electronics

MDPI

Article

Explainability and Evaluation of Vision Transformers: An In-Depth Experimental Study

Sédrick Stassin [1,*], Valentin Corduant [1], Sidi Ahmed Mahmoudi [1] and Xavier Siebert [2]

[1] ILIA Unit, University of Mons, 7000 Mons, Belgium; valentin.corduant@student.umons.ac.be (V.C.); sidi.mahmoudi@umons.ac.be (S.A.M.)
[2] Department of Mathematics and Operational Research, University of Mons, 7000 Mons, Belgium; xavier.siebert@umons.ac.be
* Correspondence: sedrick.stassin@umons.ac.be

Abstract: In the era of artificial intelligence (AI), the deployment of intelligent systems for autonomous decision making has surged across diverse fields. However, the widespread adoption of AI technology is hindered by the risks associated with ceding control to autonomous systems, particularly in critical domains. Explainable artificial intelligence (XAI) has emerged as a critical subdomain fostering human understanding and trust. It addresses the opacity of complex models such as vision transformers (ViTs), which have gained prominence lately. With the expanding landscape of XAI methods, selecting the most effective method remains an open question, due to the lack of a ground-truth label for explainability. To avoid subjective human judgment, numerous metrics have been developed, with each aiming to fulfill certain properties required for a valid explanation. This study conducts a detailed evaluation of various XAI methods applied to the ViT architecture, thereby exploring metrics criteria like faithfulness, coherence, robustness, and complexity. We especially study the metric convergence, correlation, discriminative power, and inference time of both XAI methods and metrics. Contrary to expectations, the metrics of each criterion reveal minimal convergence and correlation. This study not only challenges the conventional practice of metric-based ranking of XAI methods but also underscores the dependence of explanations on the experimental environment, thereby presenting crucial considerations for the future development and adoption of XAI methods in real-world applications.

Keywords: explainable artificial intelligence; XAI; vision transformers; deep neural networks; metrics; evaluation; computer vision; deep learning; artificial intelligence

Citation: Stassin, S.; Corduant, V.; Mahmoudi, S.A.; Siebert, X. Explainability and Evaluation of Vision Transformers: An In-Depth Experimental Study. *Electronics* **2024**, *13*, 175. https://doi.org/10.3390/electronics13010175

Academic Editor: Alberto Fernandez Hilario

Received: 14 November 2023
Revised: 18 December 2023
Accepted: 22 December 2023
Published: 30 December 2023

1. Introduction

Shortly after the emergence of the first computers, researchers became interested in developing "intelligent" systems capable of making decisions and operating autonomously [1]. Over the past decades, artificial intelligence techniques such as machine learning have made significant progress, and numerous prototypes have been studied for use in diverse fields such as personal assistants, logistics, transportation, surveillance systems, high-frequency trading, healthcare, and scientific research. Although some artificial intelligence systems have already been deployed, a limiting factor for broader adoption of this technology is the inherent and undeniable risk associated with handing control and human supervision over to autonomous systems [2]. For sensitive tasks involving critical infrastructures or impacting human well-being, health, and safety, it is crucial to limit the possibility of automated systems making inappropriate or dangerous decisions. Before deploying such systems, it is necessary to validate their behavior and establish guarantees that they will continue to function as intended when deployed in a real-world environment. The operation of simple models such as shallow decision trees or Bayesian models is easily interpretable in artificial intelligence, but their predictive capacity is limited. The latest deep

neural networks provide significantly higher predictive power but come with a "black box" behavior, where the underlying reasoning is much more challenging to extract. Therefore, new tools must be deployed to achieve this security objective, thereby allowing humans to verify the alignment between artificial intelligence decisions and their objectives.

Explainable artificial intelligence (XAI) has emerged as a subdomain of artificial intelligence to produce explanations of neural networks, thereby enabling humans to understand, trust, and effectively manage this new generation of artificially intelligent partners. Several XAI methods have recently been proposed to explain the outcomes of deep neural networks, particularly convolutional neural networks, thereby allowing researchers to better understand and interpret the decisions made by these models. In 2017, the introduction of a new artificial intelligence model architecture called "Transformers" [3] enabled several breakthroughs in state-of-the-art performance, notably with the emergence of the BERT model [4] for natural language processing (NLP) tasks. The most important contribution of transformers lies in their ability to consider the relationships between different parts of a data sequence using attention mechanisms to weigh the importance of each element relative to others. They are characterized by their flexibility, with the basic architecture being easily adaptable to multiple tasks. This success of transformer models inspired the development of an adaption for computer vision (CV), known as vision transformers (ViTs), in 2020 [5]. ViTs have emerged as significant models in computer vision, as is evidenced by the increasing citations of ViTs in the months following their introduction, as has been documented by Liu et al. [6]. Additionally, Dosovitskiy et al. [5] established that vision transformers stand out as the most performant models as model sizes increase over certain limits. Furthermore, vision transformers have exhibited great performances in diverse tasks, including tracking [7], segmentation [8], and detection [9].

In addition, the AI Act [10] established by the European Commission outlines rules and norms to govern the responsible use and deployment of AI in the European Union. This framework emphasizes three key elements that AI should adhere to: legality—as exemplified by the General Data Protection Regulation (GDPR) ensuring users' right to total transparency regarding decisions made by automated systems; ethics—to prevent biases in AI towards individuals; and robustness—thereby ensuring that deployed AI systems do not encounter critical failures with severe consequences (e.g., in autonomous cars). Thus, explaining vision transformers becomes imperative to ensure their widespread adoption. A straightforward approach for vision transformers (ViTs) involves the use of the attention weights as a demonstration of explanation. These weights effectively represent the significance assigned to each part of the input. However, research has revealed that relying solely on raw attention is inadequate for explaining transformer results, as it considers the query and key elements of the self-attention, but not the value [11–13]. This realization has prompted the development of newer explainable artificial intelligence (XAI) methods that are specifically tailored for vision transformers. In the absence of a definitive ground-truth label intrinsic to explainability, XAI method users are left with the option of evaluating them either visually or through metrics designed to assess XAI methods without relying on human judgments.

With the growing number of XAI methods and XAI metrics in the field, the main objective of this work is to assess the convergence of XAI metrics criteria, including faithfulness, coherence, robustness, and complexity, in the context of modern XAI methods applied to the ViT architecture. We explore the obtained visual results of an XAI method in detail and examine the convergence of metrics through comprehensive mean score analysis. Additionally, we delve into their concordance using Kendall's τ_b rank coefficients [14]. The findings highlight minimal to no correlation or convergence across the evaluated criteria. Due to the significant divergence in results, we argue that it is unfounded to attempt to quantify the strength of a criterion by averaging its metrics. Subsequently, we turn our attention to assessing the discriminating power of these metrics. Our analysis demonstrates that faithfulness metrics exhibit limited discrimination between various explainability methods,

which is in contrast to complexity metrics. Our main contributions can be summarized within three key points:

1. A scientific review: This work conducts a comparative analysis of XAI methods for the ViT architecture that, to the best of our knowledge, does not yet exist in the literature.
2. A framework for XAI method analysis: The development of a high-level framework allows for the integration of all available XAI methods for ViTs and evaluates them using metrics that are currently available in the literature. This framework provides visual insights into explanations and helps understand the model's functioning.
3. An experimental analysis: This study delves into the results of XAI methods and their evaluation through metrics, analyzing convergence, correlation, and the discriminative power of the metrics.

The remainder of this paper is organized as follows:

- **Section 2—Related Works:** This section begins with an exploration of the taxonomy of explainability methods, followed by an examination of the vision transformer architecture, thus culminating in the attention process. The discussion then delves into a detailed analysis of modern XAI methods adapted from CNNs or tailored specifically for vision transformers.
- **Section 3—Experimental Setup:** This section meticulously explains the protocol employed for obtaining and studying the results, thereby establishing a robust methodological foundation for the experimentation.
- **Section 4—Results:** Undertaking a comparative study, this section evaluates various XAI methods presented in the state of the art. Assessment is conducted through key criteria found in the scientific literature, thereby encompassing faithfulness, complexity, randomization, and robustness, along with associated metrics.
- **Section 5—Conclusions:** Providing reflections informed by the results of the experimentation, this section discusses the implications of utilizing XAI methods and sheds light on the limitations inherent in current evaluation systems.

2. Related Works

2.1. XAI Taxonomy

As a research discipline, explainable artificial intelligence (XAI) has seen exponential growth in recent years. This growth has made it challenging for both new and experienced researchers to navigate the constantly evolving landscape of XAI methods. To address these challenges, an increasing number of articles aim to create taxonomies, which are crucial for organizing research. However, the diversity and complexity of explainability methods make it impossible to fit them all into a single taxonomy. Each taxonomy inevitably specializes in specific aspects. This situation leads to several issues. Firstly, researchers struggle to find representative classification categories for their methods within a single taxonomy, as categories are often depicted as mutually exclusive, which rarely aligns with real-world applications. Secondly, researchers grapple with managing numerous nonuniform taxonomies. Furthermore, different taxonomies may have similar categories with different names, or the same category name may have varying definitions. To compound the confusion, scientific articles typically do not specify which taxonomy and definition they adhere to. Figure 1 summarizes the primary taxonomic approaches from the current scientific literature. There are three taxonomies detailed in the following paragraphs, with each distinguished by their approach: conceptual, function-based, and result-based.

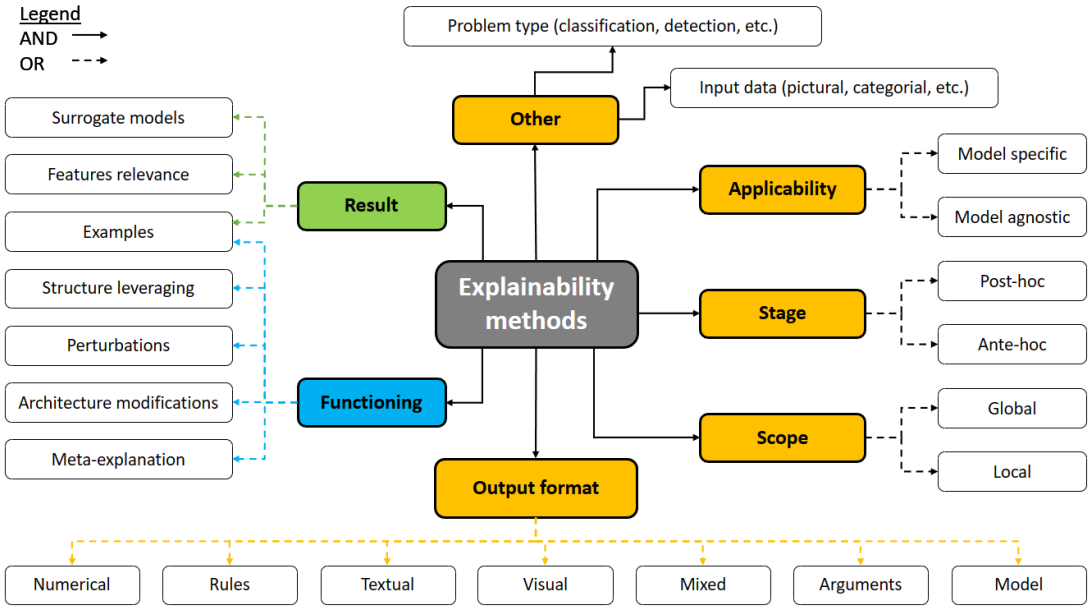

Figure 1. Overview of the proposed categorization of explainability methods modified from [15].

The **function-based taxonomy**, highlighted in blue in Figure 1 and introduced by Samek and Müller [16], takes the underlying workings of an explainability method as the essential element of its classification. Functioning refers to how an explainability method extracts information from a model, and this classification includes three categories.

- *Local Perturbations*: These methods slightly alter the model's inputs to assess the importance of specific features on the model's predictions
- *Architecture Exploitation*: These methods leverage specific properties of the model's structure to build explanations. An example of architecture exploitation is examining gradients through a backpropagation process, which provides information on the importance of input values.
- *Meta-Explanation*: These methods aggregate and compare explanations from other XAI methods to create a more comprehensive explanation than any single method alone.

Arrieta et al. [17] expanded this taxonomy by adding two additional categories:

- *Architecture Modification*: These methods simplify complex models by altering their architecture and are often referred to as model distillation.
- *Example Selection*: These methods pick representative examples that generate high or low certainty, thereby offering insights into the model's internal functioning.

McDermid et al. [18] categorized explainability according to the result obtained, which is highlighted in green in Figure 1 as the **result-based taxonomy**. It comprises three categories:

- *Feature Relevance*: Probably being the most-used type of result-based methods, these highlight the importance of input features in the model's predictions through saliency maps (or heatmaps). The maps assign a relevance score to each feature, thereby quantifying their impact on the model's output.
- *Surrogate Models:* These methods construct partial models to simplify and interpret a specific portion of the original model. They utilize local perturbations or exploit the model's architecture to extract information.
- *Example Selection*: These methods, as proposed by Arrieta et al. [17] in the function-based approach, are also part of the result-based approach.

The last taxonomy, introduced by Sayed-Mouchaweh [19], makes use of the concept applicable to explainability to divide them, which is highlighted in yellow in Figure 1 as the **concept-based taxonomy**:

- *Ante Hoc or Post Hoc*: These methods depend on the position. Post hoc methods generate explanations for all types of models and are applied after model training. Ante hoc methods, which are tailored for intrinsically interpretable models, provide explanations generated during training.
- *Local or Global*: Local scope methods explain a single prediction, while global scope methods provide explanations for the entire model. This distinction is commonly made for post hoc explainability methods but also holds relevance for complex ante hoc methods that are challenging to interpret as a whole. For instance, an individual prediction from a highly branched decision tree can be directly interpretable despite the complexity of the tree structure.
- *Agnostic or Specific*: Regarding applicability, agnostic methods work with all types of models, while specific methods apply only to particular models.

The scientific literature also mentions two additional conceptual categories:

- *Problem Type*: This category of methods is defined by the type of problem for which the method is suitable, such as classification or object detection.
- *Output Format*: This category is based on the formats in which explanations are presented, including numerical, rule-based, textual, visual, mixed, arguments, or even a model. Despite its potential for tailoring explanation results to specific objectives and audiences, this category is not yet widely referenced in the literature.

This general overview of taxonomies provides insight into the categories of explainability tools that are currently available. For this work, only selected categories will be employed, as the goal is to investigate XAI methods applied to ViTs. Specifically, the focus is on analyzing specific and agnostic post hoc methods due to the lack of transparency in transformers. Most of the examined methods will center on feature importance at a local level. The results, primarily visual, will be obtained through various techniques, such as perturbation, architecture exploitation, and modification, to explain predictions from a ViT image classification model. The choice to use these XAI methods is driven by the current research direction in this field. The categories utilized in this work are highlighted in blue in Figure 2.

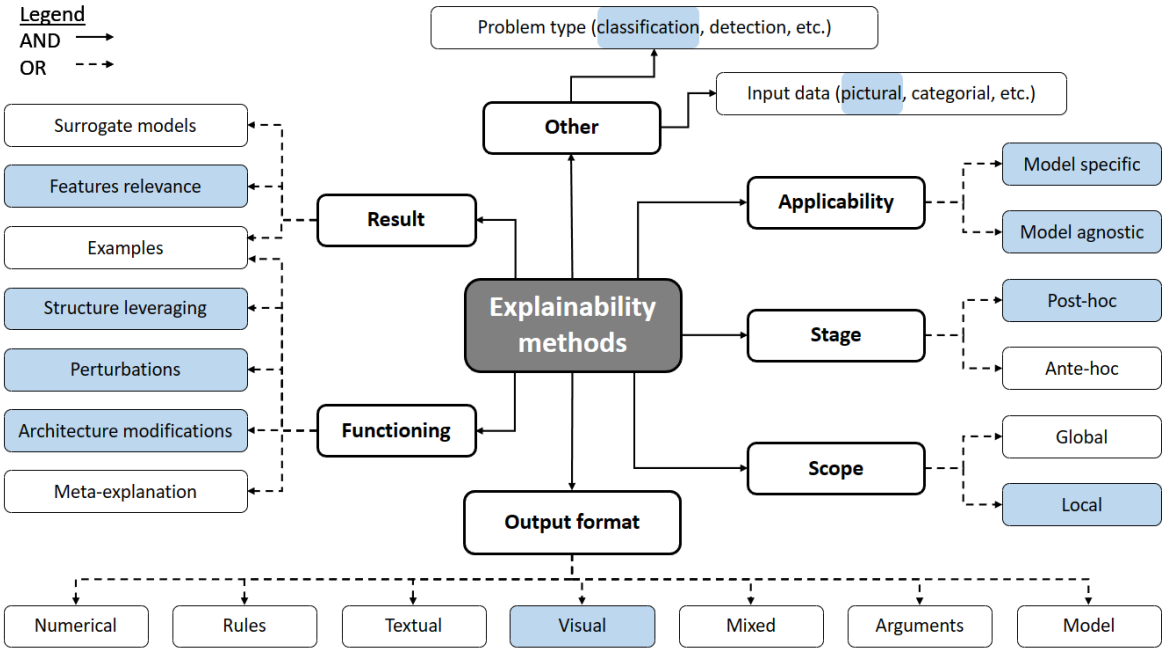

Figure 2. Part of the taxonomy of explainability methods tackled in this work, underlined in blue and modified from [15].

2.2. Transformers

Recurrent neural networks (RNNs) have played a pivotal role in solving numerous problems in fields like speech recognition and machine translation. However, despite these advancements, Amini [20] identified three key limitations associated with them:

The first limitation is the sequential encoding, which is inherent in RNNs and LSTMs. This encoding of data does not guarantee that information is accurately maintained and recorded from the beginning to the end of the sequence. In practice, there is a significant loss of information when processing long sequences, even with the addition of dedicated memory. The second limitation is the lack of parallelization, which is not computationally efficient. Sequential data processing does not make optimal use of the power of graphics processing units (GPUs), which are commonly employed for parallelizing independent computations in deep learning. The third limitation is their limited memory capacity, which prevents the storage of all the information present in long sequences containing thousands of elements.

In response to these limitations, a novel neural network architecture, known as transformers, was introduced in 2017 by Vaswani et al. [3]. Thanks to their significantly more efficient architecture, transformers can process very long sequences of data in parallel. Vaswani et al. [3] argued that the attention mechanism, a key principle of transformers, can completely replace RNNs and LSTMs. To achieve this, transformers have eliminated the need for feedback in networks, thereby using a feedforward architecture that allows for efficient parallel computation. Data are processed continuously and in parallel, thereby fully utilizing the power of GPUs. Moreover, transformers have a much larger long-term memory capacity than RNNs and LSTMs, thereby enabling them to handle much longer data sequences. Dosovitskiy et al. [5] trained a transformer architecture for image classification. They were the first to successfully demonstrate that training a transformer on ImageNet [21] leads to excellent results, particularly when the model is trained on a very large dataset, thereby surpassing conventional convolutional architectures like Xception [22], EfficientNet [23], and ResNet [24]. The ability to capture long-range dependencies in images enables them to effectively model global relationships between pixels, thereby making them well suited for tasks requiring global scene understanding. The remainder of this section focuses on describing vision transformers and their attention mechanism.

2.2.1. Vision Transformer Architecture

The transformer architecture is an encoder–decoder architecture based on multihead self-attention. This architecture is designed to address sequence data-related problems in two stages. The encoder carries out the initial data processing. Its role is to transform the input sequence into a lower-dimensional vector representation, thereby capturing the essential information of the sequence in a comprehensible and compact form for the decoder. This representation, called feature vector (embedding), is generated by passing the sequence through a series of recurrent neuron layers, convolutional filters, or attention mechanisms. The decoder performs the second phase; it processes the feature vector to produce an output sequence that corresponds to the model's specific task, such as translating a sentence or generating a sequence of words. A vision transformer (ViT) is designed for a classification task. This task does not require the generation of new information, unlike tasks like image generation or image quality improvement. The feature vector established by the encoder is directly used for classification. Therefore, a ViT only utilizes the encoder part of a transformer. ViTs have a remarkable versatility, as they can be pretrained on large-scale datasets and fine-tuned for specific tasks, thereby showcasing their potential for transfer learning.

The rest of this section focuses on data preparation (see Figure 3) for processing using the encoder and presents its internal architecture. Originally, transformers were designed for natural language processing. Hence, their encoder takes an input sequence of words grouped into a single information vector where each word is considered to be a token in that sequence. Conversely, computer vision differs from natural language processing due to the nature of the data being processed. Images consist of pixels that do not have clear relationships with each other, unlike the words in a sentence. To adapt the Transformer architecture to this domain, Dosovitskiy et al. [5] treated an image not as a grid of pixels but as a sequence of patches (equivalent to tokens in language processing). The architectural parameters specified in this section were those of the *ViT-base-patch16-224* model studied in this work. The input image needed to be resized to a $(224, 224)$ dimension through preprocessing. The attention mechanism described in the section below was employed to assess the significance of each element in the sequence relative to others. This mechanism carries a quadratic computational complexity, as each input element must be compared to all others. This $O(n^2)$ complexity makes it infeasible to directly use 50,176 image pixels as input elements in the model. To feed the encoder, each input image is divided into a sequence of nonoverlapping square patches, with each of size $(16, 16)$. In this case, there were $N = 224^2/16^2 = 196$ patches in total. Each patch was a matrix of 256 pixels, which was then reshaped into a vector of dimension $(1, 256)$. Their reliance on a fixed-size grid of nonoverlapping patches may compromise their ability to capture fine-grained details in images. This can be a disadvantage in scenarios where local information is crucial, such as in object detection tasks. The image, transformed into a sequence of data, can be used by the ViT-like transformers in natural language processing tasks. Subsequently, each patch was linearly projected into a new space of dimension $(1, 768)$. The dimension of the embedding space was selected to strike the right balance between model accuracy and encoder computational load. This new vector, known as feature vector (embedding), represents the initial patches in a vectorized form. The weights W_{embed} associated with this projection were learned during the model's training process.

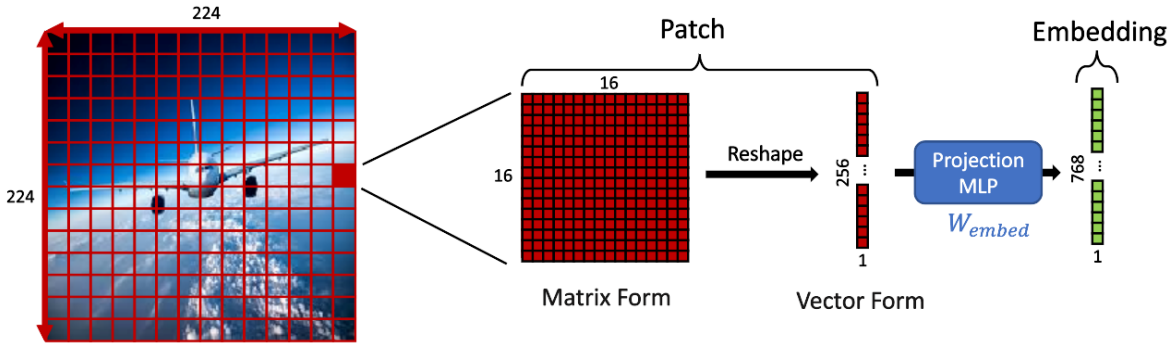

Figure 3. Vision transformer: image processing into patches and embedding.

An additional token, [CLS], was added to the beginning of the sequence of patches to represent the object class present in the image. This addition is solely related to the model's classification objective, because the ViT makes its predictions solely based on the embedding of [CLS] at the output of the network. Transformers are insensitive to permutations, which means they do not take into account the spatial position of input patches, even though this is important in image analysis. Therefore, it is necessary to include positional information for each patch in the embedding. These position details are added in the form of position embeddings, which can be either learned during training or predefined. Typically, the prediction is made by projecting the final embedding of the [CLS] token through layers of multilayer perceptrons (MLPs), as are shown in Figure 4.

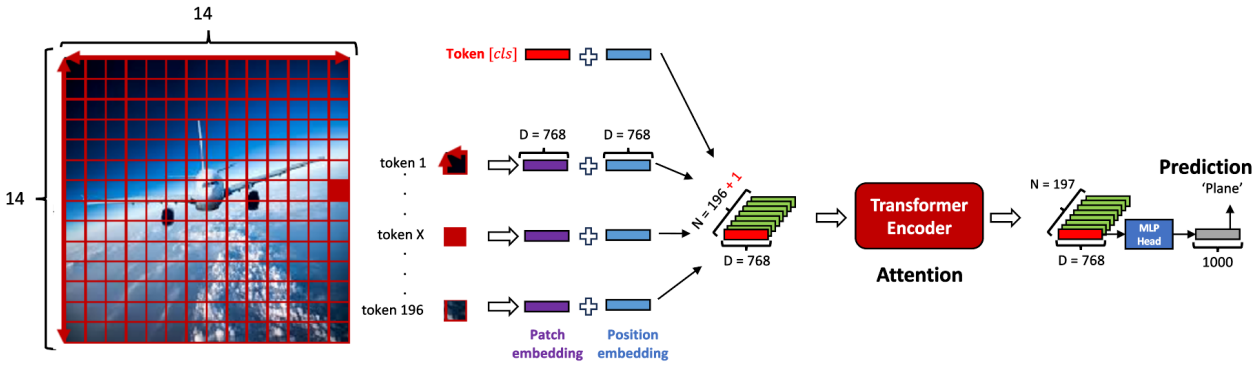

Figure 4. Vision transformer: complete process outside the encoder, which is modified from [25].

The remainder of this section introduces the encoder's architecture (refer to Figure 5), thereby intending to rework the information so that the output captures the essential elements of the input. In a transformer, embeddings pass through the encoder L times to refine and create a more precise, contextually nuanced representation. An encoder block is characterized by the following elements:

- Layer Normalization (LN): These layers enhance the learning speed and model generalization without introducing new dependencies among the embeddings.
- Multihead Self-Attention Layer (MSA): This forms the heart of the transformer encoder.
- Multilayer Perceptron (MLP): It consists of two layers with GeLu-type activation functions.
- Residual Connections: These connections enable gradients to flow directly through the network.

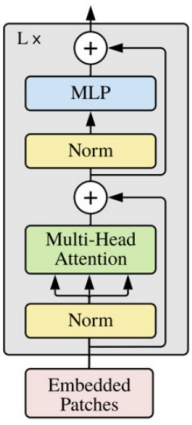

Figure 5. Vision transformer: encoder block [3].

2.2.2. Attention

The attention mechanism enriches each embedding by considering the information contained in other embeddings. This mechanism transforms the representation of the input sequence into a more contextualized one. This process is akin to how the human brain rapidly extracts the primary content from an image by scanning it with the eyes. It directly focuses on regions of interest and only subsequently identifies the precise location of the object. Guo et al. [26] have demonstrated that our brains indeed concentrate on specific regions of an image when identifying an object.

To process a sequence of embeddings, you need to create three elements: query, key, and value (Q, K, V). Vaswani et al. [3] defined these elements as the projection of the embedding into three other spaces defined by the projection matrices W_Q, W_K, and W_V, respectively. These weight matrices are learned by the model during training (see Figure 6).

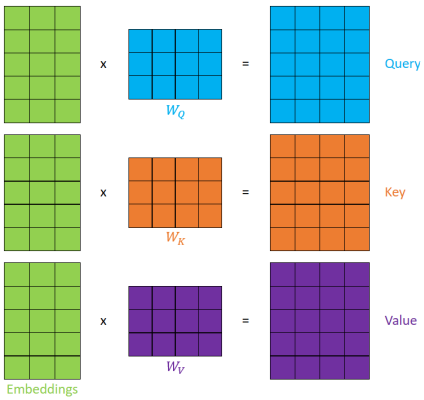

Figure 6. Embedding projection for the creation of matrices Q, K, V.

Once these elements (Q, K, V) are calculated, the attention score is determined by the similarity between queries and keys. Mathematically, a classical tool for quantifying the similarity between two vectors is the dot product. Geometrically, this is akin to identifying which keys are oriented in the same direction as the queries. Therefore, the attention score is calculated as follows:

$$\text{attention score} = A(Q, K) = softmax((Q^T K)/\sqrt{d}) \quad (1)$$

The softmax function converts the attention score into a probability distribution. The addition of a scaling factor $\sqrt{d}$ helps mitigate the "exploding gradient" problem. When the input is too large, the softmax activation function can yield an extremely small gradient, thereby slowing down the learning process. Here, d represents the dimension of an attention head, as is discussed later in this section.

Typically, the attention score is represented in matrix form. Each row of this matrix indicates the attention that a given embedding pays to all the other embeddings in the sequence. This mechanism is often referred to as self-attention in the literature. The value of the information is then weighted by the attention scores and aggregated into the enriched embeddings as follows:

$$\text{enriched embeddings}(Q, K, V) = softmax((Q^T K)/\sqrt{d}).V \quad (2)$$

Figure 7 illustrates the self-attention mechanism:

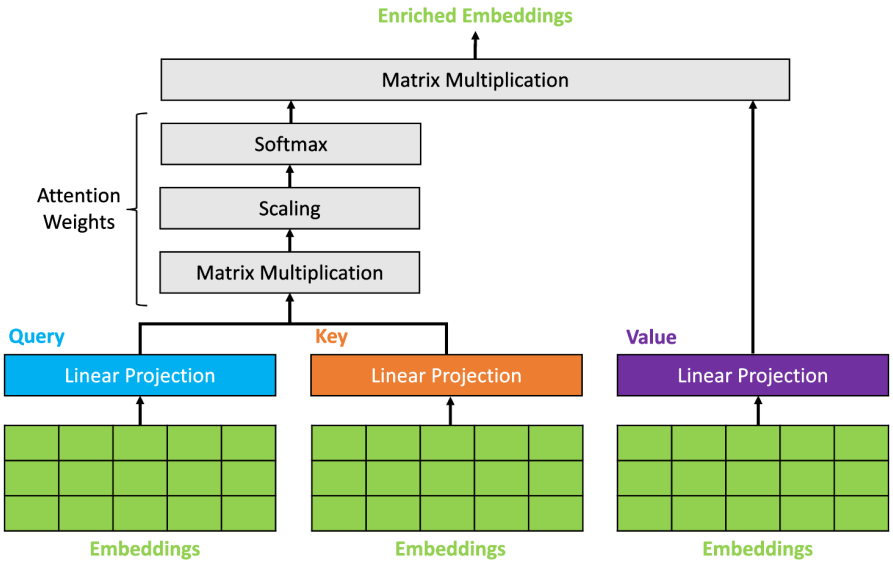

Figure 7. Self-Attention mechanism summary, inspired by [20].

Multiple self-attention processes can be employed in parallel to form a more complex network architecture. Each of these processes becomes a distinct attention head. These various heads are used to extract different information as if each could view the input sequence from a different perspective. To put this into practice, the embeddings, which are the vectorized representation of the input data, are divided based on the number of attention heads used in the network. Each attention head then applies the attention mechanism to a specific portion of the feature vector for each patch.

2.3. XAI Methods

Explainability methods tailored for tision Transformers (ViTs) can be divided into two categories. The first one includes methods initially used in computer vision for convolutional neural networks (CNNs) but can also be applied or adapted for ViTs. The second category encompasses methods specifically designed for ViTs. This section does not delve into how these methods can be utilized in tasks other than image classification, such as object detection, segmentation, visual question answering, and more. The objective is to elucidate the theoretical functioning of these methods dedicated to experimentation. The common goal of all these explainable AI (XAI) methods is to explain the classification model for a specific class by measuring the importance of each pixel in an image for the final prediction. This importance is summarized in a saliency map, with results expressed as scores within the [0, 1] range. These scores are visualized in a heatmap (also called a saliency map or an attribution map), where hot areas (represented in red) indicate the model's strongest points of interest, while cold areas (in blue) are considered less influential in the decision-making process of the model. The following conventions are adopted:

- The network's input is a vector $\mathbf{x} \in \mathbb{R}^p$, where p is the number of pixels in the image.
- The neural network f predicts a vector $\mathbf{y} = [y_1, \ldots, y_C]$, where y_i is the score of the prediction (logit) for class c.
- The vector $\mathbf{y}$ is transformed into a probability vector $\mathbf{S} = [S_1, \ldots, S_C]$ through a softmax function, where S_i represents the probability that the image belongs to class c.
- The network's error is evaluated by the cost function $\mathcal{L}$.
- The explanation of the network is a vector $\mathbf{R}^c = [R_1^c, \ldots, R_p^c]$, where R_i^c is the relevance score of pixel i in the prediction for class c.

2.3.1. Methods for CNNs

Three families of methods are commonly used for CNN explainability. Two are based on architectural exploitation: gradients and feature maps. The third is based on local perturbations. The rest of this section examines these method families in detail and their adaptation to the ViT model.

Gradient Methods: These explainability methods are based on gradient backpropagation. They involve evaluating the gradients of input features to determine their importance in prediction. If the value of a pixel changes, the prediction probability for a given class also changes proportionally to the associated gradient value. In other words, the higher the absolute value of the gradient, the more significant the impact of that pixel on the prediction.

The first method is the visualization of the loss function gradient for the class of interest, whicch is weighted according to the input pixel values. This method, called input Grad [27], can be mathematically represented by the following equation:

$$R^c = x \cdot \frac{\partial L_c(x)}{\partial x} \tag{3}$$

The second method is Smooth Grad [28], which is an extension of the previous method. It is based on the observation that, in practice, saliency maps generated by the input gradient method highlight significant areas for human observers but also reveal seemingly aberrant pixels that introduce noise to the explanation. The main idea of this method is to calculate an average of gradients over a set of image samples after applying a slight Gaussian

noise to smooth out these aberrations. Let M denote the number of noisy images, and let $x_i = x + \mathcal{N}(0, \sigma_i)$ represent the noisy image. By averaging the explanations obtained from these samples, the results are generally less noisy:

$$R^c = \frac{1}{M} \sum_{i=1}^{M} \left(\frac{\partial L_c(x_i)}{\partial x_i} \cdot x_i \right) \tag{4}$$

A third method is Integrated Grad [29], which also calculates an average of gradients. However, the result is computed over a sample of M images obtained through interpolation between the input image x and a reference image b (for example, a black or white image):

$$R^c = \frac{(x-b)}{M} \sum_{i=1}^{M} \frac{\partial f(L_c(b + \frac{i}{M}(x-b)))}{\partial x_i} \tag{5}$$

Perturbation Methods: These methods assess the relevance of pixels by modifying or removing them and observing the resulting change in prediction. They assume that the model's performance decreases when essential information is absent.

The occlusion method [30] sequentially masks groups of pixels (from left to right and top to bottom) and measures their marginal relevance to the model's prediction. However, a limitation of the occlusion method is that it solely considers the marginal relevance of pixels, while the covariance of a group of sensitive pixels also impacts the model's prediction. The iterative displacement of occlusion masks poses the risk of partially masking this group, thereby diminishing the synergy of these pixels on the prediction. Other perturbation methods vary the parameters of the perturbation window: its dimension k, sampling, and displacement stride. These selections should be carefully made, because they influence the resolution of the heatmaps, which, in turn, is intrinsically connected to the computation time of the explanation.

The rise method [31] is a saliency map generation method based on the Monte Carlo process. This perturbation method generates N masks $\{M_1, \ldots, M_N\}$ by randomly perturbing parts of the input image. The masks are generated as follows: binary masks smaller than the image size are created and then enlarged to the image size using bilinear interpolation. After interpolation, the masks, consisting of continuous values in the interval $[0, 1]$, are applied to sets of pixels. To introduce greater variety in the mask generation, they are finally randomly shifted by a few pixels in both directions. A weight P_{M_i} is assigned to each mask M_i that is proportional to the prediction of class c for the perturbed image. Finally, the saliency map is calculated as the sum of masks weighted by their respective weights:

$$R^c = \frac{1}{E(M) \cdot N} \sum_{i=1}^{N} P_{M_i} \cdot M_i \tag{6}$$

where $P_{M_i} = f(x \cdot M_i)$.

CAM Methods: Zhou et al. [32] introduced class activation mapping (CAM), thereby utilizing the activations from the convolutional layers of a CNN to obtain saliency maps. However, as vision transformers (ViTs) do not employ convolutions to extract image information, these methods need to be adapted for ViT models. In a CNN, spatial information is extracted from convolutional filters. This information is then transmitted to a fully connected (FC) layer, which concludes with a softmax layer to provide the probability of belonging to each class. The CAM method proposes modifying this architecture by adding a global average pooling (GAP) layer to synthesize the features extracted by the CNN and transmit them to the FC layer. This architecture can be summarized as follows: GAP(conv) $\rightarrow$ FC $\rightarrow$ Softmax. This method requires retraining the model, because adding a GAP layer modifies the architecture, thereby necessitating the adaptation of weights to the new model features. To obtain a saliency map, the CAM method performs a linear combination of activation maps A generated by N convolutional filters from the last layer of the network,

with each weighted by the weights $w_{n,c}$ of the FC layer, where the pair (n, c) represents the connection between the nth neuron of the GAP layer and the cth neuron of the FC layer:

$$R^c = \sum_{n}^{N} w_{n,c} A_n \tag{7}$$

The Grad CAM method [33] constitutes a generalization of the CAM method for a broader spectrum of CNN architectures, as it does not require the use of a GAP layer and, consequently, model retraining. The weights of the activation maps A are no longer obtained through the GAP layer but rather from the gradients of the prediction score for a class c with respect to each activation map A. This method only requires the use of a differentiable final activation function (e.g., softmax) and can be applied to architectures that use an MLP at the end of the network. With $A_{n,i,j}$ the neuron of activation map $A_n \in \mathbb{R}^{p \times q}$ at position (i, j), the weights $w_{n,c}$ for activation maps are calculated by averaging the gradients of the prediction score y_c with respect to all pixels of the activation map n for class c:

$$w_{n,c} = \frac{1}{p \cdot q} \sum_{i=1}^{p} \sum_{j=1}^{q} \frac{\partial y_c}{\partial A_{n,i,j}} \tag{8}$$

To obtain a saliency map, Selvaraju et al. [33] linearly combine thesed activation maps A_n with their weights $w_{n,c}$. The final result was obtained by passing this result through a ReLU activation function to retain only the positive contributions of pixels to the prediction:

$$R^c = \text{ReLU}\left(\sum_{n}^{N} w_{n,c} A_n \right) \tag{9}$$

The Grad CAM++ method [34] is based on the assumption that calculating the weights of activation maps in the Grad CAM method results in assigning the same importance to each pixel within the same activation map A_n as an average of gradients is performed over the entire map. Intuitively, if an image contains three objects of interest, the Grad CAM method highlights the object with the most pixels, because these pixels have the greatest influence on the final score for class c, thereby leading to a higher gradient. Grad CAM++ assigns an equally important score to small objects that also contribute to the model's prediction. This is achieved by adding a pixelwise contribution, thereby allowing the gradient of all objects of the same class to be in the same order of magnitude. The Grad CAM++ method adds this pixelwise contribution by no longer averaging the gradients but summing the contributions of each pixel. This summation is weighted by $w_{n,c,i,j}$, where $w_{n,c,i,j}$ represents the weight of each pixel in activation map A_n for predicting class c:

$$w_{n,c} = \sum_{i=1}^{p} \sum_{j=1}^{q} w_{n,c,i,j} \cdot \text{ReLU}\left(\frac{\partial y_c}{\partial A_{n,i,j}} \right) \tag{10}$$

Chattopadhay et al. [34] demonstrated the methodology for obtaining this weight $w_{n,c,i,j}$ in their paper.

The score CAM method [35] enhances the Grad CAM method by addressing a common limitation in gradient-based methods. Firstly, it tackles the issue of gradient saturation, which occurs when the influence of a pixel becomes too significant but does not contribute further to increasing the model's confidence. This results in a weak gradient for this feature, even though its actual importance is high. In their work, Wang et al. [35] attempted to use the integrated gradient method to alleviate this phenomenon, but the results showed that this technique did not work well for CAM methods. Secondly, the Score CAM method also addresses the issue of false confidence in gradient-based methods. This phenomenon arises when higher weights in the activation map do not lead to a more intense heatmap. In other words, even if a region has high weights, it may not be genuinely considered important for the model's prediction. To overcome these limitations, Wang et al. [35] turned to a

perturbation method where the weights $w_{n,c}$ of activation maps were determined to avoid the aforementioned problems. This method involves using the activation maps themselves as perturbation masks for the input image. The weights $w_{n,c}$ are determined by the model's response to these perturbations. The first step is to retrieve activation maps, which typically come from the last layer of the CNN. Each activation map is then used to perturb the input image. The more important a region in the activation map, the less it disturbs the input image. The weight $w_{n,c}$ corresponds to the model's prediction score for class c, with the input perturbed by activation map A_n.

In conclusion, CAM methods have initially been designed for CNNs based on the hypothesis that convolutional filters extract features from the image and localize the objects of interest. However, this information is lost in fully connected layers. Therefore, it is appropriate to use the last convolutional layer to extract the richest information. As it stands, CAM methods cannot be directly applied to ViTs, because convolution operations are not used in ViT layers. However, adaptation can be achieved by using embeddings, which are similar to activation maps, as they contain rich information for classification in the model's last layers. Like the final activation maps of CNNs, embeddings from the last blocks of the encoder are also information-rich, thereby justifying their use in ViT technology. To use these embeddings based on the activation map principle, they need to be rearranged, thereby positioning them at the locations of the initial patches while ensuring not to use the class token that does not belong to the image (see Figure 8).

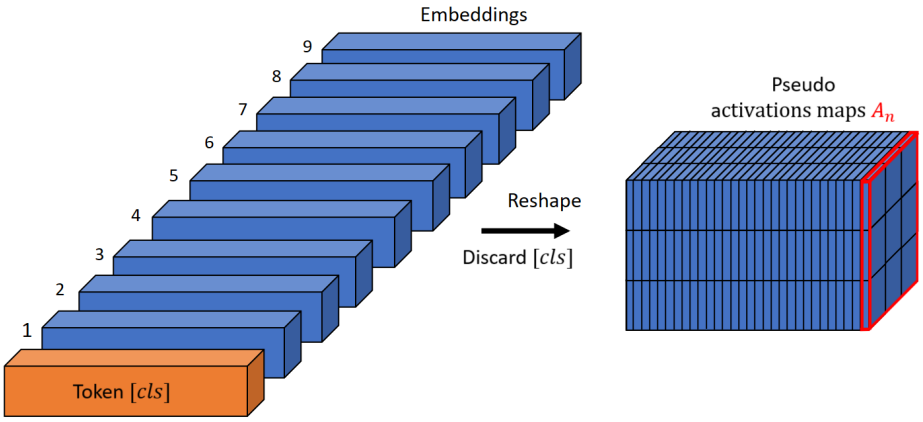

Figure 8. Adaptation of embeddings into pseudoactivation maps.

2.3.2. Methods for ViT

The main difference between CNN and ViT architectures lies in the use of an attention mechanism to analyze image information. These attention weights help represent the relationships that exist between the embeddings of tokens in an image, thus providing a better contextualization of the information. Consequently, initial attempts at interpretability for ViTs are simply based on the direct use of these attention weights, with visualization appearing to be the most straightforward way to understand a transformer's decisions and gain insights into its internal workings. For a given prediction, it is common to use the input feature with the highest attention score as an explanation. This approach is found in several articles [3,36]. Indeed, attention is a representation of the influence that tokens exert on each other. It is important to note that only the class token $[cls]$ is used to make the model's prediction. Therefore, visualizing the attention that this token directs to all others initially reflects the importance of each token in the prediction.

Upon analysis, the use of attention scores for interpretability does, however, present limitations. Firstly, the ViT architecture is composed of multiple attention heads, so how can one interpret the significance of these heads when the attention weights do not converge (see Figure 9)? Additionally, Michel et al. [37] demonstrated that one could remove most of these attention heads (pruning) without affecting the prediction accuracy, thereby indicating that they do not all have the same importance.

Furthermore, the ViT architecture consists of multiple encoder blocks. Each self-attention layer combines embeddings from the previous layer to calculate new embeddings for each token. As a result, Brunner et al. [38] observed that information from different tokens becomes increasingly intertwined across the encoder blocks. This concept is illustrated by Figure 9, which showcases the attention weights (represented by black arrows) of the embeddings (represented by red dots) in successive blocks of an encoder in a transformer with six layers and a single attention head trained to determine the agreement between the subject and the verb in the phrase "the key [*verb*] to the cabinets".

In this diagram, attention weights are more significant when the arrows are darker. In the first layer, it is visually evident that the [*verb*] token mainly focuses on the [*key*] token (see 1 in Figure 9). However, in the second layer, this attention is no longer as pronounced (see 2 in Figure 9). As we progress through the encoder blocks, the information becomes more mixed, and the weights become more uniform across different tokens. In reality, the attention weights of the network are often wrongly associated with attention between tokens, when, in fact, it is attention between the embeddings of two successive layers (Brunner et al. [38]). Furthermore, it is not accurate to directly equate attention with explanation (Jain and Wallace [13]), as this hypothesis has never been established. To formally establish that attention weights provide a faithful explanation for the model's prediction, the following additional observations should be established:

- Attention weights should be correlated with other feature importance measures, such as gradients. However, the results indicate that this correlation is generally weak.
- Alternative attention weighting configurations lead to corresponding changes in predictions (and if not, they would be equally plausible as explanations).

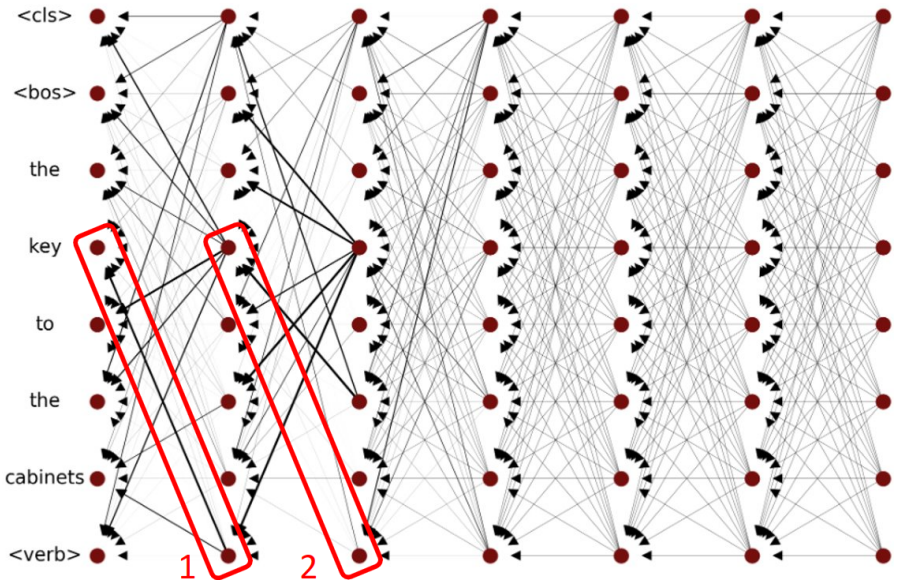

Figure 9. Attention weights from the encoder (modified from [39]).

Although attention modules consistently enhance model performance, their ability to provide transparent explanations is debatable, especially when a complex multilayer encoder is employed.

Attention Methods: From the above, it is evident that a straightforward analysis of attention weights is not a relevant method for explaining vision transformers (ViTs). The richest information is found in the last layer of the model, which is also the most mixed. To unravel this complexity, Abnar and Zuidema [39] proposed two methods for analyzing the flow of information throughout the entire model. They established a relationship between the last layer of the model and input tokens by introducing new disentangled attention weights A_{roll}. They utilized the attention weights of each layer and computed the information flow in the network. These methods are based on the same assumptions, using

the same raw attention weights A^l from each layer, but differ in how they calculate this flow characterized by the disentangled weights A_{roll}. Both methods rely on the following simplified assumptions: First, they compute the information flow based on the attention weights of a single head. For a network with multiple heads, these are aggregated by their average. Second, embeddings are linearly combined across layers based on attention weights. The attention weights correspond to the proportion of information contained in an incoming embedding $E^{(i_{\text{in}})}$ propagating to all other output embeddings $E^{(i_{\text{out}})}$ of a layer i. Third, to account for the residual connection in an encoder block, Abnar and Zuidema [39] modified the attention weights by adding an identity matrix I as follows:

$$A^l_{\text{res}} = \frac{1}{2}A^l + \frac{1}{2}I \tag{11}$$

The coefficients $\frac{1}{2}$ are used to balance the contribution of the attention and the residual connection. The weights A^l_{res} must be normalized after this transformation. These simplifying assumptions allow for the creation of an approximation of information propagation in the encoder layers. The first method is called attention rollout [39]. The disentangled weights A^l_{roll} are calculated by multiplying the A_{res} matrices from the last layer L to the first. To start, A^L_{roll} is initialized to be equal to the raw attention weights of the last layer L such that $A^L_{\text{roll}} = A^L$. Then, iteratively, the following is calculated:

$$A^{(L-i-1)}_{\text{roll}} = A^{(L-i-1)}_{\text{res}} A^{(L-i)}_{\text{roll}} \tag{12}$$

where $A^{(L-i-1)}_{\text{res}} = \frac{1}{2}A^{(L-i-1)} + \frac{1}{2}I$, and $i \in \{0, 1, \ldots, L-1\}$. Finally, the explanation R^c is the row of the matrix A^1_{roll} corresponding to the token [cls]. The second method is attention flow [39]. The unraveled attention weights A^l_{roll} are computed by solving a maximum flow problem, where the A^l_{res} weights are used as the capacity of each link. This attention flow method will not be considered in the following sections due to the significant computational resources required for its implementation.

Gradient Methods: The layerwise relevance propagation (LRP) method [40], is a technique for calculating the relative contribution of each neuron to a given point in a network concerning another neuron located at a different point. This method recursively breaks down the decision made by the network into contributions from the previous layers up to the input by using a conservation property. The conservation property ensures that what has been received by a neuron is entirely redistributed to the lower layer, regardless of the layer in question. The value of a neuron in the final layer (e.g., a class probability) thus results from the sum of contributions from all previous neurons. With z_{jk} representing the contribution of node j to the value of node k, the general rule of the conservation axiom is as follows:

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k \tag{13}$$

The partial LRP method from Voita et al. [41] suggests employing the LRP method to assess to what extent different attention heads contribute to the model's prediction instead of considering an average value of attention heads. However, the partial LRP method aims to assess the relevance of the heads only to visualize their importance and to prune the less relevant heads. It does not directly link the importance of attention heads to the importance of each token on these heads. Therefore, it is only an intermediate tool to provide a complete explanation between the prediction result and the input tokens.

Chefer [42,43] introduces two methods for ViT explanation, which we will respectively call Chefer 1 and Chefer 2 in the remainder of this paper. The Chefer 1 method [42] calculates specific explanations for a class c by incorporating the relevance of the attention heads as follows. Instead of averaging attention heads as proposed by Abnar and Zuidema [39] such as $E[A_h]$, the method calculates the gradients of attention matrices A^l for each head

with respect to the prediction of class c. Then, the method weights the attention matrices according to their respective gradients, thereby resulting in new attention matrices ∇A^l given by the following:

$$\nabla A^l = A^l \left(\frac{\partial S_c}{\partial A^l} \right) \tag{14}$$

∇A^l does not take into account many other components of a transformer encoder (attention weights multiplied by a matrix V, normalizations, linear projections, residual connections, etc.) that influence the model's prediction. To consider these components in the final result, on the one hand, a factor R^l is added to backpropagate the complete contribution of the attention matrices A^l to the final result following LRP propagation rules. On the other hand, similar to the attention method [39], an identity matrix is also added to account for residual connections. The reweighting of the attention weights for each layer is formulated as follows:

$$\overline{A}^l = \mathbb{E}[\nabla A^l \odot R^l] + I \tag{15}$$

To recover the relevance of the tokens for the final prediction, these new weights can be multiplied iteratively:

$$R^c[\text{cls}] = \overline{A}^1 \cdot \overline{A}^2 \cdot \ldots \cdot \overline{A}^L \tag{16}$$

where the notation $R^c[\text{cls}]$ refers to selecting the row of matrix R^c corresponding to token [cls].

The Chefer 2 method [43] provides a generic explanation method for all transformer architectures, even those with more than two modalities (e.g., handling both image and text in parallel). It considers residual connections through an identity matrix to calculate attention scores, as proposed in the attention rollout method, and uses gradients to obtain the relevance of each head with respect to an output class c. The LRP method used in Chefer 1 allows for considering more information about the transformer architecture, thereby yielding better results but requiring more computational resources. To propagate information through the encoder blocks, Chefer 2 redefines the propagation rules to no longer use the LRP method. To aggregate information from the attention heads, the same method as in Chefer 1 is used:

$$\nabla A^l = A^l \left(\frac{\partial S_c}{\partial A^l} \right) \tag{17}$$

To propagate the flow of information through the network, Chefer et al. [43] initialized $R^c = I$ before embeddings are mixed at the input of the network. Intuitively, the identity matrix means that each embedding is only contextualized by its own information. Then, R^c is updated by propagating information from input to output using update rules specific to each component of the encoder. For the MSA module, the update is as follows:

$$R^c = R^c + \nabla A^l R^c \tag{18}$$

The transition attention maps (TAMs) method [44], models the evolution of the representation of embeddings in the model as a Markov chain. In mathematics, a Markov chain is a modeling tool for random processes in which the probability of transitioning from one state to another depends only on the current state. A Markov chain is fully defined by its transition matrix, which, for each state, defines the probability of transitioning to another state. At each block, the representations of the output embeddings are considered as states of the Markov chain, with the state transition matrix being constructed based on the attention weights A^l. This allows them to build the flow of information between the model layers and connect the features of the final embeddings to input tokens. The initial state s_0 is defined as follows:

$$s_0[cls] = E[A_h^L] \in \mathbb{R}^{(1 \times s)} \tag{19}$$

where E is the average of the attention heads, L is the index of the last encoder block, and s is the number of tokens. Since the residual connection is not directly considered in the attention weights, an identity matrix is added to the transition matrix. Intuitively, this corresponds to a step where no transition is made, and the representation of tokens does not change.

$$s_i[cls] = \begin{cases} E[A_h^L] & \text{if } i = 0 \\ \frac{1}{2}s_{i-1} + \frac{1}{2}s_{i-1}E[A_h^{(L-i)}] & \text{else} \end{cases} \tag{20}$$

Here, $L - i \in \{l_{\text{end}}, \ldots, L\}$. The parameter l_{end} stops the propagation of the flow before reaching the early encoder blocks because Yuan et al. [44] consider the early encoder blocks as feature extractors at the local token level, thus not mixing information between tokens. A specific explanation is obtained by combining the states with integrated gradients obtained with respect to the last attention module. Integrated gradients are used to reduce noise and irrelevant features in the explanation. Multiplying these integrated gradients by the states of the Markov chain yields specific class explanations.

The method described by Chen et al. [45], which we will refer to here as the bidirectional transformer (BT) method, is based on the assumption that representing the contribution of a token to the model's prediction by a single scalar (like the gradient) is not complete and introduces noise into the explanation result. The BT method broadens the contribution of each token by examining two factors: attention perception P^L and reasoning feedback F^c. The attention perception represents the contribution of each input token to the final embedding of the [cls] token. It approximates the relationship between the input and output in attention blocks, similar to the attention rollout method, by following the rule below:

$$A^1_{\text{roll}} = (A^L + I) \ldots (A^1 + I) \tag{21}$$

However, this assumption does not consider the effect of the W_{MLP} projection on the interaction between embeddings. Chen et al. [45] showed that this information can be taken into account by redefining the attention weights as follows:

$$P^1 = (A^{\sim L} + I)W^L_{\text{MLP}} \ldots (A^{\sim 1} + I)W^1_{\text{MLP}} \tag{22}$$

where $A^{\sim l}$ can be defined in two ways, thereby using or not using averaged information from different heads. This is represented as follows:

$$\begin{aligned} \text{BT-Token:} \quad & (A^{\sim l})_{\text{token}} = \frac{\|Z^{(l-1)}W^l\|}{\|Z^{(l-1)}\|}A^l \\ \text{BT-Head:} \quad & (A^{\sim l})_{\text{head}} = \sum_{h=1}^{H} \frac{I_h^l}{\sum_h I_h^l} A_h^l \end{aligned} \tag{23}$$

The reasoning feedback represents how the $[cls]$ token is used for the prediction of a class c. It is calculated by backpropagating integrated gradients from the final decision for class c in proportion to the attention weights of the final embedding of the $[cls]$ token.

Perturbation Methods: The ViT-CX method from Xie et al. [46] adopts a different approach compared to previous methods. It no longer relies directly on attention weights A and gradients but on perturbation masks created from embeddings (similar to Score-CAM [35] using feature maps as masks for CNNs). The relevance of each mask is then crucial, which is calculated by evaluating the model with a masked image for explanation. The generation of masks M_{CX} is as follows: By convention, the input image of dimension $H \times W$ is divided into N tokens. Each token is then characterized by an embedding vector of size D. The masks are created from the final embeddings E^L. The embedding vectors are first rearranged into a 3D structure of dimensions $(\sqrt{N}, \sqrt{N}, D)$ where each embedding is positioned at the spatial coordinates of the token in the input image. Each frontal slice forms a feature map. The feature maps are then normalized to the interval

[0, 1] to obtain the masks. A set of masks M_{vit} is obtained: $M_{vit} = \{M_1, \ldots, M_D\}$, where $M_j \in \mathbb{R}^{H \times W}$ $(j = 1, \ldots, D)$. To reduce redundancy in the mask set and improve the explanation efficiency, similar masks are merged via the agglomerative clustering algorithm [47], which recursively merges masks with a minimal cosine distance between them. An implicit assumption in previous perturbation methods applied to CNNs is that only pixels not masked by a mask M_i contribute to the prediction. The score of these nonmasked pixels in a mask M_i is equal to $S^c = f(c|x \odot M_i)$, which is the prediction score for class c for the image x masked by M_i.

However, the masked pixels to which a zero value is assigned are not neutral. They provide graphical information to the model and contribute to the prediction score. The score assigned to nonmasked pixels is thus biased by the artifact of attributing a zero value to pixels under the mask. To correct this bias in the artifact, the perturbed image becomes $x \odot M_i + (1 - M_i) \cdot Z$, where Z follows a Gaussian distribution $N(0, \sigma)$. Adding noise to the complement $(1 - M_i)$ of the mask M_i mainly modifies the masked pixels and only slightly modifies the nonmasked pixels. However, even a slight modification of nonmasked pixels leads to a slight decrease in the prediction score. To correct this slight loss, the bias compensation term $f(c|x) - f(c|x + (1 - M_i) \cdot Z)$ is added to the prediction score, which is defined as follows:

$$[S^c(x, M_i) = f(c|x \odot M_i + (1 - M_i) \cdot Z) + f(c|x) - f(c|x + (1 - M_i) \cdot Z) \tag{24}$$

Finally, the attribution of a pixel in the explanation is the sum of the prediction scores of the corresponding masks $S^c(x, M_i)$. Therefore, the more a pixel is included in multiple masks, the higher its relevance value becomes. This phenomenon is the pixel coverage bias (PCB). This bias is corrected by dividing the final attribution of a pixel by its coverage frequency $\rho(x)$:

$$R^c(x) = \sum_{i=1}^{D} s^c(x, M_i) \cdot \frac{M_i(x)}{\rho(x)} \tag{25}$$

ViT-CX uses an average of fewer than 100 masks to explain an image of dimension (224, 224). The computational cost is significantly reduced compared to other presented perturbation methods (occlusion [30] with by default 196 local masks that mask pixels only once and RISE [31], which by default uses 4000 masks).

The transformer input sampling (TiS) method [48] stands out from previous explainability methods. It is based on the sampling of embeddings (token sampling) by applying masks before their introduction into a transformer. Once masked after the embedding phase, these tokens are no longer considered as input for self-attention, thereby avoiding the issue related to the choice of a replacement value encountered in perturbation-based explainability methods (RISE [31], ViT-CX [46]). As this is done just after the incorporation of the position and before any self-attention, the nonsampled tokens do not influence the output. Raghu et al. [49] showed that embeddings retain the location information of their tokens from the beginning to the end of the model thanks to residual connections. There is, therefore, no limitation to considering only a subset of tokens. This method exploits the prediction $w_{i,c}$ of the subset of tokens i associated with class c. It is worth noting that, unlike perturbation methods in input space (which modify pixel values like RISE, Score-CAM, or ViT-CX), TiS takes advantage of the transformer's ability to accept a variable-length sequence of embeddings to remove some tokens so that the model can perform calculations only on the remaining tokens. The advantage of the method is that it avoids generating aberrant images that can be produced by other perturbation methods. Indeed, Hooker et al. [50] argue that perturbation-based methods modify important parts of the original image, thereby violating the assumption that training and evaluation data come from the same distribution. The first step for TiS is to generate masks M_i to control the sampling of a token sequence $T \in \mathbb{R}^{(T \times D)}$ consisting of T tokens (except for the $[cls]$ token) of dimension D. To achieve this, a concatenation of the embeddings of each encoder layer

is performed, thus resulting in a matrix $A \in \mathbb{R}^{(T \times L \times D)}$, where L is the number of layers. K-means clustering is applied to the columns of A to reduce the number of masks, thereby producing a smaller matrix $K \in \mathbb{R}^{(T \times N_m)}$, where N_m is the number of masks. The number of centroids of K-means N_m is a parameter of the method.

$$K = \text{KMeans}(A, N_m) \tag{26}$$

The masks are then binarized into a matrix $M \in \{0,1\}^{(N_t \times N_m)}$, where each component of a mask indicates whether the targeted token is retained or not. Let N_k represent the number of tokens to keep, which is defined as follows:

$$M_{ij} = \begin{cases} 1 & \text{if } K_{ij} \in \text{topk}(K_{.j}, N_k) \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

The weight $w_{j,c}$ represents the prediction score for class c obtained for mask M_j. The explanation is derived as the weighted sum of these masks. This result is divided by the sum of the masks to account for a potential token frequency bias, which is similar to the pixel coverage bias addressed in the ViT-CX method [46]:

$$R^c = \frac{\sum_{j=1}^{N_m} w_{j,c} M_{.j}}{\sum_{j=1}^{N_m} M_{.j}} \tag{28}$$

Subsequently, the resulting saliency maps are rescaled using bilinear interpolation to match the resolution of the input image.

2.4. Taxonomy Table

Table 1 presents the main explainability methods that are examined in detail in the previous Sections 2.3.1 and 2.3.2. The methods are numerous and diverse, and sorting is necessary for various reasons. Firstly, in the rapidly evolving field of explainability research, some reference articles cited in this report were published just a few months ago. Certain implementations of these methods have not yet been published at the time of the experiments, most notably the Vit-CX methods. Therefore, they were not usable in experimentation. Secondly, some methods are only intermediate tools in the construction of others: the Partial LRP method is a component of the Chefer 1 method; the CAM method forms the basis of Grad CAM, Grad CAM++, and score CAM methods; and the occlusion method is the source of the rise method. In this study, Partial LRP and CAM are therefore not used as explainability methods. Lastly, the attention flow method was not used in experimentation due to its high computational resource cost. Only the remaining methods, i.e., those not marked with an asterisk in Table 1, were considered for experimentation.

The multitude of available methods presents a challenge in decision making. This work addresses this issue at its core and offers researchers in the field through a scientific review of explainability methods, along with the underlying evaluation criteria and associated metrics. The aim is to promote the adoption of best practices in future research, with a clear awareness of the limitations of the available tools. This comparative study is divided into two sections: Section 3 outlines the experimental framework of the study, including the objective evaluation criteria, the analyzed dataset, the XAI frameworks, and the protocol used to evaluate the methods, which are explained and justified. Section 4 presents the experimental results obtained and discusses their applicability.

Table 1. Overview of XAI methods analyzed in the state of the art. Methods not used in the experimentation are marked with an * in the table.

Methods	Publication	Section	Taxonomy	Model
Input Grad [27]	November 2016	Section 2.3.1	Architecture Exploitation	Differentiable
Smooth Grad [28]	June 2017	Section 2.3.1		
Integrated Grad [29]	July 2017	Section 2.3.1		
Occlusion * [30]	November 2013	Section 2.3.1	Perturbation	Black Box
RISE [31]	June 2018	Section 2.3.1		
CAM * [32]	June 2016	Section 2.3.1	Architecture Exploitation	Activation Maps
Grad CAM [33]	October 2017	Section 2.3.1		
Grad CAM++ [34]	March 2018	Section 2.3.1		
Score CAM [35]	June 2020	Section 2.3.1		
Attention Rollout [39]	May 2020	Section 2.3.2	Attention	Transformers
Attention Flow * [39]	May 2020	Section 2.3.2		
Partial LRP * [41]	May 2019	Section 2.3.2	Attention and Architecture Exploitation	
Chefer 1 [42]	March 2021	Section 2.3.2		
Chefer 2 [43]	April 2021	Section 2.3.2		
TAM [44]	August 2021	Section 2.3.2		
BT [45]	February 2023	Section 2.3.2		
ViT-CX * [46]	February 2023	Section 2.3.2	Perturbation	
TIS [48]	October 2023	Section 2.3.2		

3. Experimental Setup

For the continuation of this work, the objective is to apply the various XAI methods to the ViT model on the experimental dataset as defined in Section 3.2, thereby following the protocol defined in Section 3.3 and critically examining the results in Section 4. This experimentation aims to evaluate the methods based on various criteria detailed in Section 3.1.

Evaluation involves considering the criteria to apply. It is necessary to define precise and consistent criteria for evaluating the quality of an explanation of a ViT model. Ideally, there should be reference explanations (ground truth) against which the obtained explanation can be compared. Establishing such references is challenging given the nature of the task. Unlike a classification task, where reference results exist, the explanation of the functioning of a model is not known a priori. In classification, the type of object to be classified is specified a priori, thus allowing for supervised and semisupervised model training. In the case of explanations, the aim is to highlight the network's operation, which can only be known as a posteriori. Consequently, there is an intuitive temptation to turn to methods evaluating the quality of explanations based on their intelligibility to the target audience. However, introducing this human component into the evaluation introduces strong subjectivity interference, as not all audiences have equal comprehension capabilities, whereas the intrinsic quality of the explanation is what needs evaluation. Furthermore, imposing human evaluation of the explanation generates significant costs, especially in fields such as healthcare or Industry 4.0, thereby underscoring the value of automated evaluation methods.

In addition to the objectivity criterion of an explanation, Samek et al. [51] highlighted that, first, the quality of a heat map, reflecting the explanation model's quality, depends not only on the explainability algorithm used but also on the model's performance. The model's effectiveness depends largely on the architecture used, as well as the quantity and quality of available training data. An uncertain model produces uncertain explanations, thereby emphasizing the importance of having a well-trained model to generate quality explanations. Second, there is no guarantee that human explanations and those of the model perfectly coincide. An explanation always represents the model's viewpoint and is not necessarily correlated with human intuition or focused on the specific object of interest. Third, a heat

map should not be considered as a segmentation mask. Other information than the object of interest, such as its context, can be extremely important for the model's decision. The features of the object of interest can be highly discriminative, meaning that evidence for a specific class does not necessarily need to be localized over the entire object. For example, visualizing a dog's head allows one to conclude the "dog" class as a whole. Conducting a quantitative evaluation of XAI methods for comparison requires defining an experimental framework and rigorous criteria. This is the focus of the following sections. Since explanations depend on the dataset, the model used, and their implementation, these topics are addressed hereafter.

3.1. XAI Metrics

The literature offers a multitude of criteria for evaluating and measuring the quality of explanations. It also shows that these criteria can vary considerably depending on the pursued objective and the targeted user groups. Defining a "good" explanation indeed depends on the user, the type of model and data, the context of use, and the desired form of explanation. However, it is advisable to prioritize objective criteria that are less dependent on human subjectivity and, to do so, to use measurement tools specific to the explanation itself. The most commonly referenced evaluation criteria in the literature are faithfulness, complexity, randomization, robustness, and localization. Metrics based on these criteria allow for the evaluation and comparison of XAI methods. Before delving into the details of these metrics, here is a reminder of the concepts used in this work:

- An AI model is used to make predictions on classes of objects of interest.
- XAI methods generate explanations about the functioning of an AI model.
- Criteria reflect different characteristics of the explanation of a prediction on a given image.
- Metrics quantify the criteria.

In other words, metrics are applied to an explanation of a prediction of a given class using a model, which is in turn related to an image. The manner of using metric criteria to evaluate XAI methods is addressed in the following section.

3.1.1. Faithfulness Criteria

The faithfulness criterion evaluates the extent to which explanations follow the actual predictive behavior of the model, thereby ensuring that the pixels highlighted in the explanation are also crucial in the model's prediction. Metrics for the faithfulness criterion are based on the following method: perturb the original image at the highlighted pixels in the explanation and study the consequences of this perturbation on the prediction result. The various metrics (presented below) vary in terms of pixel selection order, the perturbation method used, and the measurement of the influence of the perturbation.

Faithfulness Correlation: The faithfulness correlation metric [52] is defined as follows: a set of T random pixels is replaced with a base value (either black, white, or random). By repeating this process N times, the metric score can be calculated as the Pearson correlation coefficient between the following:

- The difference in prediction for a class c between the original input and the perturbed input.
- The sum of attributions for the T pixels from the selected subset.

Intuitively, this metric can be understood as follows: By perturbing the T pixels that are crucial for the prediction, the difference in prediction between the original and perturbed images will be substantial. By definition, if the explanation highlights these same pixels, the attribution of these pixels is significant. Therefore, if the correlation between these two results is strong, the method is faithful and vice versa. This reasoning holds even when considering that the T pixels are not important for the prediction but are for the explanation (and vice versa), thereby causing the correlation to decrease. It remains equally valid if the T pixels are not important for either the prediction or the explanation, as the correlation becomes strong again. This correlation measure is thus a good indicator of the explanation's faithfulness to the prediction.

Faithfulness Estimate: The faithfulness estimate metric [53] operates similarly, except for the choice of pixels to perturb. Instead of randomly selecting pixels, they are chosen by attribution in descending order. The score for this metric is established exactly as for the faithfulness correlation. Intuitively, the advantage of this method is that if the first perturbed pixels significantly lower the score for class c, one can conclude that the explanation is faithful and vice versa. Indeed, by construction, the first perturbed pixels are those with the highest attribution, meaning they are highly important for the explanation. By perturbing them, the prediction model's reaction directly reflects the faithfulness of the method: if the prediction is strongly influenced by the initial perturbations (resulting in a high correlation), the method is faithful and vice versa.

Monotonicity Correlation: The monotonicity correlation metric [54] operates similarly to the faithfulness estimate but differs in the calculation of the prediction difference. By selecting pixels based on their absolute attribution, the prediction difference is inferred over multiple iterations. Each set of pixels is randomly perturbed several times. The implementation of this metric in the Quantus framework [55] calculates the prediction difference as the root mean square of the prediction difference for class c between the original input and the perturbed input. This choice is made because Nguyen and Martinez [54] did not specify which differentiation function to use. All three metrics presented above return a value in the range $[-1, 1]$, where higher scores indicate better fidelity.

Monotonicity: The monotonicity metric [56] is substantially different. It involves starting with a base image (white, black, or random) and gradually adding each pixel according to their attribution in ascending order. At each addition, the method evaluates the prediction score for class c. If the method is faithful, then as more pixels are added, the model predictions should improve. Furthermore, since they are added in increasing order of attribution, the prediction growth should be monotonic. This metric assesses fidelity through a simple positive or negative judgment: it is positive if the prediction growth is strictly monotonic, and it is negative otherwise. To obtain a more nuanced numerical result from this metric, Stassin et al. [57] slightly modified its operation by calculating how many times adding a pixel led to an improvement in predictions compared to the number of times this addition resulted in a decrease in performance. This percentage numerically represents the result of this metric. If the initial judgment was positive, then the proportion would be 100%. However, a negative judgment according to the basic metric would be better nuanced by the knowledge of this new metric.

Pixel Flipping: The pixel flipping metric [40] is defined as follows: the principle is to sort the pixels of the image in descending order of attribution. The image is also perturbed by a set of T pixels. Predictions for class c are recorded throughout the iterations. Finally, the metric result corresponds to the area under the curve (AUC) of these predictions. The higher the AUC, the more faithful the explanation. Intuitively, for a faithful explanation, perturbing the more important pixels (in the early iterations) causes low scores of S_c for class c, while, conversely, perturbing less important pixels (in the later iterations) has less and less of an influence on the score of class c (the curve takes on an asymptotic shape). For an unfaithful or random method, perturbing important pixels in the explanation does not significantly alter the result for class c over all iterations.

Selectivity: The selectivity metric [58] differs from other metrics in that it considers not just a single pixel at a time but a patch of pixels (e.g., size 4×4). The patches are sorted in descending order of attribution. Then, iteratively, the image is increasingly perturbed so that the previous perturbation is retained in the next iteration. The method measures the prediction score for class c at each iteration and summarizes this information by calculating the area under the curve (AUC). The explanation is faithful if its AUC is low.

3.1.2. Complexity Criteria

The notion of complexity is variously referenced in the literature. For instance, Bellucci et al. [59] define complexity as the measure of interpretability in an explanation, i.e., the measure of how easily a user can simulate and/or understand it. While the

underlying idea is plausible, this definition directly relates to the user's abilities, which are highly variable and challenging to objectively measure. In the context of this work, the adopted definition is that complexity is a concept reflecting the level of conciseness in an explanation, i.e., its ability to be most significant with the fewest possible pixels [55]. The literature offers various user-independent metrics to measure this conciseness, and thus its inverse: complexity.

Complexity: The metric complexity [52] is defined as the Shannon entropy of the distribution of attributions a_i for the i pixels in the explanation:

$$\text{entropy} = -\sum_i (a_i \cdot \log(a_i)) \tag{29}$$

The higher the Shannon entropy, the more complex the explanation and vice versa. Intuitively, Figure 10 shows that this favors explanations where attributions are either very low (close to 0, dark blue region) or very high (close to 1, red region). Conversely, explanations containing average attributions (close to 0.5 in yellow) are heavily penalized and considered complex.

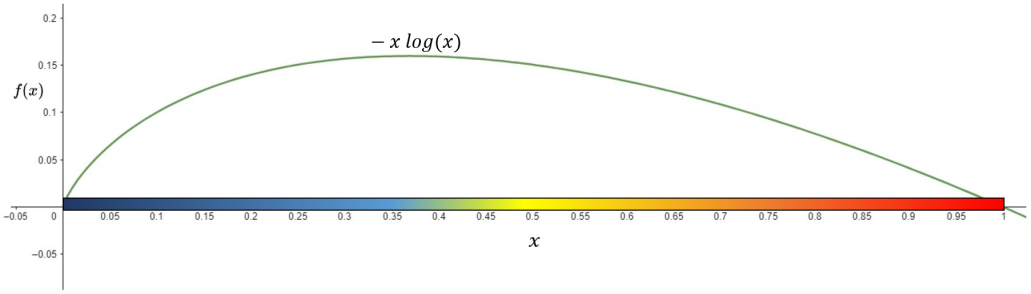

Figure 10. Interpretation of Shannon entropy.

Effective Complexity: The metric effective complexity [54] is defined as the percentage of pixels with attributions exceeding a certain threshold ε. If a pixel i has an attribution a_i greater than the threshold ε, this pixel i is assumed to be important for the prediction. The higher this percentage, the more complex the explanation.

Sparseness: The metric sparseness [60] is defined as the Gini index applied to the attributions a_i of the pixels:

$$\text{Gini} = \frac{\sum_{i=1}^{n}(2i - n - 1) \cdot a_i}{n \sum_{i=1}^{n} a_i} \tag{30}$$

The Gini index is a statistical measure that reflects the distribution of a variable within a population. This coefficient is typically used to measure income inequality in a country. It is a number in the range $[0, 1]$, where 0 represents a perfectly equal distribution of income, and theoretical 1 represents a completely unequal distribution, where one person has all the income. In its application to the complexity metric, the attribution of pixels is likened to the income of the population. The complexity is higher as it tends toward 1.

3.1.3. Randomization Criteria

The randomization criterion (also called the coherence criteria or sanity check) evaluates the extent to which the explanation evolves when its environment (e.g., model parameters) varies randomly. Intuitively, a good explainability method will yield results that depend on model parameters, and, thus, the outcomes will vary based on the chosen criteria for the randomization metrics.

Model Parameter Randomization: The model parameter randomization metric [61] measures the correlation between the original explanation and a new explanation calculated with a model of the same architecture but not trained, and with weights initialized randomly.

If the explanations depend on the learned parameters of the model, they differ significantly between the two situations (indicated by a low correlation). If they are similar (indicated by a high correlation), it means that the explanations are insensitive to the model's properties.

Random Logit: The random logit metric [62] calculates the correlation between the original explanation for a class c and another explanation for a randomly chosen class. For example, based on an image of a dog, if the explanation is similar to that of the boat class, then the XAI method is considered inconsistent. For both of these randomization metrics, if the correlation is low, the explanation is coherent and vice versa.

3.1.4. Robustness Criteria

The robustness criterion assesses the extent to which the explanation remains stable when the input image is slightly perturbed, thereby ensuring that the model's classification is not altered.

Local Lipschitz Estimate: The local Lipschitz estimate metric [63] measures the largest Lipschitz distance between the original image and N neighbors perturbed by Gaussian noise, with $a(x)$ representing the explanation for input x and x' representing a perturbed input belonging to the set N:

$$\max\left(\frac{\|a(x) - a(x')\|_2}{\|x - x'\|_2}\right) \tag{31}$$

Average Sensitivity and Maximum Sensitivity: The *average sensitivity* metric and the *maximum sensitivity* metric [64] measure the mean and maximum Frobenius distances, respectively, between the explanation of the original image and the explanations of N perturbed neighbors:

$$\text{mean or } \max\left(\|a(x) - a(x')\|_{\text{fro}}\right) \quad \text{where} \quad x' \in N \tag{32}$$

For both robustness metrics, a lower score indicates a more robust explanation.

3.1.5. Inference Time

Although not referenced as an evaluation criterion for explainability methods, it seems intuitively useful to take into account the inference time when comparing various XAI methods, especially if this comparison is intended for large-scale applications

3.1.6. Localization Metrics

The localization criterion assesses whether explanations are centered around a region of interest (ROI), which can be defined around an object by a bounding box, a segmentation mask, or a cell within a grid [55]. This criterion is used to evaluate the ability of an explanation to capture the object of interest.

The pointing game metric [65] compares the explanation to a human annotation of the relevant area of the image, thereby quantifying how similar the given explanation is to that of a human. Specifically, if the pixel with the most relevance is within the annotated bounding box, the automatic explanation earns a point. The result of this measurement on a set of images is the precision of the method defined as $accuracy = \frac{hits}{hits+misses}$. The higher the localization precision score, the more the explanation behaves comparably to human annotation.

In this work, the localization criterion was not considered for the comparison of XAI methods for two reasons: Firstly, this criterion requires human annotation of the database to locate objects in the images. This need is very costly and limits the practical use of the criterion. Secondly, this criterion deviates from the objective of explaining the model's functioning. It only reveals an analogy with human visual analysis, but, as explained earlier, the quality of an explanation should not be measured by correspondence with human judgment, which is always subject to subjectivity.

3.2. Dataset and Model

The choice of the image dataset for processing was based on the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC2012) dataset [21], which is known for its characteristics. It contains approximately 1.2 million images distributed across one thousand different categories. Each category is represented by a varying number of images, and the images are well distributed. The categories cover a wide range of objects, animals, scenes, and concepts, ranging from pets to vehicles, landscapes, household objects, and more. This diversity allows for testing recognition capabilities on a wide variety of objects and scenes. The images have varying resolutions, thereby ranging from a few hundred pixels to high-resolution images. The proven and recognized quality of this dataset allows for the assumption that the data used is not biased and will not introduce disturbances in the measurement results of the explanation criteria. However, the original reference image dataset is too vast to be used in its entirety for experimentation. Wang et al. [35] proposed a random selection of 2000 images from this dataset in their evaluation of the Score-CAM method, followed by Stassin et al. [57] for evaluating XAI methods for CNNs. To allow for the possibility of drawing analogies between results, the experimentation will be conducted on the same set of 2000 images.

The choice of the classification model was the *ViT_base_patch16_224* architecture [5]. It was pretrained on the ImageNet dataset and achieved a top-one accuracy score of 81% and a top-five accuracy score of 95%. In other words, the model can predict the correct class in 81% of cases among the thousand classes and in 95% of cases, the correct class is among the top-five classes with the highest probability scores. This level of accuracy is high enough not to significantly degrade the results of the explanation methods. Although very high, the prediction accuracy is not perfect. An analysis of this situation is discussed in Section 4.5 to verify the robustness of the analyses despite the model's imperfections in predictions. Given these factors, the risk is minimal that the evaluation and comparison of XAI methods are influenced by factors other than the XAI method itself.

3.3. Protocol

The experimentation followed the protocol described below to evaluate XAI methods based on the criteria presented in Section 3.1. Each metric was calculated for 2000 randomly selected images from the dataset for each XAI method mentioned in Table 1. The fundamental idea is to establish the final score for a method on a metric by averaging the scores obtained for the 2000 images. Then, it was verified whether the set of measures for a given criterion (across different metrics) was sufficiently discriminative to rank the methods. This protocol is similar to the studies conducted by Petsiuk et al. [31] and Xie et al. [46]. In addition to these results, the inference times per image for both the methods and metrics were measured to provide an approximation of their execution speeds, which could be an ultimate criterion for selection or ranking when two metrics yield similar results. Here are the sequential steps of the experimentation protocol:

- Generate explanations for the 2000 images using each XAI method listed in Table 1, thereby focusing on explaining the annotated class of the image.
- During this explanation generation, measure the inference times of the XAI methods (Section 4.6).
- Perform a visual evaluation of the results (Section 4.1).
- Compare these explanations to the metrics for each criterion (Section 4.2).
- Aggregate these measurements according to criterion and discuss the analysis by studying the correlation of results from different metrics within a criterion (Section 4.3).

All experiments in this study were conducted using the same hardware. The GPU used was a Nvidia GeForce RTX 3070 (8GB RAM). Note that as a verification step, each metric was applied to a random explanation, meaning that the assignment of each pixel was set randomly following a uniform distribution in the $[0, 1]$ interval. This verification checks the metric's proper behavior. If a random explanation obtains a better score than an explanation

obtained through an XAI method, the reason should be investigated, which could be an issue with the metric's implementation. Furthermore, XAI methods and evaluation metrics are typically defined with default parameters. These default parameters were used as-is in this work, since they are commonly accepted in the literature and/or proposed by their authors. The results are highly influenced by these parameters, as noted by Stassin et al. [57] regarding the replacement value in perturbation methods and faithfulness metrics.

3.4. Framework

A high-level framework has been developed to integrate diverse XAI methods from various frameworks into a unified tool. This tool is a Python application. The entire project was implemented using PyTorch, which is an open-source framework for tensor computation dedicated to machine learning developed by Facebook AI Research. The use of PyTorch is particularly relevant for the development of model-specific explainable artificial intelligence (XAI) methods. Many XAI techniques require extracting information from various layers of the neural network. In this context, PyTorch makes it easy to access this information at different network levels, thus streamlining the development and implementation process of XAI methods. This high-level development aims to apply consistent experimentation conditions to the available methods and collect their results within a unified environment. Consequently, the user selects a dataset, a specific ViT model, and an XAI method; the user specifies whether to use the CPU or GPU and, in the latter case, selects the batch size for GPU image processing. The framework then generates the corresponding explanations and stores them for future use. This framework represents the primary contribution of this work, building upon Stassin et al.'s research [57] and being tailored to the specific goal of enhancing ViT explainability. The implementation of this high-level framework is publicly available on GitHub at the following address: https://github.com/ValentinCord/TFE_XAI_ViT, accessed on 13 November 2023.

4. Results

This Section presents the results and analyses of the experimentation following the protocol outlined in Section 3.3.

4.1. Visual Results

Figure 11 illustrates that, concerning the XAI methods not specific to vision transformers (ViTs), the RISE perturbation method yields impressive visual results, in contrast to the input Grad and integrated Grad methods, which produce diffuse outcomes across the entire image. Englebert et al. [48] had previously noted the limitations of the integrated Grad method in their paper related to the explainability method TiS for ViTs. It eliminates many global pixels while keeping the objects of interest visible to the human eye compared to their method. The Grad CAM and Grad CAM++ methods also appear to generate somewhat diffuse results but to a lesser extent. Ultimately, the Score CAM method seems best suited for ViT applications, with the RISE method applicable to models of any architecture.

In Figure 12, the visual results of the methods designed for ViTs are shown. The explanations are satisfactory. The rollout method, being the most basic, exhibited more scattered attributions within the image, which likely does not accurately reflect the model's prediction. The BT, TAM, and TiS methods, on the other hand, were typically the most effective in distinguishing all objects of interest.

As a result, a visual analysis enables individuals to define the XAI methods that may not be suitable for the task, as their results are not coherent with the human perception of a good explanation. Conversely, a subset of methods with explanations aligned with human judgment can be identified visually. Nevertheless, distinguishing and defining the best XAI method based on visual criteria remains challenging and subjective.

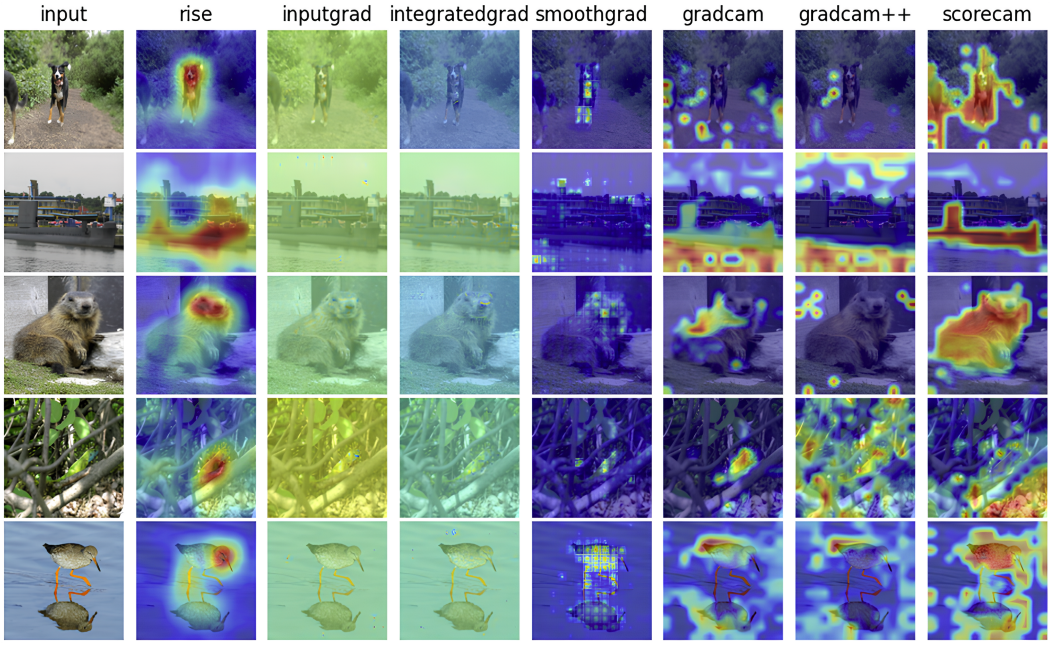

Figure 11. Explanations of five input images using XAI methods that are not specific to ViTs.

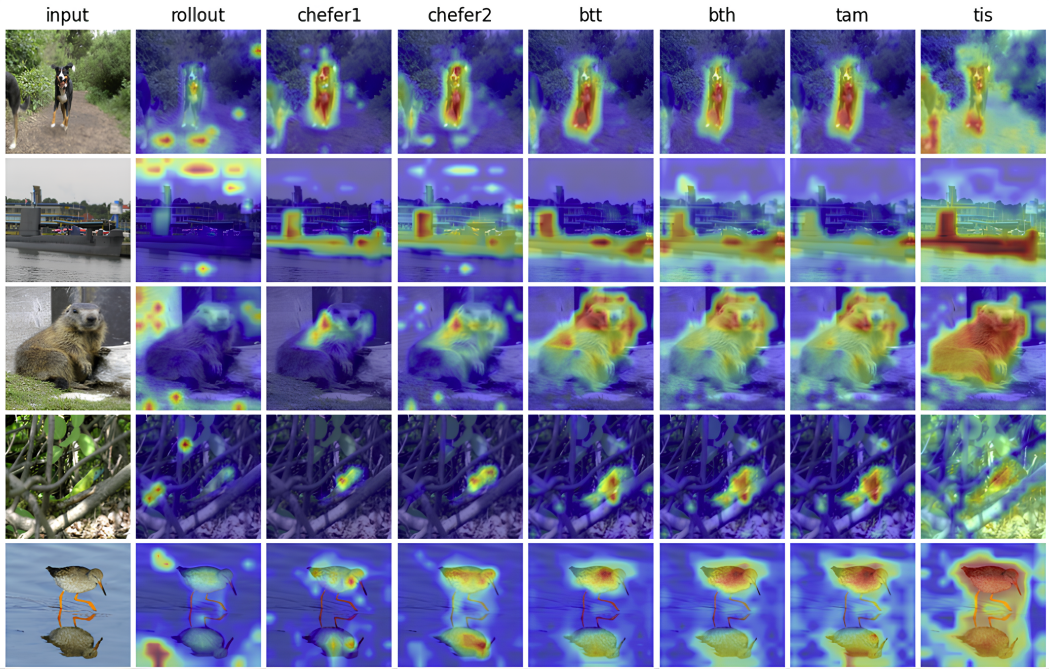

Figure 12. Explanations of five input images using XAI methods that are specific to ViTs.

4.2. Evaluation Metrics for XAI Methods

4.2.1. Robustness Metric Analysis

The robustness criterion assesses the extent to which the explanation remains stable when the input image is slightly perturbed, thereby ensuring that the model's classification is not altered (refer to Section 3.1.4). For a method to be considered robust, it must minimize the local Lipschitz estimate, maximum sensitivity, and average sensitivity metrics shown in Figure 13.

XAI	Robustness					
	Local lipschitz estimate ↓		Max-sensitivity ↓		Avg-sensitivity ↓	
	Mean	Ranking	Mean	Ranking	Mean	Ranking
Random	0.989	15	0.722	15	0.722	15
Rise	0.009	9	0.199	8	0.199	9
Grad CAM	0.006	1	0.199	8	0.173	8
Score CAM	0.006	1	0.293	12	0.274	13
Grad CAM ++	0.006	1	0.261	11	0.244	11
Input Grad	0.008	5	0.021	1	0.021	1
Integrated Grad	0.010	11	0.296	13	0.272	12
Smooth Grad	0.011	13	0.108	4	0.108	4
Rollout	0.010	11	0.315	14	0.315	14
Chefer 1	0.013	14	0.075	2	0.064	2
Chefer 2	0.007	4	0.076	3	0.076	3
BT H	0.008	5	0.149	5	0.147	5
BT T	0.008	5	0.166	7	0.158	7
TAM	0.008	5	0.155	6	0.152	6
TIS	0.009	9	0.206	10	0.201	10

Figure 13. Average scores for robustness metrics according to XAI method.

The similarity between maximum sensitivity and average sensitivity was significant across all methods, in contrast to the local Lipschitz estimate, which displayed a markedly different ranking. Therefore, there was convergence for only two of the three robustness metrics. Upon closer examination, the Chefer 1 method, for example, was ranked as the second-least robust according to the local Lipschitz estimate, even though it held the second-best position in the other metrics. In contrast, the CAM methods performed best in terms of the local Lipschitz estimate but were the least competitive in the rankings of the other two metrics. The input Grad method generally maintained a strong ranking, despite its visually diffuse results, because its explanations remained consistent across multiple tests. As anticipated, the random method consistently ranked as the least robust, as it based its explanations solely on a random foundation, which varied with each new test.

4.2.2. Complexity Metric Analysis

Complexity reflects the level of conciseness in an explanation (see Section 3.1.2). The sparseness metric should be maximized to indicate low complexity, and, conversely, the complexity and effective complexity metrics should be minimized to represent low complexity.

The complexity metrics did not exhibit convergenc, as are shown in Figure 14. The input Grad and integrated Grad methods were typically ranked lower because their diffuse explanations ran counter to the definition of conciseness. Smooth Grad ranked first in the effective complexity metric with an average score of zero, given that the ϵ threshold was higher than all the attributions calculated by the method. The random method was ranked first in terms of the complexity metric. This did not necessarily challenge the validity of the complexity metrics, as they solely assessed the conciseness of the explanation. The level of conciseness can vary significantly for the random method due to its random construction. For this particular criterion, comparing the random method to the other methods lacks genuine significance.

XAI	*Complexity*					
	Sparseness ↑		Complexity ↓		Effective complexity ↓	
	Mean	Ranking	Mean	Ranking	Mean	Ranking
Random	0.492	6	0.510	1	0.900	5
Rise	0.128	15	0.965	15	0.995	12
Grad CAM	0.586	2	0.806	8	0.693	3
Score CAM	0.565	4	0.847	13	0.769	4
Grad CAM ++	0.707	1	0.687	3	0.492	2
Input Grad	0.325	13	0.810	9	0.958	6
Integrated Grad	0.278	14	0.855	14	0.962	7
Smooth Grad	0.326	12	0.833	12	0.000	1
Rollout	0.476	7	0.688	4	0.998	14
Chefer 1	0.582	3	0.770	5	1.000	15
Chefer 2	0.446	8	0.789	6	0.997	13
BT H	0.375	10	0.816	10	0.992	9
BT T	0.394	9	0.803	7	0.993	11
TAM	0.352	11	0.826	11	0.992	9
TIS	0.548	5	0.626	2	0.979	8

Figure 14. Average scores for complexity metrics according to XAI method.

4.2.3. Randomization Metric Analysis

For these two metrics, the randomization criterion assesses the extent to which the explanation varies when the model's weights are randomized and when the target class changes (see Section 3.1.3). Randomization metrics must be minimized.

The rankings resulting from the two randomization metrics did not exhibit convergence, as are shown in Figure 15. Upon analysis, this can be explained by these two metrics focusing on different aspects: model weight modification for the first and changing the target class for the second. Therefore, it is not surprising to observe differences in the rankings of methods. The model parameter randomization metric ranked the random method in the first position, and the random logit metric ranked it in the second position. However, random was utilized here as a simple test method to validate the implementation of the metrics (see Section 3.3). In this case, it is logical to see the random method ranked among the best methods, since it will always produce a new random result, which, by definition, is expected when applying a randomization metric. However, some results remained surprising, particularly concerning the Grad CAM and integrated Grad methods. Grad CAM displayed a perfectly incoherent score in the random logit metric, thereby implying that the explanation for an image remained the same for the two different classes. The integrated Grad method ranked low in both metrics, which confirms previous observations regarding this method (see Section 4.1).

4.2.4. Faithfulness Metric Analysis

The faithfulness criterion assesses the extent to which explanations align with the actual predictive behavior of the model (see Section 3.1.1). To reflect high faithfulness, all metrics quantifying this criterion should be maximized except for the selectivity metric, which should be minimized as shown in Figure 16.

XAI	*Randomization*			
	Model parameter randomisation ↓		Random logit ↓	
	Mean	Ranking	Mean	Ranking
Random	0.067	1	0.501	2
Rise	0.533	13	0.496	1
Grad CAM	0.490	6	1.000	15
Score CAM	0.494	7	0.598	11
Grad CAM ++	0.511	11	0.588	9
Input Grad	0.549	14	0.672	13
Integrated Grad	0.567	15	0.701	14
Smooth Grad	0.431	2	0.584	7
Rollout	0.459	4	0.669	12
Chefer 1	0.457	3	0.583	6
Chefer 2	0.516	12	0.519	3
BT H	0.495	8	0.586	8
BT T	0.507	10	0.580	5
TAM	0.499	9	0.595	10
TIS	0.476	5	0.539	4

Figure 15. Average scores for randomization metrics according to XAI method.

XAI	*Faithfulness*											
	Faithfulness Correlation ↑		Faithfulness Estimate ↑		Monotonicity ↑		Monotonicity correlation ↑		Pixel flipping ↑		Selectivity ↓	
	Mean	Ranking	Mean	Ranking	Mean	Ranking	Mean	Ranking	Mean	Ranking	Mean	Ranking
Random	0.237	15	0.246	14	0.420	15	0.502	14	0.060	15	0.892	15
Rise	0.306	8	0.422	5	0.599	1	0.575	8	0.137	8	0.215	3
Grad CAM	0.304	10	0.377	10	0.445	11	0.576	7	0.171	3	0.195	1
Score CAM	0.321	1	0.379	9	0.442	13	0.526	13	0.170	4	0.221	6
Grad CAM ++	0.292	14	0.255	13	0.469	9	0.493	15	0.272	1	0.228	10
Input Grad	0.303	11	0.341	11	0.513	4	0.640	1	0.109	12	0.231	12
Integrated Grad	0.310	4	0.192	15	0.447	10	0.583	4	0.103	14	0.219	5
Smooth Grad	0.312	3	0.435	4	0.584	2	0.585	3	0.122	10	0.221	6
Rollout	0.306	8	0.314	12	0.486	8	0.542	11	0.228	2	0.333	14
Chefer 1	0.316	2	0.400	8	0.496	7	0.582	5	0.139	7	0.208	2
Chefer 2	0.309	6	0.444	2	0.549	3	0.594	2	0.119	11	0.217	4
BT H	0.297	13	0.422	5	0.511	5	0.541	12	0.154	6	0.229	11
BT T	0.308	7	0.441	3	0.424	14	0.574	9	0.131	9	0.221	6
TAM	0.299	12	0.412	7	0.444	12	0.546	10	0.155	5	0.239	13
TIS	0.310	4	0.471	1	0.498	6	0.581	6	0.109	12	0.222	9

Figure 16. Average scores for faithfulness metrics according to XAI method.

Surprisingly, the rankings of the methods based on faithfulness metrics did not exhibit convergence. The random method was consistently considered to be of low fidelity in all metrics, which is intuitively understandable given the definition of the faithfulness criterion and the random construction of explanations by random.

4.2.5. Mean Approach Discussion

The main lesson of the averaging approach is that it does not lead to convergent rankings of metric results. The only exceptions to this general observation are the maximum sensitivity and average sensitivity for the robustness criterion, which is two out of the fourteen metrics studied. In conclusion, method rankings by the mean are ultimately possible only on a per-metric basis and not for the entire criterion. Within the same criterion, the different metrics present numerical results that are not directly comparable, since they originate from different mathematical methods. Given the overall divergence in the results, it is therefore unfounded to attempt to quantify the strength of a criterion by averaging its metrics. For example, it would not be accurate to consider selectivity as a better fidelity measurement tool simply because it yields higher numerical results than other metrics within this criterion. This is further confirmed by examining the range of average results, which varied significantly from one metric to another. For instance, the average values of the faithfulness correlation all fell within a range of 0.029, while the averages of faithfulness estimate fell within a range of 0.279 (excluding random, as shown in Figure 16), even though they were measuring faithfulness for the same explanations.

The lack of metric convergence was mathematically validated, as shown in Figure 17. The latter displays Kendall's τ_b correlation [14] between method rankings across various metrics. The Kendall correlation coefficient is a nonparametric measure used to assess the dependence between two ordinal variables (here, rankings). Unlike the Pearson correlation coefficient, which evaluates the linear correlation between two quantitative variables, the Kendall coefficient is suitable for ordinal variables. This statistical tool is the most appropriate for examining rankings.

	Robustness			Complexity			Randomization		Faithfulness					
	LLE	Max	Avg	Spars	Complex	Ecomplex	MPR	RL	FaithC	FaithE	Mono	MonoCor	PF	Select
LLE	1.00													
Max	0.09	1.00												
Avg	0.08	0.98	1.00											
Spars				1.00										
Complex				0.54	1.00									
Ecomplex				0.14	0.03	1.00								
MPR							1.00							
RL							-0.12	1.00						
FaithC									1.00					
FaithE									0.27	1.00				
Mono									0.14	0.29	1.00			
MonoCor									0.36	0.25	0.39	1.00		
PF									-0.10	-0.26	-0.04	-0.42	1.00	
Select									0.35	0.18	0.17	0.31	0.07	1.00

Legend	
1.00	Positively Correlated
0.00	Not Correlated
-1.00	Negatively Correlated

Figure 17. Kendall's τ_b correlation between XAI method mean scores (ViTs).

This table confirms that there was convergence only among the metrics of maximum sensitivity and average sensitivity. The other twelve metrics were generally uncorrelated. This verification is paralleled with a study conducted by Stassin et al. [57] based on CNN models, ResNet-50 [24] and VGG16 [66]. This study similarly indicated a limited correlation among metric rankings. Upon analysis, we observe that, concerning the complexity criterion, the metrics of sparseness, complexity, and effective complexity showed a higher degree of correlation compared to a ViT model. Additionally, for robustness, the results align with those for ViTs, where maximum sensitivity and average sensitivity demonstrated a strong correlation. Faithfulness metrics, on the other hand, exhibited minimal to no strong correlation, thereby making them the group with the least convergence and the most irregularities in their findings and aligning with our analysis. In terms of randomization, relatively speaking, model parameter randomization and random logit also demonstrated slightly more correlation compared to ViTs. However, it is important to note that this correlation lacks significant meaning.

4.3. Criteria Aggregation

The previous chapter demonstrates that averaging the results does not enable the ranking of methods by criterion. This section explores another avenue to determine if raw data might still contain information that reveals potential convergence of metrics by criterion, thereby averaging the results in the loss of the data distribution richness. Therefore, a second approach is to find a way to leverage this richness by starting from the raw data. In an initial analysis for a specific XAI method (e.g., TAM shown in Figure 18), wherein the correlations between the scores of metrics assigned to the explanations of the 2000 images were calculated and analyzed according to criterion.

TAM

	Robustness			Complexity			Randomization		Faithfulness					
	LLE	Max	Avg	Spars	Complex	Ecomplex	MPR	RL	FaithC	FaithE	Mono	MonoCor	PF	Select
LLE	1.00													
Max	-0.02	1.00												
Avg	-0.02	0.99	1.00											
Spars				1.00										
Complex				0.96	1.00									
Ecomplex				0.25	0.30	1.00								
MPR							1.00							
RL							0.03	1.00						
FaithC									1.00					
FaithE									0.09	1.00				
Mono									-0.04	-0.07	1.00			
MonoCor									0.03	0.19	-0.12	1.00		
PF									-0.04	-0.18	0.38	-0.21	1.00	
Select									0.03	0.04	-0.34	0.07	-0.62	1.00

Legend	
1.00	Positively Correlated
0.00	Not Correlated
-1.00	Negatively Correlated

Figure 18. Correlation between scores of the 2000 images for each metric for the TAM method.

These results were quite similar to those obtained through the analysis of the averages in the previous section. Few metrics appeared to be correlated within the same criterion, except for maximum sensitivity and average sensitivity, as well as sparseness and complexity. An examination of the results extended to all methods (compiled in Appendix A.1) reveals that metrics within a criterion generally did not seem to align, except for gradient-based methods (input Grad, integrated Grad, and smooth Grad) and CAM methods (Grad CAM, Grad CAM++, and score CAM), which aligned for the complexity criterion. Given this observation, considering that raw data were the source of the result without intermediate aggregation, the question arises about the origin of this lack of convergence when intuitively the metrics should converge, since they are all designed to measure the same criterion.

4.4. Metric's Discriminating Power

A second analysis involves evaluating the discriminative power of a metric, meaning its ability to identify which of two given XAI methods generally offers a higher score. If it is discriminative, this metric highlights the dominant method. For example, to determine which of the two XAI methods is more faithful, the faithfulness correlation metric should generally assign a higher score to the 2000 explanations of one of the two methods to conclude its greater fidelity. Therefore, a count of cases where the explanation of method A is superior to that of method B must be performed, and its dominance percentage must be examined. Figures 19 and 20 reflect this approach for the metrics faithfulness correlation and sparseness.

The ideal figure for a perfectly discriminative metric would only show cells shaded in green and red. Appendix A.2 compiles the other comparison tables of methods across all metrics. Each metric did not possess the same discriminative power towards the XAI methods. For example, Figure 19 shows that the faithfulness correlation metric discriminated between BTT and BTH in only 54% of the images at best. This indicates a low discriminative

power of the metric, and this observation can be generalized to other faithfulness metrics, which exhibited a lower discriminative power compared to metrics in other groups. In contrast, Figure 20 demonstrates that the sparseness metric discriminated between Chefer 1 and rise in as much as 98% of the images, which is significantly discriminative. In this experiment where the images were the sole variables of the analysis system (with a constant model and metric), we can conclude that metrics have varying discriminative power, and metrics with low discriminative power exhibit a strong dependency on the input image, because the input image is the only variable in the explanations.

Faithfulness correlation

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.51	0.49	0.5	0.49	0.49	0.5	0.51	0.5	0.5	0.5
Inputgrad	0.49	/	0.51	0.49	0.49	0.48	0.5	0.5	0.5	0.49	0.49
Integrad	0.51	0.49	/	0.51	0.5	0.5	0.51	0.52	0.5	0.52	0.5
Smoothgrad	0.5	0.51	0.49	/	0.51	0.5	0.49	0.51	0.51	0.51	0.49
Rollout	0.51	0.51	0.5	0.49	/	0.48	0.5	0.5	0.5	0.51	0.49
Chefer 1	0.51	0.52	0.5	0.5	0.52	/	0.52	0.53	0.52	0.52	0.52
Chefer 2	0.5	0.5	0.49	0.51	0.5	0.48	/	0.52	0.5	0.51	0.5
BT H	0.49	0.5	0.48	0.49	0.5	0.47	0.48	/	0.46	0.47	0.48
BT T	0.5	0.5	0.5	0.49	0.5	0.48	0.5	0.54	/	0.53	0.51
TAM	0.5	0.51	0.48	0.49	0.49	0.48	0.49	0.52	0.47	/	0.48
TIS	0.5	0.51	0.5	0.51	0.51	0.48	0.5	0.52	0.49	0.52	/

Figure 19. Method A's dominance percentage over method B using the faithfulness correlation metric.

Sparseness

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.11	0.14	0.1	0.04	0.02	0.04	0.09	0.07	0.1	0.03
Inputgrad	0.89	/	0.59	0.5	0.22	0.12	0.25	0.4	0.35	0.46	0.14
Integrad	0.86	0.41	/	0.42	0.15	0.07	0.19	0.33	0.3	0.37	0.09
Smoothgrad	0.9	0.5	0.58	/	0.24	0.13	0.27	0.43	0.38	0.47	0.12
Rollout	0.96	0.78	0.85	0.76	/	0.32	0.53	0.68	0.64	0.72	0.36
Chefer 1	0.98	0.88	0.93	0.87	0.68	/	0.89	0.86	0.85	0.89	0.56
Chefer 2	0.96	0.75	0.81	0.73	0.47	0.11	/	0.67	0.63	0.71	0.29
BT H	0.91	0.6	0.67	0.57	0.32	0.14	0.33	/	0.39	0.85	0.2
BT T	0.93	0.65	0.7	0.62	0.36	0.15	0.37	0.61	/	0.73	0.22
TAM	0.9	0.54	0.63	0.53	0.28	0.11	0.29	0.15	0.27	/	0.18
TIS	0.97	0.86	0.91	0.88	0.64	0.44	0.71	0.8	0.78	0.82	/

Figure 20. Method A's dominance percentage over method B using the sparseness metric.

4.5. Validity Analysis

The analysis of the methods is based on the *ViT_b16_224 model*, which is a variant of the ViT architecture pretrained on the ImageNet dataset. This model achieved an accuracy of 81% on the first class and 95% on the top-five classes. These results were obtained on the ImageNet test dataset. However, in this experiment, only 2000 images from the ImageNet dataset were used. This random selection did not follow the same distribution as the complete dataset. In fact, out of the 2000 selected images, the model had an accuracy of 72.25% on the first class and 80.10% on the top five.

The choice to use this model was based on a dual assumption: that explanations are dependent on the quality of the model and that the model selected for the experiment is of high quality. However, in light of the results on the experimental dataset, it is observed that the model was less accurate than expected. Therefore, it is necessary to examine whether this drop in quality significantly affects the previous conclusions.

After a visual analysis of the explanations for the images where the model made prediction errors, two primary sources of errors emerged during inference. First, the model made mistakes because the image contained objects belonging to two different classes, but the annotation only indicated one of these classes (see Figure 21). When the model makes a mistake, it can still distinguish between these two classes. Therefore, even if it does not predict the annotated class correctly, the explanation for that class yields satisfactory results. Second, the model made mistakes because the object in the image belonged to a class for which there were other closely related classes (see Figure 22). For example, in the case of ImageNet, various dog breeds were annotated as distinct classes. However, the objects of interest can sometimes be similar across these different classes. Thus, the model might make an error in predicting the class (e.g., the dog breed) but still identify important features (e.g., a dog). Consequently, the explanation for the true class was generally correct given the proximity between the annotated classes.

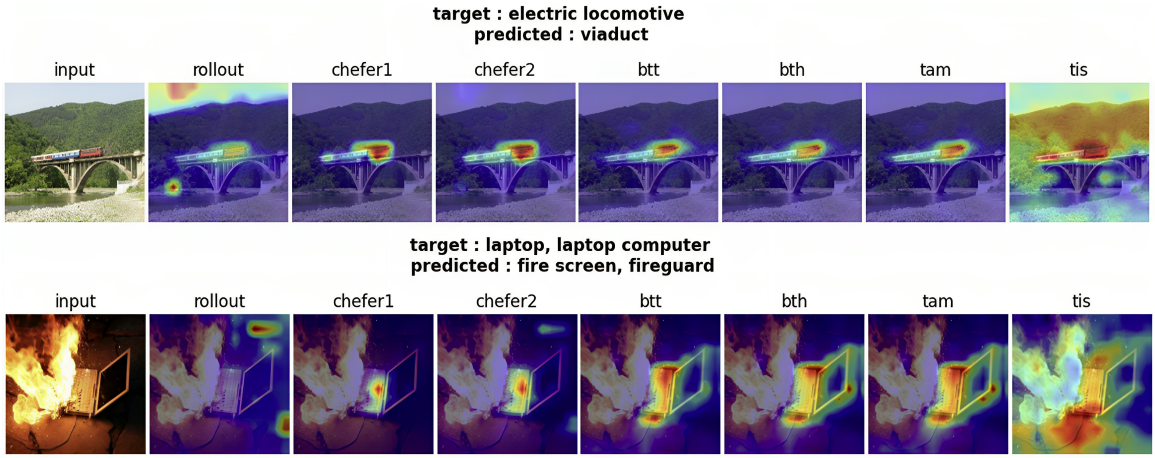

Figure 21. Images with two represented classes.

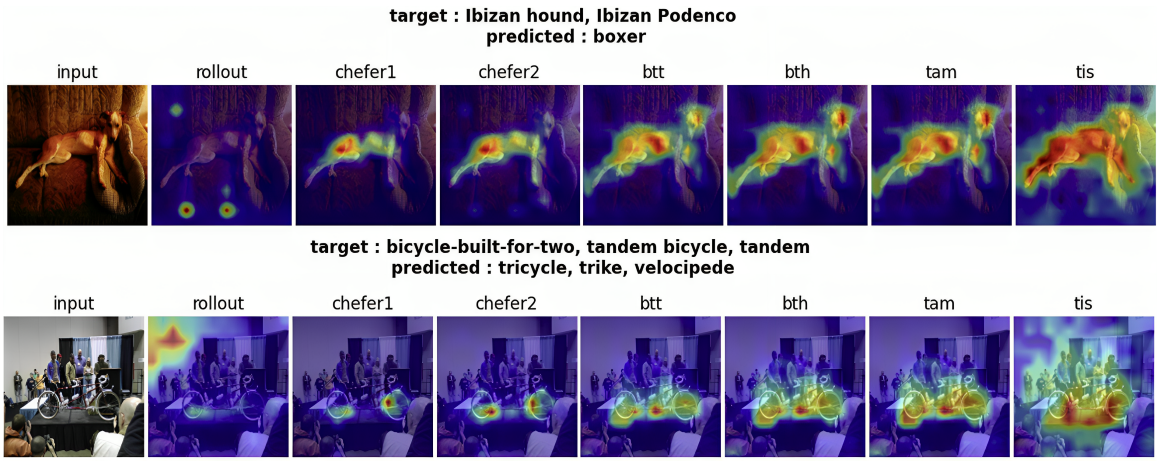

Figure 22. Images with objects of similar classes.

4.6. Inference Time

This section is dedicated to the examination of processing times for the XAI methods and evaluation metrics, the results of which are illustrated in Figure 23 and Table 2.

The benchmarks presented here do not aim to provide absolute results in terms of inference times, as these depend on the hardware used. However, these benchmarks allow for estimating processing times and identifying the slowest methods, which can be a crucial criterion depending on production constraints. Additionally, inference times vary depending on the implementation of methods and metrics. The results presented in this section provide an approximation of the processing times of this information in its current

state of implementation. An interesting observation is that, during the measurement of this information, the graphics processor was never used at more than 50%. This suggests that processing times could be significantly improved in the future by further optimizing the implementations of the methods and metrics for the graphics processors. Despite these limitations, these results demonstrate that XAI methods and metrics can be used to explain the results of deep learning models with reasonable execution times.

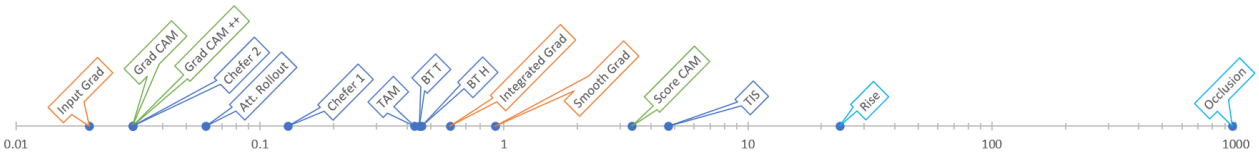

Figure 23. Inference time of XAI methods.

Table 2. Inference time of XAI metrics in seconds.

Category	Metric	Inference Time (s)
Faithfulness	Faithfulness Correlation	00.09
	Faithfulness Estimate	01.47
	Monotonicity	01.43
	Monotonicity Correlation	14.54
	Pixel Flipping	01.46
	Selectivity	01.69
Robustness	Local Lipschitz Estimate	00.58
	Max Sensitivity	00.60
	Avg Sensitivity	00.59
Complexity	Sparseness	00.26
	Complexity	00.14
	Effective Complexity	00.01
Randomization	Model Parameter Randomization	02.63
	Random Logit	00.04

Among the studied XAI methods in Figure 23, many can be calculated with processing times of less than 1 s. At the top end of the scale, input Grad, Grad CAM, Grad CAM++, and attention rollout had processing times in the hundredths of a second and are good choices for large databases. The occlusion method proved to be particularly costly for vision transformers (ViTs), thereby requiring 16 min. per iteration for a (16, 16) patch and 9 min per iteration for a (64, 64) patch. In general, perturbation methods (occlusion, rise, TiS) showed higher inference times even though RISE could lower the number of masks needed, and the TiS method parameters can be adapted to achieve reduced processing time with a minimal compromise on metric results [48].

Among the studied metrics in Table 2, the monotonicity correlation metric stands out for its high execution time, with model parameter randomization being the second slowest metric. These metrics remained suitable when working with a small database, but their usefulness may be questioned for larger databases, which would considerably slow down the experiment processing time. On the other extreme, some metrics had an execution time of the order of a few hundredths of a second (effective complexity, random logit, and faithfulness correlation) and are therefore interesting choices if the experiments performed involve very large databases.

5. Conclusions

In light of the results and as the conclusions of this work emerge, Albert Einstein's famous formula comes to mind: "The more I learn, the less I know". The overall structure to establish a ranking of XAI methods necessitates comparing them according to

criterion, which are themselves composed of various metrics, which are in line with the scientific literature.

However, the experimental results show that among the fourteen metrics tested on different methods, only two metrics (maximum sensitivity and average sensitivity) exhibited convergent results, whether in terms of ordinal ranking or, with a few exceptions mentioned in Section 4.3, in terms of raw results. This work highlights the dependence of explanations and their evaluation on the experimental environment. The results obtained for a specific criterion depend on the metric used, and they exhibit variability based on both the input image and the intrinsic parameters. The metrics employed were additionally influenced by the model used [57], thereby making it challenging to identify the best XAI method within a predefined set. While the results of the metrics may initially appear as mathematical results, and therefore inherently accurate and objective, their lack of convergence within the same criterion questions their common relevance for measuring that criterion and raises the legitimate question of which metric to prioritize. How does one select a metric within a criterion, knowing that the other metrics will provide conflicting or opposite rankings? Metrics are supposed to be objective compared to human judgment. In the absence of convergent metric results according to criterion, how can metrics aid in defining the best method for a use case? This is a limitation of current metric criteria and an open question highlighted by the findings in our paper.

Yet, in the scientific literature that presents new explainability methods, this choice is generally made without asking this question. The scientific articles referenced in this work focus on developing novel explainability methods. Logically, they benchmark their performance against that of previous methods, thereby adopting the evaluation metrics used in preceding works. Consequently, their conclusions are confined to these metrics, thus essentially comprising two faithfulness metrics (pixel flipping [40] and selectivity [58]). However, the literature that specifically deals with explanation evaluation reveals the existence of numerous faithfulness metrics, six of which were used in this work. This paper demonstrates that employing these metrics results in diverse and even divergent rankings.

In conclusion, with the current state of the art, the aggregation or weighting of metric results according to criterion appears to be a risky endeavor. Consequently, the ranking of methods, which was the initial intention of this study, is also considered challenging. The pursuit of this original goal has led to the establishment of three contributions:

- A scientific review: This work conducted a comparative study of XAI methods for the ViT architecture, thereby offering a perspective not explored to our knowledge in the existing literature.
- A scientific reserve: This study demonstrates that a broader perspective moderates the validity of current comparisons of XAI methods through metrics and highlights that they have only local and limited applicability within the study's environment. However, this observation in no way invalidates the quality of the referenced scientific articles, since their purpose is not a general classification of methods but their development.
- A framework for the analysis of XAI methods: Crafting a sophisticated framework enables the incorporation and evaluation of all existing XAI methods designed for vision transformers (ViTs) using metrics present in the current literature. This framework offers visual insights into explanations, thereby facilitating a deeper comprehension of the model's functionality.

Based on these findings, future directions in the field of explainability might explore the following:

- They might explore approaches for adapting existing metrics tied to specific properties to produce more convergent results. This adjustment aims to consolidate metrics into global criteria adapted to user needs.
- Integrating XAI metrics into human-centered studies could offer a synergistic approach to understanding model explainability. The criteria established by XAI metrics provide quantitative measures that can complement the qualitative insights gained from human-centered studies. Additionally, human-centered studies have the potential to

contribute valuable context-specific information that may help address the challenges associated with nonconvergent results observed in purely metric-based evaluations.

- Future research directions could focus on developing diverse representations of ground-truth labels for widely used datasets, such as ImageNet, to represent various perspectives on what constitutes correct explainability. This could contribute to advancing the robustness and applicability of pretrained models. It would also provide adaptability and be valuable when models are applied to user-specific tasks. Nevertheless, defining diverse ground-truth labels remains a complex task and would represent considerable resources for annotation.

Author Contributions: Conceptualization, S.S., V.C., S.A.M. and X.S.; methodology, S.S., V.C., S.A.M. and X.S.; software, V.C. and S.S.; validation, S.S., V.C. and S.A.M.; formal analysis, S.S., V.C. and S.A.M.; investigation, S.S. and V.C.; writing—original draft preparation, S.S. and V.C.; writing—review and editing, S.S., V.C., S.A.M. and X.S.; visualization, S.S. and V.C.; supervision, S.S., S.A.M. and X.S.; project administration, S.A.M. and X.S.; funding acquisition, S.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: Sédrick Stassin thanks the support of the COMP4++ project (convention no. 8565) within the Walloon Skywin pole of Belgium.

Data Availability Statement: The ImageNet 2012 validation dataset is available for download on the official ImageNet website. Once logged in, visit the downloads page at https://image-net.org/download-images.php (accessed on 13 November 2023).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Additional Figures

Appendix A.1. Correlation of Raw Data Scores Normalized

TAM

	Robustness			Complexity			Randomization		Faithfulness					
	LLE	Max	Avg	Spars	Complex	Ecomplex	MPR	RL	FaithC	FaithE	Mono	MonoCor	PF	Select
LLE	1.00													
Max	-0.02	1.00												
Avg	-0.02	0.99	1.00											
Spars				1.00										
Complex				0.96	1.00									
Ecomplex				0.25	0.30	1.00								
MPR							1.00							
RL							0.03	1.00						
FaithC									1.00					
FaithE									0.09	1.00				
Mono									-0.04	-0.07	1.00			
MonoCor									0.03	0.19	-0.12	1.00		
PF									-0.04	-0.18	0.38	-0.21	1.00	
Select									0.03	0.04	-0.34	0.07	-0.62	1.00

Legend	
1.00	Positively Correlated
0.00	Not Correlated
-1.00	Negatively Correlated

TiS

	Robustness			Complexity			Randomization		Faithfulness					
	LLE	Max	Avg	Spars	Complex	Ecomplex	MPR	RL	FaithC	FaithE	Mono	MonoCor	PF	Select
LLE	1.00													
Max	0.06	1.00												
Avg	0.06	0.99	1.00											
Spars				1.00										
Complex				0.23	1.00									
Ecomplex				0.21	0.98	1.00								
MPR							1.00							
RL							0.03	1.00						
FaithC									1.00					
FaithE									0.09	1.00				
Mono									-0.02	-0.13	1.00			
MonoCor									-0.01	0.18	0.00	1.00		
PF									-0.03	-0.19	0.34	-0.09	1.00	
Select									0.02	0.09	-0.35	-0.05	-0.60	1.00

Legend	
1.00	Positively Correlated
0.00	Not Correlated
-1.00	Negatively Correlated

BTH

	Robustness			Complexity			Randomization		Faithfulness					
	LLE	Max	Avg	Spars	Complex	Ecomplex	MPR	RL	FaithC	FaithE	Mono	MonoCor	PF	Select
LLE	1.00													
Max	-0.03	1.00												
Avg	-0.02	1.00	1.00											
Spars				1.00										
Complex				0.96	1.00									
Ecomplex				0.24	0.28	1.00								
MPR							1.00							
RL							0.03	1.00						
FaithC									1.00					
FaithE									0.08	1.00				
Mono									-0.04	-0.10	1.00			
MonoCor									0.03	0.21	-0.13	1.00		
PF									-0.03	-0.16	0.38	-0.21	1.00	
Select									0.02	0.03	-0.35	0.08	-0.63	1.00

Legend	
1.00	Positively Correlated
0.00	Not Correlated
-1.00	Negatively Correlated

BTT

	Robustness			Complexity			Randomization		Faithfulness					
	LLE	Max	Avg	Spars	Complex	Ecomplex	MPR	RL	FaithC	FaithE	Mono	MonoCor	PF	Select
LLE	1.00													
Max	-0.01	1.00												
Avg	-0.01	0.99	1.00											
Spars				1.00										
Complex				0.96	1.00									
Ecomplex				0.21	0.25	1.00								
MPR							1.00							
RL							0.03	1.00						
FaithC									1.00					
FaithE									0.08	1.00				
Mono									-0.05	-0.06	1.00			
MonoCor									0.01	0.23	-0.11	1.00		
PF									-0.04	-0.15	0.36	-0.18	1.00	
Select									0.03	0.05	-0.35	0.07	-0.64	1.00

Legend	
1.00	Positively Correlated
0.00	Not Correlated
-1.00	Negatively Correlated

Figure A1. Correlations between the 2000 images for each metric computed using XAI methods (part *a*).

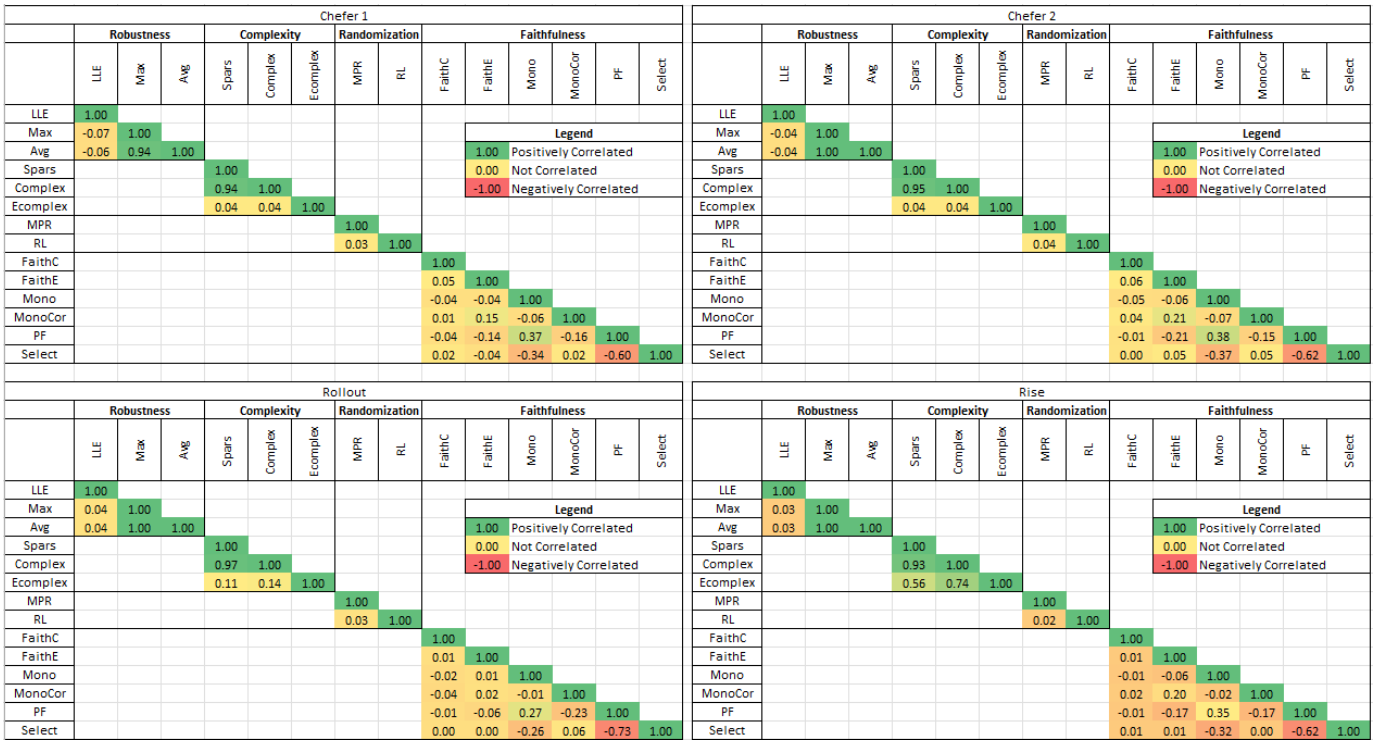

Figure A2. Correlations between the 2000 images for each metric computed by XAI methods (part *b*).

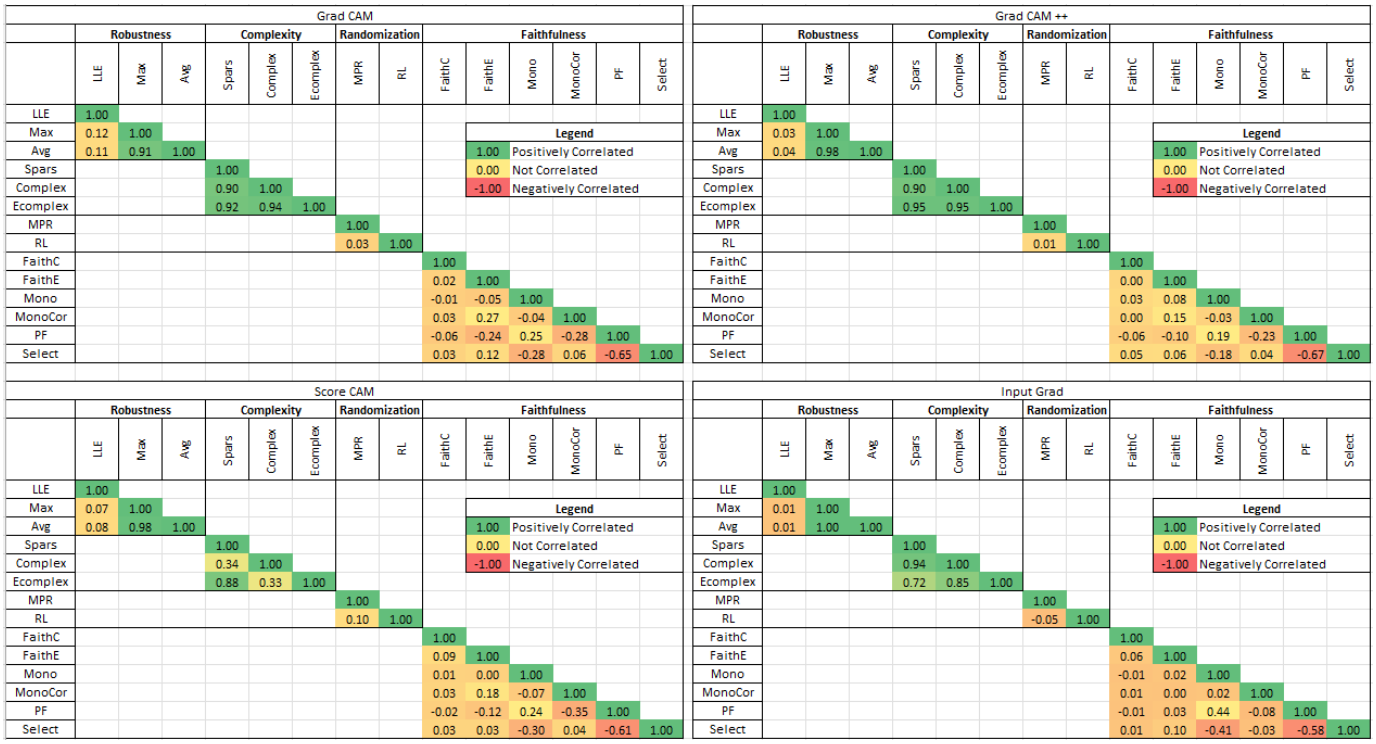

Figure A3. Correlations between the 2000 images for each metric computed using XAI methods (part *c*).

Smooth Grad

	Robustness			Complexity			Randomization		Faithfulness					
	LLE	Max	Avg	Spars	Complex	Ecomplex	MPR	RL	FaithC	FaithE	Mono	MonoCor	PF	Select
LLE	1.00													
Max	0.00	1.00												
Avg	0.00	1.00	1.00											
Spars				1.00										
Complex				0.94	1.00									
Ecomplex				0.85	0.42	1.00								
MPR							1.00							
RL							-0.01	1.00						
FaithC									1.00					
FaithE									0.04	1.00				
Mono									-0.04	-0.13	1.00			
MonoCor									0.01	0.16	-0.06	1.00		
PF									0.01	-0.16	0.37	-0.16	1.00	
Select									0.00	-0.02	-0.29	0.01	-0.57	1.00

Legend	
1.00	Positively Correlated
0.00	Not Correlated
-1.00	Negatively Correlated

Integrated Grad

	Robustness			Complexity			Randomization		Faithfulness					
	LLE	Max	Avg	Spars	Complex	Ecomplex	MPR	RL	FaithC	FaithE	Mono	MonoCor	PF	Select
LLE	1.00													
Max	0.07	1.00												
Avg	0.07	0.98	1.00											
Spars				1.00										
Complex				0.96	1.00									
Ecomplex				0.68	0.80	1.00								
MPR							1.00							
RL							0.03	1.00						
FaithC									1.00					
FaithE									0.02	1.00				
Mono									-0.04	0.00	1.00			
MonoCor									0.01	0.08	-0.02	1.00		
PF									0.00	-0.03	0.49	-0.08	1.00	
Select									0.03	-0.01	-0.42	-0.01	-0.6	1.00

Legend	
1.00	Positively Correlated
0.00	Not Correlated
-1.00	Negatively Correlated

Figure A4. Correlations between the 2000 images for each metric computed using XAI methods (part *d*).

Appendix A.2. Discriminative Power of Metrics

Faithfulness correlation

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.51	0.49	0.5	0.49	0.49	0.5	0.51	0.5	0.5	0.5
Inputgrad	0.49	/	0.51	0.49	0.49	0.48	0.5	0.5	0.5	0.49	0.49
Integrad	0.51	0.49	/	0.51	0.5	0.5	0.51	0.52	0.5	0.52	0.5
Smoothgrad	0.5	0.51	0.49	/	0.51	0.5	0.49	0.51	0.51	0.51	0.49
Rollout	0.51	0.51	0.5	0.49	/	0.48	0.5	0.5	0.5	0.51	0.49
Chefer 1	0.51	0.52	0.5	0.5	0.52	/	0.52	0.53	0.52	0.52	0.52
Chefer 2	0.5	0.5	0.49	0.51	0.5	0.48	/	0.52	0.5	0.51	0.5
BT H	0.49	0.5	0.48	0.49	0.5	0.47	0.48	/	0.46	0.47	0.48
BT T	0.5	0.5	0.5	0.49	0.5	0.48	0.5	0.54	/	0.53	0.51
TAM	0.5	0.51	0.48	0.49	0.49	0.48	0.49	0.52	0.47	/	0.48
TIS	0.5	0.51	0.5	0.51	0.51	0.48	0.5	0.52	0.49	0.52	/

Pixel-flipping

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.55	0.58	0.54	0.2	0.49	0.61	0.42	0.53	0.43	0.65
Inputgrad	0.45	/	0.52	0.46	0.22	0.44	0.5	0.4	0.45	0.4	0.54
Integrad	0.42	0.48	/	0.43	0.19	0.42	0.48	0.39	0.43	0.39	0.53
Smoothgrad	0.46	0.54	0.57	/	0.21	0.48	0.54	0.42	0.49	0.42	0.59
Rollout	0.8	0.78	0.81	0.79	/	0.79	0.86	0.75	0.81	0.75	0.88
Chefer 1	0.51	0.56	0.58	0.52	0.21	/	0.68	0.42	0.55	0.42	0.7
Chefer 2	0.39	0.5	0.52	0.46	0.14	0.32	/	0.31	0.43	0.32	0.58
BT H	0.58	0.6	0.61	0.58	0.25	0.58	0.69	/	0.71	0.49	0.75
BT T	0.47	0.55	0.57	0.51	0.19	0.45	0.57	0.29	/	0.29	0.64
TAM	0.57	0.6	0.61	0.58	0.25	0.58	0.68	0.51	0.71	/	0.75
TIS	0.35	0.46	0.47	0.41	0.12	0.3	0.42	0.25	0.36	0.25	/

Sparseness

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.11	0.14	0.1	0.04	0.02	0.04	0.09	0.07	0.1	0.03
Inputgrad	0.89	/	0.59	0.5	0.22	0.12	0.25	0.4	0.35	0.46	0.14
Integrad	0.86	0.41	/	0.42	0.15	0.07	0.19	0.33	0.3	0.37	0.09
Smoothgrad	0.9	0.5	0.58	/	0.24	0.13	0.27	0.43	0.38	0.47	0.12
Rollout	0.96	0.78	0.85	0.76	/	0.32	0.53	0.68	0.64	0.72	0.36
Chefer 1	0.98	0.88	0.93	0.87	0.68	/	0.89	0.86	0.85	0.89	0.56
Chefer 2	0.96	0.75	0.81	0.73	0.47	0.11	/	0.67	0.63	0.71	0.29
BT H	0.91	0.6	0.67	0.57	0.32	0.14	0.33	/	0.39	0.85	0.2
BT T	0.93	0.65	0.7	0.62	0.36	0.15	0.37	0.61	/	0.73	0.22
TAM	0.9	0.54	0.63	0.53	0.28	0.11	0.29	0.15	0.27	/	0.18
TIS	0.97	0.86	0.91	0.88	0.64	0.44	0.71	0.8	0.78	0.82	/

Complexity

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.06	0.07	0.06	0.03	0.04	0.04	0.06	0.05	0.07	0.02
Inputgrad	0.94	/	0.59	0.55	0.22	0.36	0.42	0.55	0.49	0.57	0.15
Integrad	0.93	0.41	/	0.45	0.13	0.26	0.34	0.45	0.4	0.47	0.1
Smoothgrad	0.94	0.45	0.55	/	0.18	0.34	0.41	0.52	0.46	0.54	0.1
Rollout	0.97	0.78	0.87	0.82	/	0.65	0.71	0.77	0.75	0.79	0.36
Chefer 1	0.96	0.64	0.74	0.66	0.35	/	0.63	0.66	0.62	0.69	0.21
Chefer 2	0.96	0.58	0.66	0.59	0.29	0.37	/	0.61	0.55	0.64	0.18
BT H	0.94	0.45	0.55	0.48	0.23	0.34	0.39	/	0.39	0.74	0.15
BT T	0.95	0.51	0.6	0.54	0.25	0.38	0.45	0.61	/	0.68	0.16
TAM	0.93	0.43	0.53	0.46	0.21	0.31	0.36	0.26	0.32	/	0.14
TIS	0.98	0.85	0.9	0.9	0.64	0.79	0.82	0.85	0.84	0.86	/

Figure A5. Method A's dominance percentage over method B (part *a*).

Local lipschitz estimate

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.38	0.51	0.59	0.51	0.6	0.33	0.37	0.4	0.4	0.44
Inputgrad	0.62	/	0.68	0.75	0.67	0.74	0.42	0.52	0.54	0.55	0.57
Integrad	0.49	0.32	/	0.63	0.49	0.59	0.3	0.35	0.36	0.38	0.45
Smoothgrad	0.41	0.25	0.37	/	0.39	0.56	0.21	0.27	0.28	0.29	0.38
Rollout	0.49	0.33	0.51	0.61	/	0.62	0.24	0.34	0.37	0.36	0.41
Chefer 1	0.4	0.26	0.41	0.44	0.38	/	0.26	0.32	0.33	0.33	0.35
Chefer 2	0.67	0.57	0.7	0.79	0.76	0.74	/	0.58	0.61	0.6	0.64
BT H	0.63	0.48	0.65	0.73	0.66	0.68	0.42	/	0.65	0.68	0.56
BT T	0.6	0.46	0.64	0.72	0.63	0.67	0.39	0.35	/	0.43	0.5
TAM	0.6	0.45	0.62	0.71	0.64	0.67	0.4	0.32	0.57	/	0.52
TIS	0.56	0.43	0.55	0.62	0.59	0.65	0.36	0.44	0.5	0.47	/

Selectivity

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.55	0.53	0.54	0.52	0.49	0.53	0.56	0.54	0.6	0.54
Inputgrad	0.45	/	0.47	0.47	0.48	0.44	0.46	0.49	0.47	0.52	0.47
Integrad	0.47	0.53	/	0.51	0.51	0.46	0.5	0.53	0.51	0.57	0.5
Smoothgrad	0.46	0.53	0.49	/	0.5	0.45	0.49	0.53	0.51	0.56	0.49
Rollout	0.48	0.52	0.49	0.5	/	0.47	0.5	0.53	0.5	0.55	0.51
Chefer 1	0.51	0.56	0.54	0.55	0.53	/	0.55	0.57	0.55	0.6	0.54
Chefer 2	0.47	0.54	0.5	0.51	0.5	0.45	/	0.54	0.5	0.56	0.51
BT H	0.44	0.51	0.47	0.47	0.47	0.43	0.46	/	0.46	0.54	0.47
BT T	0.46	0.53	0.49	0.49	0.5	0.45	0.5	0.54	/	0.58	0.51
TAM	0.4	0.48	0.43	0.44	0.45	0.4	0.44	0.46	0.42	/	0.43
TIS	0.46	0.53	0.5	0.51	0.49	0.46	0.49	0.53	0.49	0.57	/

Effective complexity

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	1	1	1	0.97	0.97	1	0.97	0.97	0.97	0.98
Inputgrad	0	/	0.44	1	0	0	1	0.01	0.02	0.02	0.02
Integrad	0	0.56	/	1	0	0	1	0.01	0.01	0.02	0.02
Smoothgrad	0	0	0	/	0	0	1	0	0	0	0.02
Rollout	0.99	1	1	1	/	0.99	1	0.99	0.99	0.99	0.99
Chefer 1	1	1	1	1	1	/	1	1	1	1	1
Chefer 2	0	0	0	1	0	0	/	0	0	0	0.02
BT H	0.96	0.99	0.99	1	0.96	0.96	1	/	1	0.98	0.96
BT T	0.96	0.99	0.99	1	0.96	0.96	1	0.96	/	0.98	0.96
TAM	0.96	0.98	0.98	1	0.96	0.96	1	0.98	0.98	/	0.96
TIS	0.98	0.98	0.98	1	0.98	0.98	1	0.98	0.98	0.98	/

Random logit

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.72	0.76	0.73	0.89	0.67	0.55	0.74	0.73	0.76	0.66
Inputgrad	0.28	/	0.58	0.35	0.43	0.25	0.31	0.4	0.38	0.41	0.36
Integrad	0.24	0.42	/	0.33	0.39	0.23	0.28	0.38	0.36	0.39	0.33
Smoothgrad	0.27	0.65	0.67	/	0.76	0.55	0.36	0.56	0.54	0.59	0.4
Rollout	0.11	0.57	0.61	0.24	/	0.39	0.22	0.39	0.37	0.42	0.25
Chefer 1	0.33	0.75	0.77	0.45	0.61	/	0.34	0.5	0.47	0.52	0.4
Chefer 2	0.45	0.69	0.72	0.64	0.78	0.66	/	0.73	0.71	0.74	0.56
BT H	0.26	0.6	0.62	0.44	0.61	0.5	0.27	/	0.44	0.7	0.3
BT T	0.27	0.62	0.64	0.46	0.63	0.53	0.29	0.56	/	0.64	0.32
TAM	0.24	0.59	0.61	0.41	0.58	0.48	0.26	0.3	0.36	/	0.27
TIS	0.34	0.64	0.67	0.6	0.75	0.6	0.44	0.7	0.68	0.73	/

Figure A6. Method A's dominance percentage over method B (part *b*).

Monotonicity

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.69	0.83	0.54	0.76	0.75	0.64	0.7	0.88	0.84	0.74
Inputgrad	0.31	/	0.74	0.3	0.57	0.55	0.4	0.51	0.74	0.68	0.55
Integrad	0.17	0.26	/	0.17	0.4	0.38	0.23	0.34	0.57	0.52	0.37
Smoothgrad	0.46	0.7	0.83	/	0.72	0.72	0.59	0.67	0.87	0.8	0.7
Rollout	0.24	0.43	0.6	0.28	/	0.47	0.35	0.43	0.65	0.58	0.47
Chefer 1	0.25	0.45	0.62	0.28	0.53	/	0.36	0.45	0.7	0.63	0.49
Chefer 2	0.36	0.6	0.77	0.41	0.65	0.64	/	0.59	0.82	0.75	0.64
BT H	0.3	0.49	0.66	0.33	0.56	0.54	0.41	/	0.75	0.68	0.5
BT T	0.12	0.26	0.43	0.13	0.34	0.3	0.18	0.25	/	0.44	0.31
TAM	0.16	0.32	0.48	0.2	0.42	0.37	0.25	0.32	0.56	/	0.37
TIS	0.26	0.45	0.63	0.3	0.52	0.51	0.36	0.44	0.69	0.63	/

Faithfulness estimate

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.6	0.78	0.48	0.63	0.55	0.47	0.52	0.48	0.53	0.44
Inputgrad	0.4	/	0.68	0.4	0.52	0.44	0.38	0.41	0.4	0.43	0.35
Integrad	0.22	0.32	/	0.23	0.34	0.25	0.22	0.22	0.22	0.23	0.19
Smoothgrad	0.52	0.6	0.77	/	0.63	0.55	0.49	0.52	0.49	0.53	0.47
Rollout	0.37	0.48	0.66	0.37	/	0.39	0.34	0.37	0.34	0.38	0.31
Chefer 1	0.45	0.56	0.75	0.45	0.61	/	0.37	0.45	0.43	0.48	0.41
Chefer 2	0.53	0.62	0.78	0.51	0.66	0.63	/	0.53	0.51	0.55	0.49
BT H	0.48	0.59	0.78	0.48	0.63	0.55	0.47	/	0.46	0.55	0.44
BT T	0.52	0.6	0.78	0.51	0.66	0.57	0.49	0.54	/	0.57	0.48
TAM	0.47	0.57	0.77	0.47	0.62	0.52	0.46	0.45	0.43	/	0.42
TIS	0.56	0.65	0.81	0.53	0.69	0.59	0.51	0.56	0.52	0.58	/

Max-sensitivity

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.06	0.75	0.22	0.79	0.21	0.21	0.4	0.42	0.42	0.56
Inputgrad	0.94	/	1	0.91	0.99	0.94	0.96	0.98	0.98	0.98	0.99
Integrad	0.25	0	/	0.06	0.56	0.04	0.03	0.1	0.08	0.12	0.27
Smoothgrad	0.78	0.09	0.94	/	0.97	0.35	0.38	0.67	0.74	0.68	0.84
Rollout	0.21	0.01	0.44	0.03	/	0.03	0.03	0.09	0.11	0.1	0.18
Chefer 1	0.79	0.06	0.96	0.65	0.97	/	0.58	0.85	0.87	0.86	0.89
Chefer 2	0.79	0.04	0.97	0.62	0.97	0.42	/	0.88	0.87	0.89	0.9
BT H	0.6	0.02	0.9	0.33	0.91	0.15	0.12	/	0.57	0.73	0.7
BT T	0.58	0.02	0.92	0.26	0.89	0.13	0.13	0.43	/	0.47	0.67
TAM	0.58	0.02	0.88	0.32	0.9	0.14	0.11	0.27	0.53	/	0.69
TIS	0.44	0.01	0.73	0.16	0.82	0.11	0.1	0.3	0.33	0.31	/

Model parameter randomisation

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.55	0.62	0.5	0.26	0.14	0.38	0.26	0.33	0.28	0.19
Inputgrad	0.45	/	0.6	0.46	0.16	0.22	0.4	0.34	0.39	0.35	0.29
Integrad	0.38	0.4	/	0.39	0.18	0.19	0.33	0.27	0.32	0.28	0.24
Smoothgrad	0.5	0.54	0.61	/	0.26	0.17	0.39	0.28	0.35	0.31	0.22
Rollout	0.74	0.83	0.82	0.74	/	0.53	0.7	0.64	0.66	0.65	0.6
Chefer 1	0.86	0.78	0.81	0.83	0.47	/	0.92	0.73	0.81	0.75	0.6
Chefer 2	0.62	0.6	0.67	0.61	0.3	0.08	/	0.31	0.38	0.35	0.21
BT H	0.74	0.66	0.73	0.72	0.36	0.27	0.69	/	0.69	0.66	0.36
BT T	0.67	0.61	0.68	0.65	0.34	0.19	0.62	0.31	/	0.37	0.28
TAM	0.72	0.65	0.72	0.69	0.35	0.25	0.65	0.34	0.63	/	0.34
TIS	0.81	0.71	0.76	0.77	0.4	0.4	0.79	0.64	0.72	0.66	/

Figure A7. Method A's dominance percentage over method B (part *c*).

Monotonicity correlation

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.38	0.5	0.48	0.56	0.5	0.46	0.6	0.51	0.58	0.49
Inputgrad	0.62	/	0.7	0.6	0.69	0.6	0.59	0.7	0.63	0.68	0.62
Integrad	0.5	0.3	/	0.46	0.58	0.47	0.45	0.57	0.49	0.55	0.5
Smoothgrad	0.52	0.4	0.54	/	0.58	0.5	0.5	0.6	0.52	0.59	0.52
Rollout	0.44	0.31	0.42	0.42	/	0.44	0.41	0.5	0.45	0.5	0.44
Chefer 1	0.5	0.4	0.53	0.5	0.56	/	0.45	0.59	0.5	0.58	0.49
Chefer 2	0.54	0.41	0.55	0.5	0.59	0.55	/	0.63	0.55	0.61	0.54
BT H	0.4	0.3	0.43	0.4	0.5	0.41	0.37	/	0.35	0.47	0.4
BT T	0.49	0.37	0.51	0.48	0.55	0.5	0.45	0.65	/	0.62	0.49
TAM	0.42	0.32	0.45	0.41	0.5	0.42	0.39	0.53	0.38	/	0.42
TIS	0.51	0.38	0.5	0.48	0.56	0.51	0.46	0.6	0.51	0.58	/

Avg-sensitivity

Method A \ Method B	Rise	Inputgrad	Integrad	Smoothgrad	Rollout	Chefer 1	Chefer 2	BT H	BT T	TAM	TIS
Rise	/	0.06	0.7	0.22	0.79	0.18	0.21	0.4	0.4	0.41	0.54
Inputgrad	0.94	/	1	0.91	0.99	0.93	0.96	0.98	0.98	0.98	0.99
Integrad	0.3	0	/	0.08	0.62	0.04	0.04	0.13	0.1	0.15	0.33
Smoothgrad	0.78	0.09	0.92	/	0.97	0.31	0.38	0.66	0.72	0.67	0.83
Rollout	0.21	0.01	0.38	0.03	/	0.02	0.03	0.09	0.1	0.1	0.18
Chefer 1	0.82	0.07	0.96	0.69	0.98	/	0.71	0.89	0.9	0.89	0.92
Chefer 2	0.79	0.04	0.96	0.62	0.97	0.29	/	0.88	0.88	0.88	0.89
BT H	0.6	0.02	0.87	0.34	0.91	0.11	0.12	/	0.55	0.66	0.68
BT T	0.6	0.02	0.9	0.28	0.9	0.1	0.12	0.45	/	0.48	0.68
TAM	0.59	0.02	0.85	0.33	0.9	0.11	0.12	0.34	0.52	/	0.68
TIS	0.46	0.01	0.67	0.17	0.82	0.08	0.11	0.32	0.32	0.32	/

Figure A8. Method A's dominance percentage over method B (part *d*).

References

1. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **1958**, *65*, 386–408. [CrossRef] [PubMed]
2. Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; Mané, D. Concrete problems in AI safety. *arXiv* **2016**, arXiv:1606.06565.
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–15.
4. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
5. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
6. Liu, Y.; Zhang, Y.; Wang, Y.; Hou, F.; Yuan, J.; Tian, J.; Zhang, Y.; Shi, Z.; Fan, J.; He, Z. A Survey of Visual Transformers. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–21. [CrossRef]
7. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8126–8135.
8. Wang, H.; Zhu, Y.; Adam, H.; Yuille, A.; Chen, L.C. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 5463–5474.
9. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 213–229.
10. European Commission. Ethics Guidelines for Trustworthy AI. 2023. Available online: https://ec.europa.eu/futurium/en/ai-alliance-consultation.1.html (accessed on 13 November 2023).
11. Pruthi, D.; Gupta, M.; Dhingra, B.; Neubig, G.; Lipton, Z.C. Learning to deceive with attention-based explanations. *arXiv* **2019**, arXiv:1909.07913.
12. Serrano, S.; Smith, N.A. Is attention interpretable? *arXiv* **2019**, arXiv:1906.03731.
13. Jain, S.; Wallace, B.C. Attention is not explanation. *arXiv* **2019**, arXiv:1902.10186.
14. Kendall, M.G. The treatment of ties in ranking problems. *Biometrika* **1945**, *33*, 239–251. [CrossRef]
15. Speith, T. A review of taxonomies of explainable artificial intelligence (XAI) methods. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, 21–24 June 2022; pp. 2239–2250.
16. Samek, W.; Müller, K.R. Towards explainable artificial intelligence. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*; Springer: Cham, Switzerland, 2019; pp. 5–22.
17. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
18. McDermid, J.A.; Jia, Y.; Porter, Z.; Habli, I. Artificial intelligence explainability: The technical and ethical dimensions. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200363. [CrossRef] [PubMed]
19. Sayed-Mouchaweh, M. *Explainable AI within the Digital Transformation and Cyber Physical Systems*; Springer: Cham, Switzerland, 2021.
20. Amini, A. MIT 6.S191: Recurrent Neural Networks, Transformers, and Attention. 2023. Available online: https://www.youtube.com/watch?v=ySEx_Bqxvvo&ab_channel=AlexanderAmini (accessed on 13 November 2023).
21. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
22. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.

23. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
25. Kamat, D. Vision Transformers (ViT). 2021. Available online: https://dkamatblog.home.blog/2021/08/05/vision-transformers-vit/ (accessed on 13 November 2023).
26. Guo, M.H.; Xu, T.X.; Liu, J.J.; Liu, Z.N.; Jiang, P.T.; Mu, T.J.; Zhang, S.H.; Martin, R.R.; Cheng, M.M.; Hu, S.M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
27. Kindermans, P.J.; Schütt, K.; Müller, K.R.; Dähne, S. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv* **2016**, arXiv:1611.07270.
28. Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; Wattenberg, M. Smoothgrad: Removing noise by adding noise. *arXiv* **2017**, arXiv:1706.03825.
29. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic attribution for deep networks. In Proceedings of the ICML, Sydney, Australia, 6–11 August 2017.
30. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 818–833.
31. Petsiuk, V.; Das, A.; Saenko, K. RISE: Randomized Input Sampling for Explanation of Black-box Models. In Proceedings of the BMVC 2018, Newcastle, UK, 3–6 September 2018.
32. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26–30 June 2016; pp. 2921–2929.
33. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
34. Chattopadhay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 839–847.
35. Wang, H.; Wang, Z.; Du, M.; Yang, F.; Zhang, Z.; Ding, S.; Mardziel, P.; Hu, X. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Workshop on Fair, Data Efficient and Trusted Computer Vision, Seattle, WA, USA, 14–19 June 2020.
36. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 7–9 July 2015; pp. 2048–2057.
37. Michel, P.; Levy, O.; Neubig, G. Are sixteen heads really better than one? *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
38. Brunner, G.; Liu, Y.; Pascual, D.; Richter, O.; Ciaramita, M.; Wattenhofer, R. On identifiability in transformers. *arXiv* **2019**, arXiv:1908.04211.
39. Abnar, S.; Zuidema, W. Quantifying attention flow in transformers. *arXiv* **2020**, arXiv:2005.00928.
40. Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.R.; Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **2015**, *10*, e0130140. [CrossRef]
41. Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; Titov, I. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv* **2019**, arXiv:1905.09418.
42. Chefer, H.; Gur, S.; Wolf, L. Transformer interpretability beyond attention visualization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 782–791.
43. Chefer, H.; Gur, S.; Wolf, L. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 397–406.
44. Yuan, T.; Li, X.; Xiong, H.; Cao, H.; Dou, D. Explaining Information Flow Inside Vision Transformers Using Markov Chain. In Proceedings of the eXplainable AI Approaches for Debugging and Diagnosis. 2021. Available online: https://openreview.net/forum?id=TT-cf6QSDaQ (accessed on 28 December 2023).
45. Chen, J.; Li, X.; Yu, L.; Dou, D.; Xiong, H. Beyond Intuition: Rethinking Token Attributions inside Transformers. *Trans. Mach. Learn. Res.* **2022**. Available online: https://openreview.net/forum?id=rm0zIzlhcX (accessed on 13 November 2023).
46. Xie, W.; Li, X.H.; Cao, C.C.; Zhang, N.L. ViT-CX: Causal Explanation of Vision Transformers. *arXiv* **2022**, arXiv:2211.03064.
47. Müllner, D. fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python. *J. Stat. Softw.* **2013**, *53*, 1–18. [CrossRef]
48. Englebert, A.; Stassin, S.; Nanfack, G.; Mahmoudi, S.A.; Siebert, X.; Cornu, O.; De Vleeschouwer, C. Explaining through Transformer Input Sampling. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 806–815.
49. Raghu, M.; Unterthiner, T.; Kornblith, S.; Zhang, C.; Dosovitskiy, A. Do vision transformers see like convolutional neural networks? *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12116–12128.

50. Hooker, S.; Erhan, D.; Kindermans, P.J.; Kim, B. A benchmark for interpretability methods in deep neural networks. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–17.
51. Samek, W.; Binder, A.; Montavon, G.; Lapuschkin, S.; Müller, K.R. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2660–2673. [CrossRef]
52. Bhatt, U.; Weller, A.; Moura, J. Evaluating and Aggregating Feature-based Model Explanations. In Proceedings of the International Conference on International Joint Conferences on Artificial Intelligence (IJCAI), Online, 7–15 January 2021.
53. Alvarez-Melis, D.; Jaakkola, T.S. Towards Robust Interpretability with Self-Explaining Neural Networks. In Proceedings of the NeurIPS, Montreal, QC, Canada, 3–8 December 2018.
54. Nguyen, A.p.; Martínez, M.R. On quantitative aspects of model interpretability. *arXiv* **2020**, arXiv:2007.07584.
55. Hedström, A.; Weber, L.; Krakowczyk, D.; Bareeva, D.; Motzkus, F.; Samek, W.; Lapuschkin, S.; Höhne, M.M.C. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *J. Mach. Learn. Res.* **2023**, *24*, 1–11.
56. Arya, V.; Bellamy, R.K.; Chen, P.Y.; Dhurandhar, A.; Hind, M.; Hoffman, S.C.; Houde, S.; Liao, Q.V.; Luss, R.; Mojsilović, A.; et al. One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques. *arXiv* **2019**, arXiv:1909.03012.
57. Stassin, S.; Englebert, A.; Nanfack, G.; Albert, J.; Versbraegen, N.; Peiffer, G.; Doh, M.; Riche, N.; Frenay, B.; De Vleeschouwer, C. An Experimental Investigation into the Evaluation of Explainability Methods. *arXiv* **2023**, arXiv:2305.16361.
58. Montavon, G.; Samek, W.; Müller, K.R. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.* **2018**, *73*, 1–15. [CrossRef]
59. Bellucci, M.; Delestre, N.; Malandain, N.; Zanni-Merk, C. Une terminologie pour une IA explicable contextualisée. In Proceedings of the EXPLAIN'AI Workshop EGC 2022, Vienna, Austria, 23–29 July 2022.
60. Chalasani, P.; Chen, J.; Chowdhury, A.R.; Wu, X.; Jha, S. Concise explanations of neural networks using adversarial training. In Proceedings of the International Conference on Machine Learning (ICML), Virtual Event, 13–18 July 2020.
61. Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; Kim, B. Sanity Checks for Saliency Maps. In Proceedings of the NeurIPS, Montreal, QC, Canada, 3–8 December 2018.
62. Sixt, L.; Granz, M.; Landgraf, T. When explanations lie: Why many modified bp attributions fail. In Proceedings of the International Conference on Machine Learning (ICML), Virtual Event, 13–18 July 2020.
63. Alvarez-Melis, D.; Jaakkola, T.S. On the robustness of interpretability methods. *arXiv* **2018**, arXiv:1806.08049.
64. Yeh, C.K.; Hsieh, C.Y.; Suggala, A.; Inouye, D.I.; Ravikumar, P.K. On the (in) fidelity and sensitivity of explanations. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–12.
65. Zhang, J.; Bargal, S.A.; Lin, Z.; Brandt, J.; Shen, X.; Sclaroff, S. Top-down neural attention by excitation backprop. *Int. J. Comput. Vis.* **2018**, *126*, 1084–1102. [CrossRef]
66. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.