Visualization Techniques for Improving Explainability in Visual Transformers

Ana Iulia Coporan

West University of Timisoara, Timisoara, Romania
ana.coporan01@e-uvt.ro

Abstract. Vision Transformers (ViTs) have emerged as a powerful alternative to convolutional neural networks (CNNs) for various computer vision tasks, demonstrating state-of-the-art performance in image classification, object detection, and segmentation. However, due to their unique self-attention mechanisms and lack of intuitive hierarchical structure, understanding the internal workings of ViTs remains a significant challenge. Visualization techniques play a crucial role in enhancing the interpretability and explainability of ViTs by providing insights into the model's decision-making process. This report reviews the current state-of-the-art visualization techniques for ViTs, following multiple categories of methods, like attention-based, gradient-based, or perturbation-based. Attention flow, transition attention maps(TAM), class activation maps (CAM), layer-wise relevance propagation (LRP) and causal explanations are explored in the context of ViTs, focusing on how these techniques can reveal the regions of an image most influential in the model's predictions. We offer implementation of two such visualization methods, LRP and CAM and do a comparison of them by looking at different types of results that they produce. By alligning the results with human interpretability but also by comparing some quantitative results we conclude that LRP performs better than CAM. The report concludes with an overview of ongoing research directions aimed at improving the interpretability and explainability of ViTs.

Keywords: ViT · XAI · CAM · LRP · Attention · Causal Explanations

1 Introduction

In the field of artificial intelligence(AI) applications span across various types of inputs, including text, images, audio, and more. Each type of input requires specialized techniques to process and analyze effectively. For computer vision tasks like image classification convolutional neural networks (CNNs) have been the go-to option because of their very accurate results.

Despite the dominance of CNNs in image classification, newer architectures like vision transformers (ViTs) are emerging. Originally developed for natural language processing (NLP) tasks because of their impressive performance they have been extended also to the vision domain. ViT have demonstrated performances that are comparable with those of CNN arhitectures.[4]. This evolution

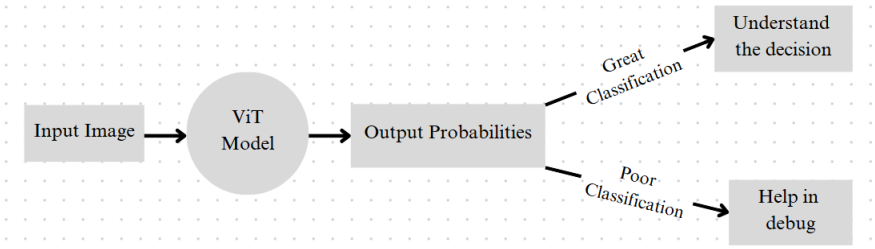

Fig. 1. Motivation of explainable visualization techniques

in AI architecture reflects a shift towards models that integrate global context in a more efficient manner, potentially offering advantages

To expand the knowledge behind ViT it is crucial to understand their inner working procedure and examine their explainability[7]. Explainable artificial intelligence (XAI) is a domain that challanges the transparency of AI models. XAI holds the promise of making AI systems more trustworthy and accountable. Not only that but it helps in better understanding the models, uncover limitations and improve the overall performance. As AI becomes more embedded in critical decision-making processes, transparency is vital to ensure ethical use and mitigate risks. This endeavor ensures that AI systems are not only powerful but also transparent, reliable, and aligned with human values and expectations. Regarding ViT XAI can help in uncovering which features are most important and which image regions have impacted the most the prediction, helping to better clarify the workings of the attention mechanism. This kind of interpretability has the potential to revolutionize fields like autonomous driving and medical imaging, where knowing the "why" behind a decision is as important as the decision itself. This understanding is crucial in applications like medical diagnostics, where knowing which specific image features led to a diagnosis can enhance interpretability and trust in AI-driven medical decisions.

Despite the promise of ViTs, their explainability poses significant challenges. Unlike CNNs, where convolutional layers explicitly capture hierarchical features, ViTs rely on self-attention mechanisms to integrate global context and local dependencies across image patches. These self-attention mechanisms allow ViTs to process long-range dependencies, but they also introduce new challenges in terms of interpretability, requiring novel approaches to decode the complex interactions within the model. This unique architecture necessitates novel approaches for interpreting how these models arrive at their predictions, highlighting which image regions and features contribute most significantly to classification outcomes.

A more detailed view on the flow which describes the motivation of using visualization techniques as explainability methods for ViT models is the following: you give a picture to a pre-trained ViT model and the model gives you as output probabilities corresponding to its classification of the input image. Following this, two scenarios may appear. The model seems to be great at making classifications, but users are afraid to use it because there is a lack of transparency.

Let us imagine the healthcare sector, in which a model says based on a MRI scan that a patient has cancer. Doctors and medical experts can use visualization techniques as a second opinion to complement diagnostic processes. This dual approach, AI predictions validated by human expertise, has the potential to create a more robust system for life-critical applications. Doctors can cross-check these regions highlighted by visualization techniques with their knowledge to validate the prediction. The second scenario is that the model makes bad decision. By using the insights provided by the visualization techniques, developers can use the heatmaps as debugging techniques on understanding what went wrong and how to improve the models performance. The model might have learned spurious correlations or irrelevant features due to biases in the training data or it might not properly capture the semantic features of the object. By identifying such issues, researchers and engineers can refine the training data, adjust model architectures, or implement additional regularization techniques to mitigate these problems. This flow is visually explained by Figure 1.

The question is, how did the model come to such a result, on which parts of the image did he focus his attention, which pixels were more relevant for his decision making? Visualization techniques help uncover that by analyzing the attention through different techniques. Ultimately, these techniques form an integral part of the ongoing efforts to ensure that AI models are not only effective but also understandable to humans in real-world scenarios.

2 Problem Formulation

The problem is finding out which parts of the input image influence the decisions of the ViT classification models in order to gain a better understanding on how the attention mechanism works in a ViT and consequently how a ViT arrives to a decision. Let us put down the problem in technical terms, describing the workflow from 2 The input of visualization techniques is a trained ViT model with its internal representations, like attention weights, feature embeddings or gradients and the input image. The output is a heatmap, where warmer colors indicate regions of high importance, and cooler colors indicate less influential areas. These heatmaps are often overlaid on the original image to provide clear context. This means that visualization techniques provide as output visual explanations that highlight important regions of the image that influence the decision of the ViT model. These techniques can be particularly useful for understanding model behavior in complex or sensitive domains, like healthcare or autonomous driving, where knowing the rationale behind a decision is critical.

The question that remains is how can you derive a heatmap with explanations on why a model came to a certain conclusion based on the model itself, how can you extract relevant information about its workings out of the model's architecture? This issue becomes more challenging as the complexity of the model architecture increases. ViTs, with their self-attention mechanisms, present a unique challenge due to their non-hierarchical structure and the need to process long-range dependencies across the image. There are some fundamental principles

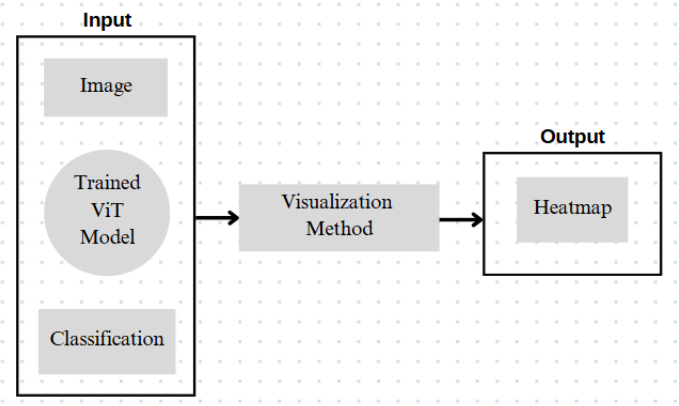

Fig. 2. Workflow of explainable visualization techniques

that offer a starting point for solving this problem. At the core of visualization methods is the idea of attributing importance to different regions of the input image, in other words, the heatmap. This involves determining which parts of the image had the greatest influence on the final decision of the model. For generating the heatmap the goal is to assign a weight to each region, whether it is a pixel, patch, or feature, that reflects its contribution to the classification output. This is where the power of explainability techniques lies, they break down complex, black-box models into more interpretable and human-understandable outputs. To determine this importance, techniques rely on the propagation of influence through the model. There are two main ways in which one can trace the flow of information through the model, either forward as the input image passes through the model or backward, starting from the model's output and propagating relevance backward through the network layers. The backward approach, such as gradient-based methods, has been widely used because it allows for the tracing of model decisions from output to input, revealing which input features were most important in forming the final output. Once the influence has been traced, the next step is to quantify the contribution of each image region. This involves calculating a score that reflects how strongly each region impacted the model's decision. There are different ways on how this quantification can be computed, for example by measuring how strongly certain features are activated within the model or by assessing how sensitive the output is to small changes in the input.

3 Theoretical Background

3.1 Vision Transformers

Vision Transformers[4] work by getting 2D images converted into 1D sequences of sequentially arranged patches. The patches get through the multi head self

attention mechanism, which learns the relationships between them. The architecture of ViT can be broken down into multiple steps, an overview being shown in Figure 3. The first step prepares the input image for the following steps, transforming it into a sequence of patch tokens. Each patch is flattened into a one-dimensional vector. The next step is about patch embedding. Each flattened patch vector is linearly projected to a lower-dimensional space using a trainable linear layer, producing patch embeddings. Positional embeddings are added to each patch embedding in order to capture the order of the patches. The use of positional embeddings is crucial for ViTs because, unlike CNNs, they do not inherently preserve the spatial structure of the image, so these embeddings enable the model to understand the relative positions of the image patches. Another important ViT step is the classification head, for which a class token gets added to the sequence of patch embeddings. This token aggregates information from all patches during the self-attention operations and is ultimately used for classification. The next part are the transformer encoder layers. These layers include multi head self attention, a normalization layer, a feed forward network in the form of the multi-layer perceptron(MLP) and residual connections.The residual connections ensure that the gradients can flow more effectively through the network during training Of importance to our study is the self attention mechanism. The self attention of each token denotes how much attention a token should pay to itself and to others. For each attention head, three vectors are computed: query, key and values. Through these, the self attention matrices are computed. The attention outputs from each attention head are concatenated along the feature dimension and then linearly projected to produce the final output.

ViTs use a self-attention mechanism as a core part of their architecture. This mechanism allows the model to weigh the importance of different parts of the input image when making predictions. What distinguishes self-attention from traditional CNN-based approaches is its ability to model long-range dependencies between distant image patches, unlike CNNs which focus more on local feature extraction. The self-attention mechanism computes attention scores, which indicate how much focus each part of the image receives. These attention scores provide insight into how the model processes information from different parts of the image, which is why attention is widely used for explainability. By analyzing these attention scores, researchers can gain a clearer understanding of which regions of an image influence the model's decision, making it a valuable tool for interpretation.

3.2 Explainability of ViT

To understand the inner work of a ViT one can choose one of the several existing approaches: visualization techniques, attention based methods, pruning-based methods, inherently explainable models. A more comprehensive review of ViT explainability is given by [7]. They do a thoroughly organization of the existing methods, taking into considerations factors like motivation, structure and application scenarios. The work of this project focuses on the visual side of explainability. For visualizing attention there is also a variety of methods that

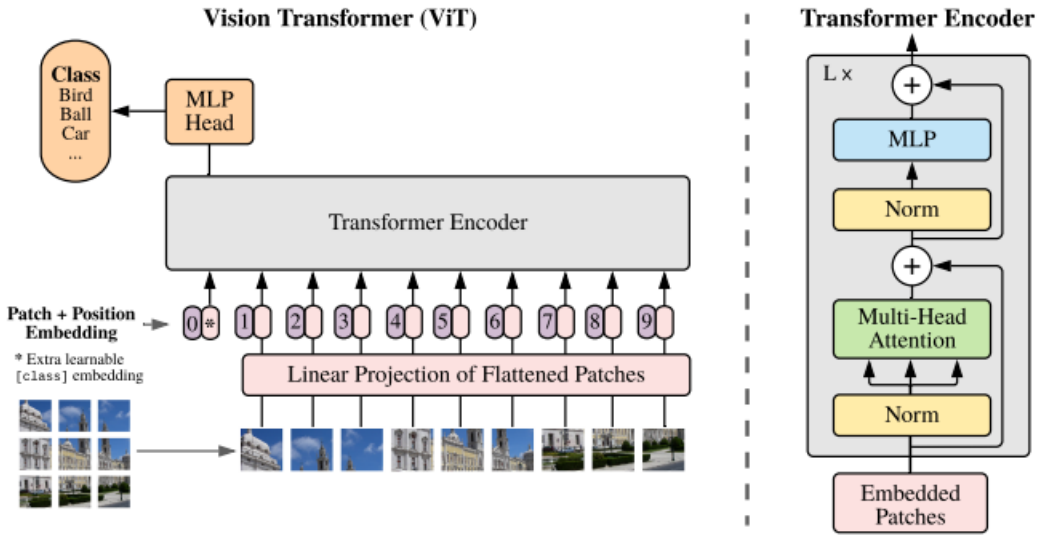

Fig. 3. A schematic representation of the ViT architecture[4]

can be chosen, from flow maps, parallel coordination plots and heatmaps[9]. Heatmaps are the most popular among them, they visually represent the intensity of attention strengths through color coding. Heatmaps provide an intuitive way to understand and analyze complex data patterns.

Visualization methods for transformer based models have been classified in a few different categories.

Gradient-based methods for interpreting deep learning models work by calculating the gradients of the model's output with respect to the input features. The logic is based on the idea that these gradients indicate how much each input feature contributes to the model's prediction. By visualizing these gradients, these methods highlight important regions or aspects of the input that most influence the model's decision. For example vanilla saliency computes the absolute value of the gradients of the output with respect to the input, by computing the absolute value of the gradient of the loss and taking the maximum gradient value across the color channels with respect to the input image.

Another category is represented by attribution propagation methods which are based on the Deep Taylor Decomposition(DTD). DTD propagates relevance scores backward from the output layer to the input layer.

3.3 Markov chains

A Markov chain[11] is a tool which is being used for modeling random processes in which the probability of transitioning from one state to another depends only

on the current state. A Markov chain is being characterized by the transition matrix, which defines the probabilities of moving from one state to another in the system. Each element of the transition matrix, represents the probability of transitioning from one state to another. The rows of the transition matrix correspond to the current states, while the columns represent the potential next states. For the matrix to be valid, the probabilities in each row must sum to 1, in order to make sure that the system transitions to some state, including the current one, is possible with total certainty.

3.4 Evaluation metrics

Key evaluation metrics are segmentation metrics such as Pixel Accuracy, F1 Score, and Jaccard Index. These metrics are essential for evaluating the quality of segmentation results, providing insights into how accurately and precisely the models can identify and localize relevant features in input images. Pixel Accuracy quantifies the overall correctness of the segmentation predictions. The F1 Score offers a balanced measure of precision and recall across different classes, highlighting the trade-off between these two aspects. The Jaccard Index (or Intersection over Union) measures the overlap between predicted segments and the ground truth, offering a comprehensive view of segmentation performance.

A lot of time segmentation metrics are being used to evaluate explainable methods because these metrics are designed to measure how well the highlighted regions in an image correspond to meaningful, ground-truth regions. While XAI methods are not performing segmentation directly, their goal is often to identify which parts of the input contributed most to the model's decision, which is conceptually similar to segmentation.

There are also explainability-specific metrics that help quantify the results of explainable methods. Most known are deletion and insertion metrics. Deletion measures the drop in model confidence when the most important pixels are removed, while insertion measure the confidence gain when pixels are gradually added in order of importance. After the input image is perturbed, either by deletion or by inserion, the model's performance is measured. The area under the curve (AUC) is computed by comparing the model's performance against the fraction of the input that has been deleted or inserted. In the case of deletion, if the explanation method correctly identifies the most important regions, deleting those regions will cause a sharp drop in model performance, which results in a lower AUC. Both of the methods are relevant by quantifying how much the highlighted regions contribute to the model's decision. Another explainability-specific metric is called pointing game. It checks if the most activated pixel in the heatmap falls within the true object region, being effective to test if the focus point aligns with the object of interest.

4 Methodology

The selection of papers for this word has been done through a systematic two-step approach. First, a keyword-based search was performed on Google Scholar

using queries such as "XAI", "ViT", "visualization techniques", "attention visualization", "causal XAI" and "understanding attention in ViT". Papers were initially selected based on their relevance to the domain of explainability and visualization techniques for ViT. This screening involved reading the abstracts, introductions, and conclusions of the selected papers to make sure that they alligned with the focus of this work. Secondly, a snowballing approach has been performed. This means that the references of the initially selected papers were examined to identify additional relevant literature. This method helped uncover the primary literature for this review but also other relevant papers, ensuring that the search covered as much as possible from the field of interest. Together, these strategies offered a way in which a representative set of papers have been chosen, that address the challenges and advancements in understanding and visualizing attention mechanisms in ViTs.

Table 1. Summary of Selected Papers

Paper	**Method**	**Model Type**	**Citations**
[12]	Grad-CAM	CNN	22,159
[16]	CAM	CNN	12,640
[2]	LRP	CNN	5,375
[1]	Attention Rollout/Flow	NLP Transformer	955
[3]	LRP	ViT	864
[8]	CAM	ViT	12
[14]	Causal Explanation	ViT	18
[13]	Causal Explanation	General black box	3338
[11]	Markov chains	General	5916
[15]	Markov chains	ViT	30
[7]	Review of ViT Explainability	ViT	14

The selected papers can be organized into three categories based on their focus and citation trends. First, foundational works like Grad-CAM [12], CAM [16], and LRP [2] were included, which were initially developed for CNN-based architectures and have high citation counts due to their contributions. Second, papers like [1], which focus on attention mechanisms in NLP-based Transformers, were included as they provide insights into understanding attention flow, a concept directly applicable to ViT. Finally, recent works like [3], [8], [10], and [14] were selected for their specific focus on ViTs. These papers have fewer citations because of their recent publication, but they are highly relevant to this work. The state-of-the-art review by [7] was also included to provide a comprehensive overview of ViT explainability. This categorization ensures a balanced representation of foundational and recent advancements in the field.

5 State-of-the-Art Review

5.1 Attention rollout and attention flow

One of the most intuitive methods to understand the decisions of a ViT model is to look at its attention component. [1] introduces two methods for mixing the attention scores across layers, attention rollout and attention flow. These methods provide insights into how information propagates through the layers of a Transformer, offering a quantitative and interpretable way to analyze attention patterns.

Attention rollout is a technique used to aggregate attention maps across multiple layers of a transformer. The method works by combining the attention matrices from each layer, taking into account the residual connections in the Transformer. Mathematically speaking, we can define attention rollout as in equation 5.1, where I is the identity matrix, representing the residual connections in the Transformer, $A^{(l)}$ is the attention matrix at layer l. The product $\prod_{l=1}^{L}$ aggregates attention weights across all layers.

$$\mathbf{AttentionRollout} = \prod_{l=1}^{L} \left(\mathbf{I} + \mathbf{A}^{(l)} \right)$$

Attention rollout provides a cumulative view of how information is propagated and aggregated throughout the layers. Although intuitive and easy appliable, it does not give a class specific explanation. However it remains one of the most popular benchmarks used for the comparison of such visualization methods.

Attention flow also aims to propagate attention scores through the layers of the transformer, and the idea behind this mechanism is to compute the maximum flow problem through a graph. This method traces the flow of information through the network, providing a more detailed understanding of how attention mechanisms contribute to the model's decisions. The latter method is very slow, which usually leads to it being left out from comparisons with other methods.

Through these methods the authors managed to offer a building block for further explainability methods for transformer based models. At that time, it was pretty unclear how to understand raw attention scores and how to trace the flow of information through the layers of transformers and with their work appeared one of the first and most known methods that offers a quantitative and interpretable result that analyses the attention patterns. These methods, particularly attention rollout represent the foundation of other visualization techniques, which combine attention with other methods to get even more insight into the model.

5.2 Transition Attention Maps (TAM)

TAM[15] is a method which models the information flow in ViTs as a Markov process. A theoretical foundation for Markov chains, including their properties and applications is provided by Norris in [11]. A Markov chain is a mathematical framework used to model stochastic processes, where the probability of transitioning from one state to another depends solely on the current state and not on the sequence of states that preceded it.

In the paper [15] a novel method for explaining the information flow in ViT is being proposed. The authors argue that attention only based methods, like attention rollout, are not a reliable sources of explanations. Instead the attention mechanism can be used to construct an information flow together with other components which lead to more accurate and trustworthy explanations. So they propose the idea of modeling the flow of information as a Markov chain. 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 and residual connections. A specific explanation is obtained by combining the states with integrated gradients, which are used in order to reduce noise and irrelevant features in the explanation. The gradients are obtained with respect to the last attention module. Multiplying the states with these integrated gradients obtains specific class explanations.

Using the Markov chain model, the authors analyze how information propagates through the layers of the ViT. This includes identifying key layers where information is aggregated or transformed. The analysis provides insights into the role of each layer in the model's decision-making process. The results of the information flow analysis are visualized as a heatmap, highlighting the most important paths and layers for information propagation.

5.3 Class activation map(CAM)

Firstly proposed as a solution for regularization training for CNN based models, CAM[16] became quite popular for the representations of the focus of CNN models that it exposed. The method proposes to compute a linear combination between the weights and feature maps from the last layer of the model.

The limitation of CAM was that it requires a specific CNN model architecture, needing a Global Average Pooling layer right before the final classification layer. This limitation opened the door to another problem, more exactly the method has much potential in the area of providing visual explanations of the decisions of a model, but it restricts users to use a specific architecture. Can the method be extended so that it overcomes it's architectural limitations? This is how a much more flexible method, GradCAM[12] came along. It keeps the idea of using the feature maps of the model, but combines them with the gradients of the target class on eny convolutional layer. The gradients are globally averaged to produce weights that reflect how important each feature map is for the class prediction.

Both works are really relevant for the domain of visual explanations but not so much for the architecture of ViT. GradCAM is offering enough flexibility and has the potential to be applied also on ViT based models. Attention guided CAM(AG-CAM)[8] is extending the GradCAM method in the field of ViT. AG-CAM is selectively aggregating gradients propagated from the classification output to each self-attention layer. The method gathers contributions of image features from various locations in the input image. Additionally, these gradients are refined using normalized self-attention scores, which represent pairwise patch correlations. These scores enhance the gradients with patch-level context information identified by the self-attention mechanism.

$$AG - CAM = \sum_k \alpha_k^c \cdot A_k(x, y) \cdot Att_k(x, y)$$

The equation 5.3 explains the formula behind the AG-CAM method. $\cdot Att_k(x, y)$ represents attention map from the k-th attention head, representing the model's focus at spatial position (x, y). The attention map is being combined with the feature map activations from the transformer layer, $A_k(x, y)$. Both of these are further combined with α_k^c which represents the global average of gradients with respect to the class c. This method generates a class-specific attention map from a ViT to visualize the regions of an input image that contribute most to a model's prediction. This formula enhances traditional Grad-CAM by weighting the feature maps not just with gradients, but also with the model's internal attention maps, making the explanations more aligned with the model's reasoning process.

The AG-CAM method is an appropriate method to provide visual explanations for several reasons. Since ViTs inherently use attention to process inputs, AG-CAM usesthis mechanism to generate explanations that are naturally aligned with the model's internal workings. Attention maps often capture global context, while feature maps capture local patterns. AG-CAM combines both, offering more insight to visual explanations. AG-CAM can provide class-specific explanations by weighting attention maps based on the target class, something original CAM or standard attention rollout doesn't do.

5.4 Layerwise Relevance Propagation (LRP)

Initially adressed for explainability problems in deep neural networks, the work done by Bach et al.[2] introduces LRP as a method for explaining the decisions of non-linear classifiers, particularly deep neural networks. The authors propose LRP as a method to attribute the model's output to its input features in a way that is computationally efficient. The method propagates the model's output back through the network to the input pixels, assigning a relevance score to each pixel. These relevance scores indicate how much each pixel contributed to the model's decision.

LRP can be adapted to propagate relevance through the self-attention mechanisms, making it fit for ViT explainability, as seen in [8], in which a class specific

LRP based visualization technique has been proposed that uses LRP based relevance to calculate scores for each attention head in every layer of the model. Comparing to other attention visualization methods like attention flow or CAM based methods, which only shows where the model "looks" this approach explains why the model makes certain predictions by quantifying the contribution of each input token to the final decision.

$$LRP = \prod_{l=1}^{L} \mathbf{A}^{(l)} \cdot \mathbf{R}^{(L)},$$

There are three main steps through which this method computes its heatmap. Firstly compute the gradients of the model's output with respect to the attention weights. This captures how small changes in attention affect the prediction. Secondly propagate the model's output back to the input, assigning relevance scores to each input token, to make sure that the contributions of all tokens are accounted for. Lastly combine the attention gradients and the LRP. The attention gradients are used to modulate the relevance scores obtained from LRP. This ensures that the explanation reflects both the model's attention and the importance of each token for the final prediction. The final relevance scores are visualized as heatmaps, highlighting the most influential regions of the input.

These steps are described by equation 5.4, in which L is the total number of layers, $R^{(L)}$ is the LRP relevance score at layer L and $A^{(l)}$ are the weighted gradients with respect to the attention weights. The process of computing the relenace score is repeated layer by layer, starting from the output layer and propagating back to the input layer. The final relevance scores at the input layer provide a heatmap that highlights the most influential image patches for the model's decision.

This method [8] provides explanations that go beyond simple attention maps, offering insights into the model's decision-making process, the author managed to extend LRP to work with Transformer architectures and because attention is being combined with LRP it produces more accurate results.

5.5 Causal explanations

The concept of causal explanations in machine learning aims to identify cause-and-effect relationships rather than mere correlations. A key paper for this concept is [13], which introduces the idea of counterfactual explanations, a form of causal reasoning that has been applied to explain deep learning models. The main point of these kind of explanations is being described by the following question: "What would the model's output be if the input were changed in a specific way?"

A novel approach on ViT models regarding causal explanations has been proposed in [14], which provides ViT-CX, a framework for causal explanation of ViT(s). Unlike the previously described methods that focus on correlation,

ViT-CX aims to identify causal relationships between input features and model predictions. This is achieved by using interventional techniques to measure the causal effect of individual image patches on the model's output.

The ViT-CX framework consists of three key steps. The first step is building a causal graph. A causal graph is constructed to represent the relationships between input patches, intermediate representations, like patch embeddings and the final prediction. This graph captures how changes in one part of the input causally affect the model's output. Let $X = \{x_1, x_2, \ldots, x_N\}$ be the input image patches, $Z = \{z_1, z_2, \ldots, z_N\}$ be the be the intermediate representations and y be the model's output. The causal graph G encodes the relationships between these variables, such as $x_i \rightarrow z_i$ each input patch influences its corresponding intermediate representation and $z_i \rightarrow y$ the intermediate representations collectively influence the output.

The next step is that of performing interventional experiments. Interventions, like modifying specific patches, are performed on the input patches to measure their causal effect on the model's prediction. These experiments help quantify the causal contribution of each patch.

Lastly, using the results of the interventional experiments, causal attribution scores are computed for each input patch. The causal effect of each patch x_i is aggregated to compute a causal attribution score. These scores indicate how much each patch causally influences the model's prediction and are visualized as a heatmap.

$$\text{Causal Attribution} = y(\text{do}(x_i = \tilde{x}_i)) - y(\text{do}(x_i = x_i)),$$

ViT-CX introduces a novel causal perspective to explainability, complementing the attention-based and gradient-based methods explored in this work. The framework is specifically designed for Vision Transformers, making it highly relevant for understanding how ViTs process visual information. The use of interventional techniques ensures that the explanations are grounded in causal relationships. The causal attribution scores provided by ViT-CX can be used to generate interpretable visualizations, helping to explain how ViTs make decisions. All these reasons make ViT-CX an appropriate candidate to be studied and compared to other methods in this report.

5.6 Theoretical Comparison

Five different types of visualization methods for explainability in ViT models have been presented. They all try to give insightful details on the decisions of the model but the ways in which they do this differ. Firstly let us explore what they all share in common and then let us underline their differences. The method of attention flow has been left out of this comparison, because even though from a theoretical point of view it provides a good result, from a practical point of view is costly and from all the researched papers is has been left out of the comparisons, so this paper follows the same pattern. While attention flow has its theoretical merits, the complexity and needed resources required to implement

it often make it impractical for real-world applications, in which efficiency is critical.

The first common point between all of the methods is that they do not require the modification of the models. All methods operate on pre-trained ViTs without needing architectural changes. They analyze internal representations like attention weights, activations, or gradients. This characteristic makes these methods highly adaptable, as they can be easily integrated into existing workflows without requiring retraining or adjustments to the underlying model. Secondly, they are all post-hoc techniques, meaning they explain decisions after the model has made a prediction, rather than influencing the model during training. Post-hoc techniques are essential because they provide an interpretability layer for models already deployed in real-world applications, helping to clarify decisions made by AI systems without needing to alter their operation or retrain them.

Additionally, each of these methods has its own way of tracing the model's decision-making process. These variations reflect the diversity of approaches in XAI. Some methods may provide more fine-grained explanations by focusing on specific image regions, while others might offer a global perspective of how different patches interact to form the model's output.

Method	**Think of it as**	**Example**
Attention Rollout	A map showing how attention spreads across the model	Highlights both the cat's ears and whiskers, showing the model focused on these parts when predicting "cat"
TAM	A probability map showing how likely information is to move between regions	Shows that information from the cat's eyes has a high probability of influencing attention to the ears in deeper layers
CAM	A heatmap highlighting important regions for a specific class	The heatmap glows over the cat's face, indicating it was crucial for predicting "cat"
LRP	A pixel-level importance map showing which individual pixels contributed most	Pinpoints specific pixels around the cat's eyes and whiskers as key contributors to the prediction
ViT-CX	A cause-and-effect map showing which parts of the image directly influenced the output	Determines that removing the cat's ear causes the model to misclassify the image, proving the ear's causal importance

Table 2. Comparison of Explainability Methods for ViTs

The explainability methods for ViTs differ in terms of what aspects of the model's decision-making process they highlight. The table 2 explains what makes each method unique by using easy to understand analogies and examples. The column "Think of it as" simplifies complex concepts by using analogies, while the other column offers practical examples to illustrate how each method highlights different parts of the input.

Attention Rollout explains the global flow of attention across layers, highlighting which parts of the input the model attended to overall. TAM take this

a step further by modeling the probabilistic flow of information between tokens using Markov chains, capturing how likely it is for information to transition from one part of the image to another within the model.

On the other hand, methods like CAM and LRP provide more input-focused explanations. Generally speaking, gradient based methods explain the contribution of the image features elicited through multiple layers, while LRP based methods capture the contribution of the independent pixels to the classification output[8]. CAM highlights image regions that are most important for a specific class prediction, generating heatmaps that localize key areas influencing the decision. LRP offers a more fine-grained analysis, attributing relevance scores to individual pixels to indicate their contribution to the model's output.

Finally, Causal Explanations with ViT-CX go beyond correlation-based methods by identifying input regions that causally impact the model's decisions. This method perturbs parts of the input to determine which regions truly cause changes in the output, providing more robust insights into the model's reasoning.

Overall, while attention-based methods such as rollout and TAM help understand how information moves through the model, LRP- and CAM-based methods focus on which parts of the input are important, and causal methods like ViT-CX reveal why specific input regions matter by establishing direct cause-effect relationships.

6 Experiments

6.1 Dataset

The Pascal Visual Object Classes (VOC) dataset has been used to test the potential of the visualization methods for segmentation tasks [5]. It was created by the VOC challange which was organized at the University of Oxford in multiple years, from 2005 until 2012. In this paper the dataset from the year 2012 is being used. The Pascal VOC 2012 dataset is a well known benchmark in visual object category recognition and detection, containing annotated images for tasks such as classification, detection, segmentation, and action recognition. The dataset includes 20 object categories and it is splitted into 2 subsets: a training one with 1,464 images and a validation one with 1,449 images. In this work we use the validation subset.

6.2 Model

In the visual classification experiments, a pretrained ViT-based model has been used. The model adopts a BERT-like architecture. The images used are of size 224x224, the size of the patches are of 16, and they are followed by flattening and linear transformations to generate a sequence of vectors. A classification token is prepended to the sequence and serves as the input for classification tasks. We use the same model as proposed in the LRP[3] method.

6.3 Implementation

Github Repository The implementation of the methods can be found under the following link: https://github.com/anacopo/TSW

The preprocessing of the dataset implies image transformation, resizing, and mask manipulation to prepare the data for ViT model. For the implementation of the methods we use the validation VOC 2012 validation dataset. These methods are applied on an already trained model, so we use the validation dataset only to test our results, with no need of training a model.

The dataset contains images with corresponding segmentation masks that label different objects within the images. The library *torchvision.datasets* offers a way through which to load the VOC dataset automatically. For model compatibility the images are converted from PIL format to PyTorch tensors.

The image as well as the mask need to be manipulated accordingly to the model. The pixel values of the image are being converted to [0,255] to improve visualization. The segmentation mask is resized to match the original image dimensions and then it is being binarized so that it focuses either on the object or on the non object regions. The image is being resized so that it is compatible with the model.

CAM Implementation For implementing this method, we provide the pseudocode with the most relevant steps that need to be followed. The most important aspect you need to know in order to implement this is how to retrieve the gradients and the attention map out of the model. Once you have these you can compute the CAM values by computing the element-wise product of them, averaging it across the batch dimension, and then ensuring all the resulting values are non-negative by applying a lower bound of 0.

Algorithm 1 CAM Pseudocode

Require: Model M, Input Image I, (Optional) Target Class Index c
  Step 1: Get Model Prediction ▷ Forward pass through the model
  Step 2: Select Target Class
  if c is not provided **then**
    $c \leftarrow \arg\max(O)$ ▷ Choose class with highest score
  end if
  Step 3: Create One-Hot Tensor for Target Class
  Step 4: Compute Gradients with Backpropagation
    ▷ This represents a score for target class
  Step 5: Obtain Gradients and Attention Maps
    ▷ Gradients of attention layers ▷ Attention maps from last layer
  Step 6: Compute the CAM Heatmap

LRP Implementation The LRP method attributes the model's output back to the input features by propagating relevance scores through the layers of the transformer. The implementation follows the steps out of the given pseudocode. Specific for this function is the use of relevance propagation, more exactly for the implementation it is needed to know which is the specific function of the model that you need to use. The relevance scores are propagated from the output layer back to the input tokens.

Algorithm 2 LRP Pseudocode

Require: Model M, Input Image I, (Optional) Target Class Index c
Step 1: Get Model Prediction ▷ Forward pass through the model
Step 2: Select Target Class
if c is not provided **then**
    $c \leftarrow \arg\max(O)$ ▷ Choose class with highest score
end if
Step 3: Create One-Hot Vector for Relevance Propagation
Step 4: Backpropagate Relevance Through the Model

Common Funtionalities Both CAM and LRP require certain preprocessing steps and visualization utilities to convert the raw attribution data into interpretable heatmaps. Key components include normalization of the results to $[0, 1]$ range to enhance contrast in visualizations and a utility function which overlays the heatmap on the original image.

6.4 Results

Visual comparisons The first part of the evaluations focus on visual comparisons of generated heatmaps, emphasizing their collective ability to offer detailed insights into the reasoning behind ViT. Our evaluation criterion for explanation quality centers on how well explanation maps align with human observations of images in visual recognition tasks. Good explanations often align with human expectations, meaning that we expect the model to focus on semantically relevant regions. This was found effective in [10] [3] [8] and was proposed in [6].

The results can be seen in Figure 4, where you can clearly see that the attention of the model lands exactly on the objects which are recognized. The first row of the figure shows the original images while the second displays the results of the heatmap generated by the attention guided CAM technique and the third one the results of LRP. As for the images, the exact index from the dataset for each of them is being provided in the figure.

For the first picture with the train, both methods seem to highlight mostly the same broader area of the train. As for the sheep image, both of them highlight the entire body of the sheep, seeming to highlight the wool texture. For the bird picture both methods show some spill-over into the surrounding background,

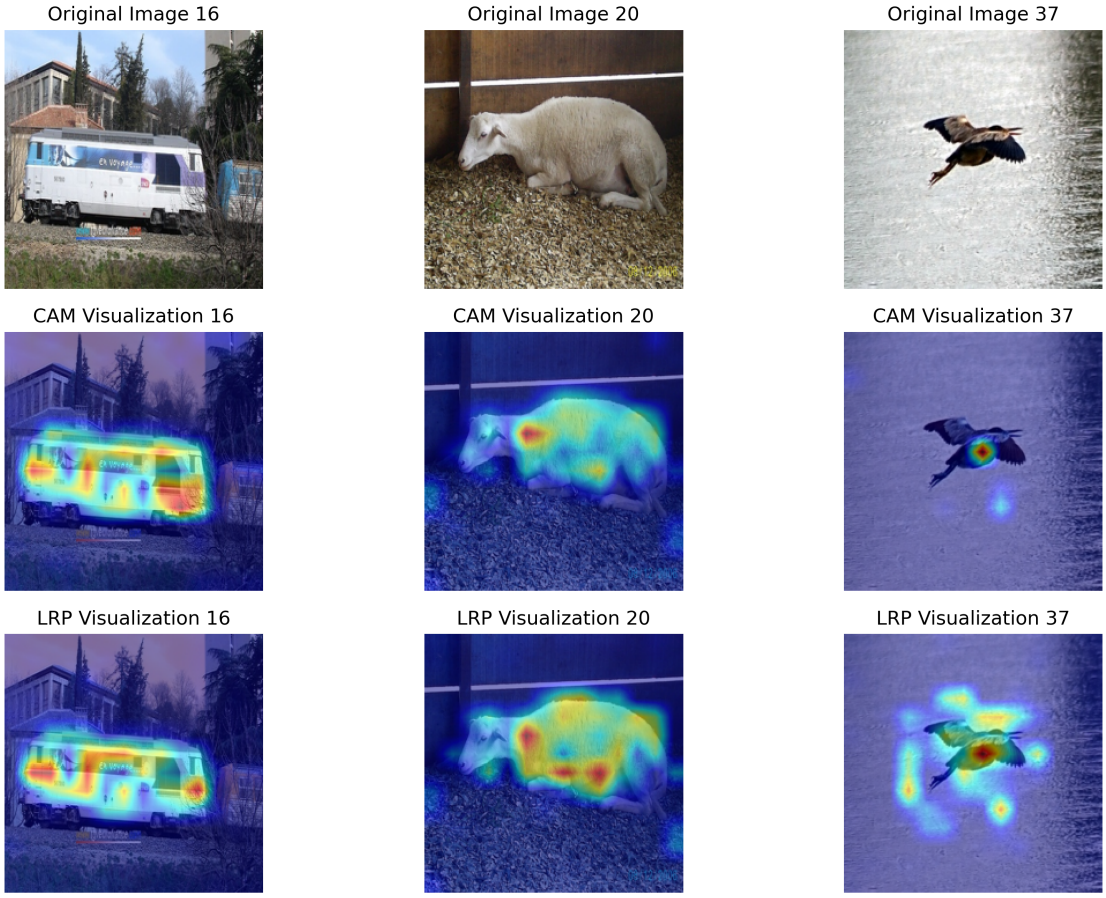

Fig. 4. Heatmaps of attention guided CAM and LRP visualization techniques

whereas CAM seems to be centered on the bird's torso, while LRP offers more relevant spots.

Evaluation Metrics In addition to directly evaluating the generated heatmaps, we employed specific metrics aimed at assessing the performance of the explainability methods. Firstly we look at the results given by some important segmentation metrics after which we also analyze explainability specific metrics. This part of the evaluation has been done using evaluation metrics. The tests were performed on a subset of Pascal VOC, where the predict probability for the main class was higher than 85%. Such measures were needed as our model was pretrained on ImageNet21k, and we had to ensure the model found the relevant parts in the tested images.

Table 3. Average Segmentation Metrics. Threshold set to 0.85 accuracy minimum.

Method	Jaccard Index	F1 Score	Pixel Accuracy
CAM	4.35	7.49	66.49
LRP	14.28	22.96	70.35

In Table 3, the performance of the CAM and LRP method is evaluated using three metrics: Jaccard Index, F1 Score, and Pixel Accuracy. For computing these metrics we used the true masks given by the dataset which we compared with the heatmaps generated by the methods. For all these metrics the ranges vary from 0 to 1 with higher values showing better performance. The Jaccard Index Measures the overlap between the predicted mask and the true mask. The F1 score shows a balance between precision and recall for the predicted mask. The Pixel Accuracy score measures the proportion of correctly classified pixels. The Jaccard Index and F1 Score are very low for both methods indicating poor overlap between generated mask and the true mask, even though it is noticable that the values are higher for LRP. These low scores for both methods suggest that while the models highlight some important areas, they struggle with precise localization. Pixel Accuracy is moderately high for both of the methods with LRP having better results then CAM. A Pixel Accuracy of approximately 70 % shows that 70% of the pixels in the prediction match the true values. While this indicates that the method correctly generates a majority of the pixels, the accuracy is not particularly high, suggesting room for improvement in correctly identifying all relevant regions. Overall we can say that LRP performs better than CAM.

Table 4. Average Explainability-Specific Metrics.

Method	Deletion AUC	Pointing Game Accuracy
CAM	24.7	44.24
LRP	12.4	71.98

The results from the CAM and LRP methods on explainable specific evaluation metrics from table 4 show the same differences in performance between methods as the segmentation metrics. The CAM method, in terms of pointing game accuracy, achieved a value of 0.4424, while the LRP method outperformed it with an accuracy of 0.7198. This suggests that the LRP method is significantly more effective in identifying and classifying points correctly in comparison to CAM, where a higher accuracy is typically preferred, closer to 1.0. Regarding the deletion AUC, LRP performed again better with AUC of 12.4 whereas CAM had a higher AUC of 24.7. A lower AUC value indicates that the explanation method is effective at identifying the most important regions of the input. When these regions are deleted, the model's performance drops quickly, resulting in a smaller AUC.

7 Conclusions and Future Work

In this report, we explored several state-of-the-art methods for explaining ViTs. attention rollout provides a cumulative view of attention across, TAM extends attention attribution methods offering a more complete information flow through the model, LRP offers a gradient-based approach to quantify the contribution of input features, making it highly interpretable, CAM highlights regions influencing class predictions and finally, ViT-CX introduces a causal framework to explain ViT decisions by identifying causal relationships between input patches and predictions.

We successfully implemented LRP and CAM to generate heatmaps that visualize the regions of the input image most influential to the model's predictions. Our visual comparisons demonstrated that both methods effectively highlight semantically relevant regions. While both methods showed alignment with human intuition, LRP provided more precise and relevant spots compared to CAM. Quantitative evaluation using segmentation metrics and explainability specific metrics revealed that both methods struggle with precise localization. However, LRP consistently outperformed CAM, demonstrating a better overlap with ground truth masks, indicating its effectiveness in identifying critical regions.

In conclusion, while both LRP and CAM provide valuable insights into the decision-making process of ViTs, LRP seems to be the more robust and precise method for generating explanations.

Future work could focus on improving localization accuracy and extending these methods to more complex datasets from various domains. Another possible future direction is exploring a combination of multiple explainability methods. Each method has its strengths and by integrating them together it may be possible to create more comprehensive explanations. Such hybrid methods could address the limitations of individual techniques, such as poor localization or lack of class specificity, and pave the way for more interpretable and trustworthy AI systems.

References

1. Abnar, S., Zuidema, W.: Quantifying attention flow in transformers. arXiv preprint arXiv:2005.00928 (2020)
2. 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 **10**(7), e0130140 (2015)
3. Chefer, H., Gur, S., Wolf, L.: Transformer interpretability beyond attention visualization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 782–791 (2021)
4. 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 preprint arXiv:2010.11929 (2020)
5. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**, 303–338 (2010)
6. Fuad, K.A.A., Martin, P.E., Giot, R., Bourqui, R., Benois-Pineau, J., Zemmari, A.: Features understanding in 3d cnns for actions recognition in video. In: 2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA). pp. 1–6. IEEE (2020)
7. Kashefi, R., Barekatain, L., Sabokrou, M., Aghaeipoor, F.: Explainability of vision transformers: A comprehensive review and new perspectives. arXiv preprint arXiv:2311.06786 (2023)
8. Leem, S., Seo, H.: Attention guided cam: Visual explanations of vision transformer guided by self-attention. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 2956–2964 (2024)
9. Li, Y., Wang, J., Dai, X., Wang, L., Yeh, C.C.M., Zheng, Y., Zhang, W., Ma, K.L.: How does attention work in vision transformers? a visual analytics attempt. IEEE transactions on visualization and computer graphics **29**(6), 2888–2900 (2023)
10. Mallick, R., Benois-Pineau, J., Zemmari, A.: I saw: a self-attention weighted method for explanation of visual transformers. In: 2022 IEEE international conference on image processing (ICIP). pp. 3271–3275. IEEE (2022)
11. Norris, J.R.: Markov chains. No. 2, Cambridge university press (1998)
12. 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 IEEE international conference on computer vision. pp. 618–626 (2017)
13. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harv. JL & Tech. **31**, 841 (2017)
14. Xie, W., Li, X.H., Cao, C.C., Zhang, N.L.: Vit-cx: Causal explanation of vision transformers. arXiv preprint arXiv:2211.03064 (2022)
15. Yuan, T., Li, X., Xiong, H., Cao, H., Dou, D.: Explaining information flow inside vision transformers using markov chain. In: eXplainable AI approaches for debugging and diagnosis. (2021)
16. 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. pp. 2921–2929 (2016)