ORIGINAL ARTICLE

Multiplex network-based representation of vision transformers for visual explainability

Michele Marchetti[1] · Davide Traini[1,2] · Domenico Ursino[1] · Luca Virgili[1]

Received: 7 November 2024 / Accepted: 12 August 2025 / Published online: 6 September 2025
© The Author(s) 2025

Abstract
The enormous growth of artificial intelligence (AI), and deep learning (DL) in particular, has led to the widespread use of these systems in a variety of contexts. One DL model capable of addressing complex computer vision tasks is the vision transformer (ViT). Despite its huge success, the reasoning behind the inferences it makes is often unclear, which poses significant challenges in critical scenarios. In this paper, we propose a new approach called MUltiplex Transformer EXplainer (MUTEX), which aims to explain the inferences made by ViTs. MUTEX combines multiplex network-based representations of attention matrices and mask perturbation approaches to provide insight into the inference process of ViTs. By mapping the attention layers of a ViT into a multiplex network, MUTEX is able to analyze the relationships between different parts of the input image and identify the image patches that most influence the inference process. We tested MUTEX on a subset of ImageNet and on BloodMNIST and compared its performance with that of existing visual explainability approaches. In addition, to assess the robustness and adaptability of MUTEX, we conducted a qualitative analysis, along with a hyperparameter and ablation study, which allowed us to further appreciate its potential in visual explainability of ViT.

Keywords Visual Explainability · Vision Transformer · Attention Mechanism · Multiplex Network · Computer Vision

1 Introduction

Artificial intelligence (AI) has experienced tremendous growth, with deep learning (DL) attracting considerable attention from both researchers and industries [1, 2]. Among DL models, transformers have achieved remarkable success. Originally introduced for natural language processing (NLP) [3], transformers use attention mechanisms to identify key input features. This model has been adapted to computer vision, resulting in vision transformers (ViTs) [4], which have demonstrated outstanding performance in several tasks. However, like most DL models, ViTs operate as black boxes where the internal reasoning behind the predictions remains unclear. This lack of explainability poses challenges in critical domains, such as healthcare [2] and autonomous systems [1], where understanding the rationale behind decisions is critical. For this reason, researchers have proposed many approaches to address this issue.

In the literature, existing approaches can be categorized based on their underlying mechanisms. Attribution-based approaches modify some parts of the input data to describe the model inference process [5–7], but struggle to capture nonlinear relationships [8]. Attention-based approaches use the attention mechanisms of the transformer to understand the inferences made by the model [9–11]. However, they have been shown to be unreliable

due to the weak correlation between attention weights and model outputs [12]. Interpretability-aware approaches modify the original architecture to create new models that are inherently interpretable [13–15]; their way of proceeding makes them unable to work on pre-trained ViTs and suitable only for specific settings. Finally, mask-based approaches selectively obscure parts of the input data and observe the resulting effects on model output [16–18]. They could suffer from a high computational load, depending on the number of masks used to explain a ViT output. To the best of our knowledge, there is no approach that is able to perform ViT explainability by mask perturbations, attention matrices, and relationships between parts of the input simultaneously. The approach we propose in this paper, called MUltiplex Transformer EXplainer (MUTEX), aims to fill this gap.

By integrating key ideas from different approaches, MUTEX has several advantages over existing solutions. It effectively addresses the nonlinearity in patch relationships by constructing a multiplex network on top of the attention matrices, capturing patch interactions across different layers and modeling the information flow within the ViT. Unlike attention-based methods that rely solely on raw attention weights, MUTEX generates a set of masks that are then fed into the ViT. This step is critical because it establishes a direct link between attention weights and model output, mitigating potential problems associated with the lack of correlation between attention and model predictions. Based on these inferences, MUTEX identifies salient areas and constructs the final heatmap.

Specifically, MUTEX provides visual explainability for the inferences made by a ViT. It performs this activity in two tasks. In the first task, it maps the attention layers of a ViT onto a multiplex network-based representation that highlights the relationships between parts of the input image. In the second task, it uses the resulting multiplex network to identify the most important patches based on centrality measures and aggregation functions, and then constructs a set of masks corresponding to specific parts of the image. Finally, it merges these masks using the ViT model's class confidence score and coverage bias formula [16].

We evaluated the performance of MUTEX on two datasets (i.e., a subset of the ImageNet validation set [7, 9, 16–19] and BloodMNIST) and on two Vision Transformer architectures (i.e., ViT and DeiT). Our results show promising performance compared to state-of-the-art approaches based on well-established visual explainability and object localization metrics. Furthermore, we found that MUTEX has a lower computational cost than other mask-based approaches; it maintains comparable efficiency to alternative solutions, but with superior performance. We also made a qualitative analysis to visually evaluate the capability of MUTEX on three examples. Finally, we performed a hyperparameter and ablation study to highlight the impact of hyperparameters and model design on the performance of MUTEX.

In summary, the main contributions of this paper are as follows:

- we introduce a multiplex network-based representation for ViTs, providing a novel structural perspective on their attention mechanisms;
- we propose an algorithm that leverages the multiplex network to generate heatmaps, effectively assessing the importance of each input region;
- we conduct a comprehensive experimental evaluation, demonstrating the effectiveness and efficiency of our approach across different datasets and ViT architectures.

This paper is organized as follows: In Sect. 2, we describe related work that addresses the issue of visual interpretability of ViTs. In Sect. 3, we present our approach to map a ViT onto a multiplex network and generate a heatmap capable of visually explaining the inferences of the ViT on an image. In Sect. 4, we illustrate the experiments we conducted to validate our approach. In Sect. 5, we describe its advantages and limitations. Finally, in Sect. 6, we draw our conclusions about our work and look at some possible future developments.

2 Related work

With the evolution of deep learning, transformer models [3], originally designed for NLP, are rapidly spreading to the field of computer vision, outperforming more traditional models such as CNNs in many tasks [4]. This transition of the field of computer vision to vision transformers makes the issue of visual explainability even more prominent. Researchers are trying to address this problem and, as mentioned in the Introduction, have proposed four main strands of approaches, namely attribution-based, attention-based, interpretable-aware, and mask-based. In the next subsections, we will examine these four strands of research and then propose some considerations on how MUTEX fits in relation to them.

2.1 Attribution-based methods

Attribution-based methods aim to identify the parts of the input data most influential in the decision-making process of the model under consideration [20]. These methods, traditionally applied to CNNs [5, 6], have been adopted in the context of ViTs [21, 22]. They provide insight into the features and patterns that appear to be most relevant to the models being evaluated. Several methods in this family, including GradCAM [5], GradCAM++ [23], ScoreCAM [24], integrated gradients [6], and its variant SmoothGrad [25], have found direct application in the context of ViTs [26]. For example, GradCAM uses gradient information, which flows into the final convolutional layer, to generate a heatmap highlighting the most important regions for predictions. Similarly, GradCAM++ uses a weighted combination of positive partial derivatives of feature maps to ensure accurate object localization and the ability to explain multiple instances of the same class in a single image. ScoreCAM, instead, uses the forward-passing score of each activation map of the target class to determine its weight; it then combines the resulting weights with the activation maps to generate class activation explanations without relying on gradients. Similarly, integrated gradients is a path-based approach that calculates the contribution of each input feature to the model output by integrating gradients along the path from a baseline to the current point.

More complex approaches, such as Shapley additive explanation (SHAP) [27] and layer-wise relevance propagation (LRP) [28] require some adaptations to be applied to ViTs. [21] adapts Shapley scores to ViT models. Similarly, the authors of [29] use Shapley values to compute the importance of each token in a transformer by decoupling self-attention and freezing unrelated values to linearize the model; this allows efficient and accurate computation of token contributions without relying on gradients. By adopting an attention masking strategy and creating a learned explainer model, it facilitates the production of Shapley score explanations for ViTs. Instead, [22] uses LRP to identify critical attention heads within the Transformer architecture. Through their analysis, the authors show that only some of these attention heads are essential because they carry out specialized and interpretable functions that are critical to the performance of the model. [7] combines LRP-based relevance computation for each attention head across all layers with a relevance propagation rule that considers both positive and negative attributions. By integrating attention with relevance score and iteratively filtering out negative contributions, it obtains class-specific visualizations for attention models.

2.2 Attention-based methods

ViT architectures, characterized by the use of attention mechanisms, have remarkably stimulated the adoption of attention-based explainability methods in computer vision. This enormous interest is largely due to the ability of these methods to provide a way to intuitively understand how models make their inferences [30].

The attention flow and rollout methods introduced in [9] provide deep insights into how information flows within ViTs. The “attention rollout” method models information flow using a directed acyclic graph and shows how the identities of input tokens are combined across layers based on attention weights. Meanwhile, the “attention flow” method views the attention graph as a flow network and applies the maximum flow algorithm to

quantify the contributions of input tokens. These methods, initially designed for NLP transformer models, have been adapted to ViTs, thus extending their application to the context of computer vision.

[10] proposes Grad-Sam, which uses attention mechanisms that focus on gradients of attention scores. The goal of Grad-Sam is to understand how changes in the model focus on specific inputs affect the model's output. To this end, it examines attention mechanisms within transformer models to identify which parts of the input data are most critical to the model's decision-making process. Another approach relevant to the visual explainability of ViTs is transition attention maps (TAM), introduced in [31]. It models the information flow in ViTs as a Markov process and maps the progress of information based on the hidden states of tokens across layers, which evolve through Transformer operations. TAM treats weights as transition matrices within a Markov chain, allowing top-down propagation of information. [32] presents an interpretation framework for estimating the contribution of individual tokens within transformer models. The framework operates via a two-step process that incorporates attention perception and reasoning feedback based on the partial derivatives of the loss function with respect to each token.

[33] introduces attentive class activation tokens (AttCAT), an approach for generating token-level explanations using features, gradients, and self-attention weights. AttCAT quantifies the influence of each token on the class-specific output using gradient information and self-attention weights. The authors further decompose the information flow within the transformer for a given token into two components, namely the intrinsic information of the token and the information about token interactions represented by the self-attention mechanism. In this way, AttCAT calculates impact scores from multiple layers to derive the final explanation. [11] proposes TokenTM, which recasts the ViT layers as weighted linear combinations of input and transformed tokens. To assess the effects of the transformations performed by the ViT model on the input tokens, the authors introduce a measure that focuses on two attributes, i.e., length and direction. TokenTM integrates transformation effects with attention weights to evaluate token contributions within each layer. It aggregates these contributions across all layers and generates a comprehensive contribution map showing token contributions throughout the model.

In [34], the authors introduce the weighted relevance accumulation approach, which aims to improve ViT explainability by adaptively balancing attention and residual connections in relevance mapping. Unlike approaches like attention rollout that uniformly average relevance, the weighted relevance accumulation approach dynamically weighs the importance of tokens using gradients and activations, thereby addressing gradient saturation and relevance bias. Saliency maps visualize token relevance for tasks like Visual Question Answering and image captioning, with relevance scores aggregated for the CLS token or captions to improve decision visualization.

The authors of [35] introduce the grounding everything module (GEM), an approach to explainability in vision-language transformers based on self-attention. GEM extends traditional value-value attention with query–query and key–key attention, clustering semantically similar visual tokens while aligning with text embeddings. Saliency maps are generated by computing the cosine similarity between vision encoder patch tokens and text embeddings, and heatmaps highlight image regions relevant to the query.

Finally, in the biometric domain, the authors of [36] use ViTs to perform vein biometric recognition. Their approach relies on self-attention mechanisms to generate saliency maps that highlight image regions most relevant to model decisions. Saliency maps are obtained by averaging attention weights from multiple heads of the final transformer layer and overlaying them on the input image.

MUTEX shares some similarities with the attention-based approaches as they all derive heatmaps from the attention maps of the transformer layers [37]. For instance, similar to AttCAT, MUTEX considers the interactions between tokens across all layers by using the self-attention weights, but models these interactions by means of a multiplex network. The latter allows for a more nuanced understanding of the relationships between tokens. In addition, similar to TokenTM, MUTEX merges the token contributions from all layers without introducing a tailored aggregation framework, but using simple aggregation functions to summarize the token contributions across the multiplex network. A key difference between MUTEX and the aforementioned methods lies in the fact that some of the latter rely solely on the classification token (CLS) values to determine the importance of image

patches, thus overlooking the insights provided by the attention values of the tokens themselves. Instead, MUTEX does not use these values directly, but to construct a multiplex network from which it generates the heatmaps that represent the importance of different patches. This feature gives MUTEX a significant advantage because, thanks to it, MUTEX can be applied to ViT models not using the CLS token (e.g., SimpleViT [38]).

Finally, we feel obliged to conclude this section by pointing out that although the above approaches are promising and widely discussed in the literature, some researchers debate whether using attention can provide accurate explanations for transformer models. [12] shows that different attention matrices can sometimes lead to the same output, raising concerns about the reliability of attention weights as an explanatory tool. This issue is also highlighted in [39], where the authors argue that higher attention weights are generally correlated with greater impact on model predictions, but there are many cases where this correlation does not hold. To address this issue, [33] uses a complex approach to compute saliency maps by leveraging encoded features, their gradients, and attention weights. Actually, MUTEX does not solely use attention weights to generate the saliency map but employs them to generate a set of masks. These masks are then weighted based on their relevance in predicting the correct class, ensuring that masks not highlighting the object to be classified have a low influence. Finally, some approaches use attention weights to prune tokens and improve the efficiency of ViTs [40]. Among them is ATSViT [41], where, for each transformer layer, the probability of retaining a specific token is directly proportional to the attention weight of the CLS token toward it. Such approaches demonstrate that, under the right conditions, attention can be used to assess the importance of individual tokens, showing that MUTEX, while using attention weights, addresses common issues and provides robust and interpretable representations.

2.3 Interpretable-aware methods

Other work focuses on the explainability of ViTs by developing models that can intrinsically provide explanations for their results. These approaches introduce variations on the original architecture aiming to integrate interpretability mechanisms directly into the model design in order to support the extraction of insights from ViT processing. They are defined as "ante-hoc" because they add interpretability methods directly into the model's structure and training process.

For instance, [13] proposes ConceptTransformer, a DL module that provides explanations of a model's output considering user-defined concepts. It consists of the generalization of attention from low-level features to high-level concepts to ensure the interpretability of attention scores. The ConceptTransformer's explanations are evaluated based on plausibility and faithfulness. Plausibility refers to how convincing the explanations are to human users and is achieved by training attention heads to represent known relationships between inputs, concepts, and outputs, as described by domain knowledge. Faithfulness measures how well the explanations match the model's reasoning process and is achieved by establishing a linear relationship between the transformer's value vectors representing the concepts and their contribution to the classification output. [14] introduces a new ViT neural tree decoder called ViT-NeT. In this architecture, a ViT serves as the backbone of the model, and its output is processed using a neural tree decoder called NeT. Specifically, a NeT aims to classify fine-grained objects by using similar inter-class correlations and differing intra-class correlations. Then, it leverages a tree structure to describe the decision process, providing a visual interpretation of the results accessible to humans. The authors then find a way to balance the tradeoff between performance and interpretability. [15] introduces the interpretability-aware ViT (IA-ViT) model. The core idea of IA-ViT is that both the CLS token and the image tokens consistently generate predicted distributions and attention maps. The IA-ViT model consists of a feature extractor, a predictor, and an interpreter that are trained together using an interpretability-aware training objective. As a result, the interpreter mimics the behavior of the predictor and retrieves an accurate explanation through its single-head self-attention mechanism.

In [42], the authors propose ICEv2 (ICEv2: Interpretability, Comprehensiveness, and Explainability in Vision Transformer), which aims to perform ViT explainability through patch-wise classification and adversarial normalization to label patches. ICEv2 labels patches as foreground or background using patch embeddings. By fine-

tuning the last three encoder layers, ICEv2 scales the visualization of class-relevant regions without altering the model structure. As an interpretable-aware approach, ICEv2 directly predicts classes and separates foreground from background, thus providing intuitive, human-understandable explanations. [43] presents an Interpretability-Aware REDundancy REDuction framework (IA-RED^2). The insight behind IA-RED^2 is that the authors observed a large amount of redundant computation on uncorrelated patches. These computations are dynamically pruned by an interpretable module, which leads to the introduction of a hierarchical training scheme for pre-trained ViTs. This scheme removes uncorrelated tokens at different stages, resulting in a significant reduction in computational cost while sacrificing some accuracy. Overall, IA-RED^2 retrieves a human-understandable interpretation, while providing a lighter model. [44] proposes B-cos-ViTs, which aim to provide comprehensive explanations for ViT outputs. The authors model each ViT component, such as the multilayer perceptrons, attention layers, and tokenization module, as dynamic linear layers. This modeling approach enables them to represent the transformer's processing as a single linear transformation. The authors evaluate B-cos-ViT on ImageNet, showing interpretable outputs while maintaining good performance. [45] introduces the eXplainable Vision Transformer (eX-ViT), an intrinsically interpretable transformer model. eX-ViT consists of three components, namely: *(i)* the Explainable Multi-Head Attention (E-MHA) module, *(ii)* the Attribute-guided Explainer (AttE) module, and *(iii)* a self-supervised attribute-guided loss. E-MHA is designed to generate explainable attention weights that learn semantically interpretable representations from local patches. AttE encodes distinctive attribute features for the target object by discovering different attributes. The self-supervised attribute-guided loss enhances representations allowing the model to localize diverse and discriminative attributes while providing more robust explanations.

Unlike interpretable-aware ViTs, which are ante hoc methods constructed to be interpretable by design, MUTEX is a post hoc method that provides interpretability after the model has been trained. This distinction leads to different application contexts. In an ante hoc scenario, a variation of a ViT model is trained to intrinsically produce interpretable outputs. However, as noted by [43], there is often a tradeoff between accuracy and interpretability because the inclusion of interpretable modules within a ViT architecture could degrade classification performance. MUTEX, on the other hand, does not affect model performance because it neither adds additional parameters nor requires retraining of the ViT model. This allows it to provide interpretability without interfering with the model's architecture.

2.4 Mask-based methods

Mask-based methods selectively obscure portions of the input data (e.g., pixels of patches in images, words in texts) and observe the resulting effects in the model's output. This process helps identify which elements of the input are most influential in driving model predictions, thus providing insights into model behavior. This category of methods has evolved over time to facilitate visual explanation of various DL models. Currently, there are both adaptions of earlier methods to ViTs, such as Randomized Input Sampling for Explanations (RISE) [18], and new methods, such as ViT-CX [17] and TIS [16], that allow for improved explainability of complex systems.

Specifically, RISE enables explainability by applying random masks to input images, obscuring different sections each time, and observing how these changes affect model predictions.

A mask-based method developed directly for visual explainability of ViTs is ViT-CX [17]. It focuses on the causal effects of patch embeddings on model outcomes. It generates and aggregates masks to create saliency maps that reflect the model's decision-making process. Another mask-based method similar to ViT-CX is TIS [16]. It exploits the ability of ViTs to process a variable number of tokens while avoiding the introduction of misleading outlier features, which was a common problem in previous methods. TIS applies perturbations after linear projections and position encoding, but before processing the transformer layer. This ensures that the perturbations do not generate outlier inputs that could distort the interpretation. The authors of [46] propose the Coarse-To-Fine Explainer (C2F-Explainer), which aims to provide explainability in ViTs through a coarse-to-fine strategy, using sequential mask refinement to accurately locate and visualize model-relevant regions while reducing noise and

improving clarity. Finally, R-Cut [47] does not conform to a specific category of explainability, but has characteristics similar to perturbation-based and mask-based approaches. It consists of two main components, namely the Relationship Weighted Out (R-Out) module, which extracts relevant features from the intermediate layers of the model, and the "Cut" module, which decomposes these features into finer details concerning position, texture, and color.

MUTEX can be seen in some ways as a mask-based method, since it generates masks from the attention values of the tokens at each layer. These masks are used to perturb the original images in order to explain which part of the input was most affected by the model's decision-making process. A notable difference between MUTEX and the other mask-based approaches is the number of masks used, which is fixed and low compared to the other methods (see Sect. 5 for all details). A smaller number of masks results in faster computation time to obtain a final heatmap, as it requires less model inference, mask aggregation, and similar processes. Moreover, the masks generated by MUTEX rely on different perspectives of token importance across the multiplex network, effectively describing the relationships between tokens in the different layers of the ViT. Therefore, MUTEX masks have specific meanings that can be easily explained. In contrast, approaches like RISE [18] generates masks randomly, resulting in a heatmap generation algorithm that is difficult to interpret. Another advantage of MUTEX is the use of a multiplex network to handle the relationships between tokens across ViT layers. The multiplex network provides a straightforward way to represent interactions between tokens, allowing for a more nuanced understanding of how information is processed across different ViT layers. MUTEX observes the behavior of tokens in the different layers and then merges these observations to identify the most influential tokens for the current classification output.

3 Proposed approach

This section provides technical details about MUltiplex Transformer Explainer (MUTEX). It is organized as follows: Section 3.1 presents some preliminary concepts necessary to understand its behavior. Section 3.2 provides an overview of it. Finally, Sections 3.3 and 3.4 illustrate the technical details of the two steps of which it consists.

3.1 Background

Network analysis is a branch of graph theory that deals with the study of networks and can also be used to build predictive models. It combines graph theory with various other concepts and tools ranging from machine learning to information visualization, from inference modeling to the study of social structure.

At the highest level of abstraction, a network models a set of relationships between entities. Relationships are represented by arcs, while entities are represented by nodes. Arcs can be oriented or not. The most common way to represent a network is the adjacency matrix. The element $A[i, j]$ of this matrix is equal to 1 if nodes n_i and n_j are connected by an arc; otherwise it is equal to 0. If the arcs are oriented, it may happen that $A[i, j]$ is different from $A[j, i]$. If the arcs are not oriented, $A[i,j] = A[j,i]$ and the adjacency matrix is symmetric. The arcs of a network may have associated weights whose semantics are given by the context represented by the network. In the case of weighted networks, the element $A[i, j]$ of the adjacency matrix indicates the weight of the arc from n_i to n_j. We define a walk from n_i to n_k as a sequence of arcs connecting n_i to n_k. A walk is open if n_i is different from n_k; otherwise it is closed. A path is an open simple (i.e., where no node is crossed twice) walk. A loop is a closed simple walk. A trail is a walk that includes each arc no more than once, while a tour is a closed walk that includes each arc at least once. We define a geodesic distance between two nodes as the length of the shortest path between them. Finally, the network diameter is the maximum geodesic distance between two nodes in the network.

An undirected network is connected if every pair of nodes is connected by a path. The components of the network are its connected subnetworks. An oriented network is strongly connected if direct paths exist for each

pair of nodes, while it is weakly connected if paths exist only by considering the arcs as if they were undirected. The degree of a node n_i is the number of arcs in which it is involved. If the network is oriented, we distinguish between indegree, which is equal to the number of arcs incoming into n_i, and outdegree, which is equal to the number of arcs outgoing from n_i. The shortest path from n_i to n_k is the path from n_i to n_k with the minimum number of arcs, if the network is unweighted; otherwise, it is the path from n_i to n_k with the minimum sum of weights. The average path length is the average length of the shortest paths between each pair of nodes in the network. A triad is a set of nodes in the network. A closed triad is a triad in which there is one arc for each pair of nodes. The average clustering coefficient of a network is the ratio of the number of closed triads to the total number of triads in the network.

The centrality of a network node is an indicator of its "importance". Since this term has a wide range of meanings, there are several measures of centrality. Specifically, there are four main measures, namely: *(i)* degree centrality, where a node is more important the higher its degree; *(ii)* closeness centrality, where a node is more important the smaller its average distance from other nodes; *(iii)* betweenness centrality, where a node is more important the more it is on the shortest paths of each pair of nodes in the network (and thus the more it can act as an intermediary); *(iv)* eigenvector centrality, where a node is more important the more links it has to other important nodes (this last definition is thus recursive).

Complex network analysis is a branch of network analysis that studies networks with non-trivial topological properties, i.e., properties that are not generally found in simple networks. Complex networks thus have connections between their elements that are neither purely regular nor purely random and are therefore very different from the mathematical models originally used in network analysis. Examples of networks investigated by complex network analysis are computer networks, biological networks, technological networks, brain networks, and social networks.

A multilayer network is a complex network consisting of multiple layers. Each layer is itself a complex network. Consequently, in a multilayer network with l layers, the set of nodes can be divided into l subsets, one for each layer of the network. The set of arcs in a multilayer network can be divided into two subsets, namely the set of intralayer arcs and the set of interlayer arcs. The set of intralayer arcs is in turn divided into l subsets, one for each layer. The subset of the intralayer arcs of the layer k, $1 \leq k \leq l$, represents the arcs connecting the nodes of layer k. The subset of interlayer arcs contains the arcs connecting nodes belonging to different layers.

A multiplex network is a special case of a multilayer network in which each layer describes a different type of interaction between the same real-world entities. Consequently, a real-world entity has a node associated with each layer (so there will be l nodes for each real-world entity). Interlayer edges simply connect nodes representing the same real-world entities. A multilayer network in which nodes in different layers represent different real-world entities (and thus a multilayer network that is not a multiplex network) is also called an interconnected network.

3.2 Overview of our approach

The workflow of MUTEX is shown in Fig. 1. As can be seen from this figure, it consists of two tasks. The first task aims to construct a multiplex network-based representation of a ViT, while the second one aims to generate a heatmap from the representation returned by the first task. The first task is called several times within the second task.

More specifically, during the first task, the ViT T of interest receives the image I to visually explain and an aggregation function ϕ belonging to a set Φ of aggregation functions. It first extracts the attention values associated with I. Afterward, it aggregates the attention heads corresponding to the extracted attention values and uses them to construct a multiplex network $\mathcal{M}_{T,I,\phi}$. The details of this procedure are discussed in Sect. 3.3.

The second task receives the image I to visually explain, a set Φ of aggregation functions for multiplex network construction, a set Γ of aggregation functions for node centrality values, and a set Ξ of centrality measures. For each aggregation function $\phi \in \Phi$, it first activates Task 1 to compute the corresponding multiplex network

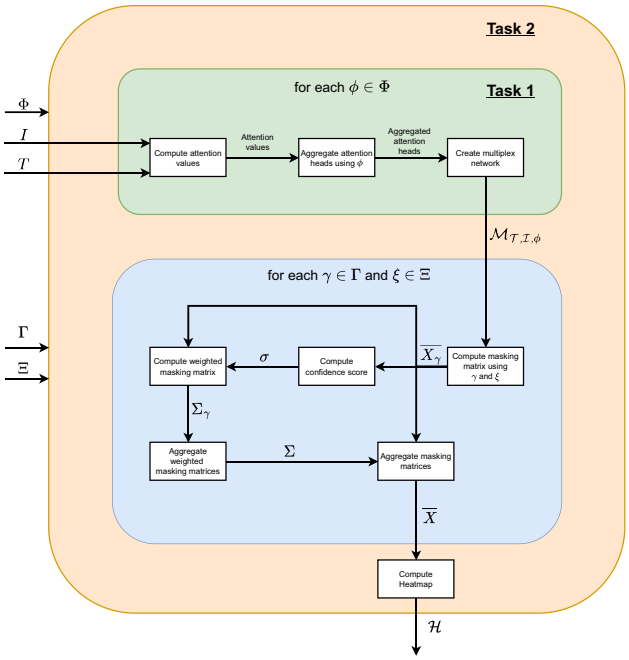


Fig. 1 Flowchart of MUTEX: for each $\phi \in \Phi$, Task 2 activates Task 1, which: *(i)* computes attention values; *(ii)* aggregates attention heads using ϕ; and *(iii)* creates a multiplex network $\mathcal{M}_{T,I,\phi}$. Then, for each $\gamma \in \Gamma$ and for each $\xi \in \Xi$, Task 2: *(i)* receives the matrix $\mathcal{M}_{T,I,\phi}$ computed by Task 1; *(ii)* computes a masking matrix $\overline{X}_\gamma$; *(iii)* computes the confidence score σ associated with $\overline{X}_\gamma$; *(iv)* computes a weighted masking matrix Σ_γ; *(iv)* aggregates the weighted masking matrices Σ_γ to obtain Σ; *(v)* aggregates the masking matrices $\overline{X}_\gamma$ to obtain $\overline{X}$; and *(vi)* computes the final heatmap $\mathcal{H}$.

$\mathcal{M}_{T,I,\phi}$. After this, for each centrality measure $\xi \in \Xi$ and for each aggregation function $\gamma \in \Gamma$, it exploits the multiplex network $\mathcal{M}_{T,I,\phi}$ to compute a masking matrix $\overline{X}_\gamma$, a confidence score σ, and a weighted masking matrix Σ_γ from them. Finally, after obtaining all masking matrices $\overline{X}_\gamma$ and Σ_γ, it constructs two overall masking matrices $\overline{X}$ and Σ from them, applies coverage bias to $\overline{X}$ and Σ, and obtains the final heatmap $\mathcal{H}$. The details of this procedure are discussed in Sect. 3.4.

3.3 Task 1: building a multiplex network of patches

In this section, we describe Task 1 of MUTEX, which deals with building the multiplex network that will be used in Task 2. In order to give both a clear and immediate idea of how this task works and all the technical details related to it, we have divided this section into two subsections. Specifically, in Sect. 3.3.1, we provide a schematic description of this task and illustrate the corresponding algorithm, while in Sect. 3.3.2, we present all its technical details.

3.3.1 Description of the approach

Task 1 takes as input: *(i)* a ViT model T; *(ii)* an image I to visually explain; *(iii)* an aggregation function ϕ; *(iv)* a value τ in the real interval [0, 1]. It returns as output a multiplex network-based representation $\mathcal{M}_{T,I,\phi}$ of T performing inference on I and using ϕ to aggregate the attention heads. It uses an initially empty set $\mathcal{N}$ of support networks. For each attention layer l_k of T: *(i)*, it extracts all the matrices of the attention heads of l_k; *(ii)* it aggregates these matrices using ϕ as the aggregation function to derive a single matrix M_k; *(iii)* it constructs a new network of $\mathcal{N}$ using the matrix obtained at step *(ii)* as the adjacency matrix; *(iv)* it computes the maximum weight w of the arcs of the new network of $\mathcal{N}$; *(v)* it removes from the latter network the arcs whose weight is less than

$w \cdot \tau$. Once $\mathcal{N}$ is obtained, it constructs $\mathcal{M}_{T,I,\phi}$ by associating a layer of it for each network of $\mathcal{N}$ (thus obtaining the nodes and the intralayer arcs of $\mathcal{M}_{T,I,\phi}$) and creating an interlayer arc for each pair of nodes corresponding to the same image patch (thus obtaining the interlayer arcs of $\mathcal{M}_{T,I,\phi}$). In Fig. 2, we provide a graphical description of this task, while in Algorithm 1 we formalize the corresponding algorithm.

Algorithm 1 Construction of the multiplex network-based representation of a ViT

Input

- T: a ViT model
- I: an image to visually explain
- ϕ: an aggregation function
- τ: a value in the real interval $[0, 1]$ for filtering the arcs from each network

Output

- $\mathcal{M}_{T,I,\phi}$: a multiplex network-based representation of T performing inference on I and using ϕ to aggregate the attention heads

$\mathcal{N}_{T,I,\phi} = \emptyset$
Store the attention layers of T in L
for l_k in L **do**
    Extract all M_{k_ϵ} matrices of the attention heads of l_k
    Aggregate all M_{k_ϵ} matrices using ϕ to obtain M_k
    Create $\mathcal{N}_k$ using M_k as adjacency matrix
    Compute the highest weight w_{max_k} in the arcs of $\mathcal{N}_k$
    Filter the arcs of $\mathcal{N}_k$ having a weight less than $w_{max_k} \cdot \tau$
    Add $\mathcal{N}_k$ to $\mathcal{N}_{T,I,\phi}$
end for
Create the sets NS and AS of $\mathcal{M}_{T,I,\phi}$ from $\mathcal{N}_{T,I,\phi}$
Create the set C of coupling edges of $\mathcal{M}_{T,I,\phi}$ for connecting the nodes corresponding to the same patches across the layers of $\mathcal{M}_{T,I,\phi}$

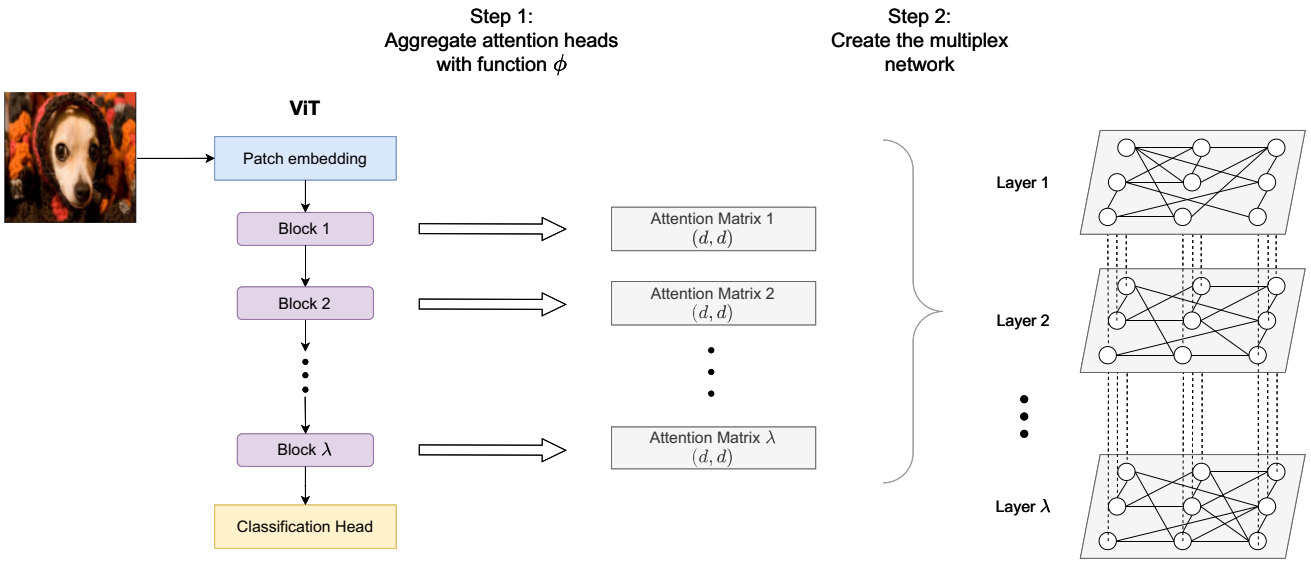


Fig. 2 Description of Task 1: In Step 1, the input image is processed by the ViT to obtain class predictions and extract attention values across all layers, which are then aggregated. In Step 2, a multiplex network is created where nodes are connected by weighted edges based on attention values. Edges with weights below a predefined threshold are pruned.

3.3.2 Technical details

Let I be an image of $b \times h$ pixels and let c be the label of the corresponding class. In the context of ViTs, an image $I \in \mathbb{R}^{b \times h}$ is split into non-overlapping squared patches. Each patch $p \in \mathbb{R}^{z \times z}$ has size $z \times z$ pixels. Consequently, I consists of a list P of patches. Specifically, I will have $r = \frac{b}{z}$ rows of patches and $m = \frac{h}{z}$ columns of patches. In total, the length of P will be $d = r \cdot m$. Since in a ViT each patch corresponds to a token, we will use these two terms interchangeably in the following.

Let T be a ViT model performing classification and let $T(c, I)$ be a function that returns the confidence of T for class c and image I. Let L be the set of attention layers present in T and let λ be the cardinality of L. An attention layer $l_k \in L$ consists of a set of η attention heads. Each attention head l_{k_ϵ}, $1 \leq \epsilon \leq \eta$, computes an attention matrix $M_{k_\epsilon} \in \mathbb{R}^{d \times d}$ that represents the attention values among the patches of I. For each attention layer, we combine all η attention matrices of the corresponding attention heads using an aggregation function ϕ. This could be one of the classical aggregation functions known in the literature, such as sum, mean, minimum or maximum [9]. As a result, we obtain an overall attention matrix $M_k \in \mathbb{R}^{d \times d}$ associated with the k-th layer of T, $1 \leq k \leq \lambda$.

Note that M_k is a squared matrix; it has one row for each patch. The i-th row of M_k indicates the attention given by the patch $p_i \in P$ to the other patches. Given its structure, M_k can be read as the adjacency matrix of a network $\mathcal{N}_k = \langle N_k, A_k \rangle$. In this case, N_k represents the nodes of $\mathcal{N}_k$; each node $n_{i_k} \in N_k$ is associated with a patch $p_i \in P$. Since there is a biunivocal correspondence between n_{i_k} and p_i, we will use these two terms interchangeably in the following. A_k represents the set of arcs of $\mathcal{N}_k$; an arc $(n_{i_k}, n_{j_k}, w_{ij_k})$ indicates that w_{ij_k} is the attention value that n_{i_k} has assigned to n_{j_k}. In this way, we map an attention matrix of an attention layer into a direct and complete network.

According to [9, 48], attention matrices tend to contain noise, which can be mitigated by removing lower attention values. This is done by removing arcs with the lowest weights from the network $\mathcal{N}_k$, which is no longer complete after this task. To this end, we use two values, namely: *(i)* w_{max_k}, which represents the maximum weight present in the arcs of $\mathcal{N}_k$, and *(ii)* τ, which is a hyperparameter whose value is in the real range [0, 1] and which serves as a tuner. The idea is to remove from $\mathcal{N}_k$ all arcs whose weight is less than $w_{max_k} \cdot \tau$. Such a removal has two advantages: it removes noise and it saves sometime in the execution of our approach. Of course, the higher τ is, the higher the number of arcs removed from $\mathcal{N}_k$.

Applying this procedure to all attention layers of L, we obtain a set of directed networks $\mathcal{N}_{T,I,\phi} = \{\mathcal{N}_1, \mathcal{N}_2, \cdots, \mathcal{N}_\lambda\}$. At this point, we can go one step further and identify the connections between all networks in $\mathcal{N}_{T,I,\phi}$. In fact, the attention matrices of the different layers correspond to the same list P of patches processed by T. Consequently, the sets $\{N_1, N_2, \cdots, N_\lambda\}$ of nodes for all networks in $\mathcal{N}_{T,I,\phi}$ correspond to the same list P of patches. To represent this peculiarity, we stack the networks and add interlayer edges to connect the nodes corresponding to the same patch $p_i \in P$, $1 \leq i \leq d$, in all of them. In this way, we obtain a multiplex network [49], where each layer corresponds to an attention layer of T.

Formally speaking, given a ViT model T, an image I, and an aggregation function ϕ, we define a multiplex network [50] as:

$$\mathcal{M}_{T,I,\phi} = \langle NS,\ AS,\ C \rangle \tag{3.1}$$

Here:

- $NS = \{N_1, N_2, \cdots, N_\lambda\}$ is the set of node sets of $\mathcal{M}_{T,I,\phi}$. The set $N_k \in NS$, $1 \leq k \leq \lambda$, is the set of nodes corresponding to the attention layer $l_k \in L$. Each node $n_{i_k} \in N_k$ is associated with a patch $p_i \in P$. Obviously, the same patch p_i is associated with λ nodes, one for each layer of the multiplex network.
- $AS = \{A_1, A_2, \ldots, A_\lambda\}$ is the set of weighted arc sets of $\mathcal{M}_{T,I,\phi}$. The set $A_k \in AS$, $1 \leq k \leq \lambda$, is the set of weighted arcs corresponding to the attention layer $l_k \in L$.

- C is the set of coupling edges, i.e., the set of interlayer edges. Each interlayer edge connects two nodes in $\mathcal{M}_{T,I,\phi}$ belonging to different layers but corresponding to the same patch $p_i \in P$.

3.4 Task 2: generating a heatmap

In this section, we describe Task 2 of MUTEX, which deals with building the Heatmap. Similar to Task 1, we have divided this section into two subsections, namely: *(i)* Sect. 3.4.1, where we provide a schematic description of this task and illustrate the corresponding algorithm, and *(ii)* Sect. 3.4.2, where we present all its technical details.

3.4.1 Description of the approach

Task 2 takes as input: *(i)* a ViT model; *(ii)* an image I to visually explain; *(iii)* a set Φ of aggregation functions for constructing the multiplex network; *(iv)* a set Γ of aggregation functions for node degree distribution; *(v)* a set Ξ of centrality measures. It returns a heatmap $\mathcal{H}$ that explains I. It uses two sets Σ and $\overline{X}$ of matrices representing image masks. For each aggregation function $\phi \in \Phi$, it generates a multiplex network $\mathcal{M}_{T,I,\phi}$ by activating Task 1. Then, for each centrality measure of Ξ and for each aggregation function $\gamma \in \Gamma$, it computes the masking matrix $\overline{X}\gamma$. Next, it takes as input the image masked according to $\overline{X}\gamma$ and returns the corresponding confidence score σ. Afterward, it calculates Σ_γ by multiplying σ and $\overline{X}\gamma$. After this, it adds the two matrices $\overline{X}\gamma$ and Σ_γ to $\overline{X}$ and Σ. Finally, after obtaining all the matrices of $\overline{X}$ and Σ, it computes the heatmap $\mathcal{H}$ from them. In Fig. 3, we provide a graphical description of this task, while in Algorithm 2 we formalize the corresponding algorithm.

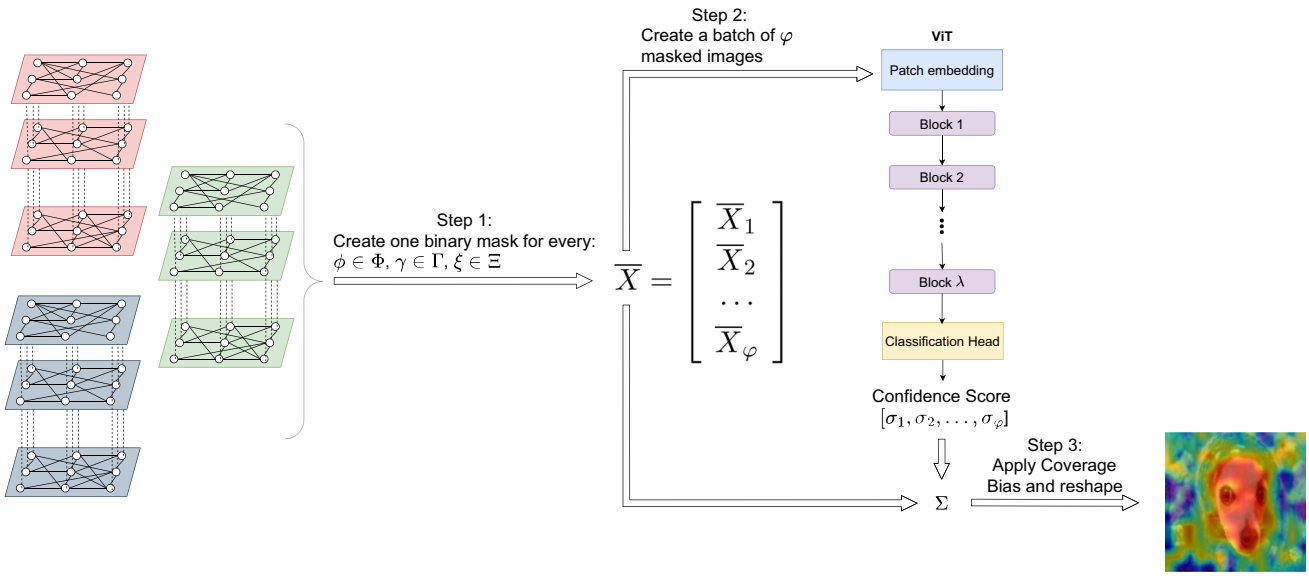


Fig. 3 Description of Task 2: In Step 1, a set of binary masks are generated, which are used in Step 2 to obscure the input image, creating a set of masked images. These are then passed through the ViT to extract the corresponding confidence scores and then are weighted using the confidence scores as weights. Finally, in Step 3 coverage bias formula is applied to obtain the final heatmap.

Algorithm 2 Heatmap generation from our multiplex network-based representation

Input

- T: a ViT model
- I: an image to visually explain
- Φ: a set of aggregation functions for the construction of the multiplex network
- Γ: a set of aggregation functions for the centrality values of nodes
- Ξ: a set of centrality measures to use in our multiplex network-based representation
- δ: the desired percentile for determining the top patches

Output

- $\mathcal{H}_{c,I}$: the heatmap explaining the class c in the image I

$\Sigma \leftarrow \emptyset$
$\overline{X} \leftarrow \emptyset$
Compute the list P of the patches of I
for $\phi \in \Phi$ **do**
    Generate $\mathcal{M}_{T,I,\phi}$ using Algorithm 1 with T, I, and ϕ as inputs
    for $\xi \in \Xi$ **do**
        for $\gamma \in \Gamma$ **do**
            Compute the masking matrix $\overline{X}_\gamma$
            Compute the confidence score σ for c, I, ϕ, and $\overline{X}_\gamma$
            $\Sigma_\gamma \leftarrow \sigma \cdot \overline{X}_\gamma$
            Stack Σ_γ into Σ
            Stack $\overline{X}_\gamma$ into $\overline{X}$
        end for
    end for
end for
$\varphi \leftarrow |\Phi| \cdot |\Gamma| \cdot |\Xi|$
for $p_i \in P$ **do**
    $\mathcal{H}_{c,I}[i] \leftarrow \frac{\sum_{q=1}^{\varphi} \Sigma[q,i]}{\sum_{q=1}^{\varphi} \overline{X}[q,i]}$
end for
Reshape $\mathcal{H}_{c,I}$ into an $r \times m$ matrix
Apply bilinear interpolation to $\mathcal{H}_{c,I}$ to obtain a $b \times h$ matrix

3.4.2 Technical details

Having defined the approach to construct $\mathcal{M}_{T,I,\phi}$, we can describe the workflow of MUTEX for generating a heatmap that provides a visual explanation of the inferences of a ViT T on an image I. We first note that $\mathcal{M}_{T,I,\phi}$ takes into account the relationships between image patches in the different attention layers. Consequently, we can use such a multiplex network to compute the importance of patches in a layer.

To achieve this, we use centrality measures commonly employed in complex network analysis, specifically weighted indegree centrality and weighted outdegree centrality. Indegree centrality reflects the attention a node receives from the other nodes and is an indicator of its importance as perceived by the other nodes. Outdegree centrality reflects the attention, and thus the importance, a node gives to the other nodes. Since these two centrality measures represent complementary perspectives on attention matrices, we decided to use both of them in our approach. Other centrality measures, such as betweenness centrality and eigenvector centrality, have high computational costs, so their use could slow down MUTEX significantly, making its execution time potentially unacceptable, while not providing meaningful information for the objectives of our approach. Instead, indegree and outdegree centrality prove to be well suited to capture the attention dynamics of Vision Transformers.

Consequently, the decision to consider only weighted indegree centrality and weighted outdegree centrality in our approach ensures the best balance between interpretability, efficiency, and relevance.

Therefore, for each node $n_{i_k} \in N_k$, we define a function $\iota(n_{i_k})$ (resp., $\omega(n_{i_k})$) that computes the weighted indegree (resp., outdegree) centrality of n_{i_k} in the layer l_k. By applying the function $\iota(\cdot)$ (resp., $\omega(\cdot)$) to all nodes of $\mathcal{M}_{T,I,\phi}$, we obtain a matrix $\mathcal{I}_{T,I,\phi}$ (resp., $\mathcal{O}_{T,I,\phi}$) of size $d \times \lambda$, which stores the weighted indegree (resp., outdegree) centrality values of all nodes of $\mathcal{M}_{T,I,\phi}$. In the following, we will use the symbol $\mathcal{M}$, instead of $\mathcal{M}_{T,I,\phi}$, and the symbols $\mathcal{I}$ and $\mathcal{O}$, instead of $\mathcal{I}_{T,I,\phi}$ and $\mathcal{O}_{T,I,\phi}$, to simplify the notation when it is not confusing. Our choice to model the ViT attention matrices as a multiplex network allows for particularly efficient operations. The multiplex network can be represented as a tensor; therefore, for each layer, the indegree (resp. outdegree) centrality of a node is computed simply by summing the corresponding elements along the rows (resp. columns).

At this point, we need an aggregation function that, given a patch $p_i \in P$, determines the overall importance of p_i in $\mathcal{M}$ by aggregating the weighted indegree (resp., outdegree) centrality values of the λ nodes corresponding to p_i in the λ layers of $\mathcal{M}$. The choice of the aggregation function is important because it defines a particular perspective from which to analyze $\mathcal{M}$. For this reason, we use a set Γ of different aggregation functions that are capable of providing different perspectives on the role that the different patches play within $\mathcal{M}$. Specifically, Γ currently contains the following aggregation functions:

- $sum(\cdot,\cdot)$: it takes as input a matrix $\mathcal{Y}$ (where $\mathcal{Y}$ can be $\mathcal{I}$ or $\mathcal{O}$) and an integer i between 1 and d, and computes the importance of the patch $p_i \in P$ by summing the values of the i-th row of $\mathcal{Y}$. Recall that this row stores the importance of the λ nodes associated with p_i in the different layers of $\mathcal{Y}$.
- $min(\cdot,\cdot)$: it takes as input a matrix $\mathcal{Y}$ and an integer i between 1 and d, and returns the importance of the patch $p_i \in P$ by computing the minimum of the values in the i-th row of $\mathcal{Y}$.
- $max(\cdot,\cdot)$: it takes as input a matrix $\mathcal{Y}$ and an integer i between 1 and d, and returns the importance of the patch $p_i \in P$ by computing the maximum of the values in the i-th row of $\mathcal{Y}$.
- $med(\cdot,\cdot)$: it takes as input a matrix $\mathcal{Y}$ and an integer i between 1 and d, and returns the importance of the patch $p_i \in P$ by computing the median of the values in the i-th row of $\mathcal{Y}$.
- $std(\cdot,\cdot)$: it takes as input a matrix $\mathcal{Y}$ and an integer i between 1 and d, and returns the importance of the patch $p_i \in P$ by computing the standard deviation of the values in the i-th row of $\mathcal{Y}$.
- $dsp(\cdot,\cdot)$: it takes as input a matrix $\mathcal{Y}$ and an integer i between 1 and d, and returns the importance of the patch $p_i \in P$ by computing the "dispersion" of the values in the i-th row of $\mathcal{Y}$. It is defined as: $dsp(\cdot,\cdot) = \frac{sum(\cdot,\cdot)}{std(\cdot,\cdot)}$. Therefore, it tends to favor patches whose nodes have high importance values and low variability.
- $skw(\cdot,\cdot)$: it takes as input a matrix $\mathcal{Y}$ and an integer i between 1 and d, and returns the importance of the patch $p_i \in P$ by computing the skewness (which is an indicator of the asymmetry) of the distribution of the values in the i-th row of $\mathcal{Y}$.
- $kur(\cdot,\cdot)$: it takes as input a matrix $\mathcal{Y}$ and an integer i between 1 and d, and returns the importance of the patch $p_i \in P$ by computing the kurtosis (which is an indicator of the tailedness) of the distribution of the values in the i-th row of $\mathcal{Y}$.
- $coh(\cdot,\cdot)$: it takes as input a matrix $\mathcal{Y}$ and an integer i between 1 and d, and returns the importance of the patch $p_i \in P$ by computing the coherency of the values in the i-th row of $\mathcal{Y}$. It is defined as:

$$coh(\cdot,\cdot) = -\sum_{k=1}^{\lambda} \frac{\mathcal{Y}[i,k]}{sum(\mathcal{Y},i)} \cdot \log_2\left(\frac{\mathcal{Y}[i,k]}{sum(\mathcal{Y},i)}\right)$$

[51]. It tends to favor patches that keep their centrality across the different layers of $\mathcal{M}$.

Considering the previous list, it is clear that Γ already has a very rich set of aggregation functions. In any case, it can be extended with additional aggregators capable of adding further perspectives to the study of $\mathcal{M}$. The current set of functions in Γ was designed to allow a balance between computational efficiency and explainability. In fact, all the functions currently in it are easy to compute and straightforward to interpret.

For each function $\gamma \in \Gamma$, we apply it to each patch $p_i \in P$ and obtain a vector X_γ representing the importance values of the patches of P in $\mathcal{M}$ computed by γ. According to [16], we binarize X_γ so that its i-th element, $1 \leq i \leq d$, indicates whether the corresponding patch $p_i \in P$ should be retained for the next step or not. To binarize X_γ, we need a value δ that belongs to the range of integers [0, 100], and a function $\psi(\cdot, \cdot)$ that takes X_γ and δ and returns the set of patches that are in the top δ percentiles. Therefore, the value of δ is in the range of integers [0, 100], because δ indicates the threshold percentile for filtering patches of P. In fact, if a patch $p_i \in P$ has an importance value $X_\gamma[i]$ within a percentile above δ, it is selected; otherwise, it is filtered out. Formally speaking, the element $X_\gamma[i]$ of X_γ, $1 \leq i \leq d$, can be binarized as:

$$\overline{X}_\gamma[i] = \begin{cases} 1, & \text{if } p_i \in \psi(X_\gamma, \delta) \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

Clearly, $\overline{X}_\gamma \in \mathbb{R}^d$. Once we have defined $\overline{X}_\gamma$, we can compute the class-specific relevance score for class c and the patches that have a value of 1 in $\overline{X}_\gamma$. To do this, we introduce the notation $I_{\overline{X}_\gamma}$ to denote the image I composed only of the patches corresponding to a value of 1 in $\overline{X}_\gamma$. Having introduced this notation, the class-specific score can be computed as:

$$\sigma_{c,I,\phi,\overline{X}_\gamma} = T(c, I_{\overline{X}_\gamma}) \tag{3.3}$$

where the function $T(c, I_{\overline{X}_\gamma})$ returns the output produced by T for class c and image $I_{\overline{X}_\gamma}$ (see Sect. 3.3). The reasoning behind Eq. 3.3 is as follows: σ represents a score that denotes the ability of our approach to classify the image I only based on a subset of patches. We evaluate its capability of classifying I into the class c to which it belongs, using not all the patches of I, but only a fraction of them (particularly those in the percentiles greater than δ). The image containing only the most significant patches is defined as $I_{\overline{X}_\gamma}$. Therefore, $T(c, I_{\overline{X}_\gamma})$ is a floating point number representing the confidence of the model T for the class c when it uses $I_{\overline{X}_\gamma}$ as input. The higher the value of $T(c, I_{\overline{X}_\gamma})$, the greater the confidence of T in classifying I into c, even when only a limited fraction of patches is available.

Finally, we multiply the score $\sigma_{c,I,\phi,\overline{X}_\gamma}$ thus obtained by $\overline{X}_\gamma$ and obtain a new mask $\Sigma_\gamma \in \mathbb{R}^d$ that takes into account the importance of the weighted patches according to the model confidence. We repeat this step for all functions $\gamma \in \Gamma$ and stack the set of masks thus obtained:

$$\Sigma_\Gamma = \begin{bmatrix} \Sigma_{\gamma_1} \\ \Sigma_{\gamma_2} \\ \cdots \\ \Sigma_{\gamma_{|\Gamma|}} \end{bmatrix} \tag{3.4}$$

We get a matrix Σ_Γ of dimension $|\Gamma| \times d$. In it, the q-th row, $1 \leq q \leq |\Gamma|$, coincides exactly with Σ_{γ_q}.

Similarly, we construct the matrix $\overline{X}_\Gamma$ by stacking all vectors $\overline{X}_\gamma$, $\gamma \in \Gamma$. The dimension of $\overline{X}_\Gamma$ is $|\Gamma| \times d$.

$$\overline{X}_{\Gamma} = \begin{bmatrix} \overline{X}_{\gamma_1} \\ \overline{X}_{\gamma_2} \\ \cdots \\ \overline{X}_{\gamma_{|\Gamma|}} \end{bmatrix} \tag{3.5}$$

The procedure we have seen so far, which leads us to the definition of Σ_{Γ} and $\overline{X}_{\Gamma}$, considers only one multiplex network and only one centrality measure. Actually, we have seen in Sect. 3.3 that it is possible to use multiple aggregation functions ϕ for the construction of the multiplex networks, each of which results in a different multiplex network. Furthermore, the computation of Σ_{Γ} was based on only one centrality measure (i.e., weighted indegree centrality), whereas there are many centrality measures that can be considered. Consequently, if we consider a set Φ of aggregation functions for the construction of the multiplex networks and a set Ξ of centrality measures, the total set of masks will be larger than $|\Gamma|$ and equal to $\varphi = |\Phi| \cdot |\Xi| \cdot |\Gamma|$. So, we will have:

$$\Sigma = \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \\ \cdots \\ \Sigma_{\varphi} \end{bmatrix} \tag{3.6}$$

Clearly, Σ is a matrix with φ rows and d columns.

Similarly for $\overline{X}$, we will have the following matrix with φ rows and d columns:

$$\overline{X} = \begin{bmatrix} \overline{X}_1 \\ \overline{X}_2 \\ \cdots \\ \overline{X}_{\varphi} \end{bmatrix} \tag{3.7}$$

To create a saliency map, we borrow the concept of coverage bias [16, 17]. Specifically, we compute the heatmap $\mathcal{H}_{c,I}$ associated with class c and image I. Such a heatmap will have d elements, one for each patch $p_i \in P$. The i-th element $\mathcal{H}_{c,I}[i]$ is obtained by summing the contributions of the scores that p_i had in the different masks Σ_{γ}, $1 \leq \gamma \leq \varphi$, and dividing it by the number of times that p_i was selected (and thus by the number of times the values associated with it in $\overline{X}_{\gamma}$ is 1, $1 \leq \gamma \leq \varphi$). Formally speaking:

$$\mathcal{H}_{c,I}[i] = \frac{\sum_{q=1}^{\varphi} \Sigma[q,i]}{\sum_{q=1}^{\varphi} \overline{X}[q,i]} \tag{3.8}$$

Once the heatmap $\mathcal{H}_{c,I}$ is obtained, we need to perform a reshape operation [16, 17]. In fact, $\mathcal{H}_{c,I} \in \mathbb{R}^d$ where $d = r \cdot m$. The image I consists of d patches arranged in r rows and m columns. The reshape operation of $\mathcal{H}_{c,I}$ has precisely the purpose of distributing the values of $\mathcal{H}_{c,I}$ in an $r \times m$ matrix. An additional step is required to move from the heatmap seen as a set of patches to the final heatmap seen as a set of pixels. Recall that the size in pixels of the original image was $b \times h$, where $b = r \cdot z$ and $h = m \cdot z$. To obtain the heatmap as a set of $b \times h$ pixels instead of a set of $r \times m$ patches, we perform a bilinear interpolation. This is a method of estimating the value of a function at a given point if the values of the surrounding points are known. It is commonly used in image processing and computer graphics to perform tasks such as image resizing, texture mapping, or remapping grid construction [16]. Bilinear interpolation works on 2D data by extending linear interpolation, which works on only one dimension. Specifically, bilinear interpolation is carried out by first performing linear interpolation in one

direction and then later performing linear interpolation in the other direction. Although each step is linear in sampled values and position, bilinear interpolation is actually nonlinear; specifically, it is quadratic in sample location. Bilinear interpolation returns smoother results than simple methods such as nearest-neighbor interpolation, which can return visible artifacts of jagged borders.

At the end of bilinear interpolation, we obtain the heatmap representing the visual explainability of the ViT T when operating on the image I and the class c.

4 Experiments

In this section, we describe the experiments conducted to evaluate the performance of MUTEX and compare it to existing approaches in the literature. Specifically, in Sect. 4.1, we describe our experimental setup, highlighting the approaches chosen for comparison, the ViT models, the explainability metrics, the computational load metrics, and the datasets. In Sect. 4.2, we report the quantitative performance obtained by MUTEX and compare it with that of other related approaches. In Sect. 4.3, we report a qualitative analysis of MUTEX performed on three examples. Finally, in Sect. 4.4, we describe the hyperparameter analysis and ablation study we carried out to evaluate the impact of some hyperparameters and model design on the performance of MUTEX.

4.1 Experimental setup

In our experimental campaign, the patch retention ratio δ (see Eq. 3.2) is set to 50. The set Φ of functions chosen to aggregate the attention matrices of the attention heads are sum, minimum, and maximum. The aggregation functions chosen to determine the importance of a patch $p_i \in P$ are all those that compose the set Γ illustrated in Sect. 3.4; thus, $|\Gamma| = 9$. Finally, the centrality measures considered are weighted indegree centrality and weighted outdegree centrality; thus, $|\Xi| = 2$. Consequently, the size φ of the vectors Σ and $\overline{X}$ will be equal to $3 \cdot 2 \cdot 9 = 54$. Therefore, we will have 54 masks to extract the final heatmap. The methods used for the comparison with MUTEX are TIS [16], ViT-CX [17], TAM [31], Chefer 1 [7], Chefer2 [19], BT H [32], BT T [32], RISE [18], IntegratedGrad [6], SmoothGrad [25], GradCAM [5], GradCAM++ [23], and ScoreCAM [24]. In calculating the performance of these methods, we used the same hyperparameters specified in the corresponding papers.

In our experiments, we used two ViT models, namely vision transformer (ViT) [4] and data-efficient image transformer (DeiT) [52], which are widely used in computer vision tasks. ViT adopts an encoder-only structure and processes an image by dividing it into d distinct patches. It then transforms these patches into a sequence of tokens by projecting them into an embedding space. It finally adds a special classification token (CLS) to the beginning of the sequence and returns the result thus obtained to its encoder. DeiT is derived from the architecture of ViT; it considers the CLS token and adds a distillation token, which is used with an additional classification head that facilitates learning by distillation. The implementations of ViT and DeiT we use are ViT-Base and DeiT-Base. Since we want to leverage the importance of the sole patches, we have removed the CLS and distillation tokens from both models.

In order to quantitatively evaluate the performance of visual explainability approaches, several faithfulness metrics have been proposed in the literature [18, 53, 54]. We use three of them mostly adopted in the state-of-the-art papers, namely Insertion, Deletion [18] and Faithfulness Violation Test [53].

The Insertion metric evaluates the probability of correctly identifying the target class using an increasing number of the most significant pixels, according to the heatmap scores. We compute the Insertion metric using different percentages of pixels to obtain the Insertion curve. Then, we calculate the Area Under the Curve (AUC) related to the Insertion curve. Its value falls in the real range [0, 1]. The higher the value, the better the performance of the explainability approach being evaluated. In contrast, the Deletion metric evaluates how the confidence of detecting the target class decreases as the pixels with the highest score on the corresponding heatmap are removed from the original image. Similar to the Insertion metric, we again compute the Deletion

curve and the corresponding AUC. Again, the value of the metric falls in the real range [0, 1]; however, in this case, the lower the value, the better the performance of the explainability approach. Finally, we also calculate the difference between Insertion and Deletion, which provides a summarization of the performance of the explainability approach. In particular, the higher the value of the difference, the better the performance of the considered explainability approach. Following the state-of-the-art papers [16], we used four ways to obfuscate pixels in the computation of the Insertion and Deletion metrics, namely Mean, Blur, Black and Rand. In fact, replacing pixel values can affect the output of the model; therefore, it is important to test different baselines. Specifically, Mean replaces pixel values with the mean values of the entire image, Blur performs pixel blurring, Black replaces pixels with the color black, and Rand replaces pixels with a random color.

The third faithfulness metric we adopted was the Faithfulness Violation Test [53]. This metric calculates the consistency between the explanation weights of the saliency map and their impact on the prediction. Following [53], we calculated the Violation Ratio as follows: after masking the best 10% of tokens, we calculated the difference between the confidence obtained by giving the original image in input to the model and the one obtained by giving the masked image in input to it. If this difference is negative there is a violation. The Violation Ratio is calculated as the number of violations on the total number of images. This value falls in the real range [0, 1], and a lower value is associated with a better explainability approach.

To evaluate the effectiveness of our approach in localizing objects, we employed the Pointing Game metric. It assesses localization accuracy by checking whether the saliency peak within a heatmap falls inside the ground truth bounding box of the object. Note that for this metric to be computed, the image bounding boxes must be defined in the dataset used for its evaluation. The values of this metric range from 0 to 1, with higher values indicating better localization accuracy and 1 indicating perfect alignment between the explanation map and the ground truth bounding box.

To assess computational load, we chose to calculate the Giga Floating Point Operations (GFLOPs), a standard measure of computational performance that quantifies the number of billions of floating point operations required to execute an approach. GFLOPs are widely used to benchmark the efficiency and performance of algorithms.

Finally, following the assessment protocols used in previous work on visual explainability approaches for Vision Transformers [7, 9, 16–19], we decided to use two datasets. The former was a random subset of the ImageNet validation set [55] consisting of 5,000 images [17, 32]. The latter was BloodMNIST [56], a dataset from the medical domain, which is an environment very different from the ones of ImageNet. BloodMNIST was designed for the development of automatic recognition systems for images of peripheral blood cells. It consists of 17,092 RGB images of individual normal cells, categorized into eight classes. Each image has a resolution of 224×224 pixels. We point out that ImageNet provides the image bounding boxes while BloodMNIST does not provide this information.

The interested reader can find the code that implements our approach at https://github.com/DavideTraini/ViT-Visual-Interpretability.

4.2 Quantitative results

In this section, we provide a quantitative comparison between our approach and other related ones already available in the literature. The comparison is made using two datasets, namely ImageNet (Sect. 4.2.1) and BloodMNIST (Sect. 4.2.2).

4.2.1 Approach comparison using ImageNet

In Table 1, we report the values of the Insertion metric, Deletion metric and their difference obtained by MUTEX and the other approaches using the four obfuscation methods mentioned in Sect. 4.1 and using ViT [4] as vision transformer. To ensure fairness, we used the same subset of the ImageNet validation set as TIS [16]. Therefore, in Table 1, the performance results of the compared approaches are directly taken from [16]. The arrow next to the

Table 1 Results of the Insertion, Deletion, and Insertion–Deletion metrics for MUTEX and other related approaches when applied to ImageNet and the ViT model using different types of obfuscation methods

	Insertion ↑				Deletion ↓				Insertion–Deletion ↑			
Method	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
MUTEX	**0.65**	0.68	**0.65**	**0.61**	0.22	0.40	0.22	0.21	**0.43**	0.28	**0.43**	**0.40**
TIS [16]	0.52	0.66	0.50	0.47	0.10	0.39	0.10	0.09	0.42	0.28	0.40	0.38
ViT-CX [17]	0.51	0.61	0.41	0.39	0.20	0.42	0.14	0.18	0.28	0.20	0.31	0.35
TAM [31]	0.43	0.61	0.41	0.39	0.14	0.43	0.14	0.13	0.28	0.18	0.27	0.26
Chefer1 [7]	0.42	0.61	0.41	0.39	0.15	0.44	0.14	0.13	0.28	0.17	0.27	0.26
Chefer2 [19]	0.43	0.61	0.41	0.39	0.15	0.44	0.14	0.13	0.28	0.17	0.27	0.26
BT H [32]	0.45	0.63	0.43	0.41	0.12	0.41	0.12	0.11	0.33	0.21	0.32	0.30
BT T [32]	0.46	0.62	0.44	0.42	0.13	0.42	0.12	0.11	0.33	0.21	0.32	0.30
RISE [18]	0.46	0.62	0.45	0.42	0.16	0.45	0.16	0.15	0.30	0.17	0.29	0.27
IntegratedGrad [6]	0.19	**0.69**	0.16	0.15	**0.08**	**0.31**	**0.06**	**0.06**	0.11	**0.38**	0.10	0.08
SmoothGrad [25]	0.37	0.59	0.36	0.35	0.10	0.45	0.10	0.09	0.27	0.14	0.26	0.26
GradCAM [5]	0.51	0.65	0.49	0.50	0.16	0.42	0.16	0.17	0.34	0.23	0.33	0.33
GradCAM++ [23]	0.39	0.58	0.38	0.38	0.30	0.51	0.29	0.29	0.09	0.07	0.09	0.10
ScoreCAM [24]	0.47	0.62	0.45	0.45	0.21	0.43	0.20	0.20	0.26	0.19	0.25	0.25

MUTEX always achieves the maximum (or at least submaximum) value for Insertion by effectively identifying and adding the most influential patches to increase model confidence. It also achieves low Deletion values, close to those of the best performing approaches. Finally, it always achieves the maximum (or at least submaximum) value for Insertion–Deletion, regardless of the perturbation. This highlights MUTEX's ability to reliably identify features that critically affect model predictions

metric name indicates whether the best performance is obtained for high values (up arrow) or for low values (down arrow). Bold values indicate the best values for the metric and the corresponding obfuscation method. Underlined values indicate suboptimal values.

From the analysis of Table 1, we can see that MUTEX has an interesting performance. In fact, regarding the Insertion metric, it obtains the best performance for three obfuscation methods, namely Mean (where its performance is 25.00% higher than the suboptimal one), Black (+30.00%) and Rand (+22.00%). On the other hand, for Blur, MUTEX obtains a suboptimal performance (which is 0.68, compared to the optimal one of 0.69 obtained by IntegratedGrad). As for the Deletion metric, MUTEX is the third best when Blur is used as an obfuscation method. For the other three obfuscation methods, the performance of MUTEX is not among the best, although it is in line with other approaches (e.g., ViT-CX with Mean and Rand obfuscation methods). Finally, if we consider the difference between Insertion and Deletion, MUTEX shows the best performance for three of the four obfuscation methods. Specifically, in the case of Mean, it achieves an improvement of 2.38% over the suboptimal value, in the case of Black this improvement is 7.50%, while in the case of Rand it is 5.26%. In the case of Blur, MUTEX obtains the suboptimal value together with TIS, reaching a value of 0.28, while the optimal value is 0.38 (+35.71%) obtained by IntegratedGrad.

We also performed the same computations described above, but using the Vvsion transformer DeiT [52] instead of ViT. Table 2 shows the results obtained. Similar to the previous experiment, we used the same subset of the ImageNet validation set as TIS [16]; hence, the performance results of the compared approaches are directly taken from [16].

From the analysis of this table, we can see that MUTEX also obtains interesting results with DeiT. In fact, regarding the Insertion metric, it achieves the optimal values for all four obfuscation methods. In particular, it improves the suboptimal approach by 16.13% in the case of Mean, 7.35% in the case of Blur, 17.74% in the case of Black, and 19.04% in the case of Rand. Similarly to what happened with ViT, the Deletion metric values of MUTEX are not among the best ones, although they are in line with those of other approaches (e.g., those of TIS and ViT-CX in the case of Blur). However, the most interesting result is the fact that in terms of the difference

Table 2 Results of the Insertion, Deletion, and Insertion–Deletion metrics for MUTEX and other related approaches when applied to ImageNet and the DeiT model using different types of obfuscation methods

	Insertion ↑				Deletion ↓				Insertion–Deletion ↑			
Method	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
MUTEX	**0.72**	**0.73**	**0.73**	**0.75**	0.28	0.41	0.28	0.30	**0.44**	**0.32**	**0.44**	**0.44**
TIS [16]	0.57	0.65	0.57	0.54	0.15	0.40	0.15	0.14	0.42	0.25	0.42	0.41
ViT-CX [17]	0.51	0.61	0.51	0.48	0.20	0.42	0.20	0.18	0.31	0.19	0.31	0.30
TAM [31]	0.50	0.59	0.50	0.46	0.23	0.45	0.23	0.19	0.27	0.14	0.26	0.26
Chefer1 [7]	0.51	0.60	0.51	0.48	0.22	0.45	0.22	0.18	0.29	0.15	0.29	0.29
Chefer2 [19]	0.50	0.60	0.50	0.47	0.23	0.45	0.23	0.19	0.28	0.14	0.27	0.28
BT H [32]	0.52	0.60	0.52	0.49	0.19	0.43	0.19	0.16	0.33	0.18	0.33	0.33
BT T [32]	0.52	0.60	0.51	0.48	0.19	0.43	0.19	0.16	0.33	0.17	0.32	0.32
RISE [18]	0.55	0.61	0.55	0.52	0.25	0.46	0.25	0.21	0.30	0.15	0.30	0.31
IntegratedGrad [6]	0.32	0.68	0.30	0.28	**0.14**	**0.38**	**0.12**	**0.13**	0.18	0.30	0.18	0.15
SmoothGrad [25]	0.45	0.62	0.43	0.43	**0.14**	0.44	0.14	**0.13**	0.31	0.18	0.30	0.31
GradCAM [5]	0.62	0.68	0.62	0.63	0.23	0.43	0.23	0.27	0.38	0.25	0.38	0.37
GradCAM++ [23]	0.49	0.59	0.49	0.52	0.41	0.54	0.42	0.45	0.07	0.05	0.07	0.07
ScoreCAM [24]	0.55	0.64	0.55	0.57	0.32	0.47	0.32	0.35	0.23	0.17	0.23	0.22

MUTEX always achieves the maximum Insertion value, regardless of the type of perturbations. Although it does not achieve the lowest Deletion values, it still shows competitive performance for this metric. MUTEX achieves the highest value of Insertion–Deletion regardless of the perturbation considered, demonstrating its ability to effectively identify the most critical input regions that contribute to model predictions

between Insertion and Deletion, MUTEX obtains the optimal values with all four blurring methods, improving the suboptimal values by 4.76% in the case of Mean, 6.67% in the case of Blur, 4.76% in the case of Black, and 7.32% in the case of Rand.

Overall, we can say that MUTEX achieves very high Insertion and Insertion–Deletion values, which makes it a very competitive solution compared to the methods already proposed in the literature. On the other hand, it shows a relatively weaker performance on the Deletion metric. This result can be explained by considering how each metric evaluates the model's explainability, on the one hand, and the intrinsic characteristics of MUTEX, on the other hand. The Insertion metric measures how quickly the model's confidence in the correct class increases as the most important pixels identified by the explainability method are added back to a perturbed image. MUTEX excels in this metric because it effectively identifies and ranks the most important patches or regions in an image that contribute positively to the model's predictions. This suggests that the attention maps and the multiplex network-based approach in MUTEX pinpoint the informative parts of an image, allowing the model to rapidly improve its prediction confidence with minimal information. On the other hand, the Deletion metric evaluates how quickly the model's confidence decreases as the most important pixels are removed from the original image. MUTEX shows relatively weaker performance on this metric, which may seem paradoxical given its success in identifying important regions for the Insertion metric. Actually, this can be explained by considering that, when creating the saliency map, MUTEX tends to assign similar values to the pixels of the same token, which means that clusters of pixels are eliminated in the Deletion calculation. However, as already observed in [16], approaches that perturb the image in a diffuse way result in better Deletion scores. Indeed, approaches that remove pixels in a scattered manner often produce a sharper decrease in model confidence because they disrupt the contextual information more uniformly across the image. MUTEX, on the other hand, identifies and retains significant regions during Insertion, where pixels and their contexts are incrementally added back, quickly restoring prediction confidence. Actually, the decrease in the Deletion metric is offset by a greater increase in the Insertion metric, as evidenced by the Insertion–Deletion metric values.

Table 3 Faithfulness Violation Test results for MUTEX, TIS, BT H, ViT-CX, and GradCAM when applied to ImageNet for ViT and DeiT

Method↓	*ViT*	*DeiT*
MUTEX	0.21	**0.10**
TIS [16]	**0.18**	0.11
BT H [32]	0.24	0.29
ViT-CX [17]	0.29	0.21
GradCAM [5]	0.28	0.18

MUTEX shows superior performance when applied to the DeiT model, where it achieves the best results, while it ranks second (still very close to the first) when applied to the ViT model. This highlights MUTEX's strong ability to minimize faithfulness violations across different transformer architectures, highlighting its robustness and effectiveness

In Table 3, we report the values of the Faithfulness Violation Test. We compared MUTEX with the four best approaches from the previous experiments, namely TIS [16], BT H [32], ViT-CX [17], and GradCAM [5]. For these comparisons, we used both ViT and DeiT vision transformers. Moreover, we adopted the Black obfuscation method. We chose BT H over BT T because the two approaches are similar, but the former is slightly better in Insertion–Deletion metric. We recall that a lower value of the Faithfulness Violation Test means a lower number of faithfulness violations and therefore a better explainability approach. Regarding the results obtained with the ViT model, the lowest value (i.e., 0.21) is returned by TIS, but the value returned by MUTEX is very close. Next are BT H, VIT-CX and GradCAM, which have higher values than TIS of 14.29%, 38.10%, and 35.71%, respectively. As for the results with the DeiT Vision Transformer, MUTEX has the lowest value with 0.10, followed by TIS (+10.00%), GradCAM (+80.00%), VIT-CX (+110.00%) and BT H (+190.00%). These results show that MUTEX and TIS yield very similar and satisfactory values for this metric; the differences in results are slight and only depend on the vision transformer model used.

The ability to locate objects by MUTEX and the other approaches is evaluated by the Pointing Game metric. We can compute this metric on ImageNet, since, in this dataset, for each image, we can access the bounding boxes representing the regions that contain the object to be classified. The Pointing Game values obtained by the different approaches when using ViT and DeiT as Vision Transformers are shown in Table 4.

This table shows that MUTEX outperforms all the other approaches with respect to this metric. In particular, compared to the second best approach, it achieves an improvement of 0.70% using ViT and 1.45% using DeiT. This means that MUTEX not only generates high-quality heatmaps, but also accurately localizes the object of

Table 4 Pointing game values obtained by MUTEX and related approaches on ImageNet using ViT and DeiT as vision transformer models

Method↑	*ViT*	*DeiT*
MUTEX	**0.861**	**0.837**
TIS [16]	0.823	0.825
ViT-CX [17]	0.700	0.700
TAM [31]	0.737	0.635
Chefer1 [7]	0.768	0.748
Chefer2 [19]	0.727	0.654
BT H [32]	0.855	0.775
BT T [32]	0.846	0.755
RISE [18]	0.753	0.766
IntegratedGrad [6]	0.633	0.297
SmoothGrad [25]	0.499	0.742
GradCAM [5]	0.433	0.415
GradCAM++ [23]	0.809	0.774
ScoreCAM [24]	0.735	0.716

MUTEX achieves the highest Pointing Game values with both models demonstrating higher effectiveness than related approaches

Table 5 GFLOPs required by MUTEX and other related approaches

Method↓	*GFLOPs*
MUTEX	1,002.55
TIS [16]	17,808.70
RISE [18]	140,927.36
ViT-CX [17]	4,918.71
Chefer1 [7]	377.67
Chefer2 [19]	105.13
BT H [32]	2,264.73
BT T [32]	2,262.70
TAM [31]	2,203.77
IntegratedGrad [6]	3,572.95
SmoothGrad [25]	3,572.93
GradCAM [5]	100.63
GradCAM++ [23]	100.63
ScoreCAM [24]	25,915.90

MUTEX requires significantly fewer GFLOPs than most related approaches. There are some approaches, such as Chefer1, Chefer2, GradCAM, and GradCAM++, that require fewer GFLOPs than MUTEX. However, MUTEX outperforms them in terms of Insertion–Deletion, providing the best tradeoff between computational efficiency and performance

interest. This also highlights the efficiency of MUTEX in improving object localization, making it a robust solution for interpretability in Vision Transformers.

Finally, we performed a computational load analysis of MUTEX and related approaches. To do this, we calculated the GFLOPs required by them to perform their tasks. The lower the values of GFLOPs, the lighter the approach. The results of our analysis are shown in Table 5.

This table shows that MUTEX is one of the most efficient approaches, requiring 1,002.55 GFLOPs. MUTEX is significantly more efficient than other approaches like TIS, VIT-CX, etc. There are some approaches, such as GradCAM, that require fewer GFLOPs than MUTEX. However, MUTEX consistently outperforms all of them on explainability metrics, especially Insertion–Deletion values. The tradeoff between computational cost and performance is largely favorable for MUTEX, as it achieves superior performance with a modest increase in GFLOPs. There are also some approaches, like TIS, which that require significantly more computational resources than MUTEX, while still achieving levels of explainability performance comparable to those achieved by MUTEX. Therefore, examining the efficiency and effectiveness of MUTEX allows us to say that it is a good compromise solution among the ViT explainability approaches available in the literature.

4.2.2 Approach comparison using BloodMNIST

In this section, we replicate the experiments performed on ImageNet using the BloodMNIST dataset. Table 6 shows the values of Insertion, Deletion and Insertion–Deletion metrics obtained by MUTEX and related approaches when applied to this dataset, considering the four types of obfuscation methods described above and the ViT model.

This table shows that the performance of MUTEX is very good. In particular, regarding the Insertion metrics, it achieves the best results with all obfuscation methods. Its improvements over the second best approach are 7.35% for Mean, 2.82% for Blur, 8.96% for Black and 3.17% for Rand. As for the Deletion metric, although MUTEX is not the best approach, its results are still in line with those of the other approaches. Finally, as for the Insertion–Deletion metric, MUTEX ranks first for all obfuscation methods, achieving gains of 4.55% for Mean, 6.25% for Blur, 2.22% for Black, and tying with TIS for Rand.

Table 6 Results of the Insertion, Deletion, and Insertion–Deletion metrics for MUTEX and other related approaches when applied to BloodMNIST and the ViT model using different types of obfuscation methods

	Insertion ↑				Deletion ↓				Insertion–Deletion ↑			
Method	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
MUTEX	**0.73**	**0.73**	**0.73**	**0.65**	0.27	0.56	0.27	0.27	**0.46**	**0.17**	**0.46**	**0.38**
TIS [16]	0.68	0.71	0.67	0.63	**0.23**	0.56	0.24	0.25	0.44	0.15	0.45	**0.38**
ViT-CX [17]	0.46	0.70	0.43	0.47	0.28	0.58	0.27	0.26	0.18	0.12	0.16	0.20
TAM [31]	0.54	0.65	0.49	0.52	0.28	**0.50**	0.29	0.29	0.16	0.15	0.20	0.23
Chefer1 [7]	0.57	0.66	0.52	0.52	0.25	0.51	0.27	0.29	0.21	0.15	0.25	0.23
Chefer2 [19]	0.58	0.64	0.51	0.53	0.27	0.52	0.28	0.31	0.21	0.12	0.23	0.22
BT H [32]	0.56	0.62	0.50	0.56	0.28	0.53	0.29	0.31	0.29	0.09	0.21	0.23
BT T [32]	0.55	0.63	0.49	0.54	0.27	0.52	0.27	0.30	0.28	0.11	0.22	0.24
RISE [18]	0.60	0.62	0.54	0.52	0.28	0.52	0.25	0.28	0.22	0.10	0.29	0.24
IntegratedGrad [6]	0.34	0.70	0.30	0.28	0.27	0.54	**0.20**	**0.23**	0.08	0.16	0.10	0.05
SmoothGrad [25]	0.42	0.66	0.32	0.31	0.29	0.54	0.28	0.27	0.13	0.12	0.04	0.04
GradCAM [5]	0.51	0.71	0.47	0.50	0.26	0.58	0.28	0.27	0.24	0.13	0.20	0.23
GradCAM++ [23]	0.43	0.67	0.41	0.40	0.39	0.65	0.38	0.38	0.03	0.02	0.03	0.02
ScoreCAM [24]	0.45	0.69	0.42	0.44	0.30	0.59	0.30	0.29	0.14	0.10	0.13	0.14

MUTEX achieves the highest Insertion values and proves to be superior to the other approaches in identifying critical features. It is not the method that provides the best Deletion values, but the values it provides are still competitive. More importantly, it ranks first for Insertion–Deletion values, highlighting its overall effectiveness in providing explanations

Table 7 Results of the Insertion, Deletion, and Insertion–Deletion metrics for MUTEX and other related approaches when applied to BloodMNIST and the DeiT model using different types of obfuscation methods

	Insertion ↑				Deletion ↓				Insertion–Deletion ↑			
Method	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
MUTEX	**0.76**	**0.68**	**0.76**	**0.77**	0.28	**0.34**	0.28	0.29	**0.48**	**0.34**	**0.48**	**0.48**
TIS [16]	0.72	**0.68**	0.72	0.73	0.26	0.35	**0.25**	0.27	0.47	0.33	0.47	0.46
ViT-CX [17]	0.52	0.66	0.49	0.52	**0.24**	0.38	**0.25**	0.30	0.28	0.28	0.24	0.22
TAM [31]	0.59	0.66	0.56	0.58	0.31	0.52	0.32	0.31	0.28	0.14	0.24	0.27
Chefer1 [7]	0.56	0.64	0.58	0.56	0.29	0.54	0.31	0.33	0.27	0.10	0.27	0.23
Chefer2 [19]	0.57	0.63	0.58	0.55	0.31	0.54	0.30	0.32	0.26	0.09	0.28	0.23
BT H [32]	0.58	0.60	0.52	0.57	0.29	0.55	0.31	0.35	0.29	0.05	0.21	0.22
BT T [32]	0.55	0.63	0.49	0.54	0.29	0.52	0.29	0.30	0.28	0.11	0.20	0.24
RISE [18]	0.66	0.65	0.58	0.51	0.29	0.52	0.29	0.29	0.27	0.13	0.30	0.22
IntegratedGrad [6]	0.39	0.58	0.38	0.31	0.33	0.44	0.28	**0.26**	0.06	0.14	0.10	0.04
SmoothGrad [25]	0.44	0.58	0.36	0.32	0.35	0.44	0.32	0.30	0.09	0.14	0.04	0.02
GradCAM [5]	0.48	0.65	0.47	0.54	0.25	0.40	**0.25**	0.31	0.23	0.25	0.22	0.23
GradCAM++ [23]	0.48	0.60	0.47	0.53	0.35	0.49	0.34	0.39	0.13	0.11	0.13	0.13
ScoreCAM [24]	0.43	0.56	0.45	0.49	0.42	0.55	0.41	0.46	0.01	0.01	0.04	0.03

MUTEX achieves the highest Insertion values, demonstrating a superior ability to identify important features. It achieves the best Deletion value for Blur, the suboptimal value for Black, and among the best values for Mean and Rand. More importantly, MUTEX achieves the best Insertion–Deletion values, highlighting its overall effectiveness in providing explanations

Next, we performed the same experiment using DeiT as the vision transformer model. The results are reported in Table 7. This table shows similar results to Table 6. MUTEX remains the best approach for the Insertion

metric, showing improvements of 5.56% for Mean and Black and 5.48% for Rand over the second best approach, and tying with TIS for Blur. As for the Deletion metric, MUTEX achieves the best performance for Blur (with a 2.86% improvement over the second best approach) and ranks second for Black. In the other two cases, the values it obtains are still among the best ones. Finally, as for the Insertion–Deletion metric, MUTEX outperforms all the other approaches, obtaining an improvement of 2.13% for Mean, 3.03% for Blur, 2.13% for Black, and 4.35% for Rand over the second best approach. These results confirm that MUTEX is the approach with the best explainability performance.

Finally, similar to what we did for ImageNet, we calculated Faithfulness Violation Test values for MUTEX, TIS, BT H, VIT-CX and GradCAM using BloodMNIST as the dataset. Again, we used both ViT and DeiT as vision transformer models. The results are shown in Table 8. From the analysis of this table, we can see that MUTEX is the best performing approach in terms of this metric. In the ViT case, it has a 33.33% improvement over the second best approach, while in the DeiT case, its improvement over the second best approach is 25.00%.

4.3 Qualitative results

As a further comparison between MUTEX and related approaches, we evaluated the heatmaps extracted by ViT for four images of the ImageNet dataset belonging to different classes. Given the heterogeneity of the approaches considered, we categorized them into two groups for this analysis, namely: *(i)* classical approaches, originally designed for convolutional neural networks (CNNs), and *(ii)* ViT-specific approaches, explicitly designed for Vision Transformers. This distinction allows for a more structured comparison of their interpretability and effectiveness in localizing objects of interest. The heatmaps for all the considered approaches are shown in Fig. 4. In particular, in Fig. 4a, classical approaches are shown, while in Fig. 4b, ViT-specific approaches are reported.

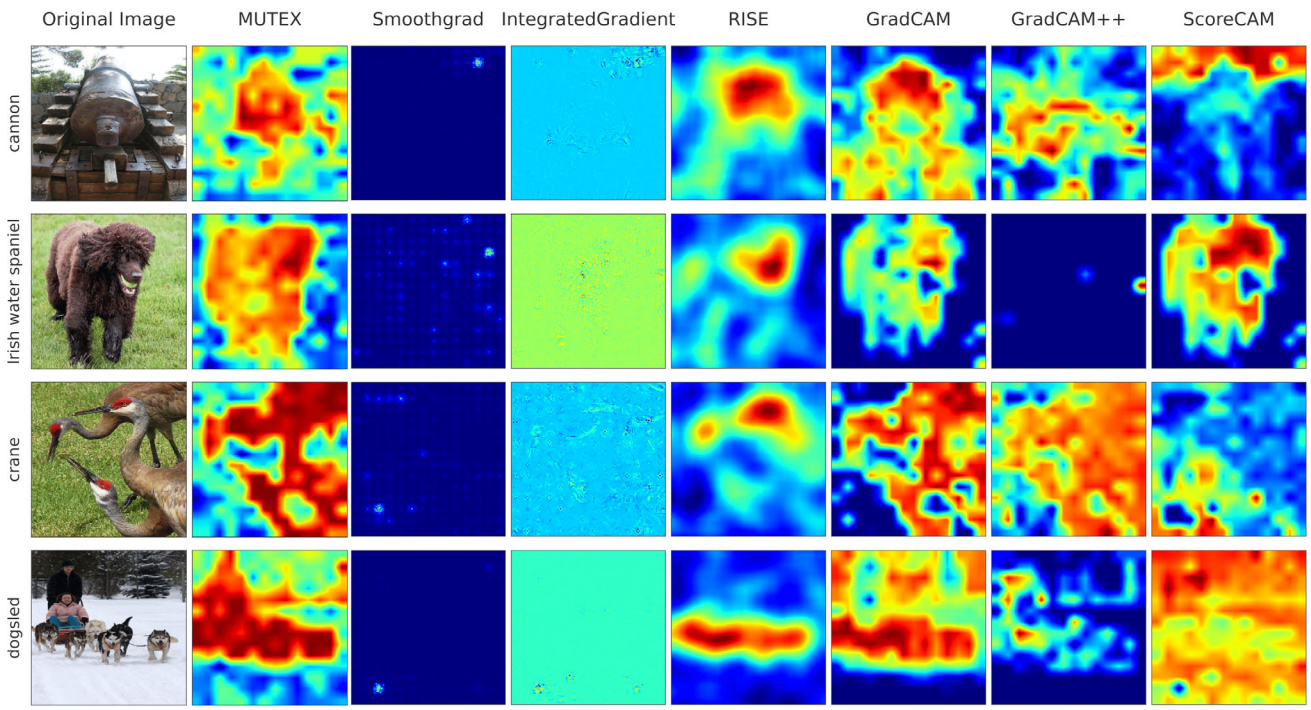
This figure shows that MUTEX is effective in identifying the salient regions of an image. In all four cases, the heatmaps it creates are highly expressive, highlighting key features relevant to classification, such as the dog's nose, the shape of the three birds, and the positions of the dogs. In addition, MUTEX heatmaps show different intensity levels, reflecting the relative importance of different regions. For instance, in the case of the Irish Water Spaniel, the dog's nose is assigned a higher intensity than its paws, while in the dogsled image, the dogs are considered more important than the two men standing behind them. In contrast, Integrated Gradients and SmoothGrad produce noisy heatmaps that are often difficult to interpret. GradCAM, while producing promising results, occasionally fails to highlight critical areas, such as the paws of the Irish Water Spaniel. Similarly, GradCAM++ and ScoreCAM struggle in specific cases, with GradCAM++ performing poorly for the Irish Water Spaniel and ScoreCAM struggling with the dogsled image. TIS and ViT-CX also yield interesting results. However, their focus is sometimes either too narrow or too wide. For example, TIS focuses too much on a small region in the Irish Water Spaniel case, while ViT-CX highlights an area that is too large, as can be seen in the crane image.

After the qualitative comparison between MUTEX and the other related approaches, we now examine in detail the results returned by MUTEX on some images related to three classes, namely chihuahua, ice cream, and

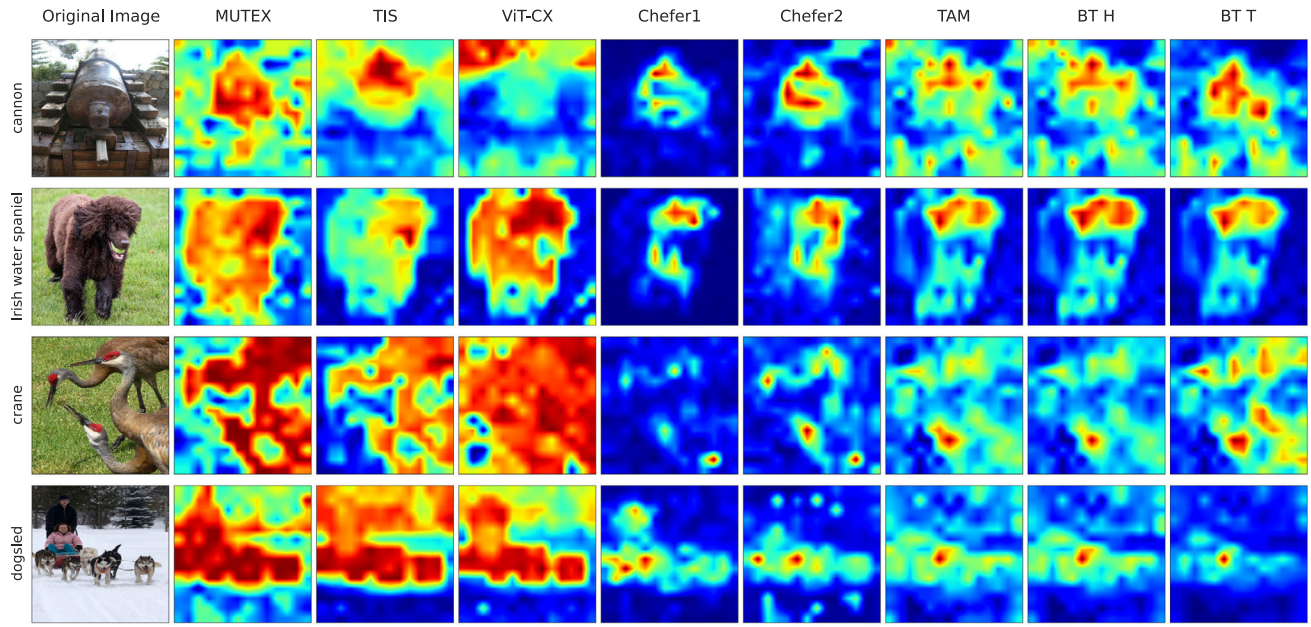
Table 8 Faithfulness Violation Test results for MUTEX, TIS, BT H, ViT-CX, and GradCAM when applied to BloodMNIST for ViT and DeiT

Method ↓	*ViT*	*DeiT*
MUTEX	**0.06**	**0.08**
TIS [16]	0.08	0.10
BT T [32]	0.11	0.12
ViT-CX [17]	0.14	0.11
GradCAM [5]	0.13	0.12

MUTEX achieves the lowest (and thus the best) Faithfulness Violation Test values, indicating that it has superior reliability in capturing the model's decision-making. It provides more consistent and faithful explanations than related approaches, which show higher inconsistency

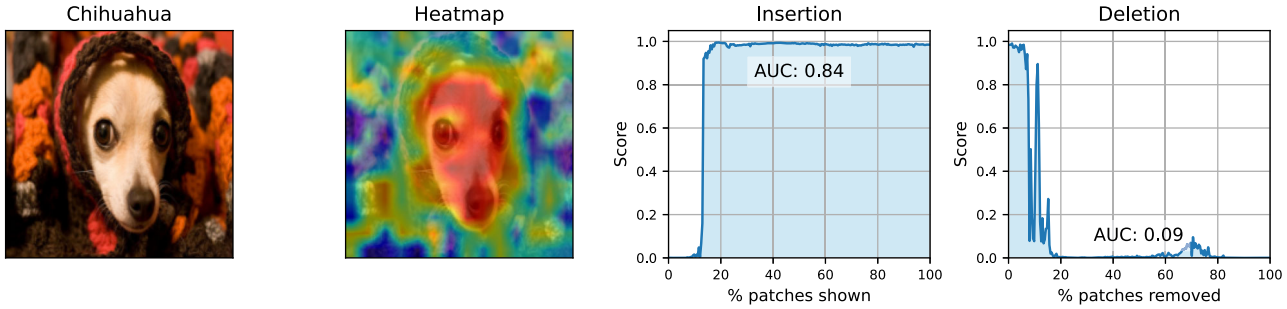


(a) Comparison with classical approaches

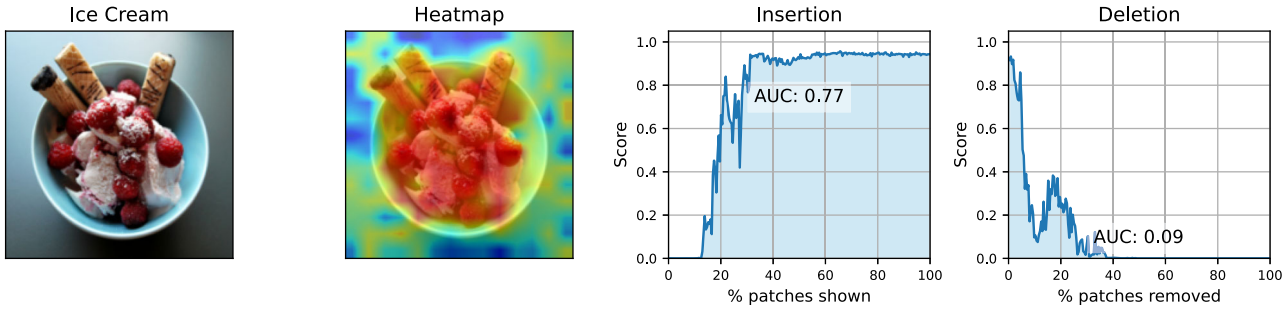


(b) Comparison with ViT-specific approaches

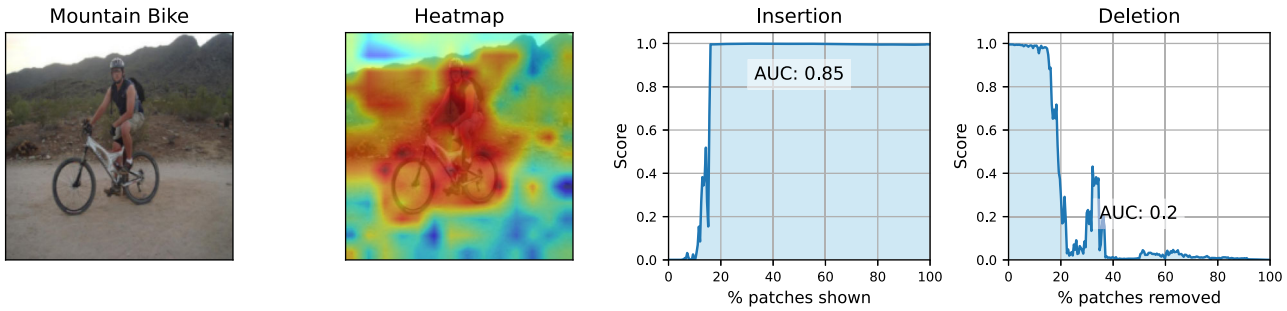
Fig. 4 Comparison of saliency maps returned by MUTEX and related approaches when applied on four images of ImageNet using ViT as vision transformer model. The figure shows that, for each image, MUTEX is able to generate a saliency map that highlights the parts of the image that contributed most to the model decision. The maps generated by MUTEX generally better emphasize the region on which the model is focused, providing more interpretable explanations than both classical methods (shown in Fig. 4a) and those developed specifically for ViT (shown in Fig. 4b).



(a) Original image of *Chihuahua*, heatmap, Insertion and Deletion curves



(b) Original image of *Ice Cream*, heatmap, Insertion and Deletion curves



(c) Original image of *Mountain Bike*, heatmap, Insertion and Deletion curves

Fig. 5 Application of MUTEX to three ImageNet images belonging to three different classes using the ViT model. The figure shows the original image, heatmaps highlighting the main subject and its features, and corresponding metrics. The Insertion curve rises rapidly with a small percentage of patches, while the Deletion curve decreases quickly, demonstrating the effectiveness of MUTEX.

mountain bike. Each selected image has its own features, backgrounds and dimensions of the object of interest. For each image, we calculated the corresponding heatmap, Insertion curve and Deletion curve obscuring the patches by applying the Black obfuscation method discussed above. For the Insertion and Deletion computation, we used d steps, where d is the number of tokens, i.e., 224. In this experiment, we used the two vision transformers mentioned in Sect. 4.1, namely ViT [4] and DeiT [52].

In Fig. 5, we show the results of MUTEX when the vision transformer used is ViT. Specifically, we considered three images with increasing difficulty of interpretation. The first image is a chihuahua; it is potentially very easy to recognize since only the chihuahua's head is seen in the foreground and the background does not refer to any other object. The second image is a bowl of ice cream; it is easy to recognize, but more difficult than the previous

image. In fact, the bowl is unique and there is no background; however, there are several sunburst cookies sticking out of the bowl, so while recognizing the inside of the bowl is easy, identifying the sunburst cookies on the outside may be difficult. The third image was deliberately chosen as one of the most difficult to interpret because it contains two different objects, namely the person and the mountain bike. In addition, the background of the image is complex and also contains two components, namely the road and the mountain. It is interesting to note that the latter can contribute to the classification of the mountain bike, since it is a typical background for this object.

From the analysis of this figure, we can first notice that the heatmaps effectively highlight the regions containing the object. This can also be witnessed by the high values of the AUC associated with the Insertion curves and the low values of the AUC associated with the Deletion curves.

Specifically, Fig. 5a shows that MUTEX correctly identifies the salient areas, as the dog's snout is particularly highlighted while the background is given little consideration. The Insertion curve reaches a high value with 15% of the patches, and the corresponding AUC value is 0.84. In contrast, the Deletion curve drops to 10% with a corresponding AUC value of 0.09. Both results confirm that ViT classifies this image with very few patches identified by MUTEX.

In Fig. 5b, the salient area covers the entire bowl of ice cream, with different nuances depending on the importance of the parts, while the background is given a low weight. Again, the Insertion curve shows that ViT is able to classify this image with 20% of the patches and reaches an AUC value of 0.77. The Deletion curve drops to 10% of the patches and the corresponding AUC value is 0.09.

Finally, in Fig. 5c, the heatmap includes the person, the bicycle, and some background. Interestingly, the importance that MUTEX assigns to the various areas is different; in particular, it assigns more importance to the bicycle components (such as the bicycle tire and frame); furthermore, it assigns a certain importance to the background mountain because this object can help to classify the mountain bike. So, MUTEX is able to identify the salient areas of the figure and the proper weight to assign to each of them. Looking at the Insertion curve, we can see that ViT is able to correctly classify the image with 20% of the patches. The corresponding AUC value is 0.85. The Deletion curve remains high up to 20% of the patches and the corresponding AUC value is 0.20. It is worth noting that, despite the fact that we deliberately chose an image difficult to interpret (see above), and despite the fact that at the qualitative level the result is worse than the other two images (because the mountain bike, the man and part of the mountains in the background are highlighted), the result obtained at the quantitative level is satisfactory.

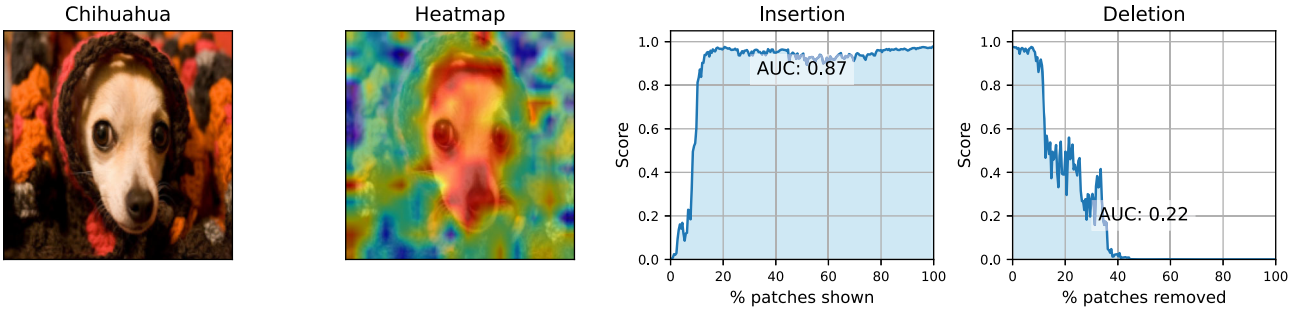
In Fig. 6, we show the heatmaps, Insertion curves and Deletion curves obtained by applying MUTEX to the same images as in Fig. 5, but using DeiT instead of ViT as the vision transformer. From the analysis of this figure, we can see that the heatmaps generally highlight the correct part of the images. We can also see that the AUC values associated with the Insertion curve and the Deletion curve are both higher than the corresponding values seen in Fig. 5.

Specifically, in Fig. 6a, the heatmap correctly identifies the object, although in this case it focuses on the higher parts of the dog's snout, while ViT mainly looked at the lower parts of it. The Insertion curve shows a high score value already with 10% of the patches while the Deletion curve drops with 20-40% of the patches removed. This results in an AUC value of 0.87 for the Insertion curve and an AUC value of 0.22 for the Deletion curve. Again, we see a high value for Insertion–Deletion.

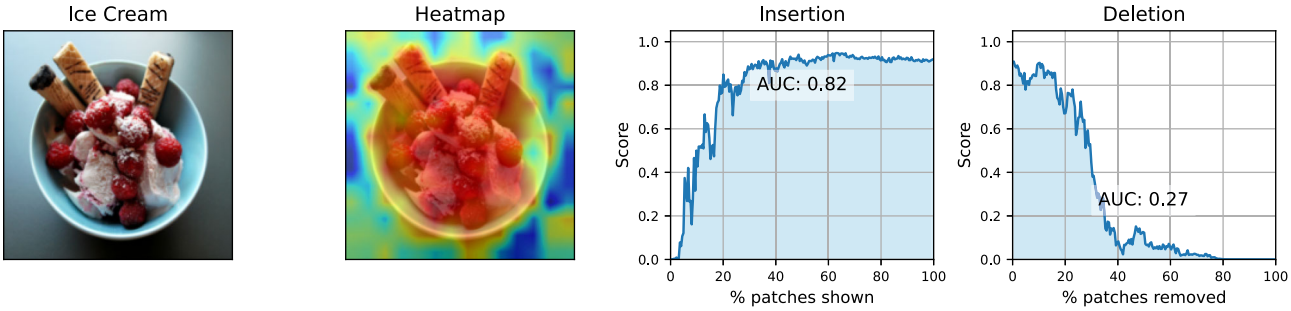
In Fig. 6b, we observe a similar result as in Fig. 5b . Again, the heatmap covers the whole bowl, the Insertion curve reaches a high value with 20% of the patches, while the Deletion curve drops to 20-40% of the removed patches. The AUC value is 0.82 for the Insertion curve and 0.27 for the Deletion curve.

Finally, in Fig. 6c, the heatmap shows more emphasis on the background and the man riding the bicycle than the heatmap in Fig. 5c . Again, examining the highlighted parts shows that MUTEX is able to give different importance to different parts of the image by putting more emphasis on the bicycle components (in particular, the bicycle tire, frame and pedals). For this image, the Insertion AUC is very high (i.e., 0.95) and this is due to the fact that DeiT is able to provide a correct classification with a very limited number of patches (i.e., 1-5%) selected

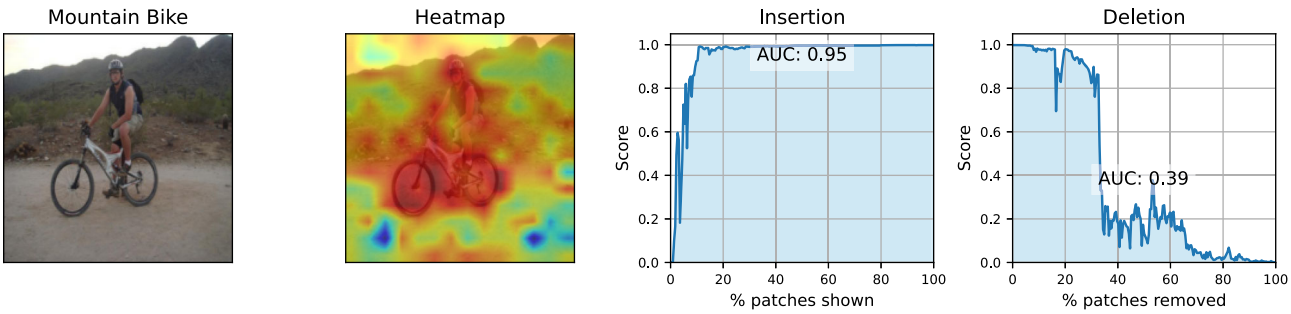
https://doi.org/10.1007/s00521-025-11591-x



(a) Original image of *Chihuahua*, heatmap, Insertion and Deletion curves



(b) Original image of *Ice Cream*, heatmap, Insertion and Deletion curves



(c) Original image of *Mountain Bike*, heatmap, Insertion and Deletion curves

Fig. 6 Application of MUTEX to three ImageNet images belonging to three different classes using the DeiT model. The figure shows the original images, heatmaps highlighting the main subject and its features, and the corresponding metrics. The Insertion curve increases sharply with only a small percentage of patches, while the Deletion curve drops quickly, showing the effectiveness of MUTEX with the DeiT model.

by MUTEX. However, the Deletion curve decreases slowly and has a drop only at 35% of the patches, resulting in an AUC value of 0.39.

This result is in line with the quantitative evaluations obtained for DeiT supported by MUTEX, which had given us very high values for Insertion and high values for Deletion, which still led to high values of the difference between Insertion and Deletion (see Sect. 4.2).

At the end of this analysis, we can see that MUTEX is able to work with different vision transformer models and to identify the salient areas of the images that contain the object to be classified. Insertion curves grew very fast, resulting in high AUC values. In contrast, the AUC values of the Deletion curves are low in some cases (e.g.,

chihuahua and ice cream images with ViT) and high in others. However, we note that in all cases the values of Insertion–Deletion remain high enough to make MUTEX competitive with existing approaches.

4.4 Hyperparameter and ablation study

In this section, we present our hyperparameter and ablation study to thoroughly evaluate the performance of MUTEX on ImageNet according to the explainability metrics. Due to space limitations, we show our analyses only with respect to ViT and the Black obfuscation method.

As a first step, we perform a hyperparameter study by testing different values of τ; recall that this hyperparameter allows us to tune the attention values that are retained and those that are filtered out (see Sect. 3.3). We also recall that such a filtering operation is useful to mitigate the noise in the distribution of attention values and to reduce execution time. Table 9 shows the results of this hyperparameter study; here, we indicate in bold the best values.

From the analysis of this table, we can see that the best values of Insertion, Deletion, and Insertion–Deletion are obtained for low values of τ. In particular, the optimal value is obtained when $\tau = 0.01$. Regarding this result, we note that a very low value of τ allows the removal of noise in the attention weight distribution. Not only does this not affect the performance in terms of Insertion, Deletion, and Insertion–Deletion, but improves it allowing also to save some time. On the other hand, when τ starts to grow, the performance in terms of Insertion, Deletion and Insertion–Deletion starts to deteriorate. This can be explained by the fact that in this case too many arcs are cut in the multiplex network, and thus MUTEX and the associated ViT become less able to identify salient areas.

As for the ablation study, we evaluate the performance of MUTEX in the case where only one type of weighted degree centrality (i.e., only weighted indegree centrality or only weighted outdegree centrality) is used to compute the importance of patches. We show the results in Table 10. In the same table, we report in bold the performance of MUTEX obtained in Sect. 4.2 using both weighted degree centralities. From the analysis of this table, we note that weighted indegree centrality gives MUTEX a higher performance than weighted outdegree centrality. In fact, the value of Insertion is higher, the value of Deletion is lower, and especially the difference between Insertion and Deletion is higher.

A possible reason for the superior performance of weighted indegree centrality lies in the interpretation of this measure. Specifically, it represents the total attention that a patch p_i receives from all the other patches in a layer of the multiplex network. A high value of weighted indegree centrality indicates that p_i is considered important by other patches, making it a good metric for identifying the most significant patches based on their influence on others. In contrast, the weighted outdegree centrality of a patch p_i reflects the attention it gives to all the other

Table 9 Study of the hyperparameter τ using the ViT model and the Black obfuscation method on ImageNet

τ	Insertion ↑	Deletion ↓	Insertion–Deletion ↑
0	0.64	0.22	0.42
0.010	**0.65**	**0.22**	**0.43**
0.030	0.64	0.22	0.41
0.050	0.63	0.23	0.40
0.075	0.62	0.24	0.38
0.100	0.61	0.25	0.36
0.150	0.60	0.26	0.34
0.200	0.58	0.26	0.32

As τ increases, the Insertion metric initially rises slightly and then falls, the Deletion metric continues to rise, while the Insertion–Deletion metric first increases slightly before decreasing. When $\tau = 0.010$, we get the optimal value of all the three metrics, which means that with this value of τ we get the best balance and performance

Table 10 Ablation study on centrality measures using the ViT model and the Black obfuscation method on ImageNet

Centrality	Insertion ↑	Deletion ↓	Insertion–Deletion ↑
Weighted Indegree	0.63	0.24	0.39
Weighted Outdegree	0.56	0.39	0.24
Weighted Indegree and Weighted Outdegree	**0.65**	**0.22**	**0.43**

The study evaluates the impact of Weighted Indegree alone, Weighted Outdegree alone, and Weighted Indegree and Weighted Outdegree together. The results show that the combined use of the two centrality measures achieves the best performance for all metrics, emphasizing the complementary role that the two measures play in MUTEX

Table 11 Ablation study on the choice of Φ using the ViT model and the Black obfuscation method on ImageNet

Φ	Insertion ↑	Deletion ↓	Insertion–Deletion ↑
{max}	0.59	0.30	0.29
{min}	0.60	0.28	0.32
{sum}	0.60	0.27	0.33
{max, min}	0.63	0.24	0.39
{max, sum}	0.62	0.24	0.38
{min, sum}	0.63	0.23	0.40
{min, max, sum}	**0.65**	**0.22**	**0.43**

The table shows the results for different definitions of Φ, specifically for {max}, {min}, {sum}, and their possible combinations. The best performing configuration is {min, max, sum}, which manages to obtain the best values for all metrics. This highlights the effectiveness of combining different aggregation functions to enhance the interpretability and robustness of the model

patches. This measure does not capture the intrinsic importance of p_i itself, but rather provides a view of its interactions with other patches. Therefore, simply using weighted outdegree centrality to compute a patch's importance is insufficient, as it only reflects the total attention paid by p_i to other patches rather than the own significance of p_i within the multiplex network.

However, the performance obtained using both centralities is better (albeit slightly better) than that obtained using weighted indegree centrality alone. This result is due to the fusion of the two perspectives provided by weighted indegree and weighted outdegree centralities applied to attention values.

Finally, we performed an ablation study on the functions for aggregating the matrices returned by the attention heads (i.e., the functions belonging to the set Φ in Algorithm 2). Clearly, the more functions we have within Φ, the more nuances of the patch interactions we can analyze and the more masks we can extract to create a better heatmap (see Algorithm 2 for all technical details). In Table 11, we report the performance of MUTEX for different configurations of Φ. Examining this table, we can see that Φ has an impact on the performance of MUTEX. In fact, when $|\Phi| = 1$, Insertion ranges from 0.59 to 0.60, Deletion ranges from 0.30 to 0.27, and finally Insertion–Deletion ranges from 0.29 to 0.33. With $|\Phi| = 1$, the highest performance is obtained when $\Phi = \{sum\}$, although this performance is worse than what MUTEX obtained in Table 1 where we set $\Phi = \{min, max, sum\}$. When $|\Phi| = 2$, we can observe that the performance improves since Insertion ranges from 0.62 to 0.63, Deletion ranges from 0.24 to 0.23, and Insertion–Deletion ranges from 0.38 to 0.40. The best performance is obtained when $|\Phi| = 3$, in which case Insertion is 0.65, Deletion is 0.22, and Insertion–Deletion is 0.43. This is also the case we had reported in Table 1. This is explained by the fact that in the latter case we have more masks and more ways to aggregate the attention matrices of the corresponding attention heads, which provides more flexibility and allows us to capture more nuances in the patch interactions.

5 Discussion

In this section, we explore the strengths and weaknesses of MUTEX and outline potential directions for future research. Specifically, Sect. 5.1 discusses the advantages of MUTEX, Sect. 5.2 describes its limitations, and Sect. 5.3 suggests some possible future developments.

5.1 Advantages

Experimental results show that MUTEX is able to identify the salient areas of an image that led a vision transformer to predict a particular class for it. MUTEX considers the importance of the attention layers of ViTs as in previous approaches [9, 10, 31], but models attention matrices as multiplex networks to highlight relationships between parts of the input. It then uses the concepts of mask perturbation [16–18] and the coverage bias formula [16] to generate a valid heatmap using the multiplex network-based representation defined previously.

MUTEX introduces several significant benefits over existing explainability approaches for Vision Transformers. It is the first approach to use a multiplex network-based representation of a Vision Transformer. This innovative representation allows for a detailed analysis of the relationships between image patches as the vision transformer processes input data. By modeling attention matrices as multiplex networks, MUTEX effectively highlights the connections between different parts of the input, thereby enhancing the interpretability of the model's decision-making process.

Another key advantage of MUTEX is its efficiency in generating masks. Compared to other perturbation-based approaches, such as [16, 17], MUTEX significantly reduces the number of masks needed to visually explain an image. In our experiments, it used only 54 masks to achieve state-of-the-art performance, while alternative perturbation methods required from 70 to 1024 masks. This feature is confirmed by the GFLOPs analysis, which highlights MUTEX as one of the most computationally efficient approaches while achieving the highest performance in terms of visual interpretability metrics. This balance between efficiency and effectiveness underscores MUTEX's ability to create high-quality explanations with minimal computational overhead, making it useful for real-world applications.

In addition, the centrality measures used by MUTEX, namely weighted indegree centrality and weighted outdegree centrality, are straightforward and easy to interpret. Indegree centrality reflects the degree to which a token is important to the others, while outdegree centrality represents the attention a token gives to the others. These measures, combined with aggregation functions, facilitate intuitive evaluation of token behavior across layers. This makes MUTEX's output more accessible and understandable to users, enhancing its usefulness as an explainability tool. Moreover, the set of aggregation functions strikes a balance between computational efficiency and interpretability, as they are both easy to compute and straightforward to interpret. At the same time, this set is not closed, but open for future extensions with additional aggregators.

Despite the use of a multiplex network, MUTEX maintains a manageable number of nodes and arcs that correspond directly to the tokens processed by the ViT. This balance ensures that the network remains interpretable without overwhelming the user with excessive detail, thereby preserving the clarity and effectiveness of the explanations provided.

Finally, since MUTEX is a mask-based approach, it inherits the advantages of this category of explainability approaches. Indeed, approaches based on linear attribution, such as Integrated Gradients [6], have been criticized in the literature because they fail to provide meaningful insights into the model behavior, especially in nonlinear scenarios [8]. Similarly, attention-based approaches have been shown to be unreliable, as attention weights are often uncorrelated with model results [12]. Conversely, mask-based approaches, such as ViT-CX [16], TIS [17], and of course MUTEX, have shown superior performance in explainability tasks. These methods are well suited to capture and analyze complex, nonlinear relationships within the data, providing more reliable insights into the decision-making processes of neural models.

In conclusion, we observe that MUTEX integrates key ideas from the other categories of approaches. It effectively addresses the potential nonlinearity in patch relationships by constructing a multiplex network on top of attention matrices, which captures patch interactions across multiple layers. However, instead of relying only on attention weights, MUTEX generates a set of masks that are then fed into the Vision Transformer. This step is critical because it establishes a link between attention weights and model output, mitigating potential uncorrelation problems between them.

5.2 Limitations

In the previous section, we have seen that MUTEX has several advantages. However, it also has some limitations. First, it is currently incompatible with hierarchical ViT models, such as Swin [57]. These models involve progressive downsampling of feature maps and variable token counts across layers, whereas MUTEX requires the same number of tokens across all attention layers. This constraint prevents its direct application to hierarchical architectures.

Another limitation is the dependence of MUTEX on access to attention maps. Some ViT models do not provide direct access to their attention matrices, making it challenging to construct the corresponding multiplex network. Hook functions can be employed to capture these attention matrices, but this requires access to the internal structure of the model. In scenarios where such access is not feasible, the applicability of MUTEX is hindered. Therefore, the use of our framework is possible in those models where attention maps are readily available or can be extracted through additional mechanisms.

Finally, there are scalability concerns when applying MUTEX to very large Vision Transformers, with thousands of tokens and dozens of attention layers. As the size and complexity of these models increase, the multiplex network representation may become more computationally intensive and difficult to interpret. While MUTEX effectively handles complexity for standard models, scaling to exceptionally large or intricate models poses challenges both in terms of the computational resources required and the clarity of the explanations generated.

5.3 Future work

It is possible to identify several promising future research developments that can be pursued from MUTEX. A first development concerns the possibility of extending MUTEX to work with hierarchical Vision Transformers. To this end, we plan to replace the current multiplex network-based representation at the core of MUTEX with a multilayer network-based representation. This change in data structure will allow MUTEX to handle different numbers of tokens in different layers and allow flexible interlayer connections, making our framework compatible with hierarchical Vision Transformers.

Another interesting future development is the application of MUTEX to natural language processing transformers. These handle large numbers of input tokens, and adapting MUTEX to efficiently represent and manage the correspondingly large multiplex networks resulting from them will be essential to providing meaningful explanations in this domain. This extension will broaden the applicability of MUTEX beyond visual data.

Furthermore, MUTEX could be extended to cross-modal images to highlight their salient regions. Cross-modal images are those images captured by different modalities, such as images captured by an RGB camera and an infrared camera depicting the same scene or object [58, 59]. Since MUTEX constructs a set of multiplex networks for both a vision transformer and an image, there are several intuitive ways to adapt it for cross-modal images. For instance, MUTEX could be applied separately to each image in the cross-modal images and the heatmaps it returns could be merged. Alternatively, if multiple ViTs are used (e.g., one ViT per image), the corresponding multiplex networks could be constructed and then integrated to form a unified multiplex network. A further approach could be to create a set of masks for each modality and use them to create the final heatmap.

Finally, strategies for reducing the computational load of a ViT could be explored by leveraging a MUTEX-inspired approach. This could be used to identify and remove less relevant tokens based on a saliency metric derived from multiplex networks. This token pruning strategy has the potential to significantly reduce the computational cost of the ViT model by reducing the number of tokens to be processed, thereby improving efficiency.

6 Conclusion

In this paper, we proposed MUTEX, an explainability framework for vision transformers that operates through multiplex network-based representations of attention matrices. MUTEX receives a ViT and an image to be visually explained. As a first step, it creates a multiplex network-based representation of the attention layers by extracting the corresponding attention heads, aggregating the associated matrices, and creating a suitable multiplex network representing them. Next, it computes the importance of patches based on weighted indegree and weighted outdegree centralities and aggregates them using aggregation functions to obtain a set of weighted masks. It also uses the patch coverage bias formula to determine the importance of each patch based on its presence in the set of weighted masks. Finally, it applies a reshaping and bilinear interpolation algorithm to construct and return a heatmap of the same size as the input image.

We evaluated MUTEX on the ViT and DeiT vision transformer models using ImageNet and BloodMNIST. In these experiments, MUTEX achieved excellent performance in Insertion and Insertion–Deletion metrics, and satisfactory performance in Deletion metric. In addition, it returned very satisfactory results in the Faithfulness Violation Test, especially with the DeiT model. The Pointing Game metric confirmed that MUTEX effectively identifies the object of interest, while the GFLOPs analysis showed that it maintains a low computational cost. Qualitative analysis demonstrated the capabilities of MUTEX in three examples and highlighted its strengths. The hyperparameter analysis and the ablation study allowed us to analyze the impact of the few hyperparameters and model designs on MUTEX and identify the best possible configuration. Therefore, our experiments, which included quantitative evaluation, qualitative analysis, and ablation study, confirmed that MUTEX is able to address the vision transformer explainability task with satisfactory performance.

Acknowledgements We acknowledge the support of the PNRR project FAIR—Future AI Research (PE00000013), Spoke 9–AI, under the NRRP MUR program funded by the NextGenerationEU.

Funding Open access funding provided by Università Politecnica delle Marche within the CRUI-CARE Agreement.

Data availability The dataset used for our study is publicly available at the link https://github.com/DavideTraini/ViT-Visual-Interpretability.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Ma Y, Wang Z, Yang H, Yang L (2020) Artificial intelligence applications in the development of autonomous vehicles: a survey. IEEE/CAA J Automatica Sinica 7(2):315–329 (**IEEE**)
2. Loh HW, Ooi CP, Seoni S, Barua PD, Molinari F, Acharyaa UR (2022) Application of explainable artificial intelligence for healthcare: a systematic review of the last decade (2011–2022). Computer Methods and Programs in Biomedicine. Elsevier, Amsterdam, p 107161
3. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is All you Need. In: Proceedings of the international conference on advances in neural information processing systems (NIPS'17), p 30, Long Beach, CA, USA, Curran Associates
4. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al (202) An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929,
5. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the international IEEE conference on computer vision (ICCV'17), pp 618–626, Venice, Italy, IEEE
6. Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In: Proceedings of the international conference on machine learning (ICML'17), pp 3319–3328, Sydney, Australia, PMLR
7. Chefer H, Gur S, Wolf L (2021) Transformer interpretability beyond attention visualization. In: Proceedings of the international conference on computer vision and pattern recognition (CVPR'21), pp 782–791, Virtual event
8. Bilodeau B, Jaques N, Koh PW, Kim B (2024) Impossibility theorems for feature attribution. Proc Natl Acad Sci 121(2):e2304406120 (**National Acad Sciences**)
9. Abnar S, Zuidema W (2020) Quantifying attention flow in transformers. In: Proceedings of the annual meeting of the association for computational linguistics (ACL'20), pp 4190–4197, Virtual event, Association for Computational Linguistics
10. Barkan O, Hauon E, Caciularu A, Katz O, Malkiel I, Armstrong O, Koenigstein N (2021) Grad-Sam: explaining transformers via gradient self-attention maps. In: Proceedings of the ACM international conference on information & knowledge management (CIKM'21), pp 2882–2887, Goald Coast, Queensland, Australia,
11. Wu J, Duan B, Kang W, Tang H, Yan Y (2024) Token transformation matters: towards faithful post-hoc explanation for vision transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR'24), pp 10926–10935, Seattle, WA, USA
12. Jain S, Wallace B.C (2019) Attention is not explanation. In: Proceedings of the conference of the North American chapter of the association for computational linguistics: human language technologies (NAACL-HLT '19), pp 3543–3556, Minneapolis, MN, USA, ACL
13. Rigotti M, Miksovic C, Giurgiu I, Gschwind T, Scotton P (2021) Attention-based interpretability with concept transformers. In: Proceedings of the international conference on learning representations (ICLR'21), Virtual Event
14. Kim S, Nam J, Ko B.C (2022) Vit-net: interpretable vision transformers with neural tree decoder. In: Proceedings of the international conference on machine learning (ICML'22), pp 11162–11172, Baltimora, MD, USA, PMLR
15. Qiang Y, Li C, Khanduri P, Zhu D (2023) Interpretability-aware vision transformer. arXiv preprint arXiv:2309.08035
16. Englebert A, Stassin S, Nanfack G, Mahmoudi SA, Siebert X, Cornu O, De Vleeschouwer C (2023) Explaining through transformer input sampling. In: Proceedings of the international conference on computer vision (ICCV'23), pp 806–815. Paris, France
17. Xie W, Li X, Cao CC, Zhang NL (2023) ViT-CX: causal explanation of vision transformers. In: Proceedings of the international joint conference on artificial intelligence (IJCAI'23), pp 1569–1577, Macao, China
18. Petsiuk V, Das A, Saenko K (2018) RISE: Randomized input sampling for explanation of black-box models. arXiv preprint arXiv:1806.07421
19. Chefer H, Gur S, Wolf L (2021) Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV'21), pp 397–406, Virtual event
20. Alicioglu G, Sun B (2022) A survey of visual analytics for explainable artificial intelligence methods. Comput Graph 102:502–520 (**Elsevier**)
21. Covert I, Kim C, Lee S (2022) Learning to estimate shapley values with vision transformers. arXiv preprint arXiv:2206.05282
22. Voita E, Talbot D, Moiseev F, Sennrich R, Titov I (2019) Analyzing multi-head self-attention: specialized heads do the heavy lifting, the rest can be pruned. In: Proceedings of the annual meeting of the association for computational linguistics (ACL'19), pp 5797–5808, Florence, Italy
23. Chattopadhay A, Sarkar A, Howlader P, Balasubramanian VN (2018) Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. In: Proceedings of the international IEEE conference on applications of computer vision (WACV'18), pp 839–847, Lake Tahoe, NV/CA, USA, IEEE

24. Wang H, Wang Z, Du M, Yang F, Zhang Z, Ding S, Mardziel P, Hu X (2020) Score-CAM: score-weighted visual explanations for convolutional neural networks. In: Proceedings of the international IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW'20), pp 111–119, Los Alamitos, CA, USA, IEEE Computer Society
25. Smilkov D, Thorat N, Kim B, Viégas F, Wattenberg M (2017) Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825
26. Leem S, Seo H (2024) Attention guided CAM: visual explanations of vision transformer guided by self-attention. In: Proceedings of the international conference on artificial intelligence (AAAI'24), vol 38, pp 2956–2964, Vancouver, British Columbia, Canada
27. Lundberg S.M, Lee SI (2017) A unified approach to interpreting model predictions. In: Proceedings of the International conference on neural information processing systems (NIPS'17), pp 4768–4777, Long Beach, CA, USA, Curran Associates
28. Bach S, Binder A, Montavon G, Klauschen F, Müller KR, Samek W (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE 10(7):e0130140 (**Public Library of Science**)
29. Sun T, Chen H, Qiu Y, Zhao C (2023) Efficient shapley values calculation for transformer explainability. In: Proceedings of the Asian conference on pattern recognition (ACPR'23), pp 54–67, Kitakyushu, Japan, Springer
30. Hao Y, Dong L, Wei F, Xu K (2021) Self-attention attribution: interpreting information interactions inside transformer. In: Proceedings of the AAAI international conference on artificial intelligence (AAAI'21), vol 35, pp 12963–12971, Virtual event
31. Yuan T, Li X, Xiong H, Cao H, Dou D (2021) Explaining information flow inside vision transformers using Markov chain. In: Proceedings of the international workshop on eXplainable AI approaches for debugging and diagnosis (XAI4Debugging@NeurIPS2021), Virtual event, 2021
32. Chen J, Li X, Yu L, Dou D, Xiong H (2023) Beyond intuition: rethinking token attributions inside transformers. Trans Mach Learn Res, https://openreview.net/forum?id=rm0zIzlhcX
33. Qiang Y, Pan D, Li C, Li X, Jang R, Zhu D (2022) Attcat: explaining transformers via attentive class activation tokens. In: Proceedings of the international conference on neural information processing systems (NIPS'22), vol 35, pp 5052–5064, New Orleans, LA, USA
34. Huang Y, Jia A, Zhang X, Zhang J (2023) Generic attention-model explainability by weighted relevance accumulation. In: Proceedings of the ACM international conference on multimedia in Asia (MMAsia'23), pp 1–7, Tainan, Taiwan
35. Bousselham W, Petersen F, Ferrari V, Kuehne H (2024) Grounding everything: emerging localization properties in vision-language transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR'24), pp 3828–3837, Seattle, WA, USA
36. Albano R, Giusti L, Maiorana E, Campisi P (2024) Explainable vision transformers for vein biometric recognition. IEEE Access 12:60436–60446 (**IEEE**)
37. Bibal A, Cardon R, Alfter D, Wilkens R, Wang X, François T, Watrin P (2022) Is attention explanation? An introduction to the debate. In: Proceedings of the annual meeting of the association for computational linguistics (ACL'23), pp Vol 1, 3889–3900, Dublin, Ireland
38. Beyer L, Zhai X, Kolesnikov A (2022) Better plain ViT baselines for ImageNet-1k. arXiv preprint arXiv:2205.01580
39. Serrano S, Smith NA (2019) Is attention interpretable? In: Proceedings of the annual meeting of the association for computational linguistics (ACL '19), pp 2931–2951, Firenze, Italy, ACL
40. Haurum JB, Escalera S, Taylor GW, Moeslund TB (2023) Which tokens to use? investigating token reduction in vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV'23), pp 773–783, Paris, France
41. Fayyaz M, Koohpayegani SA, Jafari FR, Sengupta S, Joze HRV, Sommerlade E, Pirsiavash H, Gall J (2022) Adaptive token sampling for efficient vision transformers. In: Proceedings of the European conference on computer vision (ECCV'22), pp 396–414, Tel Aviv, Israel, Springer
42. Choi H, Jin S, Han K (2024) ICEv2: interpretability, comprehensiveness, and explainability in vision transformer. International Journal of Computer Vision, pp 1–18, Springer
43. Pan B, Panda R, Jiang Y, Wang Z, Feris R, Oliva A (2021) IA-RED^2: interpretability-aware redundancy reduction for vision transformers. In: Proceedings of the international conference on neural information processing systems (NIPS'21), vol 34, pp 24898–24911, Virtual event
44. Böhle M, Fritz M, Schiele B (2023) Holistically explainable vision transformers. arXiv preprint arXiv:2301.08669
45. Yu L, Xiang W, Fang J, Chen YPP, Chi L (2023) ex-vit: a novel explainable vision transformer for weakly supervised semantic segmentation. Pattern Recogn 142:109666 (**Elsevier**)
46. Ding W, Cheng X, Geng Y, Huang J, Ju H (2024) C2F-explainer: explaining transformers better through a coarse-to-fine strategy. IEEE Trans Knowl Data Eng 36:7708–7724
47. Niu Y, Ding M, Ge M, Karlsson R, Zhang Y, Carballo A, Takeda K (2024) R-cut: enhancing explainability in vision transformers with relationship weighted out and cut. Sensors 24(9):2695 (**MDPI**)
48. Tan N, Bensemann J, Benavides-Prado D, Chen Y, Gahegan M, Lee L, Peng AY, Riddle P, Witbrock M (2021) An explainability analysis of a sentiment prediction task using a transformer-based attention filter. In: Proceedings of the annual conference on advances in cognitive systems (ACS'21), pp 1–7, Virtual event

49. Boccaletti S, Bianconi G, Criado R, Del Genio CI, Gómez-Gardenes J, Romance M, Sendina-Nadal I, Wang Z, Zanin M (2014) The structure and dynamics of multilayer networks. Phys Rep 544(1):1–122 (**Elsevier**)
50. Kanawati R (2015) Multiplex network mining: a brief survey. IEEE Intell Inform Bull 16(1):24–27
51. Battiston F, Nicosia V, Latora V (2014) Structural measures for multiplex networks. Phys Rev E 89(3):032804 (**APS**)
52. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégo H (2021) Training data-efficient image transformers & distillation through attention. In: Proceedings of the international conference on machine learning (ICML'21), pp 10347–10357, Virtual event, PMLR
53. Liu Y, Li H, Guo Y, Kong C, Li J, Wang S (2022) Rethinking attention-model explainability through faithfulness violation test. In: Proceedings of the international conference on machine learning (ICML'22), pp 13807–13824, Baltimora, MD, USA, PMLR
54. Wu J, Kang W, Tang H, Hong Y, Yan Y (2024) On the faithfulness of vision transformer explanations. In: Proceedings of the international conference on computer vision and pattern recognition (CVPR'24), pp 10936–10945, Seattle, WA, USA
55. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) Imagenet large scale visual recognition challenge. Int J Comput Vision 115:211–252 (**Springer**)
56. Yang J, Shi R, Wei D, Liu Z, Zhao L, Ke B, Pfister H, Ni B (2023) Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. Sci Data 10(1):41 (**Nature Publishing Group UK London**)
57. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV'21), pp 10012–10022, Virtual event
58. Fang A, Zhao X, Yang J, Zhang Y, Zheng X (2021) Non-linear and selective fusion of cross-modal images. Pattern Recogn 119:108042 (**Elsevier**)
59. Liu F, Gao C, Sun Y, Zhao Y, Yang F, Qin A, Meng D (2021) Infrared and visible cross-modal image retrieval through shared features. IEEE Trans Circuits Syst Video Technol 31(11):4485–4496 (**IEEE**)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Michele Marchetti[1] · Davide Traini[1,2] · Domenico Ursino[1] · Luca Virgili[1]

✉ Luca Virgili
luca.virgili@univpm.it

Michele Marchetti
m.marchetti@pm.univpm.it

Davide Traini
davide.traini@unimore.it

Domenico Ursino
d.ursino@univpm.it

[1] DII, Polytechnic University of Marche, Via Brecce Bianche, 60131 Ancona, Italy

[2] CHIMOMO, University of Modena and Reggio Emilia, Largo del Pozzo, 41125 Modena, Italy