ORIGINAL ARTICLE

Generating masks from similarity-based graphs for vision transformer explainability

Michele Marchetti · Davide Traini · Domenico Ursino · Luca Virgili

Received: 26 October 2025 / Accepted: 12 July 2026

Abstract
Due to their outstanding performance, Vision Transformers (ViTs) are becoming one of the most widely used architectures for computer vision tasks. However, the interpretation of the output returned by a ViT is still a challenging problem, and yet its solution is crucial for users to gain confidence in these new architectures. In this setting, some methods to address this issue rely only on attention scores; they cannot offer any causal guarantee because the highlighted patches do not necessarily drive the prediction. Other methods use mask-based perturbations; they are computationally expensive because they must repeatedly test many mask combinations of the image and rerun the model to measure confidence changes. To address these limitations, we present SImilarity-based GRAphs for Vision Transformer Explainability (SIGRATE), a hybrid framework that combines attention cues with graph-based mask generation to leverage the strengths of both approaches and overcome their weaknesses. For each attention layer of a ViT, SIGRATE extracts the embeddings of the image patches and constructs the corresponding similarity graph; then, it generates a set of binary masks starting from specific patches and performing walks in the similarity graph. Finally, it merges the masks of all attention layers into a heatmap using the coverage bias formula. We have tested SIGRATE on two ViT models (i.e., ViT-Base and DeiT-Base) and on two datasets, i.e., a subset of the ImageNet validation set and BloodMNIST. Our experiments show that SIGRATE provides promising performance in terms of Insertion, Deletion, and Pointing Game. Finally, we have performed a qualitative analysis demonstrating SIGRATE's capabilities, as well as a hyperparameter analysis and an ablation study showing the impact of design choices on SIGRATE's performance and efficiency.

Keywords Visual explainability · Vision transformer · Patch embedding · Similarity graph · Computer vision

1 Introduction

Vision Transformers (ViTs) [1] have quickly become a reference architecture for computer vision tasks, flanking and sometimes replacing traditional Convolutional Neural Networks (CNNs) [2].

The self-attention mechanism on which they are based allows them to model long-range dependencies between regions of an image. This makes them achieve high performance in image classification, object detection,

Michele Marchetti, Davide Traini and Domenico Ursino have contributed equally to this work.

Published online: 22 August 2026

segmentation, and, more generally, in any task where global context must be taken into account [3–6]. Unlike CNNs, which use local convolutional filters for feature extraction, ViTs consider the entire image as a sequence of patches. However, this way of proceeding increases the complexity of the process and leads to major challenges in interpreting how ViTs make their inferences. This is critical for applications where transparency and trust must be achieved.

The need to make deep learning predictions explainable is ever-growing since they impact an increasing number of critical sectors, such as healthcare, autonomous driving, and security systems [7–9]. Model explainability can also help create trust in safe and effective deployment. For this reason, many different methods have been developed within classical deep learning architectures to provide at least some insight into the decision-making process [10, 11]. However, the peculiar architecture of ViTs, based on self-attention mechanisms, makes their interpretation even more difficult [12–14]. In fact, in contrast to earlier models, which attend mostly to specific regions of the input, the attention of ViTs lies across the entire image, making it difficult to identify the critical factors that led to their decisions. It is precisely this complexity that has led to the need for new explainability techniques capable of taking the peculiarities of ViTs into account. Some of them adapt to the ViT context the explainability techniques designed for Transformer models operating in the context of Natural Language Processing. Other techniques extend explainability approaches designed for other deep learning models, such as CNNs, to ViTs. Finally, other techniques are specifically designed for ViTs and are not derived from the adaptation of previous approaches. In any case, all explainability techniques working on ViTs can be grouped into three strands [15]. The first strand includes attribute-based techniques that focus on identifying the most influential input elements responsible for specific inferences [16]. The second strand comprises techniques that use the Transformer's self-attention mechanism to interpret how attention patterns affect model predictions [17]. The third strand includes techniques that work by masking parts of the input and evaluating the output of the resulting model, which helps to assess the importance of different input regions and their contribution to the decision-making process [18].

However, the latter two strands have some limitations. Attention-based explanations provide no causal guarantee that the highlighted patches actually drive the prediction, and highly attended regions can sometimes be removed without changing the output [19, 20]. Instead, mask-based explanations are computationally expensive because the construction of a mask requires a certain number of forward passes, which can result in thousands of forward passes for each image.

To address these limitations and overcome the corresponding weaknesses, we introduce SImilarity-based GRAphs for Vision Transformer Explainability (SIGRATE), a hybrid explainability framework combining attention information and graph-based mask generation. In SIGRATE, each ViT layer is modeled as a patch-similarity graph whose edge weights represent cosine similarity of patch embeddings. SIGRATE selects the most and least relevant patches according to the attention score provided by the classification token (CLS). Then, it uses these patches to initialize a greedy walk that adds the most similar neighbors iteratively while reducing the weight of the edges that have already been visited. The walk identifies semantically similar regions and generates masks highlighting different aspects of the image, such as foreground, context, and background. Afterward, SIGRATE weights these masks to explain the model's prediction. Finally, it merges the masks from all attention layers to create an overall heatmap that visually represents the areas most critical to the model's prediction. This fusion combines the global context captured by attention with the causal insights of the perturbation, resulting in better explanation heatmaps.

Beyond the technical components described above, the key innovation of SIGRATE lies in the explainability mechanism that it introduces. Unlike existing attention- or perturbation-based approaches, SIGRATE neither directly uses attention weights nor relies on random mask sampling. Rather, it generates explanation heatmaps through similarity-graph traversal, where greedy walks extract coherent regions of patches identified through embedding similarity. To the best of our knowledge, this graph-based mask generation mechanism has not been explored in prior ViT explainability methods, making it an original contribution of SIGRATE. Furthermore, SIGRATE integrates CLS-driven relevance with graph-derived masks, combining global importance cues with

the similarity structure encoded in patch embeddings. This design defines a new explainability framework that generates coherent and semantically meaningful heatmaps.

According to what is typically done in the literature [21, 22], we evaluated SIGRATE's performance on two widely used ViT backbones (i.e., ViT-Base and DeiT-Base) using a subset of images from the ImageNet validation set. Moreover, we evaluated SIGRATE's performance using the BloodMNIST dataset [23] as well, to analyze its generalizability across different domains. We also employed three standard explainability metrics, i.e., Insertion, Deletion and Pointing Game. We found that, in most cases, SIGRATE consistently outperformed existing approaches while requiring fewer computational resources. Furthermore, we performed a qualitative analysis to highlight SIGRATE's capabilities. Finally, we conducted a failure case analysis as well as an extensive hyperparameter and ablation studies to quantify the influence of each design choice on the effectiveness and efficiency of our framework.

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

- We combine: *(i)* modeling ViT token embeddings as a patch-level similarity graph, and *(ii)* adapting a greedy walk algorithm, to cluster highly similar patches.
- We fuse CLS-token attention insights with mask-based perturbations to generate saliency masks that capture both global importance and local patch relationships, which generates more coherent explanation heatmaps.
- We conduct extensive experiments on two ViT models using a subset of ImageNet and the BloodMNIST dataset to evaluate the performance of SIGRATE in terms of Insertion, Deletion, and Pointing Game. Furthermore, we provide a comprehensive hyperparameter analysis and ablation study to quantify the contribution of each component.

The rest of this paper is organized as follows: Section 2 reviews the existing literature on explainability methods for ViTs. Section 3 provides the technical description of SIGRATE. Section 4 presents the results of our experimental campaign, compares SIGRATE with other existing approaches in the literature on two different datasets, and illustrates a hyperparameter analysis and an ablation study. Section 5 provides a discussion of the results obtained. Finally, Sect. 6 draws our conclusions about our work and discusses some possible future developments.

2 Related works

As mentioned in the Introduction, the literature on visual explainability of ViTs falls into three main categories: *(i)* attribution-based, *(ii)* attention-based, and *(iii)* mask-based. These categories offer distinct perspectives on model interpretability. Attribution- and attention-based approaches are among the most widely adopted, while mask-based approaches are gaining relevance due to their ability to provide causal insights into model behavior [15].

The goal of attribution-based approaches is to identify the influential parts of the input data that lead to a specific classification [24–26]. These methods were initially proposed for CNNs. For instance, Grad-CAM [25] generates a coarse localization map by weighting the feature maps of the last convolutional layer with the corresponding gradients. This process identifies regions that positively impact the target class. Integrated Gradients (IntegratedGrad) [27] uses a rigorous approach to overcome the gradient saturation problem. It calculates the path integral of gradients along a linear interpolation from a non-informative base input to the actual input. This ensures that the attribution satisfies fundamental axioms for attribution methods, such as sensitivity and implementation invariance. Similarly, SmoothGrad [26] addresses the problem of visual noise in sensitivity maps by assuming that the signal of interest is stable while noise fluctuates. Therefore, it repeatedly adds Gaussian noise to the input image and averages the resulting gradients to produce a clearer explanation. Attribution-based approaches were initially proposed for CNNs only and were subsequently adapted to ViTs. An example is described in [16], where the authors propose an attention masking strategy to generate Shapley scores [28] for ViT explanations. In [29], the authors use Layer-wise Relevance Propagation (LRP) [30] to identify the most important attention

heads within Transformer layers. In [17], the authors extend LRP-based relevance computation by applying it to each attention head across all layers and by including a relevance propagation rule that considers both positive and negative attributions. In this way, they obtain class-specific visualizations. In [31, 32], the authors integrate intermediate representations of the ViT gradients to generate explanation heatmaps. This integration facilitates the extraction of insights from each attention map of the ViT. In [33], the authors introduce Explaining using Verified perturbation Analysis (EVA), an attribution method that ensures that all possible perturbations have been exhaustively searched. To this end, the approach propagates perturbation bounds through a neural network to test a set of perturbations in a single forward pass. In [34], the authors propose Iterated Integrated Attribution (IIA), which is similar to IntegratedGrad but more general than it. IIA iteratively integrates the input image, the learned representations returned by the model, and the corresponding gradients to generate an explanation heatmap. Recently, other CAM-inspired methods have been developed. KPCA-CAM [35] replaces standard PCA with Kernel PCA to generate class activation maps that capture nonlinear relationships in layer activations. On the other hand, Finer-CAM [36] extends Grad-CAM by calculating the gradients not only of the logit of the target class, but also of the difference between the logit of the target class and the logits of visually similar classes. In this way, it highlights truly discriminative regions by excluding shared features and provides more accurate and targeted explanations for fine-grained classification tasks.

Attention-based explainability approaches exploit the self-attention mechanism of ViT architectures [37–39]. For example, the authors of [40] present two approaches, known as Attention Rollout and Attention Flow, which provide insights into the information flow of a ViT. Attention Rollout models information flow by tracking how input token identities are propagated to higher layers. It considers the model as a Directed Acyclic Graph (DAG) and determines the weight of a path by multiplying the attention weights along its edges. The total influence of an input token on a higher-layer embedding is then calculated by multiplying the attention matrices of all intermediate layers and then summing over all possible paths. Importantly, to capture full signal propagation, Attention Rollout explicitly accounts for residual connections by augmenting the attention matrices with an identity matrix. Attention Flow takes a different perspective by using the attention graph as a flow network with capacities and applying a maximum flow algorithm to quantify the contribution of input tokens. Both approaches were originally developed for Transformer models in Natural Language Processing, but have since been extended to ViTs. The authors of [41] introduce Gradient Self-Attention Maps (Grad-Sam), an approach that uses attention scores and gradients to understand the impact of changes in the focus of a model by examining the output that the model returns when it receives specific inputs. The authors of [42] propose an interpretation framework for estimating the contribution of individual tokens within a ViT model. This framework uses a two-step process that integrates attention perception and reasoning feedback by relying on the partial derivatives of the loss function associated with each token. The authors of [43] propose TokenTM, a post-hoc explanation method by which token transformation across ViT layers can be integrated into the interpretation approach. TokenTM quantifies token transformation by measuring changes in token lengths and correlations in their directions before and after the transformation. It also integrates attention weights to provide a holistic view of token contributions to model prediction. The authors of [44] present LeGrad, an explainability method designed for ViTs. LeGrad calculates the gradient with respect to the attention maps of each model layer and interprets this value as a relevance signal that measures the sensitivity of the prediction to changes in attention patterns. For each layer, LeGrad extracts this signal, eliminates negative contributions, and aggregates it across heads and tokens. Then, it integrates the resulting maps across layers and combines them with intermediate and final activations to obtain the explanation. This multi-level formulation allows LeGrad to capture how ViTs distribute feature formation across network's depth. Using gradient sensitivity instead of raw attention reduces background artifacts related to token registers and produces maps that focus on regions actually influencing the model's output. The authors of "Explaining Information Flow Inside Vision Transformers Using Markov Chain" [45] propose Transition Attention Maps (TAM), which model the information flow in ViTs as a Markov process, mapping the progression of information based on the hidden states of tokens across layers. Another attention-based approach, called Gradient-Driven Multi-Head Attention Rollout (GMAR), is presented in [46]. GMAR improves the interpretability of ViTs by weighting the

contribution of individual attention heads. It calculates the gradients of the predicted class logit with respect to the attention maps and interprets these values as important signals that discriminate between different heads. These scores are normalized using L1 or L2 norms at each layer and applied to modulate the attention matrices before they are recursively aggregated across the network. This hybrid formulation combines the class-specific sensitivity of gradient-based approaches with the hierarchical information flow of attention rollout. The result consists of saliency maps that reflect the regions critical to the model's decision.

Mask-based methods selectively mask parts of the input data, such as pixels in an image or words in a text, and observe the effects on the model output. As for this strand, both adaptations of old methods to ViTs [18] and new methods have been proposed in the literature. For example, Randomized Input Sampling for Explanations (RISE) [18] performs explanations by applying random masks to the input images, obscuring different parts each time, and observing how these changes affect the model's prediction. A mask-based method developed specifically for ViT explainability is ViT-CX [47]. It focuses on the causal effects of patch embeddings on model output rather than on attention mechanisms. It generates and aggregates masks to produce saliency maps that reflect model output.

Another mask-based method similar to ViT-CX is Transformer Input Sampling (TIS), proposed in "Explaining through Transformer Input Sampling" [21]. It exploits the ability of ViTs to process a variable number of tokens while avoiding the introduction of misleading outlier features. TIS applies perturbations after linear projections and position encoding, but before the Transformer layer, so that perturbations do not generate outlier inputs that could affect the corresponding interpretation. ATMAN [48] is a perturbation-based approach to generative ViT that manipulates the Transformer's attention mechanisms to generate heatmaps for input with respect to output. To identify tokens that produce a target, ATMAN relies on the detection of token neighborhoods performed by computing the cosine similarity of the corresponding embeddings. In this way, it obtains similar tokens that could have been used during the generation of a part of the image. In [49], the authors propose VISION DIFFMASK, a post-hoc explainability method that uses hidden layer activations to predict the relevant parts of the input according to a given classification. VISION DIFFMASK uses a gating mechanism to identify the minimal subset of the original input that preserves the predicted distribution over classes. In [50], the authors define class distortion as the set of unexpected changes in the network prediction during the mask optimization process. They then propose a visual interpretation framework that is robust to class distortion. For this purpose, they use a cross-checking mask optimization strategy to perturb the target predictions without affecting non-target ones, which could lead to more reliable visual explanations.

Another method that leverages patch masking is Jigsaw-ViT [51]. Although its primary goal is not post-hoc explainability, it introduces an auxiliary, self-supervised task. In this task, the model learns to solve jigsaw puzzles by predicting the correct spatial arrangement of permuted image patches. This process involves applying masks and permutations to the input patches to force the ViT to develop a stronger understanding of spatial relationships and object compositions. As a result, Jigsaw-ViT can generate more semantically meaningful attention maps, which indirectly contributes to model transparency and robustness. Finally, R-Cut [52] does not belong to a specific category of explainability methods, but shares characteristics with 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 extracts finer details from these features.

SIGRATE is a hybrid approach that combines attention-based and mask-based explainability techniques. It uses attention scores from the CLS token to select the initial tokens for each walk. This use of attention allows it to identify the most and least relevant patches in an image according to the focus of the model. After this, SIGRATE generates masks over the walks created on token similarity graphs, making it inherently mask-based. These masks are then used to perturb the input, with the goal of discovering how different regions of the image contribute to model prediction. Compared to recent mask-based approaches, such as TIS, ViT-CX, ATMAN, and VISION DIFFMASK, SIGRATE appears to be more comprehensive and nuanced. While TIS perturbs input tokens to measure their effects, and ViT-CX focuses on the causal effects of patch embeddings, SIGRATE uses

attention to guide the mask generation process, allowing it to capture both direct input effects and relational information between multiple layers.

Compared to ATMAN, which manipulates attention mechanisms to generate a heatmap, and VISION DIFFMASK, which identifies minimal subsets of input to preserve predictions, SIGRATE integrates contributions from all attention layers to generate detailed and accurate saliency maps that reflect complex interactions within the ViT model.

3 Proposed approach

This section describes our approach. In particular, Sect. 3.1 provides a concise overview of it. Section 3.2 details how the similarity graph is constructed from the ViT model. Section 3.3 explains how patch importance is computed from graph walks and how the corresponding binary masks are generated. Finally, Sect. 3.4 describes how the final heatmap is obtained from the set of binary masks.

3.1 Overview

Before detailing the technical aspects of SIGRATE, we highlight its core methodological innovation. Unlike traditional attention- or perturbation-based explainability approaches, SIGRATE generates explanations through similarity-graph traversal. For each layer, it constructs a patch-similarity graph from patch embeddings. It then performs greedy walks to identify coherent groups of patches. This process produces sets of structured masks that reflect the coherence encoded in the embedding space. This graph-driven mechanism for mask generation, together with its integration of global relevance signals derived from the CLS token, defines the distinctive features of SIGRATE's explainability strategy.

Specifically, the goal of SIGRATE is to determine which patches of an image a ViT considers when classifying it. For each attention layer, SIGRATE builds a cosine-weighted similarity graph from the patch embeddings. Then, it uses the CLS attention token to select the patches with the highest and lowest attention and follows greedy and dynamic walks in the graph. Each walk becomes a binary mask that is reweighted by the model's confidence when only the corresponding patches are visible. Thus, each layer returns a set of masks capturing its most relevant content. SIGRATE then merges all masks using coverage bias normalization, reshapes them to the patch grid and smooths them using a Gaussian filter to leverage patch locality. Finally, it upsamples them using bilinear interpolation. After these steps, it generates a single heatmap that highlights the regions of the image responsible for the ViT's predicted label.

3.2 Similarity graph construction

The idea behind SIGRATE's behavior is to cluster patches sharing the same semantic content. In order to address this issue, relying on raw attention alone (thus using the only attention matrix as the adjacency matrix) is insufficient because attention weights would only derive from query–key interactions, ignoring the value vectors that encode each patch's actual representation. Therefore, two semantically similar patches could receive little mutual attention. To capture content directly, SIGRATE uses a patch similarity graph, where the edge weights are the cosine similarities between the patch embeddings. In this graph, each patch at a given ViT layer is represented by a node. Edges connect patches with high semantic affinity, providing a solid foundation for creating coherent clusters.

Formally speaking, let T be a ViT model used for image classification and let L be the set of λ attention layers of T. Let I be an image consisting of $b \times h$ pixels and let c be its class label. I is divided into a set P of d distinct and non-overlapping square patches, each consisting of $z \times z$ pixels.

To generate a set M_l of masks for each attention layer $l \in L$, we analyze the embeddings generated by the patches of I according to l. Specifically, we extract the embeddings from the output of the intermediate Feed-Forward Network (FFN) layer within l, immediately after the Gaussian Error Linear Unit (GELU) activation function (see Fig. 1).

Each patch p_i is represented by an embedding vector $e_{p_{i_l}} \in \mathbb{R}^H$ in l; proceeding in this way for all patches yields an embedding matrix $E_l \in \mathbb{R}^{d \times H}$.

We did not use the output of the FFN, since it returns an embedding of size typically smaller than H, potentially resulting in a loss of information.

Starting from E_l, we construct a similarity matrix $S_l \in \mathbb{R}^{d \times d}$ that quantifies the degree of similarity between each pair of patch embeddings in the layer l. Specifically, let p_i and p_j be two patches, and let $e_{p_{i_l}}$ (resp., $e_{p_{j_l}}$) be the embedding of p_i (resp., p_j) in l; to compute the similarity between p_i and p_j in l, and thus the value of $S_l[i, j]$, we compute the cosine similarity [53] between $e_{p_{i_l}}$ and $e_{p_{j_l}}$:

$$sim_l(p_i, p_j) = cos(e_{p_{i_l}}, e_{p_{j_l}}) = \frac{e_{p_{i_l}} \cdot e_{p_{j_l}}}{\|e_{p_{i_l}}\| \, \|e_{p_{j_l}}\|} \tag{3.1}$$

The value of the cosine similarity $cos(e_{p_{i_l}}, e_{p_{j_l}})$ belongs to the real range $[-1, 1]$, where higher values indicate higher similarity between $e_{p_{i_l}}$ and $e_{p_{j_l}}$.

It is worth noting that S_l is a symmetrical matrix; as mentioned above, it represents the similarity between patches; therefore, it can be interpreted as the adjacency matrix of a graph that has a node for each patch and whose edges represent the degree of similarity between patches. More specifically, we can define the graph G_{S_l} associated with S_l as follows:

$$G_{S_l} = \langle N_l, A_l \rangle \tag{3.2}$$

N_l is the set of nodes of G_{S_l}; each node $n_i \in N_l$ corresponds to a patch $p_i \in P$ in l. Since there is a biunivocal correspondence between the nodes of N_l and the patches of P in l, we will use the terms "node" and "patch" interchangeably in the following. A_l is the set of edges of G_{S_l}; an edge $a_{ij} = (n_i, n_j, w_{ij}) \in A_l$ indicates the similarity in l between the patch p_i corresponding to n_i and the patch p_j corresponding to n_j

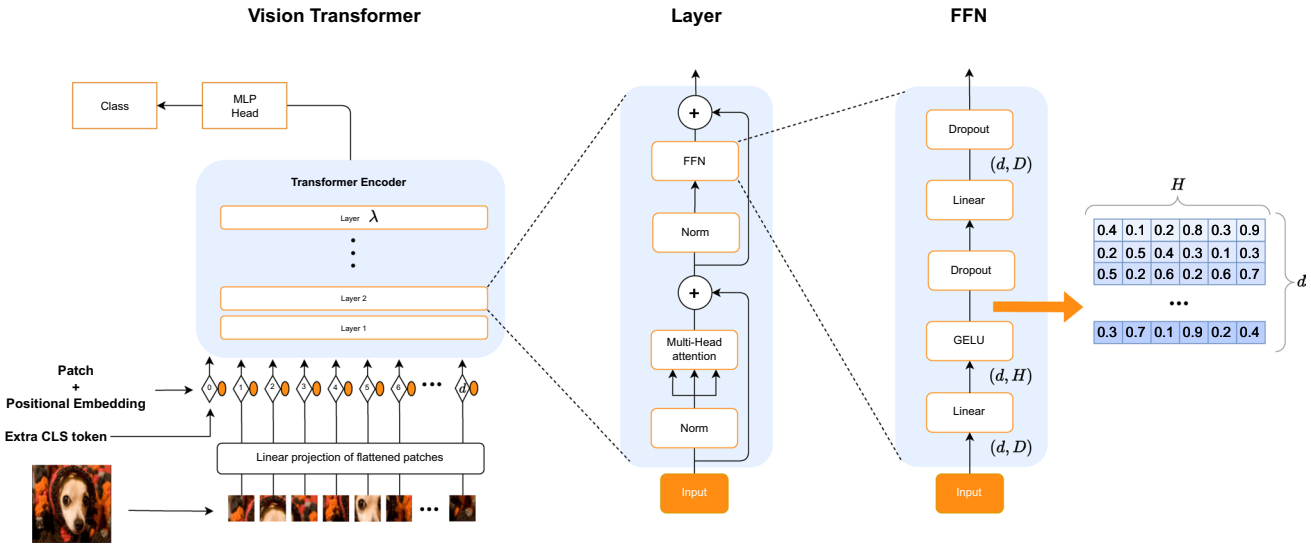


Fig. 1 Schematic representation of the embedding extraction process in SIGRATE. Patch embeddings are extracted before the FFN output after the GELU function to preserve detailed information about patches

with weight $w_{ij} = S_l[i,j] = sim_l(p_i, p_j)$. The self-loop edges do not provide any useful information, since $w_{ii} = S_l[i,i] = sim_l(p_i, p_i) = cos(e_{p_{i_l}}, e_{p_{i_l}}) = 1$, and are therefore removed from G_{S_l}.

3.3 Graph walk patch scoring

After constructing the similarity graph, we extract clusters of related patches that will serve as masks. To accomplish this, we perform a greedy walk implementing a graph-based, region-growing, edge-decay algorithm. In particular, starting from the source patches, we iteratively add the nearest neighbors with the highest cosine similarity, thereby aggregating semantically coherent regions. By always following the edge with the highest weight and decreasing the weights of the edges already traversed, the walk remains biased toward semantic contiguity. The resulting clusters provide masks that delineate meaningful image components, such as the target object, contextual cues, or background areas. These components can then be weighted according to their contribution to the model's prediction. Formally speaking, after constructing G_{S_l}, we proceed to determine the importance of the patches related to *l*. For this purpose, we use the attention matrix of *l*, which contains the attention scores assigned by the classification token (CLS) of *T* to the patches of *I* in *l*. The higher the attention scores that CLS assigns to a patch p_i in *l*, the higher the importance it is given for the correct classification of *I*.

Let μ, $1 \leq \mu \leq d$, be the number of masks we want to generate for each layer. μ is a hyperparameter of our approach; the higher its value, the more accurate the results will be, but also the longer our approach will take to obtain them. A hyperparameter analysis related to μ is proposed in Sect. 4.4. We first select the $\frac{\mu}{2}$ patches that received the highest attention from CLS in *l* and the $\frac{\mu}{2}$ patches that received the lowest attention from CLS in *l*. The patches thus selected serve as starting tokens for the construction of μ walks related to the layer *l*. This strategy ensures a more complete exploration of the input space than considering only the μ patches with the highest attention of CLS in *l*. In fact, selecting only the top μ patches could introduce redundant information, as the highly attended regions may overlap. By including the most and least important patches, we reduce the risk of redundancy, avoid losing potentially useful information present in the early layers with a low score, and ensure a more heterogeneous and complete explanation of the model's inference process.

Let p_i be one of the selected μ patches. Starting from p_i we compute a walk π_{i_l} on G_{S_l} of length $\lfloor d \cdot r \rfloor + 1$, where *r* is a real hyperparameter in the interval [0, 1] used to define the desired maximum length of the walks (in fact, each walk can have a different length) in all layers of *L*.

At this point, we start an iterative procedure. At each step, we select the node with the highest similarity score, append it to the walk π_{i_l}, and mark it as the current node. Since our walk allows revisiting nodes, the corresponding mask varies dynamically (in fact, each patch may appear only once in a mask). Therefore, different walks in the similarity graph can yield masks of different sizes. To promote exploration, we introduce a decay factor $\delta \in [0, 1]$. Whenever the walk traverses an edge a_{ij} (i.e., it first visits the node n_i and then the node n_j), we decrement the corresponding weight as follows:

$$w_{ij} = w_{ij} \cdot \delta \qquad w_{ji} = w_{ji} \cdot \delta \tag{3.3}$$

This reduces the probability of immediately retracing that step and steers the walk toward new nodes in the next iterations. The walk construction ends after $\lfloor d \cdot r \rfloor$ steps. Since a walk is a sequence of nodes connected by edges, this way of proceeding guarantees that each of the μ walks related to the layer *l* has at most $\lfloor d \cdot r \rfloor + 1$ nodes. Now, for each walk π_{i_l} in *l* we generate a binary vector $m_{i_l} \in \mathbb{R}^d$, whose *j*-th position, $1 \leq j \leq d$, is equal to 1 if n_j is included in π_{i_l}, and 0 otherwise. This vector represents just one of the masks associated with the layer *l*. The set of μ vectors thus obtained constitutes the set M_l of masks associated with *l*. This procedure is summarized in Fig. 2.

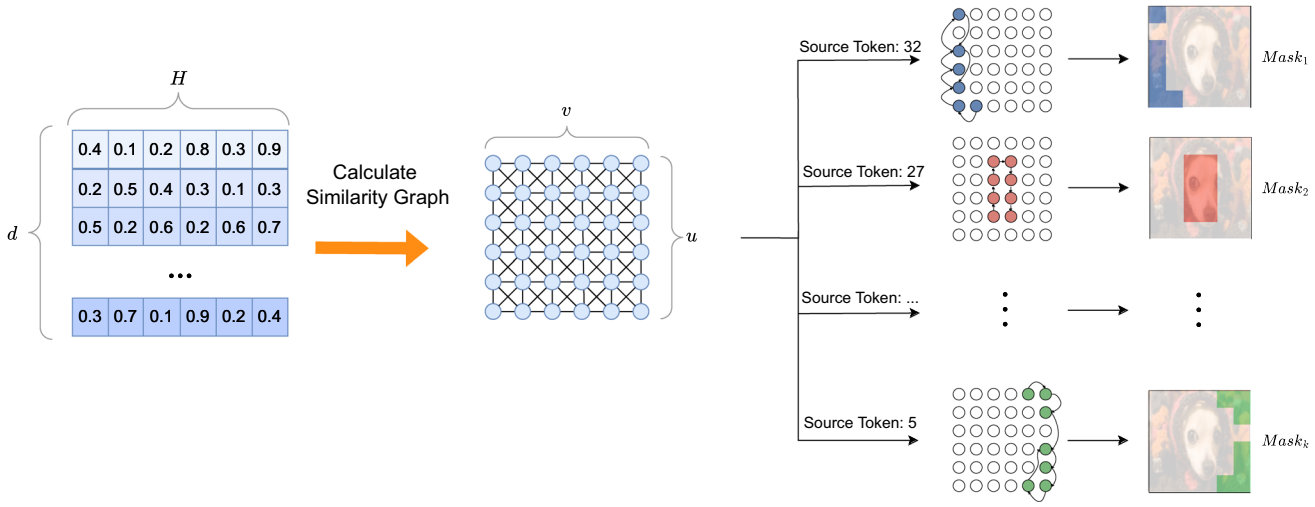


Fig. 2 The mask generation process in SIGRATE. Starting from the similarity graph, source patches are selected from among those most and least attended by the CLS token. From each of these patches, a greedy walk is constructed to generate a binary mask that highlights regions considered relevant by the model

3.4 Heatmap generation

Now, for each layer $l \in L$ and for each mask $m_{i_l} \in M_l$, let $I_{m_{i_l}}$ be the image consisting only of the patches of I selected by m_{i_l}; we compute the class-specific relevance score σ_{i_l} related to the class c for the mask m_{i_l} as:

$$\sigma_{i_l} = \tau(T, c, I_{m_{i_l}}) \tag{3.4}$$

where $\tau(T, c, I_{m_{i_l}})$ is a function that returns the confidence of T for the class c and the image $I_{m_{i_l}}$.

Afterwards, we multiply σ_{il} by m_{il} to obtain a new mask wm_{il} that takes the model confidence into account. We call "binary mask" the original mask m_{i_l} and "weighted mask" the new mask wm_{i_l}. Repeating this procedure for all layers and all masks in each layer, we obtain $\varphi = \mu \cdot \lambda$ masks weighted according to the confidence of T for c and I. Finally, we stack the set of weighted masks thus obtained into a matrix $WM \in \mathbb{R}^{\varphi \times d}$ as follows:

$$WM = \begin{bmatrix} wm_{1_1} \\ \cdots \\ wm_{\mu_1} \\ wm_{2_1} \\ \cdots \\ wm_{\mu_2} \\ \cdots \\ wm_{1_\lambda} \\ \cdots \\ wm_{\mu_\lambda} \end{bmatrix} \tag{3.5}$$

Similarly, we stack the set of binary masks into a matrix $BM \in \mathbb{R}^{\varphi \times d}$ as follows:

$$BM = \begin{bmatrix} m_{1_1} \\ \cdots \\ m_{\mu_1} \\ m_{2_1} \\ \cdots \\ m_{\mu_2} \\ \cdots \\ m_{1_\lambda} \\ \cdots \\ m_{\mu_\lambda} \end{bmatrix} \tag{3.6}$$

A critical challenge in mask-based explanation approaches, like SIGRATE, is Pixel Coverage Bias (PCB) [18, 47, 54]. In these approaches, certain patches may be selected more frequently than others simply due to the generation mechanism. For instance, they may be selected because they act as structural hubs in the similarity graph G_{S_l} rather than because of their semantic importance. Without correction, the final saliency map reflects the path's coverage frequency (i.e., how often the path was visited), rather than its actual contribution to the model's decision. This issue is particularly pronounced in cases of causal overdetermination [47]. Standard aggregation methods, such as weighted averaging or entropy-based fusion, generally rely on the accumulation of importance scores or the statistical distribution of signals across masks. Consequently, they fail to distinguish between a genuinely relevant patch and a patch that appears frequently. This leads to an exaggeration of background patches that appear in many masks despite having low individual impact, forming visual artifacts. To generate a saliency map that addresses this limitation, we adopt the coverage bias formula originally introduced for ViTs in ViT-CX [47] and subsequently applied in TIS [21].

Specifically, we compute the heatmap $\mathcal{H}_{c,I}$ associated with the class *c* and the image *I* by normalizing the accumulated relevance scores. This heatmap contains *d* elements, each of which corresponds to a patch $p_i \in P$. The *i*-th element, denoted as $\mathcal{H}_{c,I}[i]$, is obtained by summing the contributions of the scores that the patch p_i received in the various masks contained in *WM*. This sum is then divided by the number of times p_i was selected (and thus by the number of times the corresponding entry in *BM* is 1). Unlike the aforementioned fusion strategies, this formulation effectively estimates the expected causal impact of a patch, given its presence in a mask. Coverage bias formula decouples the patch's impact from its selection probability. This ensures that the final heatmap highlights regions based on their intrinsic influence on prediction confidence, thus filtering out the structural artifacts introduced by the mask generation process. A specific limitation of the coverage bias formula lies in its uniform treatment of masks across the depth of the Transformer. Unlike weighted averaging methods that could prioritize layers differently, this formula aggregates all masks with equal weighting, potentially overlooking the hierarchical evolution of features as they transition from local patterns to global semantic structures. However, the heatmaps generated through this process remain consistent and semantically coherent. Furthermore, implementing a weighted average would require the introduction of layer-specific hyperparameters, which could either be highly dependent on the characteristics of each individual image or require the development of complex heuristics. Such approaches would risk suppressing the contribution of specific layers or requiring excessive hyperparameter tuning. Instead, the current uniform aggregation adopted by SIGRATE ensures a more robust and stable representation of the model's internal processing. Formally, the previous reasoning for the computation of $\mathcal{H}_{c,I}[i]$ is formalized as follows:

$$\mathcal{H}_{c,I}[i] = \frac{\sum_{q=1}^{\varphi} WM[q,i]}{\sum_{q=1}^{\varphi} BM[q,i]} \tag{3.7}$$

After computing the heatmap $\mathcal{H}_{c,I}$, a reshape operation is required as described in [21, 47]. Specifically, $\mathcal{H}_{c,I} \in \mathbb{R}^d$; therefore, it is a vector of *d* elements, one for each patch of *P*. However, *I* consists of a matrix of patches with $u = \frac{h}{z}$ rows and $v = \frac{b}{z}$ columns such that $u \cdot v = d$. The reshape operation of $\mathcal{H}_{c,I}$ has precisely the purpose of placing the values of $\mathcal{H}_{c,I}$ in an $u \times v$ matrix. Finally, to reduce noise and exploit the intrinsic locality between

adjacent patches, we apply a Gaussian filter to the resized $\mathcal{H}_{c,I}$ matrix. After this task, an additional step is necessary to move from a heatmap defined over patches to a final heatmap defined over the pixel space. Given that the original image I has dimensions $b \times h$ pixels, where $b = v \cdot z$ and $h = u \cdot z$, the final heatmap is obtained by performing bilinear interpolation. This kind of interpolation estimates a function's value at each two-dimensional (2-D) point using the four surrounding samples. In particular, it first performs linear interpolation along one axis and then along the other. This 2-D extension of 1-D linear interpolation is common in image processing tasks, such as resizing, texture mapping, and grid remapping [21] because it yields smoother, less jagged results than nearest neighbor methods. Although each interpolation step is linear with respect to the sampled values, the combined operation is quadratic with respect to the pixel position, resulting in a nonlinear overall mapping. By adopting bilinear interpolation, we convert the heatmap from $u \cdot v$ patches to $b \cdot h$ pixels. The result is a heatmap that represents the visual explainability of the ViT T when classifying the image I into the class c.

Algorithm 1 summarizes the approach used by SIGRATE to generate a heatmap related to the classification of I in c by T.

Require: a ViT T; an image I; a number d of patches; a target class c; a token ratio r; a number μ of masks per layer; an image width b; an image height h; a decay factor δ
Ensure: a heatmap $\mathcal{H}_{c,I}$
$BM = []$
$WM = []$
$P = \textsc{Patchify}(I, d)$ ▷ split I into d square patches of size $z \times z$
$u = \frac{b}{z}$, $v = \frac{h}{z}$
$L = \textsc{GetAttnLayers}(T)$ ▷ get the attention layers of T
for $l \in L$ **do**
  $E_l = \textsc{Embeddings}(I, l)$ ▷ construct $d \times H$ patch embeddings at the layer l
  $S_l = \frac{E_l E_l^{\mathsf{T}}}{\|E_l\| \, \|E_l\|^{\mathsf{T}}}$
  $G_{S_l} = \textsc{GraphFromAdj}(S_l)$ ▷ construct a similarity graph at the layer l
  $\Pi_l = \textsc{TopCLSAttn}(l, \mu/2) \cup \textsc{BottomCLSAttn}(l, \mu/2)$ ▷ get the top $\frac{\mu}{2}$ and the bottom $\frac{\mu}{2}$ patches receiving the highest attention
  for $p_i \in \Pi_l$ **do**
    $\pi_{i_l} = \textsc{GreedyWalk}(G_{S_l}, p_i, \delta, \lfloor d \cdot r \rfloor + 1)$ ▷ construct a greedy walk with source pi, decay δ and length $\lfloor d \cdot r \rfloor + 1$
    $m_{i_l} = \textsc{MaskFromWalk}(\pi_{i_l})$ ▷ construct a mask corresponding to the greedy walk
    $\sigma_{i_l} = \tau(T, c, I_{m_{i_l}})$
    $wm_{i_l} = m_{i_l} \cdot \sigma_{i_l}$
    $BM = \begin{bmatrix} m_{i_l} \\ BM \end{bmatrix}$
    $WM = \begin{bmatrix} wm_{i_l} \\ WM \end{bmatrix}$
$\lambda = |L|$, $\varphi = \mu \cdot \lambda$
for $p_i \in P$ **do**
  $\mathcal{H}_{c,I}[i] = \frac{\sum_{q=1}^{\varphi} WM[q,i]}{\sum_{q=1}^{\varphi} BM[q,i]}$
$\mathcal{H}_{c,I} = \textsc{Reshape}(\mathcal{H}_{c,I}, u, v)$ ▷ Reshape $\mathcal{H}_{c,I}$ from a vector of size d to a matrix of size $u \times v$
$\mathcal{H}_{c,I} = \textsc{GaussianFilter}(\mathcal{H}_{c,I})$ ▷ apply a Gaussian filter
$\mathcal{H}_{c,I} = \textsc{Bilinear}(\mathcal{H}_{c,I}, b, h)$ ▷ apply bilinear interpolation to obtain a heatmap of size $b \times h$

Algorithm 1 SIGRATE algorithm for generating an explanatory heatmap

4 Experiments

In this section, we illustrate our experimental campaign aimed at evaluating the performance of SIGRATE and comparing it with that of other related approaches already proposed in the literature. Specifically, in Sect. 4.1 we describe the experimental setup, specifying the dataset, ViT models, and explainability metrics we adopted. In Sect. 4.2, we present the quantitative performance obtained by SIGRATE and compare it with that of related approaches. In Sect. 4.3, we report a qualitative analysis of the results obtained by SIGRATE on some images as well as a failure case analysis. In Sect. 4.4, we analyze the hyperparameters used in SIGRATE and evaluate their impact on its performance. Finally, in Sect. 4.5 we conduct an ablation study to evaluate the graph construction strategy underlying the SIGRATE's approach.

4.1 Experimental setup

In our experiments, we used the Vision Transformer (ViT) [1] and Data-efficient Image Transformer (DeiT) models [55] as reference ViT models. Both of them use a classification token (CLS) at the beginning of the input sequence, which is requested for the execution of SIGRATE. In addition to the CLS token, the DeiT architecture includes a distillation token, which is used with an extra classification head to facilitate learning by distillation. The implementations of ViT and DeiT we use are ViT-Base[1] and DeiT-Base[2].

Regarding the hyperparameters of SIGRATE, in this experimental campaign we set r to 0.75, μ to 12, and δ to 0.50. As mentioned above, we report a hyperparameter analysis in Sect. 4.4. ViT-Base and DeiT-Base consist of 12 layers, so the total number of masks is 144. The approaches used for comparison with SIGRATE are TIS [21], ViT-CX [47], TAM [45], Chefer1 [17], Chefer2 [56], BT H [42], BT T [42], RISE [18], IntegratedGrad [27], SmoothGrad [26], Finer-CAM [36], KPCA-CAM [35], GradCAM [25], LeGrad [44], Attention Rollout [40] and GMAR [46]. To evaluate the performance of these approaches, we used the hyperparameter values suggested in the corresponding papers. We selected these approaches because they are the most widely recognized approaches in the ViT explainability literature. Furthermore, as highlighted by recent works including TIS [21] and ViT-CX [47], most of them are commonly used as reference baselines. Including them allows us to ensure a fair and transparent comparison, in line with the experimental protocols established in this research area. In our experimental campaign, we employed two datasets, namely ImageNet[57] (Sect. 4.2.1), and BloodMNIST [23] (Sect. 4.2.2). For ImageNet, we followed the protocol of previous works [21, 42, 47] and used the same subset of 5,000 images from the ImageNet validation set. BloodMNIST contains images of peripheral blood cells divided into eight distinct categories. We selected it to evaluate the robustness of SIGRATE in the medical domain. The interested reader can find the subset index list and the code implementing SIGRATE at https://github.com/DavideTraini/SIGRATE-ViT-XAI.

The faithfulness of all approaches is measured by Insertion and Deletion explainability metrics [18]. The former assesses the probability of correctly identifying the target class using an increasing number of pixels, starting with the most significant ones from 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 varies in the real range [0, 1]; the higher the value and the better the performance of the approach. Instead, the Deletion metric considers how the confidence in finding the target class decreases as the pixels with the highest score on the corresponding heatmap are removed from the original image. Again, we compute the Deletion curve and the corresponding AUC; the lower the AUC value, the better the performance of the explainability approach. To have a metric capable of synthesizing the previous two metrics, we calculate the difference between the AUCs of the Insertion and Deletion metrics and we call it Insertion-Deletion. The higher the value of this metric, the better the performance of the approach. Following the literature [21], we used four

[1] https://github.com/huggingface/transformers/blob/v4.42.0/src/transformers/models/vit/modeling_vit.py.

[2] https://github.com/huggingface/transformers/blob/v4.42.0/src/transformers/models/deit/modeling_deit.py.

different methods to obfuscate the pixels when calculating the Insertion and Deletion metrics. These methods are: *(i)* Mean, which replaces pixel values with the mean values of the entire image; *(ii)* Blur, which performs pixel blurring; *(iii)* Black, which replaces pixels with the color black; and *(iv)* Rand, which replaces pixels with a random color. Finally, regarding the target label used for evaluation, in Sects. 4.2.1 and 4.2.2 we calculated the faithfulness metrics using the ground-truth labels, whereas in Sect. 4.2.3 we computed them using the class predicted by the model.

We also tested our approach using the Pointing Game localization metric, which assesses how accurately a method can localize the regions of an image that contribute most to the model decision. Since BloodMNIST does not provide bounding boxes, we calculated this metric only on the subset of the ImageNet validation set. Analogously to what is done in TIS [21], we excluded images whose bounding box covered more than 50% of the image, so that only the most complex images were considered. As a result of this policy, we excluded 2892 images in the calculation of this metric and retained 2,108 images. Given an image, if the pixel with the highest value in the corresponding saliency map, and thus the most important pixel, falls within one of the bounding boxes, then the test for that image passes and it is labeled as "hit"; otherwise, it is labeled as "miss". The result of the Pointing Game localization metric for the entire dataset is calculated as follows:

$$\text{Pointing Game} = \frac{No.of\ hits}{No.\ of\ hits + No.of\ misses} \tag{4.1}$$

As we can see from Eq. 4.1, this metric varies in the real range [0, 1], where 0 indicates the complete absence of "hits", while 1 denotes that in all the images analyzed, the most important pixel in the saliency map falls within the bounding box.

4.2 Quantitative results

4.2.1 ImageNet dataset

In Table 1, we show the performance achieved by SIGRATE and related approaches using the four obfuscation methods mentioned above and the ViT-Base model [1]. We use arrows to indicate which values give better performance; in particular, if there is an upward (resp., downward) arrow next to a measure, it means that for that measure the best performance is obtained with the highest (resp., lowest) values. To ensure fairness, we used the same subset of the ImageNet validation set as TIS [21].

Table 1 shows that SIGRATE performs very well. It achieves the highest Insertion values for all four obfuscation methods. Specifically, it improves upon the second-best approach by 16.07% (resp., 1.45%, 25.00%, and 19.23%) with Mean (resp., Blur, Black, and Rand).

Regarding Deletion, SIGRATE is the second-best approach with Blur, while, with the other obfuscation methods, its performance aligns with that of the other approaches (e.g., the performance of ViT-CX with Mean and Rand).

Finally, for Insertion-Deletion, SIGRATE is the best approach with Mean, Black, and Rand. In particular, we observe improvements of 11.90%, 17.50% and 21.05% respectively, compared to the second-best approach. With Blur, SIGRATE is the second-best approach after IntegratedGrad.

Next, we performed the same experiment using the DeiT-Base model [55]. The results are shown in Table 2. Similar to the previous experiment, we used the same subset of the ImageNet validation set as TIS [21].

An analysis of this table shows that SIGRATE performs very well even with the DeiT-Base model. In particular, it achieves the highest Insertion values with all obfuscation methods, outperforming the second-best approach by 16.13% with Mean, 5.88% with Blur, 16.13% with Black, and 15.87% with Rand. Regarding Deletion, SIGRATE shows competitive performance despite not appearing in the top positions. Its performance aligns with the performance of other approaches, such as Chefer2 and RISE. Finally, for Insertion-Deletion, SIGRATE is the

https://doi.org/10.1007/s00521-026-12392-6

Table 1 Comparison of explanation approaches using Insertion, Deletion, and Insertion-Deletion as metrics, ViT-Base as model, and ImageNet as dataset

Approach	*Insertion*↑				*Deletion*↓				*Insertion-Deletion*↑			
	Mean	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
SIGRATE	**0.65**	**0.70**	**0.65**	**0.62**	0.19	0.39	0.19	0.17	**0.47**	0.30	**0.47**	**0.46**
TIS [21]	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 [47]	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 [45]	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 [17]	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 [56]	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 [42]	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 [42]	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 [27]	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 [26]	0.37	0.59	0.36	0.35	0.10	0.45	0.10	0.09	0.27	0.14	0.26	0.26
Finer-CAM [36]	0.51	0.68	0.49	0.46	0.15	0.40	0.15	0.15	0.36	0.28	0.34	0.31
KPCA-CAM [35]	0.47	0.68	0.46	0.46	0.21	0.40	0.20	0.19	0.26	0.29	0.26	0.27
GradCAM [25]	0.51	0.65	0.49	0.50	0.16	0.42	0.16	0.17	0.34	0.23	0.33	0.33
LeGrad [44]	0.56	0.68	0.52	0.52	0.21	0.40	0.15	0.15	0.35	0.28	0.37	0.37
Attention Rollout [40]	0.31	0.55	0.30	0.29	0.29	0.52	0.28	0.27	0.02	0.03	0.02	0.02
GMAR [46]	0.39	0.57	0.32	0.34	0.34	0.50	0.28	0.29	0.05	0.07	0.04	0.05

As for Insertion, SIGRATE outperforms all other explanation approaches. As for Deletion, it shows competitive results. Finally, as for Insertion-Deletion, it achieves the best performance with three out of four obfuscation methods, confirming the effectiveness of its explanations

The best value in each column is shown in bold, and the second-best value is underlined

https://doi.org/10.1007/s00521-026-12392-6

Table 2 Comparison of explanation approaches using insertion, deletion, and insertion-deletion as metrics, DeiT-Base as model, and ImageNet as dataset

Approach	*Insertion*↑				*Deletion*↓				*Insertion-Deletion*↑			
	Mean	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
SIGRATE	**0.72**	**0.72**	**0.72**	**0.73**	0.24	0.39	0.25	0.26	**0.48**	**0.34**	**0.48**	**0.48**
TIS [21]	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 [47]	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 [45]	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 [17]	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 [56]	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 [42]	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 [42]	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 [27]	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 [26]	0.45	0.62	0.43	0.43	**0.14**	0.44	0.14	**0.13**	0.31	0.18	0.30	0.31
Finer-CAM [36]	0.56	0.62	0.56	0.53	0.19	0.36	0.19	0.22	0.37	0.26	0.37	0.31
KPCA-CAM [35]	0.56	0.67	0.56	0.52	0.32	0.44	0.32	0.34	0.23	0.24	0.23	0.18
GradCAM [25]	0.62	0.68	0.62	0.63	0.23	0.43	0.23	0.27	0.38	0.25	0.38	0.37
LeGrad [44]	0.60	0.67	0.59	0.63	0.17	**0.34**	0.17	0.24	0.42	**0.34**	0.42	0.39
Attention Rollout [40]	0.37	0.54	0.37	0.34	0.41	0.53	0.41	0.37	−0.04	0.01	−0.05	−0.03
GMAR [46]	0.53	0.63	0.52	0.57	0.21	0.37	0.21	0.27	0.31	0.25	0.31	0.30

As for Insertion, SIGRATE outperforms all other explanation approaches. As for Deletion, it shows competitive results, with values comparable to those of the best explanation approaches. Finally, as for Insertion-Deletion, it achieves the best results with all obfuscation methods, confirming the superiority of its explanations

Bold values indicate the best (highest for Insertion and Insertion-Deletion, lowest for Deletion) results in each column, while underlined values indicate the second-best results

best approach with Mean, Black and Rand, achieving improvements over the second-best approach of 14.29%, 14.29%, and 17.07% respectively. It ranks first alongside LeGrad with Blur, improving upon the second-best approach by 13.33%.

Experiments with both ViT-Base and DeiT-Base show that, although SIGRATE performs competitively in the Insertion metric, its performance in the Deletion metric lags behind the best related approaches. This discrepancy is explained by the nature of the mask generation process on which SIGRATE is based. Rather than simply selecting regions with the highest attention scores, this process constructs walks connecting patches in similarity graphs. These walks include areas that, while not strictly decisive, are structurally or semantically related to the most important ones. As a result, the generated masks tend to cover larger sets of patches, including not only the minimum ones necessary for classification, but also others related to them. This feature leads to asymmetric behavior between the two metrics. In fact, Insertion benefits from the presence of related patches in the patch sets obtained from the masks because they enable the model to quickly regain confidence in the prediction when new patches are added. Conversely, Deletion is penalized by the presence of related patches because they could be removed in place of those most significant for the image classification. This could cause patches relevant for classification to remain in the image to be classified for a long time as patches are removed for the Deletion computation, which prevents a rapid decrease in the ViT classification model's confidence and a consequent worsening of Deletion values.

Furthermore, SIGRATE's mechanism, which selects both high- and low-attention patches to generate walks in graphs, favors a more complete and distributed representation of patch importance, avoiding focusing exclusively on the most obvious areas of an image. This improves SIGRATE's ability to locate objects in complex images and increases its interpretative robustness. However, it results in more conservative behavior when removing patches, which slows the decrease in confidence, and consequently worsens the Deletion values. We would like to point out that this feature is a deliberate design choice of our approach. Indeed, we want SIGRATE to highlight not only the areas of an image decisive for classification, but also those functional for the model's decision-making process, which allows us to capture the relationships between patches within the attention levels.

Table 3 shows the comparison between SIGRATE and the other ViT explainability approaches considered above with respect to the Pointing Game metric. Again, in this experiment, we used ViT-Base and DeiT-Base as the underlying ViTs. Additionally, we used the same subset of the ImageNet validation set as TIS [21].

From the analysis of this table, we can see that, as far as the ViT-Base model is concerned, only 2 out of 16 approaches achieve higher performance than SIGRATE. These approaches are BT H and BT T, whose Pointing

Table 3 Results of the Pointing Game metric for the ViT-Base and DeiT-Base models and the ImageNet dataset

In the case of ViT-Base, SIGRATE achieves a very competitive performance, ranking among the best approaches and outperforming TIS. In the case of DeiT-Base, SIGRATE achieves the best performance

The best value in each column is shown in bold, and the second-best value is underlined

Approach	*ViT-Base*	*DeiT-Base*
SIGRATE	0.844	**0.866**
TIS [21]	0.823	<u>0.825</u>
ViT-CX [47]	0.700	0.700
TAM [45]	0.737	0.635
Chefer1 [17]	0.768	0.748
Chefer2 [56]	0.727	0.654
BT H [42]	**0.855**	0.775
BT T [42]	<u>0.846</u>	0.755
RISE [18]	0.753	0.766
IntegratedGrad [27]	0.633	0.297
SmoothGrad [26]	0.499	0.742
Finer-CAM [36]	0.805	0.784
KPCA-CAM [35]	0.796	0.768
GradCAM [25]	0.433	0.415
LeGrad [44]	0.761	0.789
Attention Rollout [40]	0.118	0.127
GMAR [46]	0.348	0.612

Game values are 0.855 and 0.846, respectively. The value of SIGRATE is 0.844, 1.29% lower than the value of BT H, i.e., the best approach, which is very satisfactory, comparable to or higher than several other approaches under consideration (such as BT T and TIS). If we switch from ViT-Base to DeiT-Base, the scenario changes. In fact, in this case SIGRATE achieves the highest Pointing Game value, i.e., 0.866, which is 4.97% higher than the second-best approach (i.e., TIS) that achieves a value of 0.825. BT H and BT T values are 0.775 and 0.755, corresponding to decreases of 11.74% and 14.70%, respectively, compared to the SIGRATE value.

In conclusion, as for the Pointing Game metric, SIGRATE shows robust performance in identifying relevant regions in an image with both ViT-Base and DeiT-Base. It ranks among the top approaches for this metric with very high absolute values.

4.2.2 BloodMNIST dataset

To evaluate the robustness of SIGRATE in a domain significantly different from ImageNet, we expanded our analysis to the medical field with the BloodMNIST dataset [23]. Designed for classifying peripheral blood cells, this dataset includes 17,092 RGB samples of normal cells divided into eight classes and resized to 224 $\times$ 224 pixels. Since there are no bounding boxes as ground truth in the dataset, we focused our analysis on faithfulness metrics only, as we could not consider Pointing Game.

Table 4 shows the quantitative results obtained using ViT-Base as the underlying ViT model. As in the previous experiments, SIGRATE achieves the highest Insertion values with all obfuscation methods. It outperforms the second-best approach by 10.29% with Mean, 5.63% with Blur, 11.94% with Black, and 4.69% with Rand. Regarding Deletion, although TIS and IntegratedGrad achieve slightly better values than SIGRATE with certain obfuscation methods, our approach remains highly competitive. Most importantly, when considering Insertion-Deletion values, SIGRATE demonstrates a significant advantage over all the other approaches, outperforming the second-best one by 11.36% with Mean, 37.50% with Blur, 8.89% with Black and 5.26% with Rand.

The performance trends observed for the DeiT-Base model, shown in Table 5, further validate the effectiveness of SIGRATE. In fact, our approach achieves the best results for the Insertion metric, outperforming the second-best approach by 8.33% with Mean, 2.94% with Blur, 8.33% with Black and 8.22% with Rand. Regarding the Deletion metric, SIGRATE shows the best performance with Blur, with a value 5.71% lower than the second-best approach[3]. Furthermore, it is the second-best approach with Mean, Black, and Rand, with values very close to the best ones. As a result, the gap between SIGRATE and the other approaches in terms of Insertion-Deletion obtained for the DeiT-Base model is even more pronounced than the gap obtained for the ViT-Base model. In particular, SIGRATE outperforms the second-best approach by 12.77% with Mean, 12.12% with Blur, 10.64% with Black, and 13.04% with Rand.

The analysis of Tables 4 and 5 confirms that SIGRATE is effective not only in highlighting salient regions, but also in accurately ignoring irrelevant background noise. This is a critical property for medical image analysis.

4.2.3 Performance on predicted classes

In the experiments presented in Sects. 4.2.1 and 4.2.2, we calculated the faithfulness metrics using ground truth labels. Indeed, we essentially asked the model to identify regions of the image containing features associated with the correct class. To provide a more comprehensive evaluation of SIGRATE, we extend this analysis by shifting the objective and asking the model to identify the image regions that it considers most important to the class it actually predicted, even if that class differs from the ground truth. This way of proceeding allows us to evaluate the quality of the generated heatmaps in relation to the model's specific decision, providing a complementary perspective to the ground truth-based evaluation. We repeated the evaluation on both ImageNet and BloodMNIST, calculating the faithfulness metrics using the top-1 class predicted by the model. Due to space limitations, Table 6

[3] Recall that, with regard to Deletion, lower values indicate better performance.

https://doi.org/10.1007/s00521-026-12392-6

Table 4 Results of the insertion, deletion, and insertion-deletion metrics obtained by SIGRATE and other related approaches when applied to the BloodMNIST dataset with ViT-Base as ViT model

Approach	*Insertion* ↑				*Deletion* ↓				*Insertion-Deletion* ↑			
	Mean	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
SIGRATE	**0.75**	**0.75**	**0.75**	**0.67**	0.26	0.53	0.26	0.27	**0.49**	**0.22**	**0.49**	**0.40**
TIS [21]	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 [47]	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 [45]	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 [17]	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 [56]	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 [42]	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 [42]	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 [27]	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 [26]	0.42	0.66	0.32	0.31	0.29	0.54	0.28	0.27	0.13	0.12	0.04	0.04
Finer-CAM [36]	0.51	0.64	0.51	0.54	0.28	0.52	0.28	0.29	0.23	0.12	0.23	0.25
KPCA-CAM [35]	0.49	0.64	0.48	0.51	0.27	0.52	0.27	0.28	0.22	0.12	0.21	0.23
GradCAM [25]	0.51	0.71	0.47	0.50	0.26	0.58	0.28	0.27	0.24	0.13	0.20	0.23
LeGrad [44]	0.62	0.69	0.63	0.64	0.26	0.53	0.25	0.26	0.36	0.16	0.39	0.37
Attention Rollout [40]	0.37	0.60	0.35	0.33	0.29	0.54	0.28	0.28	0.08	0.06	0.07	0.05
GMAR [46]	0.37	0.59	0.37	0.38	0.28	0.55	0.27	0.29	0.09	0.04	0.10	0.09

SIGRATE achieves the highest Insertion values and proves superior performance to the other approaches in identifying critical features. Its Deletion values are still competitive. More importantly, SIGRATE is the best approach for Insertion-Deletion values with all obfuscation methods, highlighting its effectiveness in providing explanations

The best value in each column is shown in bold, and the second-best value is underlined

https://doi.org/10.1007/s00521-026-12392-6

Table 5 Results of the insertion, deletion, and insertion-deletion metrics obtained by SIGRATE and other related approaches when applied to the BloodMNIST dataset and the DeiT-Base model with different obfuscation methods

Approach	*Insertion*↑				*Deletion*↓				*Insertion-Deletion*↑			
	Mean	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
SIGRATE	**0.78**	**0.70**	**0.78**	**0.79**	0.25	**0.33**	0.26	0.27	**0.53**	**0.37**	**0.52**	**0.52**
TIS [21]	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 [47]	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 [45]	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 [17]	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 [56]	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 [42]	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 [42]	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 [27]	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 [26]	0.44	0.58	0.36	0.32	0.35	0.44	0.32	0.30	0.09	0.14	0.04	0.02
Finer-CAM [36]	0.58	0.65	0.56	0.56	0.30	0.52	0.29	0.31	0.28	0.13	0.27	0.25
KPCA-CAM [35]	0.56	0.63	0.53	0.54	0.30	0.53	0.30	0.32	0.26	0.10	0.23	0.22
GradCAM [25]	0.48	0.65	0.47	0.54	0.25	0.40	**0.25**	0.31	0.23	0.25	0.22	0.23
LeGrad [44]	0.65	0.68	0.64	0.63	0.27	0.44	**0.25**	0.27	0.38	0.24	0.39	0.36
Attention Rollout [40]	0.44	0.58	0.41	0.38	0.32	0.50	0.30	0.30	0.12	0.08	0.11	0.08
GMAR [46]	0.45	0.59	0.43	0.41	0.31	0.51	0.31	0.32	0.14	0.08	0.12	0.09

SIGRATE achieves the highest Insertion values, demonstrating its superiority in identifying important features over the other approaches. As for Deletion, it achieves the best value with Blur, the second-best value with Black, and one of the best values with Mean and Rand. Most importantly, SIGRATE achieves the best Insertion-Deletion values with all obfuscation methods, highlighting its effectiveness in providing explanations

The best value in each column is shown in bold, and the second-best value is underlined

Table 6 Comparison of SIGRATE and other related approaches using the Insertion-Deletion metric calculated on the predicted classes for the ViT-Base and DeiT-Base models on the ImageNet and BloodMNIST datasets

Models	*Approaches*	*ImageNet*				*BloodMNIST*			
		Insertion-Deletion				*Insertion-Deletion*			
		Mean	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
ViT-Base	SIGRATE	**0.55**	**0.38**	**0.55**	**0.53**	**0.50**	0.22	**0.49**	**0.43**
	TIS [21]	0.52	0.36	0.51	0.51	0.46	0.17	0.46	0.39
	ViT-CX [47]	0.38	0.29	0.36	0.36	0.19	0.12	0.16	0.21
	TAM [45]	0.44	0.29	0.44	0.41	0.30	0.24	0.29	0.27
	Chefer1 [17]	0.49	0.27	0.41	0.39	0.32	**0.25**	0.31	0.29
	Chefer2 [56]	0.42	0.28	0.42	0.39	0.29	0.24	0.29	0.27
	BT H [42]	0.45	0.29	0.45	0.42	0.31	**0.25**	0.31	0.29
	BT T [42]	0.45	0.29	0.44	0.42	0.31	**0.25**	0.31	0.29
	RISE [18]	0.42	0.27	0.38	0.38	0.28	0.22	0.27	0.31
	IntegratedGrad [27]	0.16	0.41	0.16	0.02	0.07	0.17	0.10	0.04
	SmoothGrad [26]	0.17	0.17	0.16	0.14	−0.03	0.02	−0.05	−0.04
	Finer-CAM [36]	0.32	0.24	0.30	0.31	0.26	0.23	0.25	0.25
	KPCA-CAM [35]	0.08	0.12	0.07	0.10	0.12	0.15	0.12	0.13
	GradCAM [25]	0.41	0.30	0.40	0.39	0.30	**0.25**	0.29	0.28
	LeGrad [44]	0.36	0.26	0.35	0.35	0.27	0.23	0.26	0.25
	Attention Rollout [40]	0.10	0.08	0.11	0.07	0.12	0.10	0.12	0.10
	GMAR [46]	0.03	0.07	0.04	0.05	0.08	0.11	0.09	0.09
DeiT-Base	SIGRATE	**0.57**	**0.42**	**0.57**	**0.61**	**0.54**	**0.37**	**0.53**	**0.53**
	TIS [21]	0.49	0.37	0.49	0.51	0.48	0.35	0.48	0.48
	ViT-CX [47]	0.32	0.26	0.31	0.29	0.29	0.28	0.24	0.23
	TAM [45]	0.37	0.30	0.37	0.37	0.28	0.24	0.27	0.26
	Chefer1 [17]	0.38	0.31	0.37	0.37	0.29	0.25	0.28	0.27
	Chefer2 [56]	0.36	0.30	0.35	0.33	0.26	0.23	0.25	0.24
	BT H [42]	0.36	0.29	0.36	0.36	0.26	0.23	0.26	0.25
	BT T [42]	0.36	0.30	0.36	0.37	0.26	0.23	0.26	0.25
	RISE [18]	0.49	0.36	0.47	0.45	0.37	0.31	0.36	0.30
	IntegratedGrad [27]	0.11	0.19	0.13	0.05	0.06	0.14	0.10	0.04
	SmoothGrad [26]	0.01	0.07	0.00	−0.02	−0.13	−0.12	−0.14	−0.15
	Finer-CAM [36]	0.35	0.25	0.33	0.34	0.25	0.22	0.24	0.24
	KPCA-CAM [35]	0.16	0.15	0.16	0.17	0.18	0.16	0.18	0.18
	GradCAM [25]	0.47	0.32	0.47	0.46	0.32	0.27	0.31	0.30
	LeGrad [44]	0.41	0.32	0.40	0.37	0.29	0.25	0.28	0.27
	Attention Rollout [40]	−0.02	0.02	0.04	0.02	0.05	0.07	0.08	0.07
	GMAR [46]	0.31	0.25	0.31	0.30	0.24	0.21	0.24	0.23

The best value in each column is shown in bold, and the second-best value is underlined

only reports the Insertion-Deletion score to compare the results obtained by SIGRATE with those of other related approaches.

As shown in Table 6, SIGRATE performs very well on ImageNet with the ViT-Base model. Indeed, it achieves the highest values of Insertion-Deletion with all obfuscation methods. It outperforms the second-best approach by 5.77% with Mean, 5.56% with Blur, 7.84% with Black, and 3.92% with Rand. It also achieves the highest Insertion-Deletion values with all four obfuscation methods using the DeiT-Base model and outperforms the second-best approach by 16.33%, 13.51%, 16.33%, and 19.61% with Mean, Blur, Black, and Rand, respectively. For BloodMNIST with the ViT-Base model, SIGRATE confirms its superiority in most cases. It achieves the best performance with Mean, Black, and Rand, surpassing the second-best approach by 8.70%, 6.52%, and 10.26%, respectively. Although it does not rank first with Blur, it still achieves competitive results. Finally, with the DeiT-Base model and the BloodMNIST dataset, SIGRATE demonstrates robust performance by achieving the first

position across all obfuscation methods. It improves upon the second-best approach by 12.50%, 5.71%, 10.42%, and 10.42% with Mean, Blur, Black, and Rand, respectively.

The results reported in Table 6 demonstrate the robustness of SIGRATE even when the explanation refers to the class predicted by the approach rather than the ground truth, as in the experiments of the previous sections. Using both ViT-Base and DeiT-Base as models and operating on both ImageNet and BloodMNIST, SIGRATE consistently achieves better Insertion-Deletion values than related approaches. These results confirm that our framework can detect the areas the model uses to classify an image by effectively isolating the regions that drive the specific prediction, regardless of whether the latter matches the ground truth.

4.3 Qualitative results

In this section, we report a qualitative analysis of the heatmaps returned by SIGRATE. This analysis is divided into two parts; in the first part, we make a qualitative comparison between SIGRATE and related approaches, while in the second part, we focus only on the results of SIGRATE when applied to three images belonging to different classes and four images containing objects of two different classes.

For the qualitative comparison between SIGRATE and other approaches available in the literature, we extracted heatmaps for four images of ImageNet belonging to different classes using ViT-Base. The results are shown in Fig. 3. For reasons of space, we do not show the corresponding results obtained with DeiT-Base, which are, however, similar. From the analysis of this figure, we can see that SIGRATE identifies salient areas for the image class. In the rows related to the *starfish* and the *toaster*, SIGRATE is able to focus very well on the shapes of the objects, while giving much less importance to the background. Similar reasoning can be observed in the case of the *table lamp*, where SIGRATE identifies its shape and discards the table area.

In the case of the *bullet train*, SIGRATE gives weight only to the shape of the train. TIS and ViT-CX also produce interesting results, but they focus on larger areas and give importance to objects that are not closely related to the image classes (e.g., the background in the *bullet train* and the slices of bread in the *toaster*). Finally, IntegratedGrad and SmoothGrad return heatmaps very noisy and difficult to interpret. Overall, we can say that SIGRATE identifies the most important areas for the corresponding image class and retrieves heatmaps that are in line with (and sometimes better than) those obtained by related approaches.

To ensure a comprehensive and fair evaluation, we also examined scenarios in which the explanation quality is compromised. Figure 4 provides a failure analysis that illustrates scenarios in which SIGRATE and other approaches produce unsatisfactory heatmaps. From the analysis of this figure we can observe that failure can be attributed to two main factors. The first occurs when the underlying classification model (ViT-Base) misclassifies the ground truth, as illustrated in the first two rows of the figure. For instance, in the first row, the ground truth is *plate*, but the model incorrectly predicts *meat loaf*. Since the model fails to identify the target class, the resulting explanations cannot fully highlight the correct figure. However, we note that SIGRATE demonstrates strong noise suppression capabilities. In fact, even in cases of failure, it partially ignores features associated with the predicted class (e.g., *meat loaf* and *espresso*) and partially delineates the boundaries of the correct class (e.g., *plate* and *cup*). The second type of failure occurs when multiple instances of the same class appear in an image. In these cases, SIGRATE focuses primarily on one instance or a subset of instances that the model recognizes with high confidence. This behavior is due to the graph-based mask generation process described in Sect. 3.3. Specifically, SIGRATE constructs masks by performing greedy walks on the similarity graph G_{S_l} (see Eq. 3.2), starting from the patches with the highest and the lowest attention values. When an object is composed of highly cohesive patches, the greedy walk tends to remain confined within that object because the intra-object cosine similarity between adjacent patches is often higher than the similarity required to jump to a spatially distant object of the same class. Consequently, the generated masks may cover the same dominant object. When these masks are merged using the coverage bias formula, the final heatmap emphasizes that object more strongly while assigning significantly lower importance to the others. This phenomenon is visible in the third row (*granny smith*) of Fig. 4, where the model correctly predicts the class but SIGRATE focuses on the relevance of the central cut apple and

 https://doi.org/10.1007/s00521-026-12392-6

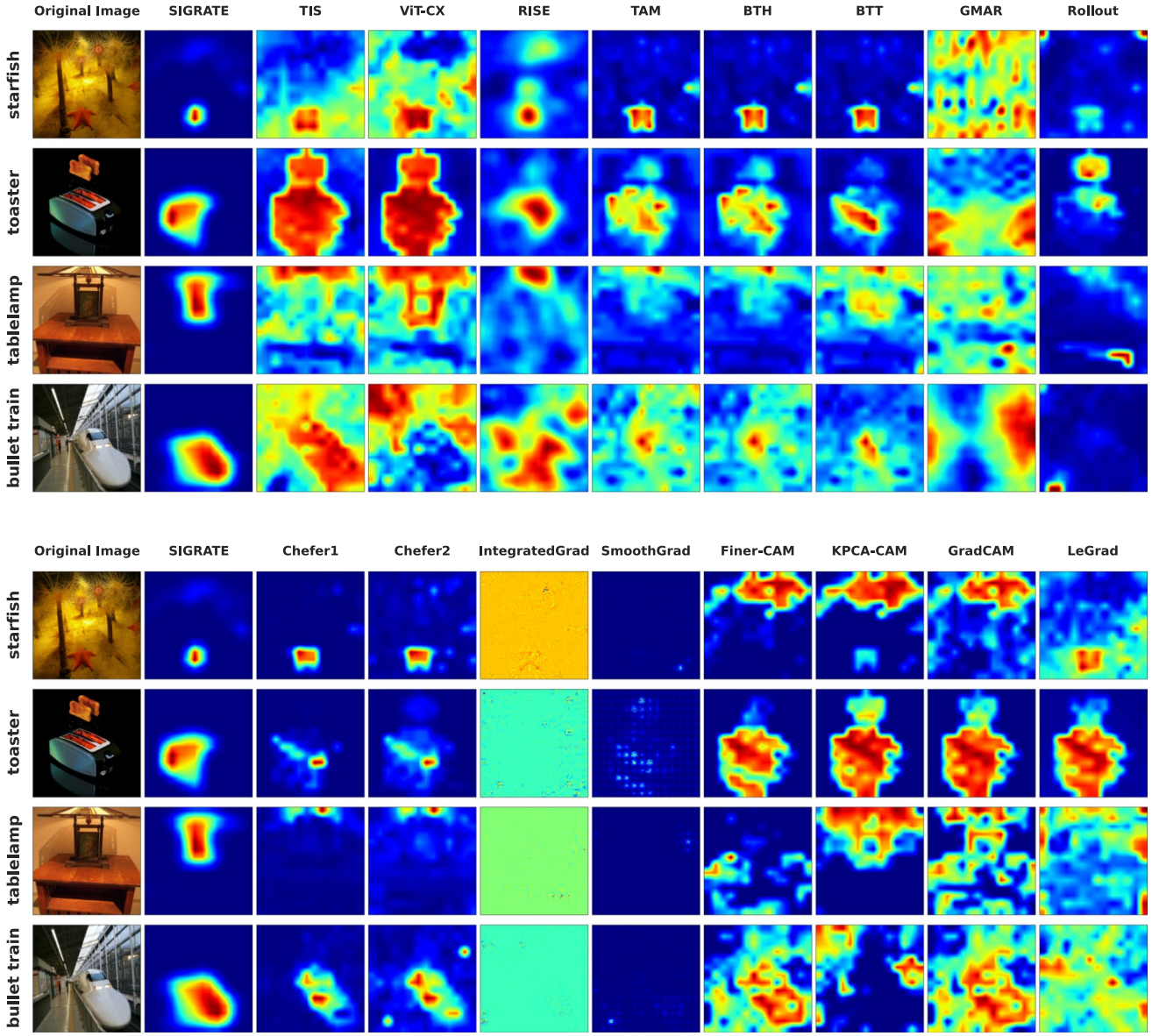


Fig. 3 Comparison of the heatmaps returned by SIGRATE and related approaches for four images of ImageNet using ViT-Base. SIGRATE effectively highlights the salient regions of each image and suppresses most of the background

gives less importance to the apples surrounding it. A similar situation occurs in the fourth row (*saltshaker*) of Fig. 4, where only one of the two shakers is highlighted.

In summary, these cases demonstrate that SIGRATE functions as a faithful interpreter of the behavior of the underlying ViT model. Its tendency to isolate specific instances or follow the predicted class (even when incorrect) confirms that the generated masks are driven by the internal decision-making process of the ViT underlying SIGRATE. While these heatmaps may appear incomplete compared to human annotations, they accurately reveal the specific, often localized visual evidence on which the underlying ViT model relies to make predictions. This further validates the consistency of the similarity-based approach.

In the second part of the qualitative comparison, we selected three images of ImageNet belonging to the *ski*, *pickup*, and *spoonbill* classes. Each image has its own features, background, and size of the object of interest, with bounding boxes provided by ImageNet indicating the location of the object. We computed the heatmap, Insertion

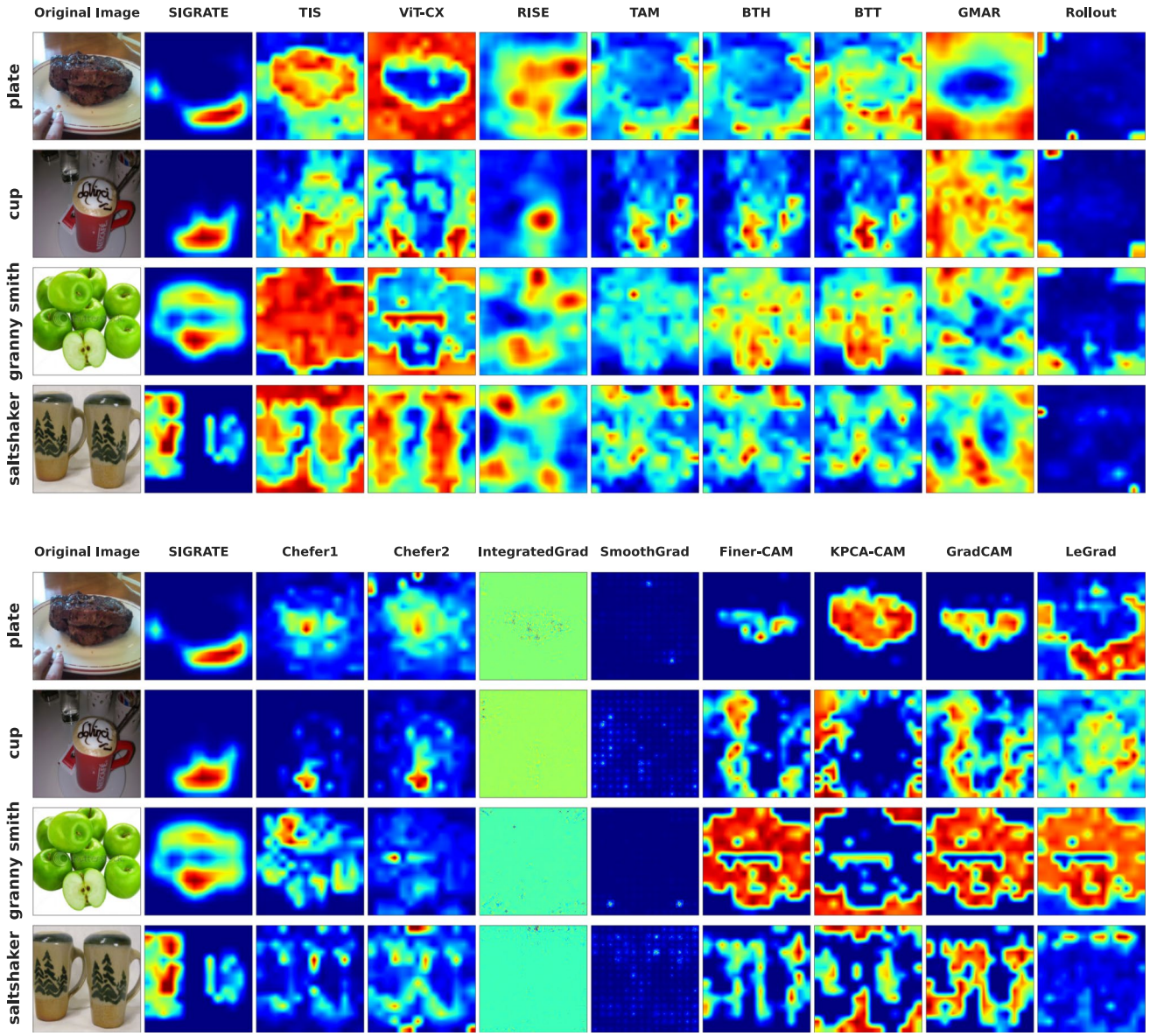


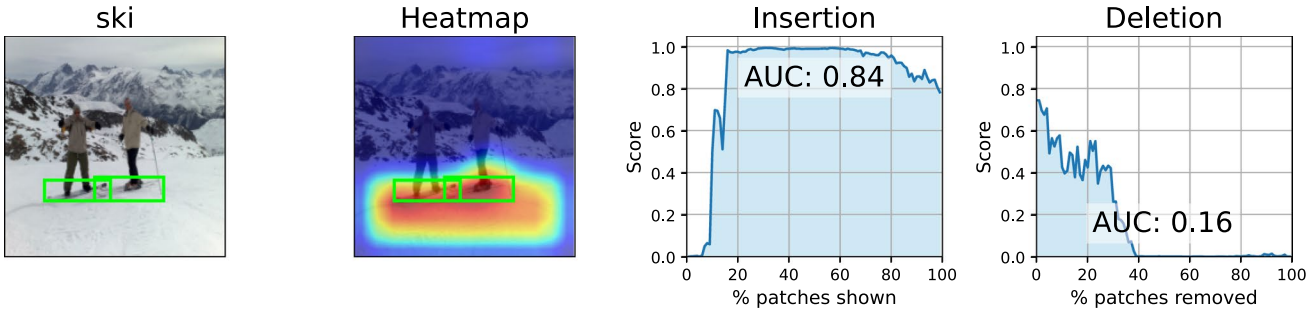
Fig. 4 Comparison of the heatmaps returned by SIGRATE and related approaches for four images of ImageNet using ViT-Base. The first two rows (*plate* and *cup*) show misclassifications, with the predicted classes being *meat loaf* and *espresso*, respectively. The third and fourth rows (*granny smith* and *saltshaker*) show scenarios with multiple instances of the same object, in which SIGRATE may focus on a single instance

curve, and Deletion curve for each image. For these analyses, we obscured the patches by applying the Black obfuscation method. Finally, we used both ViT-Base and DeiT-Base as underlying ViTs.

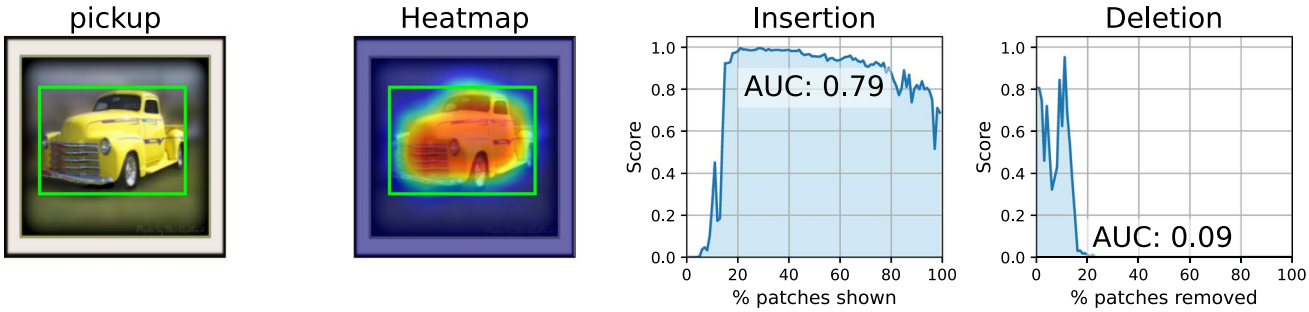
Figure 5 shows the results returned by SIGRATE when the underlying ViT model is ViT-Base. From the analysis of this figure, we can draw some interesting conclusions about SIGRATE's behavior. In fact, looking at the *ski* image (Fig. 5a), we see that SIGRATE gives great importance to the skis, shoes, and legs of the two people, as well as the snow around them.

The Insertion curve reveals that SIGRATE needs only 10% of the patches to correctly classify the image. This indicates that our approach can correctly identify the most salient areas. As proof of this, it achieves a very high AUC Insertion value of 0.84. When the number of patches exceeds 70%, the values of the Insertion curve decrease. This can be explained by the fact that the last patches are very noisy and have been correctly discarded

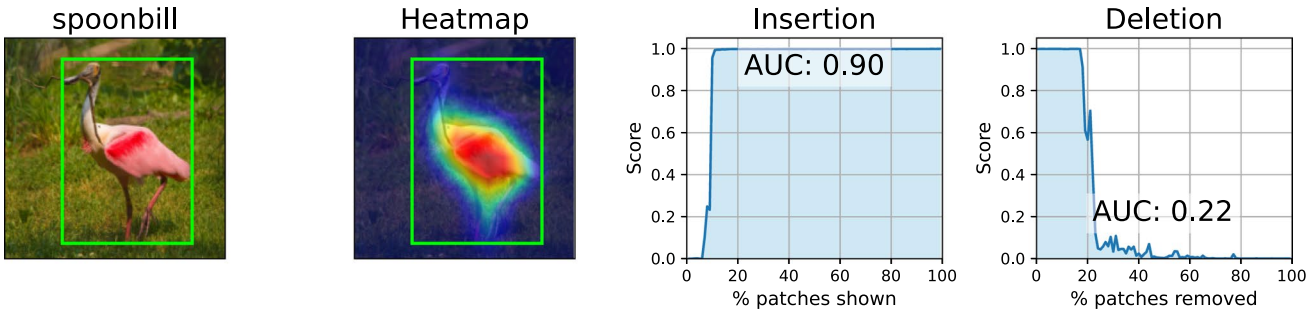
 https://doi.org/10.1007/s00521-026-12392-6



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



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



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

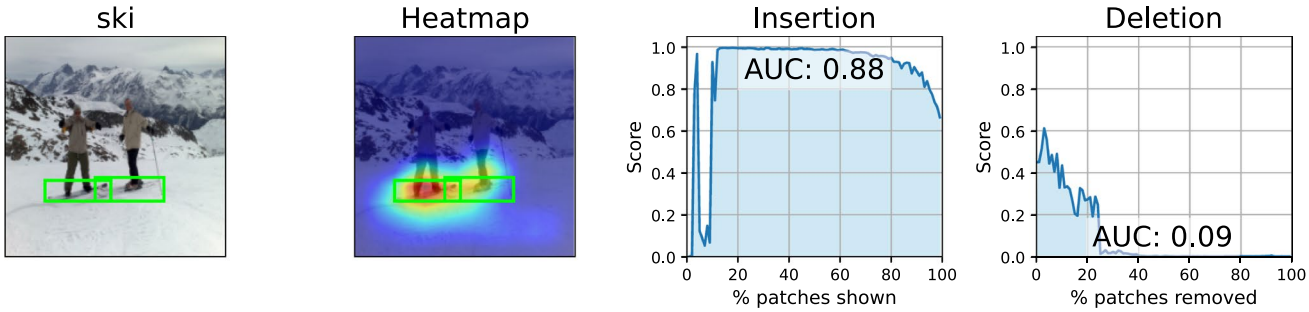
Fig. 5 Heatmaps, Insertion curves, and Deletion curves obtained by SIGRATE on three images using ViT-Base. The bounding boxes in the images indicate the location of the object to be classified

by SIGRATE. Adding them gradually will only worsen ViT-Base's confidence in classifying images. The Deletion curve decreases very rapidly with the removal of only 5–10% of the patches, corresponding to a low AUC value of 0.16.

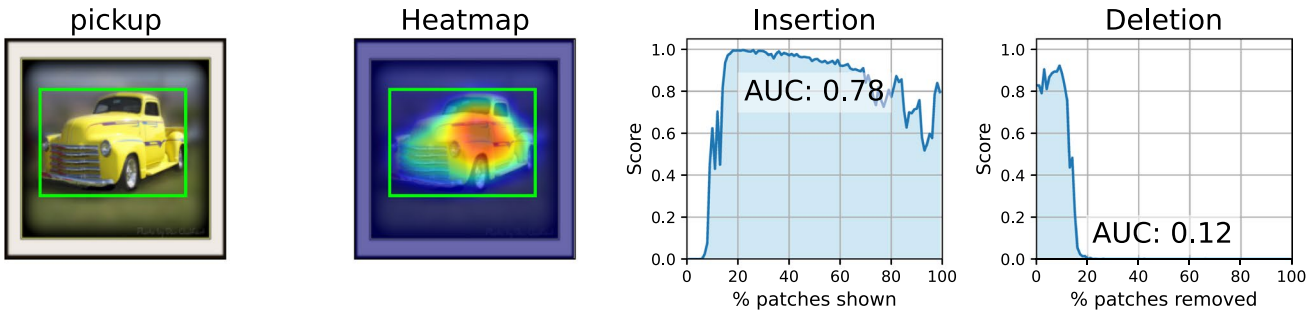
Regarding the *pickup* class (Fig. 5b), we observe that SIGRATE gives very high importance to the whole shape, while giving no importance to the background. Considering the Insertion curve, SIGRATE needs 15–20% of the patches to correctly classify this image. Beyond 20%, Insertion decreases as the percentage of patches increases. The reason is the same as for the previous image. Overall, we get an AUC value of 0.79 for this curve. If we consider the Deletion curve, we can see that it decreases rapidly with the removal of 10–15% of the patches, which corresponds to a very low AUC value of 0.09.

As far as the *spoonbill* image (Fig. 5c) is concerned, the heatmap correctly highlights the spoonbill's shape, to which SIGRATE assigns the most importance. SIGRATE gives less importance to the background near the spoonbill and even less importance to the background far from it. The Insertion curve rises rapidly as the number of selected patches increases from 10% to 15%, and then settles around the maximum value. This trend indicates that SIGRATE is very good at identifying the most salient parts of an image for classification purposes. This is confirmed by the AUC for this curve, which has a value of 0.90. Finally, the Deletion curve decreases rapidly as the number of patches removed increases from 15% to 20%. The overall AUC for this curve has a value of 0.22, which is quite higher (and therefore worse) than the previous two cases, although it is still acceptable.

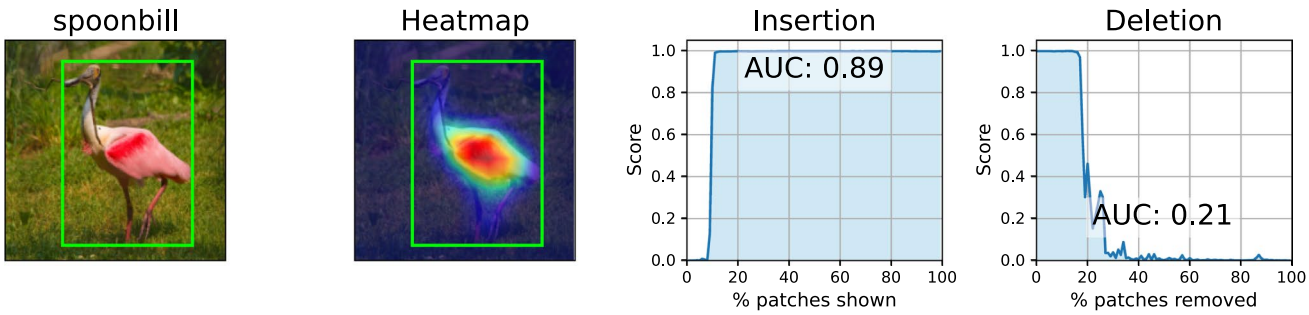
We now consider the performance of SIGRATE on the same images, but with DeiT-Base as underlying ViT model. Fig. 6 shows the results obtained.



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



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



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

Fig. 6 Heatmaps, Insertion curves, and Deletion curves obtained by SIGRATE on three images using DeiT-Base. The bounding boxes in the images indicate the location of the object to be classified

From the analysis of this figure, we can make some interesting considerations. Regarding the *ski* image (Fig. 6a), SIGRATE assigns a lot of importance to the skis. Compared to the ViT-Base heatmap, the DeiT-Base one focuses more on salient areas of the image class while giving even less weight to the background. The Insertion curve reaches a very high value with very few patches and has an AUC value of 0.88, which is higher than that achieved in the ViT-Base case. The Deletion curve has a steep slope at 5–10% of the patches and remains very low as the percentage of patches increases. The resulting AUC is very low at 0.09.

Looking at the *pickup* image (Fig. 6b), we can see that SIGRATE focuses on the center of the pickup, the edges of the pickup are less important, and the background is ignored. The Insertion curve reaches a high value at 20% of the patches and increases as the percentage of patches increases. In this case, the AUC value is 0.78. Similarly, the Deletion curve has a steep slope at 10% of the patches and reaches an AUC of 0.12.

Finally, considering the *spoonbill* image (Fig. 6c), we can see that SIGRATE correctly identifies the spoonbill's shape. The Insertion curve shows high growth already with only 10% of the patches; the corresponding AUC value is very high at 0.89. The Deletion curve shows a steep decrease at 15–20% of the patches and remains very low as the percentage of patches increases. In this case, the overall AUC value is 0.21.

After these analyses an important consideration is in order. The analysis based on the Pointing Game that we have seen in Sect. 4.2 already showed that the explanations generated by SIGRATE tend to align with human annotations, since the most relevant areas identified by the model often fall within the bounding boxes provided by ImageNet. Now, by analyzing the heatmaps, we can confirm that SIGRATE consistently focuses on the most informative areas of the images, i.e., those also highlighted by human-labeled bounding boxes. These results demonstrate that SIGRATE not only can identify salient features for classification but also can ensure that the results are consistent with human perceptions.

To conclude our qualitative analysis, we report some examples that highlight another peculiarity of SIGRATE. Specifically, we want to test whether it is able to work with images containing two objects belonging to two different classes. It often happens that an image contains objects belonging to different classes, and the presence of more than one class could change the explainability results. We performed this analysis by selecting four images containing two objects from different classes (e.g., an *elephant* and a *zebra*) and computing the saliency map for each class. The results are shown in Figs. 7, 8, 9 and 10.

From the analysis of these figures, we can see that SIGRATE can work with complex images containing more than one class of objects. For example, if we consider the class *elephant* in Fig. 7, the saliency map returned by SIGRATE correctly highlights the shape of the elephant, while giving little importance to the background and the zebra. In contrast, if we consider the class *zebra*, the saliency map returned by SIGRATE gives great importance to the shape of the zebra, while the background and the elephant are almost discarded. The same behavior can also be observed in the other figures, where the saliency map obtained by SIGRATE only highlights the object

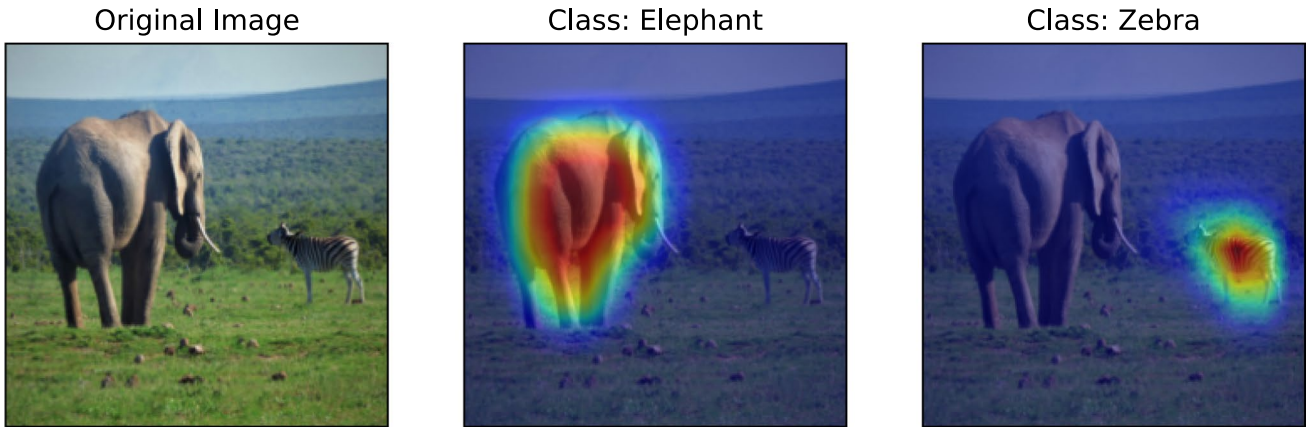


Fig. 7 Saliency maps in an image containing two objects from different classes (an elephant and a zebra)

Fig. 8 Saliency maps in an image containing two objects from different classes (a golden retriever and a tabby cat)

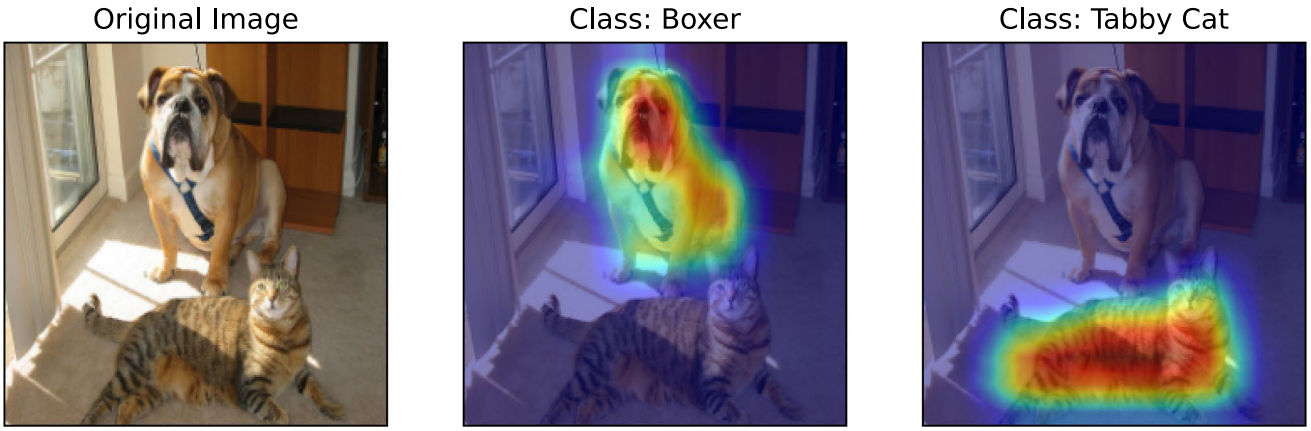


Fig. 9 Saliency maps in an image containing two objects from different classes (a boxer and a tabby cat)

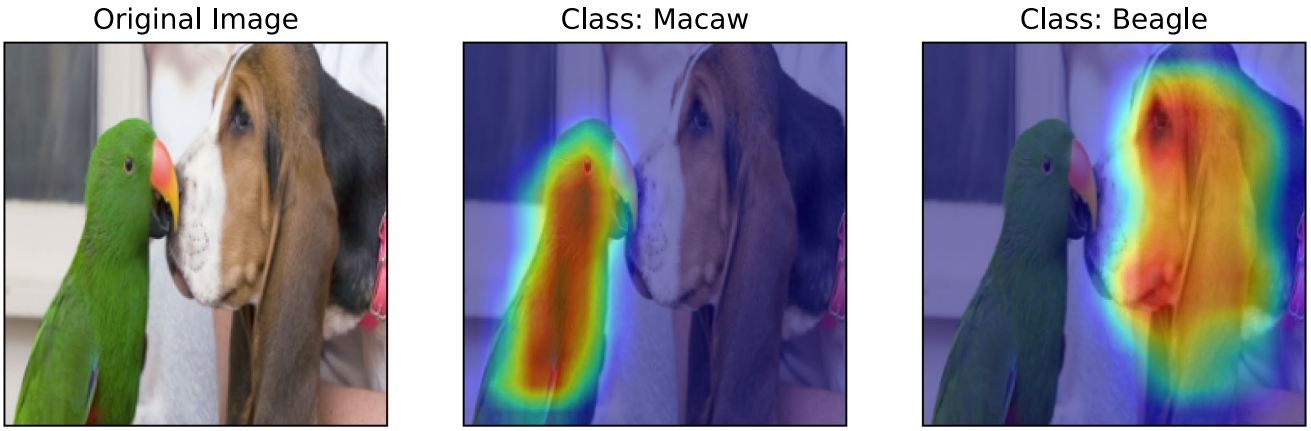


Fig. 10 Saliency maps in an image containing two objects from different classes (a macaw and a beagle)

of the selected class. This demonstrates SIGRATE's ability to accurately distinguish multiple classes of objects within a complex image.

4.4 Hyperparameter analysis

In this section, we analyze the impact of SIGRATE's hyperparameters on its performance. Due to space limitations, our study focuses only on ViT-Base; similar behaviors can be observed in the presence of DeiT-Base.

As a first analysis, we want to test the behavior of SIGRATE with different values of the token ratio *r*. Recall that *r* defines the maximum length of the walk performed on the similarity graph G_{S_l} (see Sect. 3). The higher the value of *r*, the longer the walk and the greater the number of patches selected to construct a mask. During the construction of the walk, it is possible to pass over the same node (and therefore the same patch) several times. As a consequence, since a patch can be taken only once, it is possible to obtain masks (much) smaller than the maximum size. In Table 7, we report the values of the Insertion, Deletion, and Insertion-Deletion metrics returned by SIGRATE when the value of *r* is 0.25, 0.50, and 0.75. We also report the average mask length for each possible value of *r*. Finally, we show in bold the best value obtained for each column. In these experiments, we assume that the value of μ is equal to 12.

From the analysis of this table, we can see that the average mask length is directly proportional to the value of *r*, though a scaling factor is present due to the masks' dynamic generation mechanism. In fact, when $r = 0.75$ (resp., 0.50, 0.25), rather than using 147 (resp., 98, 49) patches for each mask, only 76 (resp., 56, 33) are used on average. This emphasizes the effect of the dynamic component behind walk generation. In fact, a significant decrease in mask length implies a decrease in computational complexity compared to approaches with a fixed mask size (such as ViT-CX [47], RISE [18] or TIS [21]). This table also shows that SIGRATE achieves the best performance when $r = 0.75$.

When $r = 0.25$, its performance decreases for all metrics and obfuscation methods. This decrease is due to the low number of patches selected along the walks. Conversely, when $r = 0.50$, there is only a slight decrease in performance, as can be seen from the values of Deletion and Insertion-Deletion.

Having carried out the analysis on the hyperparameter *r*, we now present the same study on the other hyperparameter of SIGRATE, i.e., μ, which indicates the number of masks generated for each layer of the ViT of interest (see Sect. 3). Again, we considered the Insertion, Deletion, and Insertion-Deletion metrics and the four obfuscation methods used in the other experiments. In this analysis, we set *r* equal to 0.75, since this is the optimal value identified in the previous experiment. Again, for space reasons, we only report the results obtained with ViT-Base. We considered three values of μ, namely $\mu = 4$, $\mu = 8$, and $\mu = 12$. The results obtained are shown in Table 8; as in the previous table, we have bolded the best values in each column.

From the analysis of this table, we can see that, as μ increases from 4 to 12, all metrics improve. In particular, regarding Insertion, SIGRATE performs well for all values of μ, with slight improvements as μ increases.

Regarding Deletion, SIGRATE achieves optimal results at $\mu = 12$; these results often coincide with the values at $\mu = 8$. At $\mu = 4$, Deletion values are slightly higher, and therefore SIGRATE's performance is slightly worse, suggesting that fewer masks per layer limit SIGRATE's ability to isolate the most critical patches. A similar trend is obtained for Insertion-Deletion values. Again SIGRATE achieves optimal values at $\mu = 12$, equal or slightly lower values at $\mu = 8$, and further slightly lower values at $\mu = 4$. Since the best results are always obtained with $\mu = 12$, we chose this value as the reference one.

In addition to the parameters μ and *r*, SIGRATE requires setting the decay factor δ, which controls how much greedy walks in the similarity graph are encouraged to explore new nodes instead of revisiting previously selected ones. Lower values of δ promote broader exploration by progressively reducing the importance of traversed edges, while higher values of δ lead to walks that tend to remain in highly similar regions. We carried out a hyperparameter study to analyze the impact of different values of δ on SIGRATE's performance. Specifically, we tested three values of δ, i.e., $\delta = 0.2$, $\delta = 0.5$, and $\delta = 0.8$. Table 9 reports the results of this experiment.

https://doi.org/10.1007/s00521-026-12392-6

Table 7 Values of Insertion, Deletion, and Insertion-Deletion, and average mask length returned by SIGRATE with different values of the hyperparameter r. As the results show, setting $r = 0.75$ yields the best overall performance across all metrics

r	Avg.	Insertion↑				Deletion↓				Insertion-Deletion↑			
	length	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
0.25	33	0.62	0.69	0.63	0.60	0.22	0.44	0.21	0.20	0.40	0.25	0.42	0.40
0.50	56	**0.65**	0.69	**0.65**	**0.62**	0.20	0.40	0.20	**0.17**	0.45	0.29	0.45	**0.46**
0.75	76	**0.65**	**0.70**	**0.65**	**0.62**	**0.19**	**0.39**	**0.19**	**0.17**	**0.47**	**0.30**	**0.47**	**0.46**

The best value in each column is shown in bold.

https://doi.org/10.1007/s00521-026-12392-6

Table 8 Values of insertion, deletion, and insertion-deletion returned by SIGRATE with different values ofμ, which defines the number of masks generated for each layer

μ	Insertion↑				Deletion↓				Insertion-Deletion↑			
	Mean	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
4	0.63	0.68	0.64	0.61	0.20	0.41	0.20	0.18	0.43	0.27	0.44	0.43
8	0.64	0.69	0.64	0.61	**0.19**	0.40	**0.19**	**0.17**	0.46	**0.30**	0.46	0.44
12	**0.65**	**0.70**	**0.65**	**0.62**	**0.19**	**0.39**	**0.19**	**0.17**	**0.47**	**0.30**	**0.47**	**0.46**

As shown, an increase in μleads to higher values in all metrics, indicating that a larger number of masks improves the quality of explanations

The best value in each column is shown in bold

Table 9 Performance of SIGRATE with different values of the decay factorδ

δ	Insertion↑				Deletion↓				Insertion-Deletion↑			
	Mean	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
0.2	0.63	**0.70**	0.63	0.60	0.20	0.40	0.21	0.19	0.43	**0.30**	0.42	0.41
0.5	**0.65**	**0.70**	**0.65**	**0.62**	**0.19**	**0.39**	**0.19**	**0.17**	**0.47**	**0.30**	**0.47**	**0.46**
0.8	0.64	0.68	0.64	0.60	0.19	0.40	0.20	0.18	0.45	0.28	0.44	0.42

The table reports the results in terms of Insertion, Deletion, and Insertion-Deletion. It highlights that $\delta = 0.5$provides the best tradeoff across all metrics and obfuscation methods, confirming its suitability as the default setting

The best value in each column is shown in bold

After analyzing this table and seeking a tradeoff between Insertion and Deletion, we chose $\delta = 0.5$ as the default value of this hyperparameter in SIGRATE. This value ensures a balanced behavior, promoting a sufficient exploration degree while maintaining the focus on the most relevant regions of the image.

Next, we analyzed the computational load of SIGRATE. The previous experimental results have focused on a standard input resolution of 224×224 pixels. To evaluate the scalability of our approach and the related ones, we analyzed their computational cost to generate explanations both for images of 224×224 pixels and for larger images of 384×384 pixels. Table 10 reports the number of Giga FLOating Point Operations (GFLOPs) and the processing time per image (averaged over 100 images) for three values of μ with both resolutions. It also compares these values with those of the other related approaches considered in Sect. 4.2.

The results shown in Table 10 for standard resolution (i.e., 224×224 pixels) highlight that, even for the most computationally demanding configuration (i.e., $\mu = 12$), SIGRATE is significantly more efficient than most existing approaches. Its computational cost is notably lower than that of perturbation-based approaches, such as TIS [21], RISE [18], and ViT-CX [47], as well as gradient-based approaches, such as IntegratedGrad [27]. At the same time, its faithfulness performance is comparable to or superior to that of these approaches, as seen in Tables 1 and 3. As μ increases from 8 to 12, the number of GFLOPs required by SIGRATE increases slightly. However, this additional cost is more than offset by the corresponding improvements in Insertion, Deletion and Insertion-Deletion, as shown in Table 8. Some methods, such as Chefer1 [17], Chefer2 [56], Finer-CAM [36], and KPCA-CAM [35], require fewer GFLOPs than SIGRATE. However, they achieve significantly lower faithfulness performance, as highlighted in Tables 1, 2 and 3.

Regarding the processing of higher-resolution images (i.e., 384×384 pixels), the results in Table 10 confirm SIGRATE's scalability. The computational load inevitably increases with this higher image resolution; however, execution time remains low. The combination of this scaling behavior and the very high faithfulness performance values achieved by SIGRATE (see Tables 1 and 3 again) makes our framework particularly suitable for processing high-resolution images where computational resources are a constraint and, at the same time, a high faithfulness performance is desired.

Table 11 provides a more detailed analysis of the computational overhead required by SIGRATE and other mask-based approaches to generate a single mask. Since SIGRATE uses an adaptive mechanism that creates masks of variable length based on image content, the values in the table represent the average time and the

Table 10 Comparison of the execution time and GFLOPs required by SIGRATE and related approaches to process a single image

Approach	224×224 pixels		384×384 pixels	
	Time (s)↓	*GFLOPs*↓	*Time (s)*↓	*GFLOPs*↓
SIGRATE ($\mu = 4$)	0.31 ± 0.03	512.28	1.17 ± 0.08	1,573.79
SIGRATE ($\mu = 8$)	0.52 ± 0.04	1,085.59	1.70 ± 0.11	3,360.27
SIGRATE ($\mu = 12$)	0.71 ± 0.05	1,703.61	2.21 ± 0.14	5,316.72
TIS [21]	5.42 ± 0.38	17,808.70	16.01 ± 1.12	50,822.06
RISE [18]	36.39 ± 2.45	140,927.36	128.18 ± 8.67	396,006.97
VIT-CX [47]	1.40 ± 0.09	4,918.71	4.60 ± 0.31	9,859.79
Chefer1 [17]	0.59 ± 0.04	377.67	0.68 ± 0.05	1,158.66
Chefer2 [56]	0.04 ± 0.01	105.13	0.12 ± 0.01	336.36
BT H [42]	1.04 ± 0.07	2,264.73	2.73 ± 0.19	7,197.32
BT T [42]	1.11 ± 0.08	2,262.70	2.60 ± 0.17	7,146.51
TAM [45]	0.82 ± 0.06	2,203.77	2.35 ± 0.16	6,965.76
IntegratedGrad [27]	3.15 ± 0.24	3,572.95	11.63 ± 0.89	9,836.21
SmoothGrad [26]	3.36 ± 0.26	3,572.93	12.69 ± 0.95	9,836.16
Finer-CAM [36]	0.06 ± 0.01	85.06	0.48 ± 0.03	294.74
KPCA-CAM [35]	0.10 ± 0.01	170.98	0.21 ± 0.02	98.71
GradCAM [25]	0.07 ± 0.01	100.63	0.18 ± 0.01	118.12
LeGrad [44]	0.05 ± 0.01	35.97	0.19 ± 0.02	106.37
Attention Rollout [40]	0.13 ± 0.01	35.50	0.19 ± 0.02	100.12
GMAR [46]	0.34 ± 0.03	142.28	0.95 ± 0.07	498.06

Two image resolutions are considered, i.e.,224×224 pixels and 384×384 pixels. The table shows that SIGRATE is computationally more efficient than most related approaches (especially, the mask-based ones), even when the value of μ and the image resolution increase

Table 11 Comparison of the time and GFLOPs required by SIGRATE and other mask-based approaches for generating a single mask

Approach	224×224 pixels		384×384 pixels	
	Time (s)↓	*GFLOPs*↓	*Time (s)*↓	*GFLOPs*↓
SIGRATE	$(6.16 \pm 0.05) \times 10^{-5}$	0.10	$(2.12 \pm 0.03) \times 10^{-4}$	0.39
TIS [21]	$(3.21 \pm 0.02) \times 10^{-4}$	1.55	$(8.79 \pm 0.04) \times 10^{-4}$	2.08
VIT-CX [47]	$(5.88 \pm 0.04) \times 10^{-4}$	2.05	$(1.96 \pm 0.02) \times 10^{-3}$	4.20
RISE [18]	0 ± 0	0	0 ± 0	0

Two resolutions are considered: 224×224 pixels and 384×384 pixels. The results demonstrate that SIGRATE guarantees a higher efficiency than the other approaches slightly increasing both runtime and computational load as resolution increases. The low values of RISE are offset by the extremely high values of time and GFLOPs required by this approach to process a single image (see Table 10)

Table 12 Analysis of SIGRATE scalability on images of size 224×224 pixels as the hyperparameters r, δ and μ vary

The values show that the computational cost (GFLOPs) is mainly influenced by the increase in μ and r, while remaining substantially stable as δ varies

r	δ	*GFLOPs*		
		$\mu = 4$	$\mu = 8$	$\mu = 12$
0.25	0.2	384.02	690.59	1014.94
	0.5	384.37	688.70	1015.12
	0.8	378.55	681.67	1006.53
0.50	0.2	447.94	886.46	1362.83
	0.5	448.60	889.98	1359.76
	0.8	447.38	879.12	1353.23
0.75	0.2	511.26	1,080.16	1,707.02
	0.5	512.28	1,085.59	1,703.61
	0.8	512.79	1,076.91	1,700.20

average GFLOPs needed to generate each mask. These average values were computed using 100 images. As can be seen from the table, SIGRATE demonstrates remarkable efficiency and negligible processing time at standard resolution, consistently outperforming TIS and ViT-CX. Although the generation cost naturally increases with the resolution of the input, it remains orders of magnitude smaller than the inference time of the underlying model (i.e., ViT-Base or DeiT-Base). Indeed, most of the computational time used for the execution of the approach (see Table 10) is required for the classification of the generated masks. Finally, we note that RISE has virtually zero cost for generating masks thanks to its random sampling strategy. However, it typically requires a significantly larger number of masks to achieve comparable faithfulness to the other approaches, thus shifting the computational load to the classification phase (see Table 10).

To conclude the hyperparameter analysis, Table 12 shows an analysis of SIGRATE's scalability as hyperparameters r, μ, and δ vary. This analysis reveals that the number μ of masks and the token ratio r primarily influence computational cost. These factors lead to approximately linear growth in the GFLOPs required as their values increase. When compared to the performance metrics in Tables 7, 8 and 9, it is evident that the increase in computational requirements is justified by the substantial improvements in the Insertion, Deletion, and Insertion-Deletion metrics. Instead, the decay factor δ has a negligible impact on the GFLOPs required by SIGRATE to perform its tasks. This implies that δ can be adjusted to optimize the exploration strategy without incurring additional computational overhead.

4.5 Ablation study

A key step in SIGRATE is the construction of the similarity graph G_{S_l} on which greedy walks are computed to generate saliency masks. To analyze the impact of this component, we conducted an ablation study comparing four different strategies to weigh the edges between the nodes representing patch embeddings. These strategies are:

- a similarity-based graph (*sim*), which uses the cosine similarity between patch embeddings;
- an attention-based graph (*att*), which uses the attention scores of each attention layer of the ViT;
- a combined graph (*comb*), which integrates similarity and attention information;
- a Euclidean distance-based graph (*dist*), which inverts the distance between embeddings and uses this value as a similarity measure.

Table 13 reports the Insertion, Deletion, and Insertion-Deletion values obtained with these configurations. In the computation of these values, we set $\mu = 12$ and $r = 0.75$.

The analysis of this table shows that *sim* achieves the best overall performance. It obtains competitive values for Insertion and achieves the best values for Deletion with all obfuscation methods. This leads it to obtain the highest Insertion-Deletion values, confirming the effectiveness of the similarity graph in producing both informative and robust explanations.

att shows Insertion values similar to the ones of *sim*, but higher Deletion values. This suggests that attention alone is insufficient to capture the relational structure between patches and tends to focus exclusively on the regions most highlighted by the model, neglecting the surrounding ones.

comb does not improve the performance of *att*; in fact, it achieves similar results. This indicates that combining attention and similarity signals may introduce redundancies or conflicts that reduce the effectiveness of walks in selecting the most relevant patches. *dist* obtains slightly higher absolute values than the other three approaches for Insertion. However, this comes at the expense of higher Deletion values, resulting in lower Insertion-Deletion values than *sim*. Based on these results, we chose the similarity-based graph as the default configuration for SIGRATE, because it is the best tradeoff between the four approaches.

5 Discussion

The experiments presented in the previous section show SIGRATE's ability to identify areas of the images that led a ViT to predict a particular class. For each attention layer, SIGRATE extracts a set of masks representing the relevant and irrelevant areas with respect to that specific part of the model. Each mask is constructed thanks to a similarity graph that records the similarity between pairs of patches, and the analysis of the attention that the classification (CLS) token places on all the other patches.

SIGRATE starts with the patches that have the highest and lowest attention given by the CLS token and builds a set of walks that are the basis for the construction of the masks. It then merges these masks using the coverage bias formula. Finally, it obtains a heatmap containing the salient areas of an image that led the ViT to a given classification result. A closer visual inspection of the heatmaps, particularly those shown in Figs. 5 and 6, reveals a strong correspondence between the areas highlighted by SIGRATE and the actual regions of the objects defined by the bounding boxes. Rather than producing coarse or sparse heatmaps, SIGRATE accurately outlines the semantic structure and contours of target objects. For example, it captures the specific shape of the pickup truck or the skis and legs of the skiers. This precise alignment suggests that similarity-based walks effectively aggregate semantically coherent patches. This ensures that SIGRATE strictly focuses on relevant features and successfully suppresses background noise. These qualitative results are validated quantitatively by the results obtained for the Pointing Game metric in Table 3. Since this metric is an indicator of localization accuracy, its high values obtained by SIGRATE confirm that the masks generated by our framework effectively focus on the actual area of the object. These results demonstrate that SIGRATE aligns well with human perception of salience. Furthermore, our framework demonstrates remarkable robustness in complex scenarios containing multiple objects of different classes, as illustrated in Figs. 7, 8, 9 and 10. In these multi-class images, SIGRATE successfully separates the visual content, isolating and highlighting only the specific region relevant to the selected target class, while ignoring other co-occurring objects.

https://doi.org/10.1007/s00521-026-12392-6

Table 13 Ablation study on the performance of SIGRATE with different graph construction strategies

Graph Type	Insertion↑				Deletion↓				Insertion-Deletion↑			
	Mean	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*	*Mean*	*Blur*	*Black*	*Rand*
sim	0.65	**0.70**	0.65	0.62	**0.19**	**0.39**	**0.19**	**0.17**	**0.47**	**0.30**	**0.47**	**0.46**
att	0.65	**0.70**	0.65	0.61	0.22	0.42	0.22	0.20	0.43	0.27	0.43	0.41
comb	0.65	**0.70**	0.65	0.61	0.22	0.42	0.22	0.21	0.43	0.27	0.43	0.40
dist	**0.66**	**0.70**	**0.66**	**0.63**	0.23	0.42	0.23	0.21	0.43	0.29	0.43	0.42

The similarity-based graph (sim) performs best, achieving the optimal tradeoff between Insertion and Deletion with all obfuscation methods. The att, comb and dist variants show lower or unbalanced performance

The best value in each column is shown in bold

The main advantage of SIGRATE is that it considers the contribution of each attention layer in the resulting heatmap. Each attention layer learns different patterns from the image. To represent this knowledge, SIGRATE builds a set of masks for each attention layer using the similarity graph and the corresponding walks. In this way, it obtains a set of masks from each model layer, which are then merged. Another advantage of SIGRATE over mask-based approaches [21, 47] concerns the number of masks required to achieve the performance reported in the previous section. SIGRATE uses only 144 masks, while the other approaches require between 70 [47] to 1024 [21] masks. Moreover, the masks generated by SIGRATE are smaller, thus reducing computational cost, as shown in Table 10.

6 Conclusion

In this paper, we proposed SIGRATE, an explainability approach for Vision Transformers based on mask generation. SIGRATE combines the strengths of attention- and mask-based approaches while overcoming their respective weaknesses. It addresses this issue through a hybrid framework that: *(i)* leverages CLS token attention scores to select the most and least influential patches, *(ii)* constructs a layer-specific patch-similarity graph to capture patch relationships, and *(iii)* performs greedy walks on each graph to generate a set of binary masks that perturb specific image areas. It then merges these masks using the coverage bias formula. Finally, it reshapes and upsamples the resulting heatmap to the original image resolution, merging global context with causality-driven insights.

Overall, SIGRATE's main contribution is the generation of explanations through similarity-graph traversal. Here, greedy walks on patch-embedding similarity produce structured and consistent sets of masks. To the best of our knowledge, this mechanism is not employed by existing ViT explainability approaches. SIGRATE combines global importance cues with the similarity structure encoded in patch embeddings by integrating CLS-driven relevance with these graph-derived masks. This produces consistent and semantically meaningful heatmaps, as confirmed by our experimental evaluation.

We tested SIGRATE using the ViT-Base and DeiT-Base models on a subset of the ImageNet validation set and on the BloodMNIST dataset. From the experimental results, we observed that it achieves state-of-the-art performance for Insertion, Insertion-Deletion and Pointing Game while it is in line with most related approaches for Deletion. Taking into account the required execution time and GFLOPs, SIGRATE shows a better performance than most of the tested approaches. The only approaches performing better than SIGRATE on this aspect show much lower performance on faithfulness metrics. Finally, we presented a qualitative analysis demonstrating SIGRATE's capabilities on several images, conducted a hyperparameter analysis, and performed an ablation study over different graph construction strategies. The whole experimental campaign confirmed that SIGRATE can visually highlight the salient areas that led a ViT to a specific classification.

However, in addition to the aforementioned strengths, SIGRATE has some limitations. First, it can only work with ViTs that use the classification (i.e., CLS) token. While most ViTs (and Transformers in general) use this token, some architectures do not employ it [58, 59]. In these cases, SIGRATE cannot interpret these models because the walk computation is based on identifying the most relevant and irrelevant patches according to the CLS token. One possible solution is to compute relevance at the penultimate layer by estimating each token's contribution to the final class embedding (e.g., via dot-product similarity or local attention flow). These relevance signals could then replace CLS attention in the mask-generation process, preserving the core structure of SIGRATE.

Another limitation is that SIGRATE is not applicable to Hierarchical Vision Transformers, such as Swin [59]. These architectures use sliding-window attention, which progressively reduces the number of tokens across layers. This prevents semantic consistency between tokens at different layers. As a consequence, SIGRATE cannot apply the coverage bias formula directly to merge all generated masks into a unified heatmap since the masks produced at different layers are not aligned in size or meaning. To address this issue, a mechanism would be

needed to estimate how tokens in one layer contribute to those in the next layer so that masks can be upscaled or propagated consistently.

A further limitation of SIGRATE is its dependence on the quality of patch embedding similarity. Since similarity graphs are constructed from single-layer embeddings, walks implicitly assume that cosine similarity reliably reflects semantic relatedness. However, embedding spaces can vary across layers, as early layers may be noisy, while deeper layers may compress or distort information. This can lead to the aggregation of semantically inconsistent regions. One possible solution is to incorporate multi-layer representations by combining embeddings or similarity signals from multiple layers (e.g., through weighted averages or cross-layer graph fusion). This would allow SIGRATE to rely on a more stable and robust representation of patch similarity, reducing its sensitivity to the distribution of layer-specific embeddings.

The proposal for overcoming the three limitations mentioned above represent three potential future developments of the research presented in this paper. Another possible development is the extension of our approach to Natural Language Processing Transformers, where a higher number of tokens could lead to longer walks on the similarity graph, which might be more difficult to interpret.

Author contributions In this paper Michele Marchetti, Davide Traini, Domenico Ursino, and Luca Virgili interacted with each other in all the tasks connected with the presented research. Their contribution is equal and this is testified by the alphabetical order used in the Author List.

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

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

Declarations

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

Competing 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.

References

1. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2021) An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Proc. of the International Conference on Learning Representations (**ICLR'21), Virtual event, 2021. OpenReview.net**)
2. Li Z, Liu F, Yang W, Peng S, Zhou J (2021) A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2021.3084827
3. Yang J, Li C, Zhang P, Dai X, Xiao B, Yuan L, Gao J (2021) Focal attention for long-range interactions in vision transformers. Adv Neural Inf Process Syst 34:30008–30022

4. Zhou D, Yu Z, Xie E, Xiao C, Anandkumar A, Feng J, Alvarez JM (2022) Understanding the robustness in vision transformers. In Proc. of the International Conference on Machine Learning (ICML'22), pages 27378–27394, Baltimore, MD, USA. PMLR
5. Gao H, Hu C, Han G, Mao J, Huang W, Guan Q (2024) Point-level feature learning based on vision transformer for occluded person re-identification. Image Vis Comput 143:104929 (**Elsevier**)
6. Thalhammer S, Weibel JB, Vincze M, Garcia-Rodriguez J (2023) Self-supervised vision transformers for 3d pose estimation of novel objects. Image Vis Comput 139:104816 (**Elsevier**)
7. Chong Z, Mo L (2022) ST-VTON: Self-supervised vision transformer for image-based virtual try-on. Image Vis Comput 127:104568 (**Elsevier**)
8. Bravo-Ortiz MA, Holguin-Garcia SA, Quiñones-Arredondo S, Mora-Rubio A, Guevara-Navarro E, Arteaga-Arteaga HB, Ruz GA, Tabares-Soto R (2024) A systematic review of vision transformers and convolutional neural networks for Alzheimer's disease classification using 3D MRI images. Neural Comput Appl 36(35):21985–22012
9. Poornam S, Angelina JJR (2024) VITALT: a robust and efficient brain tumor detection system using vision transformer with attention and linear transformation. Neural Comput Appl 36(12):6403–6419 (**Springer**)
10. Xu F, Uszkoreit H, Du Y, Fan W, Zhao D, Zhu J (2019) Explainable AI: A brief survey on history, research areas, approaches and challenges. In Proc. of the International Conference in Natural Language Processing and Chinese Computing (NLPCC'19), pages 563–574, Dunhuang, China . Springer
11. Şahin E, Arslan NN, Özdemir D (2025) Unlocking the black box: an in-depth review on interpretability, explainability, and reliability in deep learning. Neural Comput Appl 37(2):859–965 (**Springer**)
12. Bai B, Liang J, Zhang G, Li H, Bai K, Wang F (2021) Why attentions may not be interpretable? In Proc. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21), pages 25–34, Virtual Event
13. Bhan M, Achache N, Legrand V, Blangero A, Chesneau N (2023) Evaluating self-attention interpretability through human-grounded experimental protocol. In Proc. of the World Conference on Explainable Artificial Intelligence (XAI'23), pages 26–46, Lisbon, Portugal. Springer
14. Cheng L, Liang Y, Lu Y, Cheung YM (2025) GradToken: Decoupling tokens with class-aware gradient for visual explanation of Transformer network. Neural Netw 181:106837 (**Elsevier**)
15. Kashefi R, Barekatain L, Sabokrou M, Aghaeipoor F (2023) Explainability of Vision Transformers: A Comprehensive Review and New Perspectives. arXiv preprint arXiv:2311.06786
16. Covert I, Kim C, Lee S (2022) Learning to estimate shapley values with vision transformers. arXiv preprint arXiv:2206.05282
17. Chefer H, Gur S, Wolf L (2021) Transformer interpretability beyond attention visualization. In Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPR'21), pages 782–791, Virtual event,
18. Petsiuk V, Das A, Saenko K (2018) RISE: Randomized Input Sampling for Explanation of Black-box Models. In Proc. of the British Machine Vision Conference (BMVC'18), page 151, Newcastle, UK . BMVA Press
19. Jain S, Wallace BC (2019) Attention is not explanation. In Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '19), page 3543–3556, Minneapolis, MN, USA . ACL
20. Bibal A, Cardon R, Alfter D, Wilkens R, Wang X, François T, Watrin P (2022) Is attention explanation? An introduction to the debate. In Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL'23), pages Volume 1, 3889–3900, Dublin, Ireland
21. Englebert A, Stassin S, Nanfack G, Mahmoudi SA, Siebert X, Cornu O, De Vleeschouwer C (2023) Explaining through Transformer Input Sampling. In Proc. of the International Conference on Computer Vision (ICCV'23), pages 806–815, Paris, France
22. Marchetti M, Traini D, Ursino D, Virgili L (2025) Multiplex Network-Based Representation of Vision Transformers for Visual Explainability. Neural Comput Appl 37(29):24385–24420 (**Springer**)
23. 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. Scientific Data 10(1):41 (**Nature Publishing Group UK London**)
24. Alicioglu G, Sun B (2022) A survey of visual analytics for explainable artificial intelligence methods. Computers & Graphics 102:502–520 (**Elsevier**)
25. 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 Proc. of the International IEEE Conference on Computer Vision (ICCV'17), pages 618–626, Venice, Italy . IEEE
26. Smilkov D, Thorat N, Kim B, Viégas F, Wattenberg M (2017) Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825
27. Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In Proc. of the International Conference on Machine Learning (ICML'17), pages 3319–3328, Sydney, Australia . PMLR
28. Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In Proc. of the International Conference on Neural Information Processing Systems (NIPS'17), pages 4768–4777, Long Beach, CA, USA. Curran Associates

29. 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 Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL'19), pages 5797–5808, Florence, Italy
30. 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
31. Barkan O, Elisha Y, Weill J, Asher Y, Eshel A, Koenigstein N (2023) Deep integrated explanations. In Proc. of the ACM International Conference on Information and Knowledge Management (CIKM'23), pages 57–67, Birmingham, UK . ACM
32. Barkan O, Elisha Y, Weill J, Asher Y, Eshel A, Koenigstein N (2023) Stochastic integrated explanations for vision models. In Proc. of the IEEE International Conference on Data Mining (ICDM'23), pages 938–943, Shanghai, China, IEEE
33. Fel T, Ducoffe M, Vigouroux D, Cadène R, Capelle M, Nicodème C, Serre T (2023) Don't lie to me! robust and efficient explainability with verified perturbation analysis. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'23), pages 16153–16163, Vancouver, British Columbia, Canada. IEEE
34. Barkan O, Asher Y, Eshel A, Koenigstein N (2023) Visual explanations via iterated integrated attributions. In Proc. of the IEEE/CVF International Conference on Computer Vision (CVPR'23), pages 2073–2084, Vancouver, British Columbia, Canada
35. Karmani S, Sivakaran T, Prasad G, Ali M, Yang W, Tang S (2024) KPCA-CAM: Visual explainability of deep computer vision models using kernel PCA. In Proc. of the IEEE International Workshop on Multimedia Signal Processing (MMSP'24), pages 1–5, West Lafayette, IN, USA. IEEE
36. Zhang Z, Gu J, Chowdhury A, Mai Z, Carlyn D, Berger-Wolf T, Su Y, Chao WL (2025) Finer-cam: Spotting the difference reveals finer details for visual explanation. In Proc. of the Computer Vision and Pattern Recognition Conference (CVPR'25), pages 9611–9620, Nashville, TN, USA. Computer Vision Foundation / IEEE
37. Hafiz AM, Parah SA, Bhat RUA (2021) Attention mechanisms and deep learning for machine vision: A survey of the state of the art.arXiv:2106.07550 arXiv preprint
38. Hao Y, Dong L, Wei F, Xu K (2021) Self-attention attribution: Interpreting information interactions inside transformer. In Proc. of the AAAI International Conference on Artificial Intelligence (AAAI'21), volume 35, pages 12963–12971, Virtual event
39. Ma J, Bai Y, Zhong B, Zhang W, Yao T, Mei T (2023) Visualizing and understanding patch interactions in vision transformer. IEEE Transactions on Neural Networks and Learning Systems, pages 1–10 . IEEE
40. Abnar S, Zuidema W (2020) Quantifying Attention Flow in Transformers. In Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL'20), pages 4190–4197, Virtual event. Association for Computational Linguistics
41. 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 Proc. of the ACM International Conference on Information & Knowledge Management (CIKM'21), pages 2882–2887, Goald Coast, Queensland, Australia
42. Chen J, Li X, Yu L, Dou D, Xiong H (2023) Beyond Intuition: Rethinking Token Attributions inside Transformers. Transactions on Machine Learning Research . https://openreview.net/forum?id=rm0zIzlhcX
43. Wu J, Duan B, Kang W, Tang H, Yan Y (2024) Token Transformation Matters: Towards Faithful Post-hoc Explanation for Vision Transformer. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'24), pages 10926–10935, Seattle, WA, USA
44. Bousselham W, Boggust A, Chaybouti S, Strobelt H, Kuehne H (2025) Legrad: An explainability method for vision transformers via feature formation sensitivity. In Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV'25), pages 20336–20345, Honolulu, Hawaii, USA
45. Yuan T, Li X, Xiong H, Cao H, Dou D (2021) Explaining information flow inside vision transformers using Markov chain. Proc. of the International Workshop on eXplainable AI approaches for debugging and diagnosis (XAI4Debugging@ NeurIPS2021), Virtual event
46. Jo S, Jang G, Park H (2025) GMAR: Gradient-Driven Multi-Head Attention Rollout for Vision Transformer Interpretability. In Proc. of the IEEE International Conference on Image Processing (ICIP'25), pages 582–587, Anchorage, AK, USA, 2025. IEEE
47. Xie W, Li X, Cao CC, Zhang NL (2023) ViT-CX: Causal Explanation of Vision Transformers. In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI'23), pages 1569–1577, Macao, China
48. Deiseroth B, Deb M, Weinbach S, Brack M, Schramowski P, Kersting K (2023) Atman: Understanding transformer predictions through memory efficient attention manipulation. In Proc. of the Advances in Neural Information Processing Systems (NIPS'23), volume 36, pages 63437–63460, New Orleans, LA, USA
49. Nalmpantis A, Panagiotopoulos A, Gkountouras J, Papakostas K, Aziz W (2023) Vision diffmask: Faithful interpretation of vision transformers with differentiable patch masking. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'23), pages 3756–3763, Vancouver, British Columbia, Canada
50. Kim J, Kim S, Kim ST, Ro YM (2021) Robust perturbation for visual explanation: Cross-checking mask optimization to avoid class distortion. IEEE Transactions on Image Processing, 31:301–313, IEEE

51. Chen Y, Shen X, Liu Y, Tao Q, Suykens JAK (2023) Jigsaw-ViT: learning jigsaw puzzles in vision transformer. Pattern Recognit Lett 166:53–60
52. Niu Y, Ding M, Ge M, Karlsson R, Zhang Y, Takeda K (2023) R-cut: Enhancing explainability in vision transformers with relationship weighted out and cut. arXiv preprint arXiv:2307.09050
53. Li B, Han L (2013) Distance weighted cosine similarity measure for text classification. In Proc. of the International Conference of Intelligent Data Engineering and Automated Learning (IDEAL'13), pages 611–618, Hefei, China, Springer
54. Sattarzadeh S, Sudhakar M, Lem A, Mehryar S, Plataniotis KN, Jang J, Kim H, Jeong Y, Lee S, Bae K (2021) Explaining Convolutional Neural Networks through Attribution-Based Input Sampling and Block-Wise Feature Aggregation. In Proc. of the AAAI Conference on Artificial Intelligence (AAAI'21), pages 11639–11647, Virtual Event . AAAI Press
55. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégo H (2021) Training data-efficient image transformers & distillation through attention. In Proc. of the International Conference on Machine Learning (ICML'21), pages 10347–10357, Virtual event . PMLR
56. Chefer H, Gur S, Wolf L (2021) Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV'21), pages 397–406, Virtual event
57. Deng J, Dong W, Socher R, Li LJ, Kai L, Li FF 2009 ImageNet: A large-scale hierarchical image database. In Proc. of the International IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), pages 248–255, Miami, FL, USA, IEEE
58. Beyer L, Zhai X, Kolesnikov A (2022) Better plain ViT baselines for ImageNet-1k. arXiv preprint arXiv:2205.01580
59. 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 Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV'21), pages 10012–10022, Virtual event

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