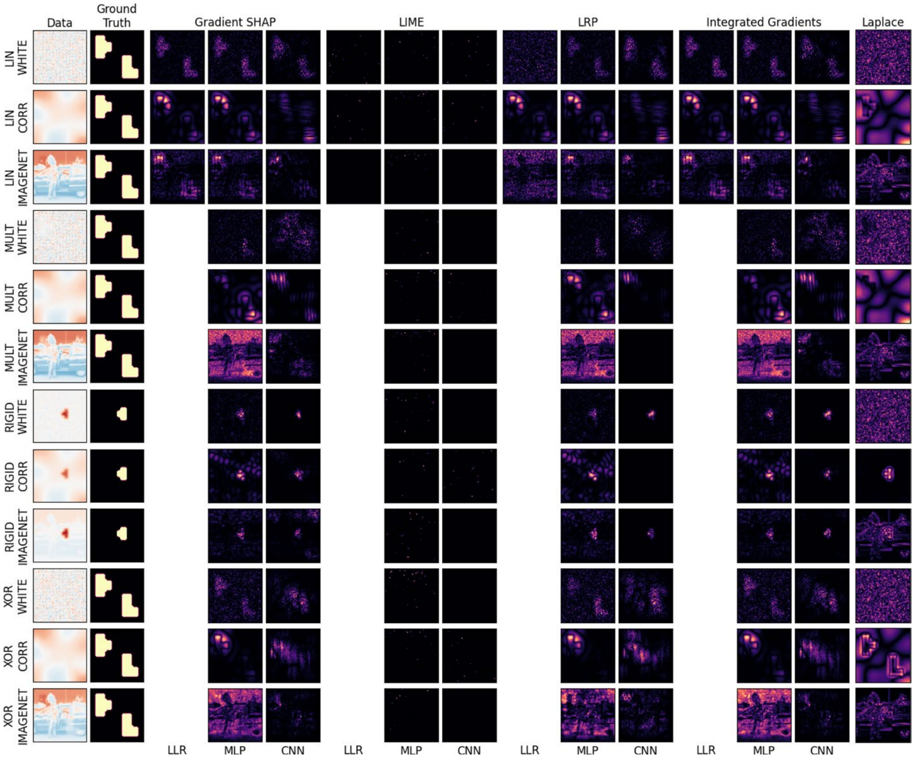
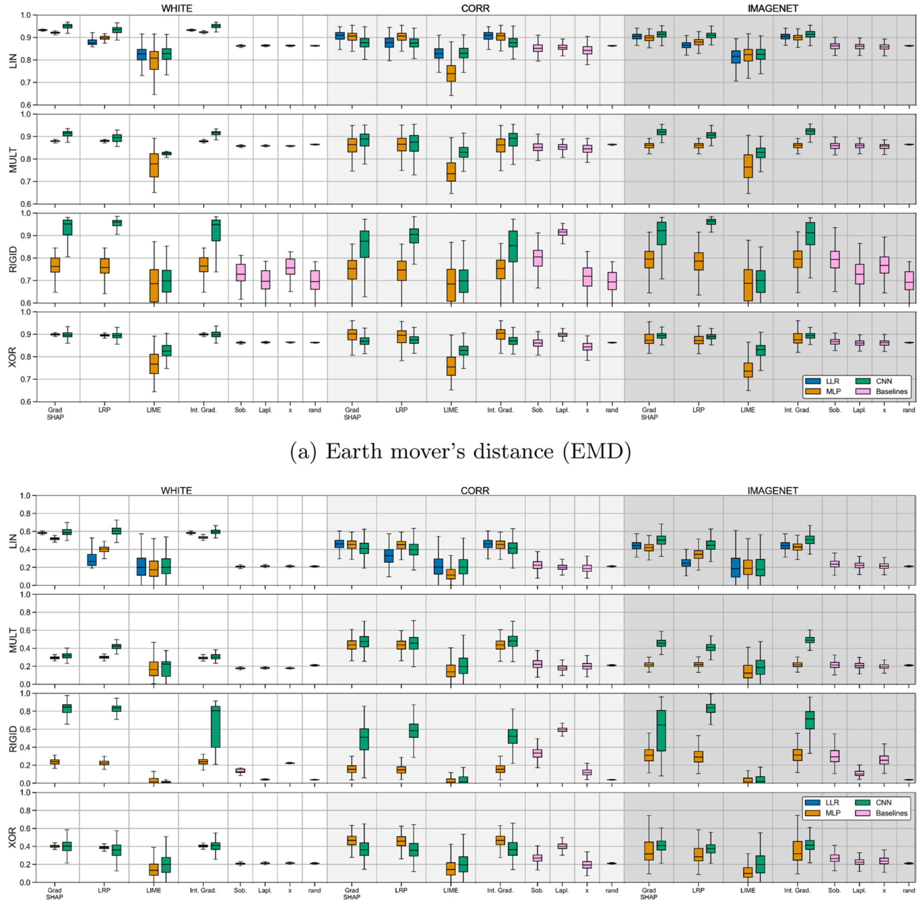


Fig. 5 Absolute-valued importance maps obtained for a random correctly-predicted data sample, for selected XAI methods and baselines. Recovery of the ground truth pattern across all scenarios is best shown by XAI methods applied to a Linear Logistic Regression (LLR) model. The Multi-Layer Perceptron (MLP) tends to focus on noise in the case of ImageNet backgrounds, and LIME often fails to produce sensical explanations across all model architectures (Color figure online)



(a) Earth mover's distance (EMD)

(b) Importance mass accuracy (IMA)

Fig. 6 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 EMD (top) and IMA (bottom) metrics. Depicted are boxplots of median explanation performance, with upper and lower quartiles as well as outliers shown. The white areas (left) show results for white background noise (WHITE), whereas the light gray shaded areas (middle) shows results for the correlated background noise (CORR) scenarios and the darker gray areas (right) for ImageNet (IMAGENET) backgrounds (Color figure online)

combination with correlated backgrounds (CORR). IMA results for baseline methods in the RIGID scenario show a lot less variance in the boxplots of Fig. 6b than for the EMD equivalents in Fig. 6a.

The results show massive variability in performance for all methods across different problems and model architectures, so we cannot necessarily declare one specific 'best' method. For the linear case, we can recommend PatternNet and PatternAttribution as being able to recover signal optimally in the presence of suppressor variables, however this property does not translate well to non-linear cases. Looking at Fig. 6, no method performs consistently near the, perhaps impossible, perfect EMD or IMA score. In many cases, particularly for IMA, the scores are low (≤ 0.5) across the board, signaling around or less than half of the total image attribution going to truly important features. Four exceptions are for CNN results in the RIGID case, where LRP for WHITE and IMAGENET, GradSHAP for WHITE, and Integrated Gradients for WHITE achieved an average above 0.8.

LIME fails in all cases at this dimensionality, so may be unsuitable for a user looking to implement explainability on higher dimensional image data. Plotting results without LIME may help to improve visual spread of results, however as such a popular method of the field, it is important to study it. From Figs. 9, 10, and 11, we can see that gradient-based methods tend to perform similarly to one another, as well as SHAP variants to one another. While this may not be a surprise, when each of these variants and formulations are supposed by their creators to possess benefits over other XAI methods, the reality shows a different story. For example, one might hope that DeepSHAP shows stronger performance for deeper architectures such as the CNN than GradSHAP, however this is not necessarily the case. It is also possible that the CNN architecture studied here is not deep enough to provide benefit to such a method like DeepSHAP.

We also observe that comparison in terms of the EMD metric are made difficult by the metrics comparably narrow range of values, with the rand method (sampling from a uniform distribution) averaging between 0.7 for RIGID and 0.875 otherwise being evidence for this.

In answer to experimental question 1, for the purposes of this study - assessing the false-positive attribution of feature importance to non-important variables such as suppressors - it is not clear which individual XAI method can be considered the ‘best’, however we have shown that the random performance baselines can achieve competitive or even improved explanation performance scores over many XAI methods. It is important for strong XAI methods to outperform such baselines to justify their use in practice, and future methods should be developed with such baselines in mind.

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?

Here we can see again that some methods vary in performance depending on the type of non-linearity (most perform better for MULT with the fixed position non-linearity than for RIGID), with a larger spread of EMD and IMA scores (seen in the size of boxes and whiskers of Fig. 6) for non-linear scenarios than for LIN.

The results for PatternNet and PatternAttribution (Kindermans et al., 2018) shown in the appendix (Figs. 9, 1011, 17, and 18) were proposed in part for solving the suppressor problem, and we can see how this is not necessarily always the case. These methods show strong performance for LIN as proposed, and as was seen in Wilming et al. (2022), but do not look to generalize as well in most non-linear scenarios. Notably when the pattern signal is not in a fixed position (i.e., RIGID), these methods perform worse than when the signal is in a fixed position (i.e., MULT and XOR). More specifically, they also look to learn the complete pattern signal (i.e., the tetromino shapes for both classes), so in the XOR case where both shapes are present and fixed in each sample, they do outright perform the best as one might expect.

The results for the RIGID scenarios may be taken with a pinch of salt, as the high signal-to-noise ratios (SNRs) lead to highly salient tetrominoes in sample images. Notably, explanations produced for CNNs in this case tend to perform very well for both the EMD and IMA metrics compared to most results for any other model architecture and problem scenario. While this problem itself (identifying a pattern with rotation and scaling invariance) is the most realistic of the four presented here, particularly when applied to CNNs, the high saliency of tetrominoes is perhaps not wholly akin to realistic problem settings, where the relative saliency of individual objects of interest is usually far lower. The high saliency of the tetrominoes derives from our experimental choice to adjust SNRs to achieve a predefined minimal classification performance threshold, which required high SNR in this setting. An alternative approach could be to reverse this and fix the SNR for all scenarios and background types.

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

When looking at results from WHITE to CORR, we can spot a decrease in performance and increase in spread in most cases. This can be attributed to the fact that the imposed correlations (induced through Gaussian smoothing) between background pixels correlated with those overlapping with $\mathcal{F}^+$ cause background pixels to act as suppressor variables. One can control the strength of this effect by increasing/decreasing the strength of the Gaussian smoothing's sigma parameter. This effect can be most strongly observed when comparing RIGID WHITE to RIGID CORR for the IMA metric, suggesting that correlations in the background do indeed increase false-positive attribution in model explanations.

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

For LIN, explanation performance of all methods for all architectures is similar in most cases. When moving to non-linear scenarios, we can see little consistency in how architectures perform - the CNN can be seen to perform best in the RIGID case, but the MLP performs relatively better for the fixed tetromino position cases of MULT and XOR.

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. This can perhaps be explained by the CNN architecture tending itself well to rotation/translation invariance, whereas the properties of the MLP work better for a fixed-position ground-truth class-conditional distribution. However, in the RIGID setting nearly all XAI methods are outperformed by a simple Laplace edge detection filter for correlated backgrounds results. In this case, the discrepancy between the MLP and CNN performance is amplified for the IMA metric, with the CNN performing relatively better for a few XAI methods. The CNN also performs well in the case of the more-complicated IMAGENET backgrounds.

We can also note that when multiple models present similar classification performance for a task, a user may assume or just not realize that explanation performance could be vastly different, as seen in the MLP vs CNN results of RIGID in Fig. 6, and qualitatively in Fig. 5 across all architectures.

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 to suppressors 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. It is therefore important that future XAI methods be developed to either highlight only truly important features, or to inform the user of whether an importantly-attributed variable is truly important, a suppressor, or otherwise.

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; Hedström et al., 2022)) or subjective or potentially circular criteria (such as fidelity and faithfulness (Gevaert et al., 2022; Nauta et al., 2023)). It was shown in Wilming et al. (2023) that faithfulness as a concept, when treated as an XAI method in itself, promotes the attribution of importance to suppressor variables. We therefore doubt that such secondary 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. If suppressors are present in the data and are highlighted as important by an XAI method, the user may seek to use these variables as a target for intervention (e.g. as a genetic manipulation or drug target in the context of a genome wide association experiment). However, any attempt to manipulate suppressor features to influence the prediction target would be futile. In the worst case, time and money would be wasted, and in any case, the false-positive attribution of importance to suppressor features has provided no value to the user. In contrast to other notions of faithfulness, the expectation that an XAI method identifies features truly related to the target is an objectively quantifiable property of an XAI method, and we here propose various linear and 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 and 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.

One potential limitation of the EMD metric is the strictness of limiting the ground truth feature set $\mathcal{F}^+$ to the specific pixels of tetrominoes $\boldsymbol{a}^{\mathrm{T/L}}$ compared to, say, the set of features outlining $\boldsymbol{a}^{\mathrm{T/L}}$. Alternative definitions of $\mathcal{F}^+$ could be conceived to more flexibly adapt to different potential ‘explanation strategies’. Figure 7 in the appendices outlines four ‘explanation strategies’ and how the EMD metric varies with each. Notably, an ‘outline’ explanation performs worse than an explanation highlighting a subset of $\mathcal{F}^+$. This highlights two interesting features of our novel metric. Firstly, a strongly performing ‘subset’ explanation shows that EMD does not penalize false negatives (not attributing high importance to some truly important features) as harshly as Precision and other ‘top-k’ metrics do. Secondly, the 'outline' explanation functions in a presumably similar way to some model-ignorant edge detection methods, and performs the worst of any explanation strategy shown in Fig. 7. Yet, we have shown such edge detection methods to be capable of outperforming many XAI methods in some problem scenarios. Our IMA metric also complements this potential limitation of EMD, where it does not matter if the attribution of importance to features of $\mathcal{F}^+$ is spread across all features, or just more intensely attributed to a subset. This metric directly measures false-positive attribution of importance to features outside of $\mathcal{F}^+$, and assists the user in understanding the role that suppressors play in model explanations. We have also seen that the EMD metric produces scores over quite a small range, where a 'low' EMD score is hard to achieve, even for a truly random explanation (rand). This contributes to the close distribution of results of Figs. 6a and 9, making experimental question 1 tougher to answer. Future work will look into the development of improved metrics to quantitatively evaluate XAI methods more robustly. For example, we can use the null distribution to normalize/standardize the EMD metric, either by subtracting the null mean or by subtracting the mean and dividing by the null standard deviation, thus widening the range of realistically attainable scores. This benchmark focuses on the issue of 'correctness' of explanations, so we will also unify the XAI-TRIS benchmarks and (improved) metrics with 'secondary' quality metrics such as robustness and, potentially, faithfulness/fidelity. Doing so will widen the characteristics studied by each given metric, and will provide a more comprehensive overview of the performance of XAI methods beyond 'correctness'.

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.

Future work will also focus on integration of the XAI-TRIS benchmarks with other related benchmarks (Wilming et al., 2022; Oliveira et al., 2024) into one platform, aiming to test the performance of XAI methods across a suite of domains and problems. We also plan to extend this with the creation of more realistic benchmarks in the domains of medical imaging and natural language processing. With the availability of such a unified benchmark suite, the possibility of developing fit-for-purpose and goal-driven XAI methods is open to researchers.

6 Conclusion

We have used a data-driven generative definition of feature importance to create XAI-TRIS, synthetic datasets 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 have shown 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. Finally, we have shown 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.

Appendix A: MLJ contribution information sheet

What is the main claim of the paper? Why is this an important contribution to the machine learning literature?

We claim that many post-hoc explanation methods consistently and reproducibly highlight certain input features that have no statistical dependency to the target variable predicted by the model. The existence of such so-called suppressor variables, and the false positive attribution of such variables as important, can lead to severe misinterpretations, which raises concerns regarding the correctness and utility of â€˜explanationsâ€™ provided by explanation methods.

We create benchmark image datasets for one linear and three non-linear classification scenarios, in which the important class-conditional features are known by design. These scenarios are based on different types and combinations of tetrominoes (Golomb, 1996), overlaid on one of three types of noisy backgrounds. One of these background types, white noise smoothed by a Gaussian filter, induces the presence of suppressor variables through the correlation of background pixels overlapping the tetromino with those just of the noisy background. In all cases, ground truth explanations are explicitly known through the location of the tetrominoes in the sample.

we develop novel performance metrics, one based on the Earth mover’s distance of transforming the ‘energy’ of a given explanation into the ground truth explanation, and use this to show that in many cases, the presence of induced suppressor variables hinders explanation performance for many popular XAI methods. Another metric directly measures the false positive attribution of model explanations through the proportion of importance attributed to ground truth features over the total attribution of the explanation. These two metrics complement each other well.

Through our experimental results we draw other conclusions, including that explanations produced for different equally performing ML architectures can be very inconsistent. We show that popular explanation methods are sometimes unable to outperform random performance baselines and edge detection methods. We highlight that secondary metrics such as faithfulness are currently not sufficient to assess ML explanation quality compared to objective metrics focused on the ‘correctness’ of explanations, such as those presented here.

The importance of these claims is that machine learning model explanations are prone to misinterpretation under such inconsistencies. For example, one may assume that equally performing models would produce equally performing explanations, however this is not always true. One may have chosen a particular architecture based on other properties of it, and end up with misleading or nonsensical explanations. We necessitate that for XAI to be deployed in high-stakes fields, such risks should be mitigated. Our approach is a rigorous and objective evaluation of the performance of current explanation methods, which can lead to the development of stronger and more reliable methods in the future.

What is the evidence you provide to support your claim? Be precise.

We conduct an extensive set of empirical experiments across 4 image classification problem scenarios, 3 background types, 3 model architectures, 16 explanation methods, 4 performance baselines, and 3 metrics. We carefully construct the important class-conditional features in each problem, which can serve as ground truth explanations. We assess many popular post-hoc XAI methods and quantify their 'explanation performance' using metrics from signal detection theory such as Earth mover's distance, IMA, and precision, and show that such methods attribute importance to suppressor variables and can lead to misleading interpretations.

Through our experimental results we observe behavior including that explanations produced for different equally performing ML architectures can be very inconsistent. We show that popular explanation methods are sometimes unable to outperform random performance baselines and edge detection methods for our developed performance metrics. We discuss, using related literature, that secondary metrics such as faithfulness are currently not sufficient to assess ML explanation quality compared to objective metrics focused on the 'correctness' of explanations, such as those presented here.

What papers by other authors make the most closely related contributions, and how is your paper related to them?

Several works in the XAI field have moved towards quantitative evaluation of XAI methods using ground truth data (Tjoa & Guan, 2020; Li et al., 2021; Zhou et al., 2022; Arras et al., 2022; Gevaert et al., 2022; Agarwal et al., 2022). However, these studies are limited in the extent to which they perform quantitative assessment, and many such studies do not construct their benchmark datasets in a way that realistic correlations between class-dependent and class-agnostic features (i.e., the foreground/object in an image vs. the background) are included. In practice, these correlations can give rise to features acting as suppressor variables. These works do not focus on such variables and our previous work is the only such work to do so.

Wilming et al. (2022), published in ECML 2022, took a similar approach to that shown here, yet focused on a linear problem for one model architecture, and did not make use of random performance baselines to compare XAI methods to. Wilming et al. (2023) also looked into quantifying explanation performance in the presence of suppressors using a two-dimensional linear example, however the focus there was on analytically deriving the exact influence of suppressors on produced explanations.

Have you published parts of your paper before, for instance in a conference? If so, give details of your previous paper(s) and a precise statement detailing how your paper provides a significant contribution beyond the previous paper(s).

The content of this paper is entirely original. Some ideas discussed in this paper have already been voiced in our prior work (Haufe et al., 2014; Wilming et al., 2022, 2023). However, our current paper goes beyond these through focusing on an extensive set of empirical experiments across 4 image classification problem scenarios, 3 background types, 3 model architectures, 16 explanation methods, 4 performance baselines, and 3 metrics.

Suggested Reviewers Pieter-Jan Kindermans (pikinder@google.com): Author of PatternNet and PatternAttribution.

Moritz Grosse-Wentrup (moritz.grosse-wentrup@univie.ac.at): Expert in XAI and causality.

Max Welling (M.Welling@uva.nl): Esteemed machine learning expert with interest in XAI.

Robert Jenssen (robert.jenssen@uit.no): Professor of machine learning with track record in XAI.

Appendix B

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.

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.

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 $N = 40,000$ 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 Sect. B.8, 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

$$\texttt{SCENARIO\_JdKp\_}\alpha\texttt{\_BACKGROUND.pkl}$$

containing a python dictionary

$$\texttt{\{SCENARIO\_JdKp\_}\alpha\texttt{\_BACKGROUND : DataRecord(...)\},}$$

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()`

is a Python

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

collection specified as

```
DataRecord.
```

Each field can be accessed programmatically via the name, for example

```
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.

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 $N = 2000$ test set, this took between 24 and 48 h of compute time per GPU on the cluster. Quantitative analysis took roughly a further 24 h 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 Sect. B.8 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 Sect. Appendix B.2.

Data

Here, we expand on Fig. 2 with Fig. 5, which shows an example of each scenario across four choices of signal-to-noise ratio (SNR), parameterized by α.

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.

Earth mover's distance

Explanation performance

This section further elaborates results of our experiments on validating the performance of XAI methods. In Figs. 9 and 11 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

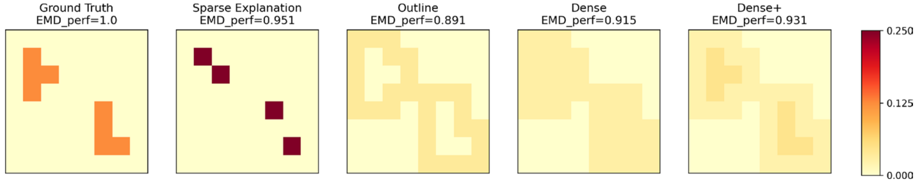


Fig. 7 EMD scores for the 8 × 8 ground truth as well as four 'explanation strategies'. Here, we can see that the EMD metric does not penalize an explanation highlighting a subset of truly important features compared to an explanation highlighting the outline of the ground truth. This shows that the EMD penalizes false negatives (not attributing high importance to truly important features) less than a 'top-k' metric like Precision would. The 'outline' strategy in the third column produces an explanation presumably similar to a model-ignorant edge detector, which has the lowest EMD score of the strategies shown, yet we have shown such edge detectors can outperform many XAI methods in some problem scenarios

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.

Qualitative results

In Fig. 8, 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.

Quantitative results

In Figs. 9, 10, and 11 we can see the full quantitative results for the EMD, IMA, 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).

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 12 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.

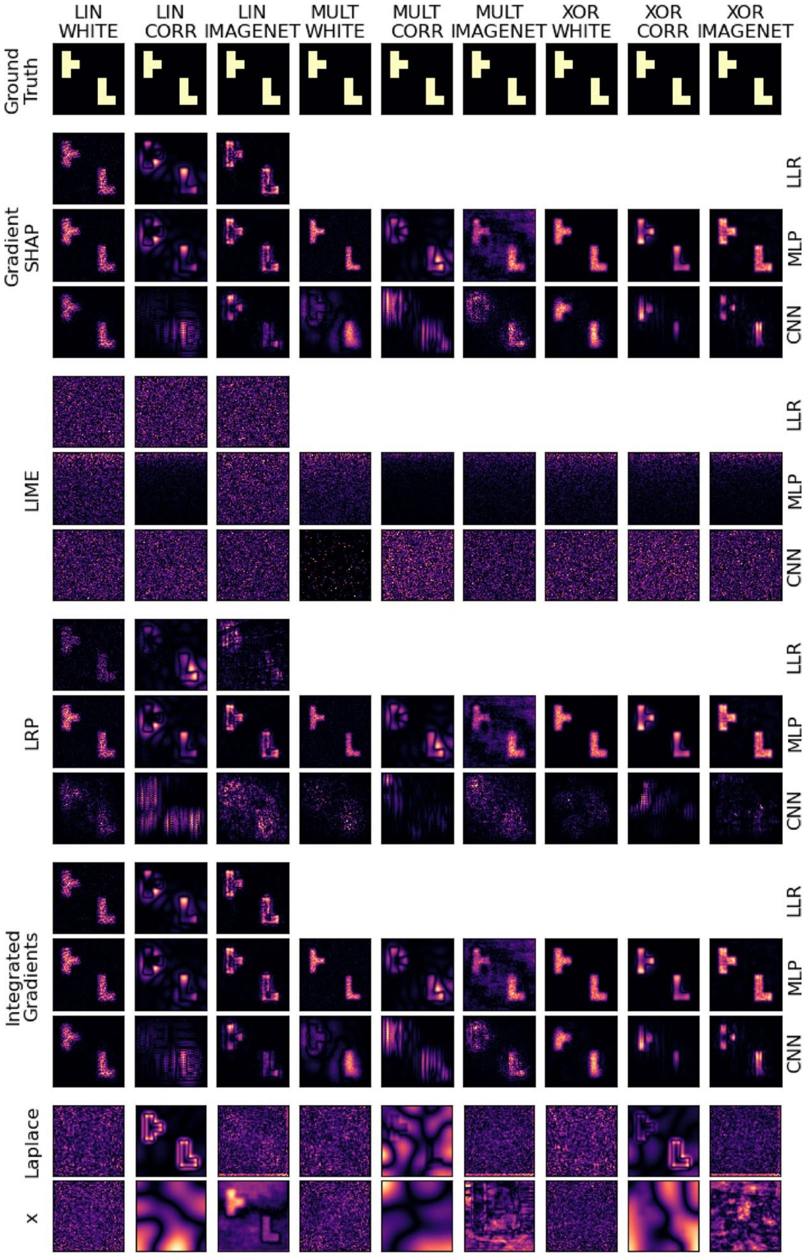


Fig. 8 Absolute-valued global importance maps calculated as the mean importance value over all correctly predicted samples, for selected XAI methods and baselines. RIGID scenarios involving translations and rotations of the tetromino signal pattern are not included as they have no fixed ground truth 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 Sect. B.8 show better performance for LIME with the smaller 8 × 8 -px image benchmark

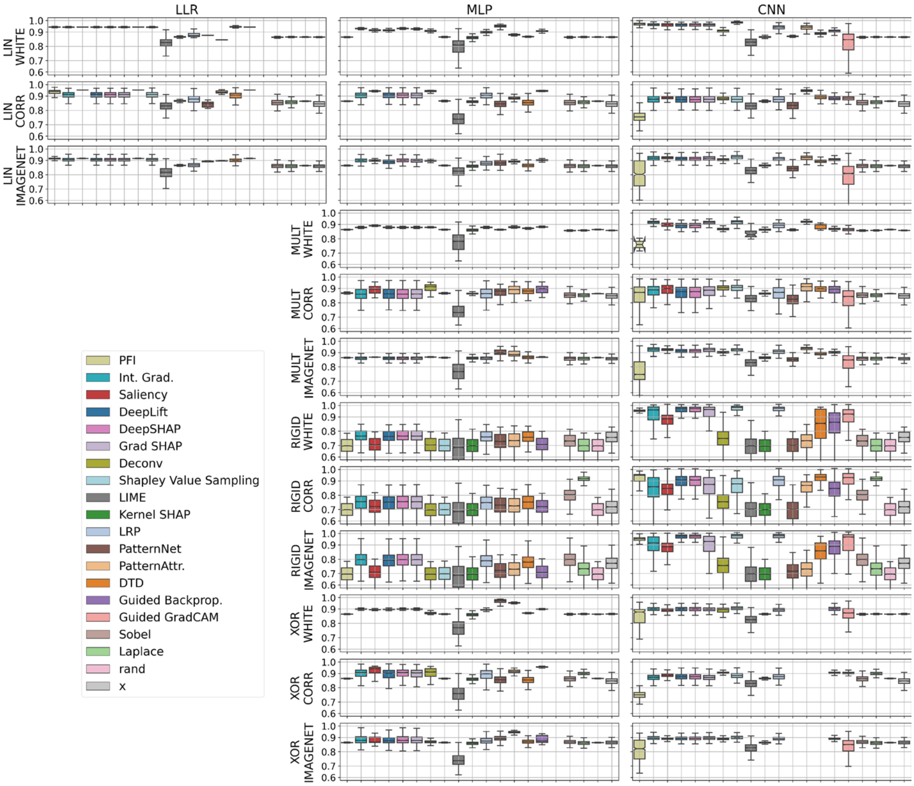


Fig. 9 EMD metric based on the Earth Mover's Distance (EMD) for every XAI method tested, separated by model architecture and depicted as boxplots of median and quartile performance scores. Guided GradCAM is only implemented for CNN architectures, and Keras models required for PatternNet, PatternAttribution, 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

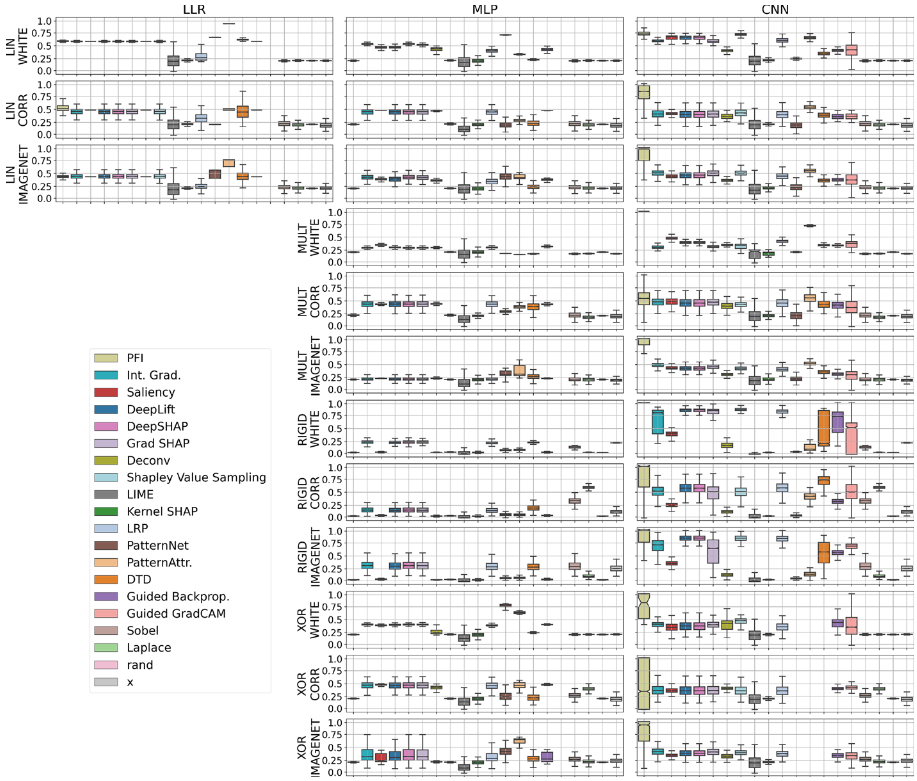


Fig. 10 IMA metric results for every XAI method tested, separated by model architecture and depicted as boxplots of median and quartile performance scores. Guided GradCAM is only implemented for CNN architectures, and Keras models required for PatternNet, PatternAttribution, and Deep Taylor Decompostion (DTD) struggled to converge for the XOR scenarios as stated above, so these are excluded from the corresponding sub-plots. For the most part, results are relatively consistent with the above EMD results of Fig. 9. 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. One noticeable difference between the EMD results of Fig. 9 and the results shown here is that PatternAttribution performs outright best for LIN WHITE under the LLR and MLP, and XOR WHITE under the MLP. In contrast, PFI performs strongly for many scenarios under the CNN, but poorly under the MLP. No XAI method performs outright the best across all scenarios

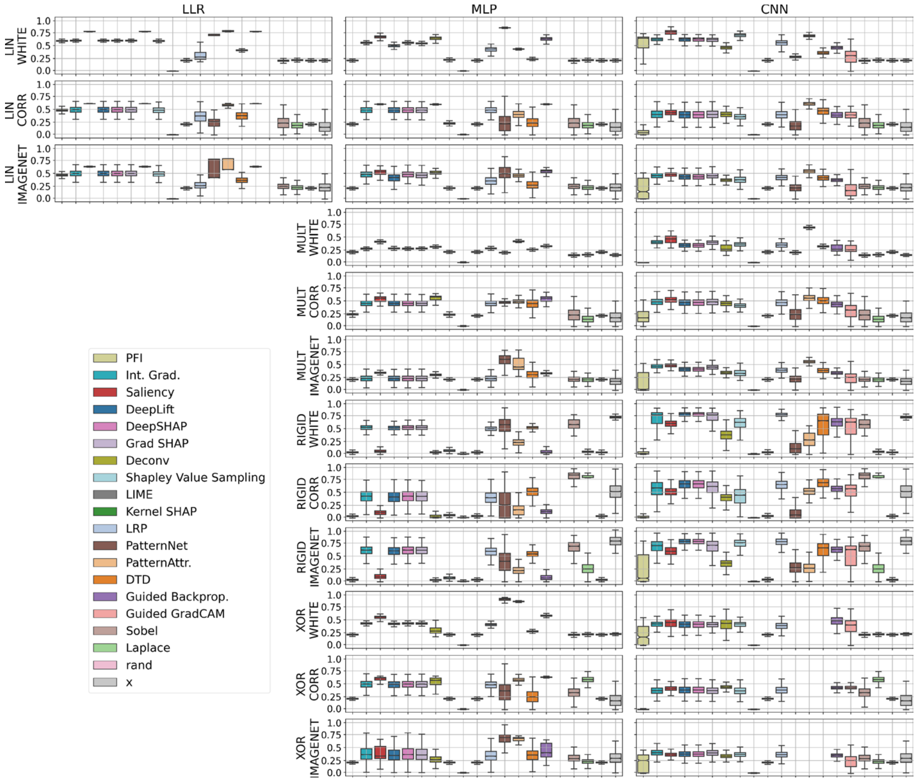
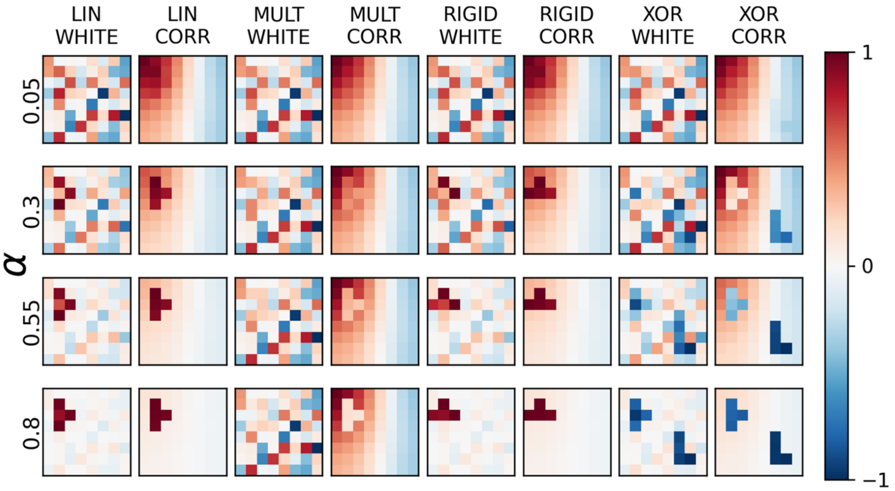
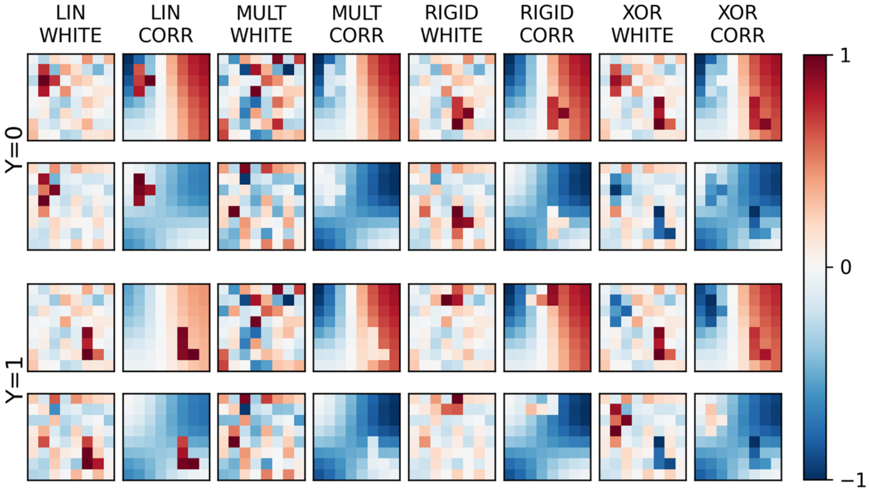


Fig. 11 Precision score for every XAI method tested, separated by model architecture and depicted as boxplots of median and quartile performance scores. Most methods outperform the baseline methods for most model-scenario parameterization pairs. The ‘x’ method, using input data as reference point of 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 Sect. B.8, with the smaller 8 × 8-px image benchmark. Similarly to the results of 9, 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.

Fig. 12 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)

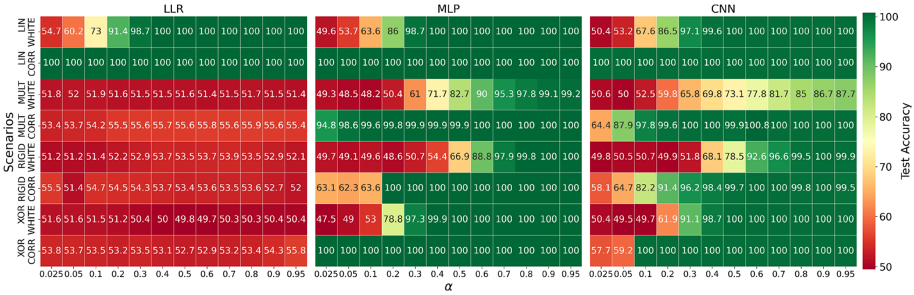


Fig. 13 Average test accuracy over 10 model trainings for each problem scenario and model architecture of the 8 × 8-px setting, for a fixed range of signal-to-noise ratios (SNRs). As expected, the Linear Logistic Regression (LLR) model cannot perform above chance level for non-linear scenarios. The Convolutional 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

Table 3 Results of the model training process for each classification setting, model architecture, and background type in the 8 × 8-px setting

		White		CORR	
		α	ACC	α	ACC
	LLR	0.1800	88.9	0.0125	99.9
LIN	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

These results are depicted as chosen Signal-to-noise ratios (SNRs), parameterized by α, as well as the average test accuracy (ACC, %)

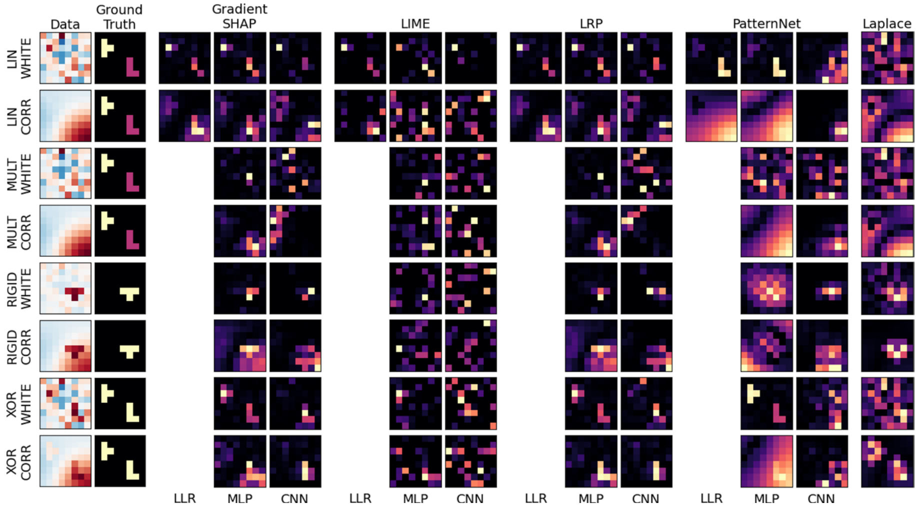
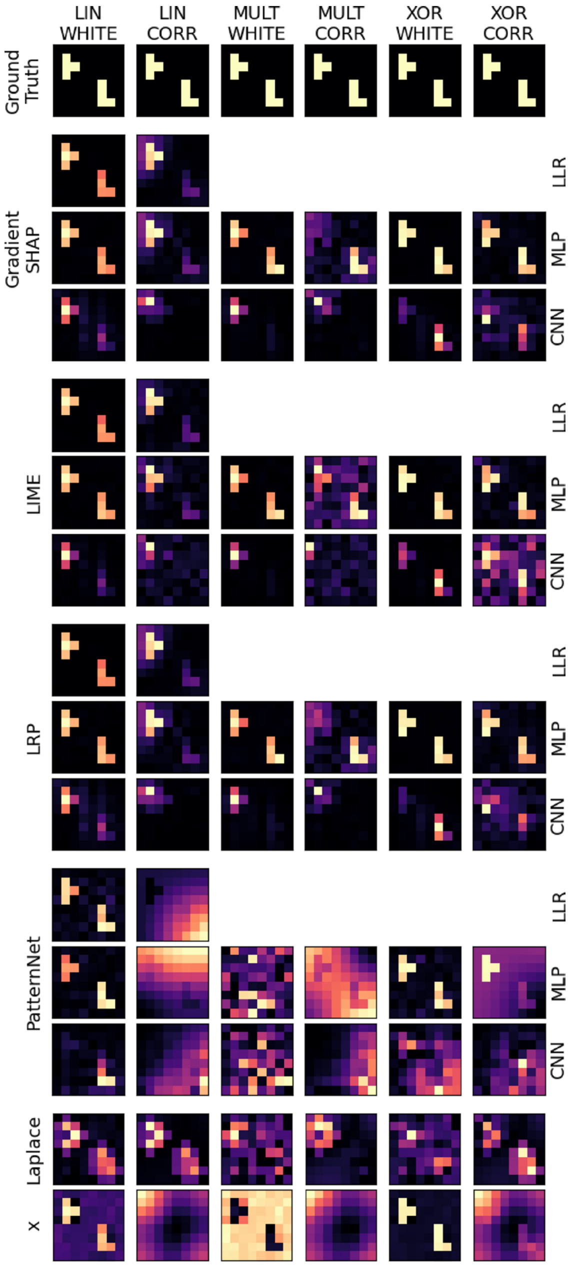


Fig. 14 Absolute-valued importance maps obtained for a random correctly-predicted 8 × 8-px data sample, for selected XAI methods and baselines. Recovery of the ground truth pattern across all scenarios is best shown by XAI methods applied to a Linear Logistic Regression (LLR) model

Fig. 15 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



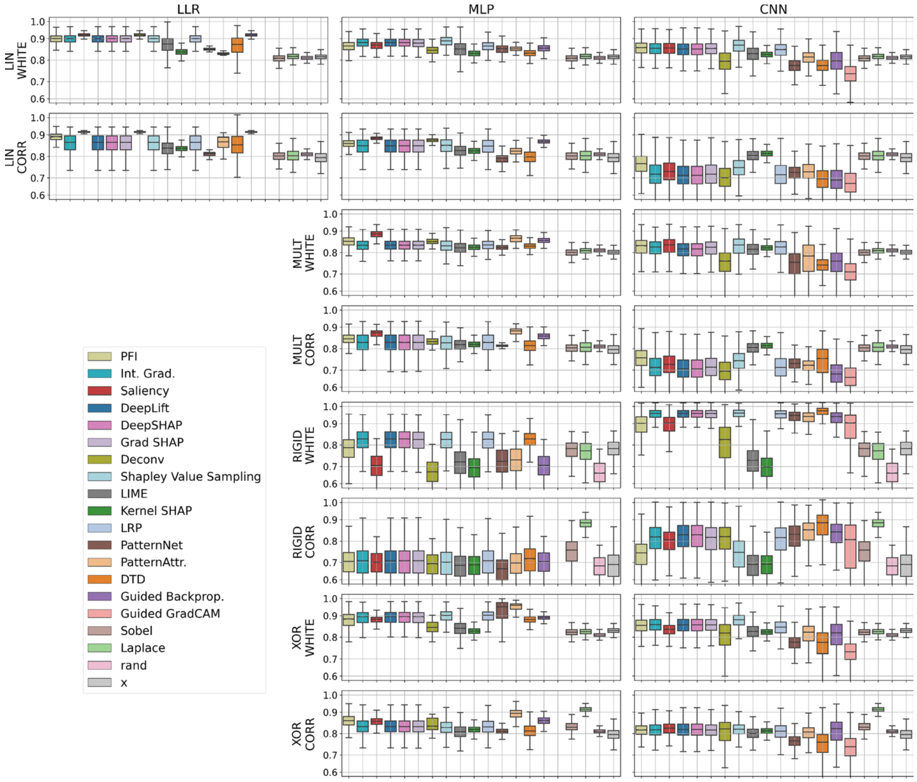


Fig. 16 EMD metric based on the Earth Mover's Distance (EMD) for every XAI method tested in the 8 × 8 -px setting, separated by model architecture and depicted as boxplots of median and quartile performance scores. Consistent with the results of Fig. 18, explanation performance tends to decrease as model 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

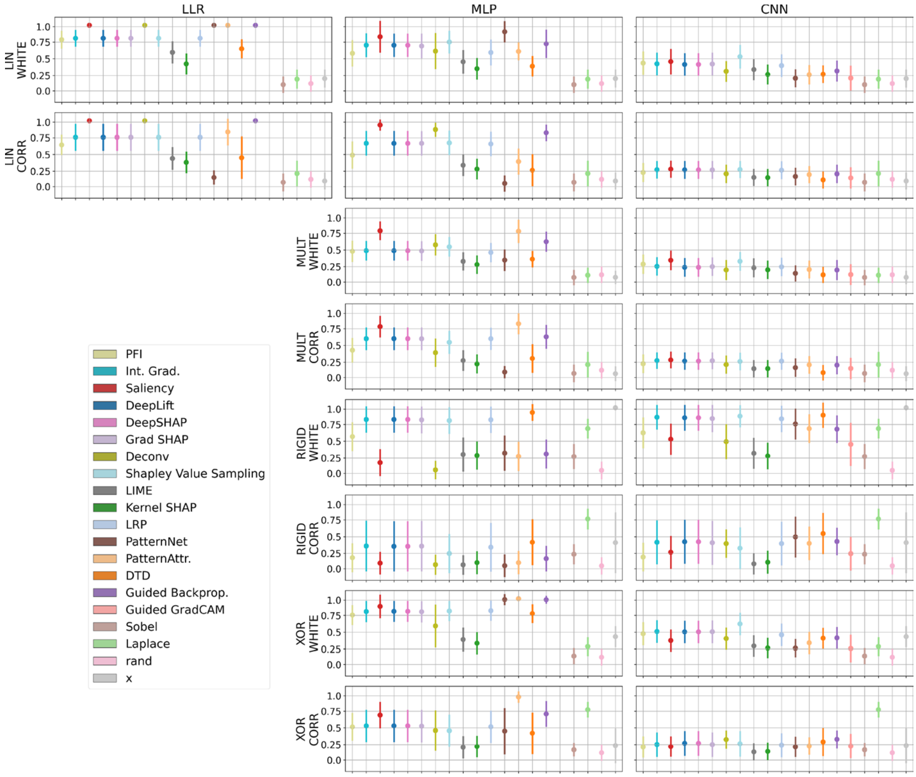


Fig. 17 Precision score for every XAI method tested in the 8 × 8-px setting, separated by model architecture and depicted as mean and standard deviation performance scores. Most methods outperform the baseline methods for most model-scenario parameterization pairs. The 'x' method, using input data as 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

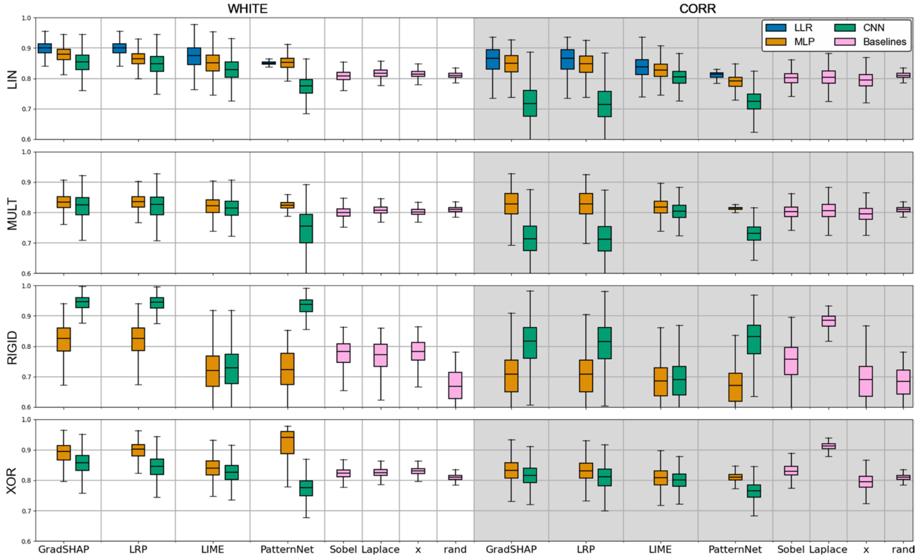


Fig. 18 Quantitative explanation performance of individual sample-based feature importance maps produced by various XAI approaches and baseline methods on correctly-predicted 8 × 8-px scenario test samples, as per the EMD metric. Depicted are boxplots of median explanation performance, with upper and 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 (Wilming et al., 2022), PatternNet and PatternAttribution do not outright outperform other XAI methods for most configurations

Figure 13 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 14 and 15 show qualitative results for local and global explanations respectively, and Figs. 16 and 17 show quantitative results for the EMD and Precision metrics respectiv ely.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Benedict Clark. The first draft of the manuscript was written primarily by Benedict Clark, and all authors commented on and edited all previous versions of the manuscript. All authors read, edited, and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. 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 in the frame of the Junior Research Group "Machine Learning and Uncertainty".

Availability of data and material All data used here can be generated using the provided code.

Code availability https://github.com/braindatalab/xai-tris.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

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

Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Mon-tavon, G., Samek, W., Müller, K.-R., Dähne, S., & Kindermans, P.-J. (2018). iNNvestigate neural networks!, arXiv:1808.04260

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

Asano, Y. M., Rupprecht, C., Zisserman, A., & Vedaldi, A.(2021). Pass: An imagenet replacement for self-supervised pretraining without humans. NeurIPS track on datasets and benchmarks

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

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

Castro, J., Gómez, D., & Tejada, J. (2009). Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research, 36*(5), 1726–1730.

Conger, A. J. (1974). A revised definition for suppressor variables: A guide to their identification and interpretation. *Educational and Psychological Measurement, 34*(1), 35–46.

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

Fisher, A., Rudin, C., & Dominici, F. (2019). 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.

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.,& Vayer, T. (2021). Pot: Python optimal transport. *Journal of Machine Learning Research, 22*(78), 1–8.

Friedman, L., & Wall, M. (2005). Graphical views of suppression and multicollinearity in multiple linear regression. *The American Statistician, 59*(2), 127–136.

Gevaert, A., Rousseau, A. J., Becker, T., A.-J., Becker, T., Valkenborg, D., De Bie, T., &Saeys, Y. (2022). Evaluating feature attribution methods in the image domain. arXiv e-prints arXiv:2202.12270. [cs.CV]

Golomb, S. W. (1996). *Polyominoes: Puzzles, patterns, problems, and packings* (Vol. 111). Princeton University Press.

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

Hedström, A., Weber, L., Bareeva, D., Motzkus, F., Samek, W., Lapuschkin, S., & Höhne, M. M. C. (2022). Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations. *Journal of Machine Learning Research*. https://doi.org/10.48550/ARXIV.2202.06861

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

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds,J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., & Reblitz-Richardson, O. (2020). Captum: A unified and generic model interpretability library for PyTorch, arXiv:2009.07896

Li, X.H., Shi, Y., Li, H., Bai, W., Cao, C. C., & Chen, L. An (2021). 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).

Lundberg, S. M., Lee, S. I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio et al (eds.), *Advances in neural information processing systems* (pp. 4765–4774), Curran Associates, Inc.

Mamalakis, A., Barnes, E. A., & Ebert-Uphoff, I. (2022). Carefully choose the baseline: Lessons learned from applying xai attribution methods for regression tasks in geoscience. *Artificial Intelligence for the Earth Systems, 2*(1), e220058.

Montavon, G., Bach, S., Binder, A., Samek, W., & Müller, K.-R. (2017). Explaining NonLinear classification decisions with deep taylor decomposition. *Pattern Recognition, 65*, 211–222.

Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt,Y., Schlötterer, J., van Keulen, M., & Seifert, C. (2023). From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI. *ACM Computing Surveys*. https://doi.org/10.1145/3583558

Nintendo of America. (1989). Tetris. https://search.library.wisc.edu/catalog/9910796303302121

Oliveira, M., Wilming, R., Clark, B., Budding, C., Eitel, F., Ritter, K., &Haufe, S. (2024). Benchmarking the influence of pre-training on explanation performance in mr image classification. *Frontiers in Artificial Intelligence*. https://doi.org/10.3389/frai.2024.1330919

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

Ribeiro, M.T., Singh, S., Guestrin, C. (2016). 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).

Rosenfeld, E., Ravikumar, P., & Risteski, A. (2021). The Risks of Invariant Risk Minimization, arXiv:2010.05761

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., &Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE international conference on computer vision (ICCV)*, (pp. 618–626).

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In *ICML*.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at international conference on learning representations*.

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

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In *ICML*.

Tjoa, E., Guan, C. (2020). Quantifying explainability of saliency methods in deep neural networks, arXiv:2009.02899

Wilming, R., Budding, C., Müller, K. R., & Haufe, S. (2022). Scrutinizing XAI using linear ground-truth data with suppressor variables. *Machine Learning, 111*(5), 1903–1923.

Wilming, R., Kieslich, L., Clark, B., & Haufe, S. (2023). Theoretical behavior of XAI methods in the presence of suppressor variables. In A. Krause, E. Brunskill, K. Cho, et al (eds.), *Proceedings of the 40th international conference on machine learning, proceedings of machine learning research*, vol. 202. PMLR, (pp. 37091–37107), https://proceedings.mlr.press/v202/wilming23a.html

Zeiler, M. D., & Fergus, R., et al. (2014). Visualizing and Understanding Convolutional Networks. In D. Fleet, T. Pajdla, & B. Schiele (Eds.), *Computer vision -ECCV 2014, Lecture notes in computer science*, (pp. 818–833). Springer International Publishing.

Zhou, Y., Booth, S., Ribeiro, M. T., & Shah, J. (2022). Do feature attribution methods correctly attribute features? In *Proceedings of the AAAI conference on artificial intelligence*, (pp. 9623–9633).