Explaining through Transformer Input Sampling

Alexandre Englebert[1,2] Sédrick Stassin[3] Géraldin Nanfack[4] Sidi Ahmed Mahmoudi[3]
Xavier Siebert[3] Olivier Cornu[2] Christophe De Vleeschouwer[1]

Abstract

Vision Transformers are becoming more and more the preferred solution to many computer vision problems, which has motivated the development of dedicated explainability methods. Among them, perturbation-based methods offer an elegant way to build saliency maps by analyzing how perturbations of the input image affect the network prediction. However, those methods suffer from the drawback of introducing outlier image features that might mislead the explainability process, e.g. by affecting the output classes independently of the initial image content. To overcome this issue, this paper introduces Transformer Input Sampling (TIS), a perturbation-based explainability method for Vision Transformers, which computes a saliency map based on perturbations induced by a sampling of the input tokens. TIS utilizes the natural property of Transformers which permits a variable input number of tokens, thereby preventing the use of replacement values to generate perturbations. Using standard models such as ViT and DeiT for benchmarking, TIS demonstrates superior performance on several metrics including Insertion, Deletion, and Pointing Game compared to state-of-the-art explainability methods for Transformers. The code for TIS is publicly available at `https://github.com/aenglebert/Transformer_Input_Sampling`.

1. Introduction

Recent advances in deep learning create an increasing need for explanation techniques to evaluate the prediction quality of neural networks. This is especially required in areas where a black box model is not desired for ethical or security reasons such as, for example, health or for granting a loan. Lately, the rise of the transformer architecture [28] in multiple modalities provides a new challenge in terms of explainability. Especially in the field of computer vision, where the convolutional neural network (CNN) has been the dominant architecture type since AlexNet in 2012 [14], with many explainability methods targeting these CNN architectures [21, 30, 8]. The Vision Transformer arrived as an adaptation of the transformer architecture [28] from Natural Language Processing (NLP) to computer vision. The main contribution of the transformer architecture is the use of the attention mechanism between different tokens at each layer. The increasing adoption of Vision transformers [32] has motivated the design of explainability methods addressing these types of architectures. The available methods targetting transformers are based on a combination of attention weights with or without gradient-based modulation [4, 5, 29, 1, 35, 6]. Multiple works have shown that raw attention is not enough to provide a valid explanation since it takes into account the query and key elements of the self-attention, but not the value [17, 22, 10]. An alternative has been proposed by the ViT-CX method [32] that utilizes the model embeddings to produce multiple perturbed input images and probe the vision transformer model with them to compute a saliency map by weighting the perturbations as a function of how they impact the model output [16, 30]. This kind of process has the disadvantage of producing outlier images, i.e. images that contain structures that are not related to the initial image content, which might end up misleading the saliency map construction. We argue that a better alternative consists in employing the natural ability of transformers to utilize an arbitrary number of tokens as input. This property is used on some pretraining methods such as the Masked Autoencoders strategy in self-supervised learning [9], but not, to our knowledge, to produce an explanation.

In contrast to previous perturbation-based methods, the main contribution of our method is to define perturbations as a sampling of the tokens before the first transformer layer, but after the linear projection and position encoding of the patches. This definition avoids the generation of outlier inputs, thereby limiting the risk of misleading the interpretation of the transformer predictions. Another advantage is that the reduction of the number of tokens at the trans-

[1]ELEN / ICTEAM, Louvain-la-Neuve, UCLouvain, Belgium
[2]Service de chirurgie orthopédique et traumatologie / CUSL & NMSK / IREC, UCLouvain, Belgium
[3]ILIA unit, UMons, Mons, Belgium
[4]University of Concordia, Montreal, Canada

former input also increases the inference speed for each perturbation, enabling more samples to be evaluated with the same computing power. This also renders the method more versatile, for example with multimodal transformers. Although the multimodal aspect has not yet been tackled in this work, it represents a preliminary step towards applying a perturbation-based method to multimodal transformers.

The rest of this paper is organized as follows. Section 2 presents the state-of-the-art concerning explainability methods in computer vision, as well as those applied to transformer models. Section 3 introduces our proposed approach. Section 4 describes the experimental setup, while Section 5 discusses our experimental results. Finally, Section 6 discusses the results and concludes our paper.

2. State-of-the-Art

This Section reviews the state-of-the-art methods used to explain the outcome of black-box deep learning models, with a focus on the explainability of vision transformers.

2.1. Explainability in Computer Vision

2.1.1 Gradient-based Methods

Among the first applicable methods to explain the results of deep learning models are the gradient-based methods. They explain the prediction of a model by performing a backpropagation from an output neuron (e.g., a probability obtained for a class) to the input features [23]. This produces a so-called saliency map (or heatmap), providing a visualization of the most important areas for the decision of black-box models. Smilkov et al. introduced **SmoothGrad** [24] which augments the input samples by adding Gaussian noise and calculates the average of the results obtained for each backpropagation. **Integrated Gradient** [26] also computes a backpropagation average, but the result is obtained based on an interpolation between the input image and a baseline image (e.g., black, white image).

2.1.2 Perturbation-based Methods

Next to the gradient-based methods, there are also methods that perturb the input image and analyze how the model response is impacted by those changes to produce an explanation (e.g., **Occlusion** [36] using square patches). Those methods are known as perturbation-based methods. **RISE** [16] is a popular state-of-the-art method that produces small random binary masks, then scaled to the size of the image. The saliency map is computed as a linear combination of the perturbation masks and their relevance, measured based on their impact on the prediction.

2.1.3 CAM-based Methods

Class Activation Maps-based methods (CAM) use the activations of the convolutional layers of CNNs to obtain saliency maps. The most popular method is **Grad-CAM** [21], which weights the activation maps by the gradients obtained by a backpropagation from the output neuron of a class to the last convolutional layer. Variants aggregate the results for the input image at different scales (**CAMERAS** [11]), combine the activations from different layers (**Poly-CAM** [8], **Layer-CAM** [12]), or predict the relevance of masks created from the activations (**Score-CAM** [30]). Since Vision Transformers employ the CLS token for downstream tasks, this limits the application of CAM methods that require the use of the embeddings before a last pooling layer.

2.2. Explainability of Vision Transformers

The key difference between a CNN and a transformer lies in the calculation of attention scores for the latter. These attention scores help in representing the relationships that can appear between each of the input features. Consequently, the first attempts to explain the results of visual attention were based on saliency maps created through an upsampling of these attention scores [33]. However, the use of attention scores as explainability scores has limitations [1, 17, 22] (e.g., attention takes into account the query and key, but not the value of the self-attention) that have led to specific explanation methods designed for transformers.

2.2.1 Attention-based Methods

The first one came from Abnar [1] who presented the **Attention Rollout** method. This approach computes the saliency map based on a combination (e.g., average; minimum; maximum) of the attention heads with the addition of an identity matrix representative of the residual connections, arguing that the latter is crucial to compute the propagation of information through the layers. However, this approach does not take into account the fact that some attention heads may be more relevant than others.

2.2.2 Gradient-based Methods

Partial LRP [29] solved this issue by calculating the importance of each attention head using the Layer-wise Relevance Propagation (LRP) [2] method. **Chefer 1** [5] argued that the use of LRP by [29] provided only partial information on the attention head relevance as the LRP rule was not utilized back to the input features. The Chefer method computes class-specific explanations by incorporating relevance (LRP) and gradient information with specific rules designed to handle the skip connections. **Chefer 2** [4] provided a

generic solution that can be applied to any transformer-based architecture and to more than two modalities. The latter takes into account the residual connections through an identity matrix to compute attention scores (as proposed by [1]) and utilizes the gradients to obtain the relevance of each head related with respect to a desired class output. The **Transition Attention Maps (TAM)** [35] method takes inspiration from the Markov process. At each block, the representations of the output tokens are considered as states of the Markov chain, with the state transition matrix being constructed based on the attention weights. A class discriminative explanation is achieved by combining the states with the Integrated Gradients obtained with respect to the last attention module. **Bidirectional Transformers (BT)** [6][1] compute an element-wise product between two terms to obtain a saliency map. The first is Reasoning Feedback. It represents how the classification token (CLS) is used for a class prediction and is calculated with the Integrated Gradients of a chosen class back to the last attention map using a black baseline. The second is Attention Perception. It represents the learning process of the input tokens through the attention blocks. It approximates the relationship between the input and output of the attention blocks and derives two attention maps from it: BT-T (T for token) and BT-H (H for head).

2.2.3 Perturbation-based Methods

ViT-CX [32] adopts a different approach compared to the previous transformer explainability methods. It no longer relies directly on attention weights and gradients but on masks created from patch embeddings (such as Score-CAM [30] using feature maps as masks for CNNs) and the relevance of each mask, computed by evaluating the model with a masked image to obtain a saliency map. This method is similar to perturbation-related methods such as RISE [16] but provides a smaller number of more focused masks because first they are not randomly generated but use transformer embeddings, and second ViT-CX adds a clustering of the embeddings to further reduce the number of masks.

We differ from the previous works as follows: we propose an explainability method based on the masking (sampling) of the tokens given to a vision transformer. When masked after the embedding phase, these tokens are no longer considered as input to the self-attention, which avoids the problem related to the choice of a replacement value encountered in perturbation-based explainability methods [16, 32] or metrics [16, 34, 25]. This replacement value, depending on the images, can correspond to input features that are independent of the class of interest but might trigger (by accident) other classes, impacting the score of all classes (including the one of interest due to the softmax), and thus misleading the explainability metrics.

[1]The method is not named in the paper but is referred to as "Bidirectional Transformers" in InterpretDL (https://github.com/PaddlePaddle/InterpretDL).

3. Our Transformer Input Sampling Approach

Section 3.1 introduces useful notations. Section 3.2 gives a general overview of our proposed method. Section 3.3 details the generation of masks, and its corresponding token sampling process. Section 3.4 explains the mask scoring process, leveraging the variable input length property of transformers, and the saliency map computation as a score-based weighted sum of masks.

3.1. Notations

Let $f(X)$ denote a vision transformer model [7, 27] applied to an image X. This model is composed of an embedding computation module (patch and positional embedding) denoted $\mathrm{embedding}(X)$, whose result is a matrix $T \in \mathbb{R}^{N_t \times D}$ composed of N_t tokens of dimension D, and a transformer encoder [28] with a task-focused head denoted $\mathrm{transformer}(X)$, such as $f(X) = \mathrm{transformer}(\mathrm{embedding}(X))$. In the following, the result of $f(X)$ is a vector of dimension C, defined as the output of a softmax function, and $f_c(X)$ corresponds to the score given by the model to a particular class c for the image X. Let $A_{i\cdot}$ be the i-th row of a given matrix A, and $A_{\cdot j}$ the j-th column of a given matrix A. Consider $\oslash$ as the element-wise division operator, and $\odot$ as the element-wise product operator. Let $\mathrm{topk}(A, n)$ be the set of n largest elements in a given set A.

3.2. General Overview

The proposed method computes class-specific saliency maps. It relies on the output score associated with the class of interest when inputting different subsets of the input tokens in the transformer part of the model. A schematic illustration of the process is depicted in Figure 1. The tokens are sampled before the transformer encoder. This is similar in principle to the Masked Autoencoders [9], with masks being generated based on the activations of the transformer model. Previous works have shown that, even if the multi-head attention modules of a vision transformer are position invariant, the tokens keep the localization information from the beginning up to the end of the model thanks to the multiple residual connections [18]. This location-preserving property in the embedding space enables the use of the embedding to guide the masking process, similarly to what is done by Score-CAM for a convolutional neural network [30]. It is worth noting that unlike perturbations methods in the input space that modify the pixels values such as RISE [16], Score-CAM [30] or ViT-CX [32], our method leverages the ability of the transformer to accept a sequence of tokens with variable length to completely remove a portion of the

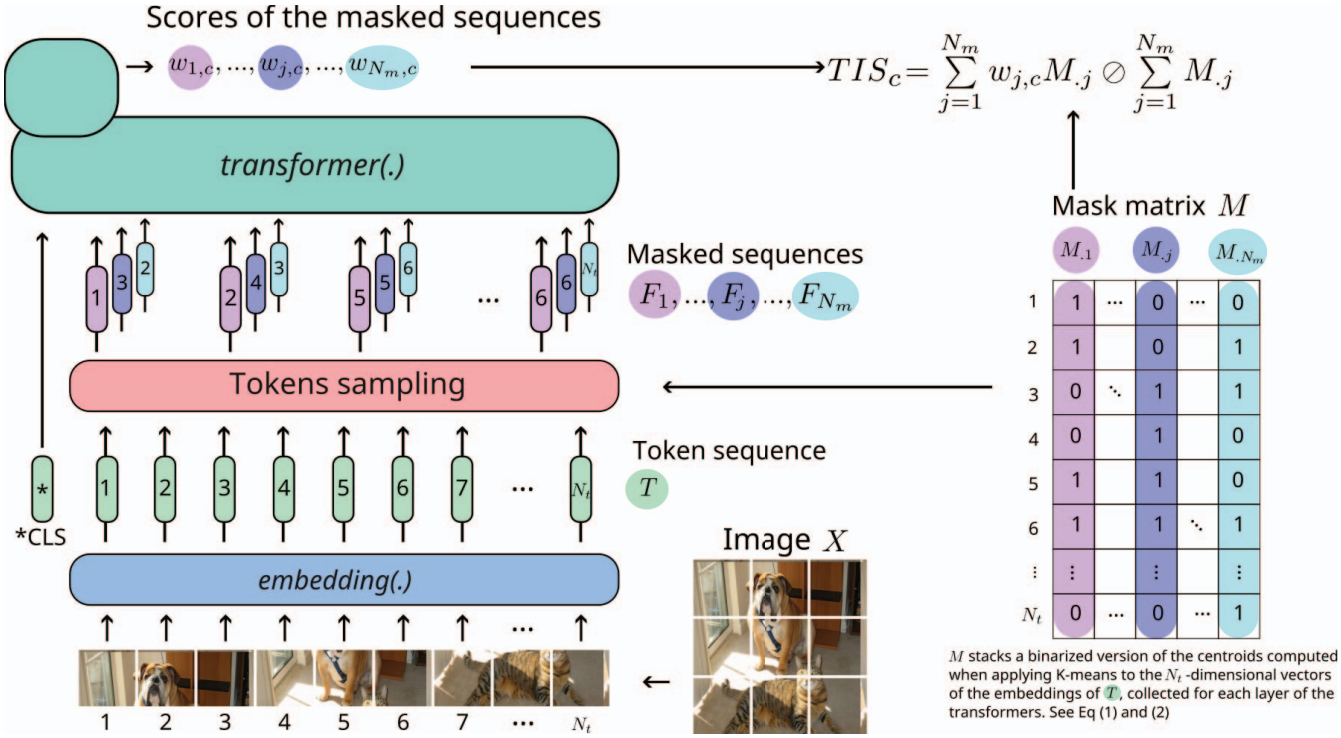


Figure 1: **Illustration of the Transformer Input Sampling (TIS) process.** The columns $M.j$ of the matrix M are the masks used to produce each sampled sequences F_j. The scores $w_{j,c}$ are the scores for each sequences F_j for a target class c.

tokens (i.e., the patches) in a way that the model can only perform computations on the remaining tokens. Since this is done just after the positional embedding and before any self-attention, the non-sampled tokens do not have any influence on the output. This is in contrast with the generation of outlier images that can be produced when corrupting the input.

3.3. Mask Generation and Token Sampling

The first step when generating a mask to control the sampling of a token sequence $T \in \mathbb{R}^{N_t \times D}$, composed of N_t tokens (excluding the CLS classification token) with dimension D, is to concatenate the activation/embeddings from every layer in the transformer into a matrix $A \in \mathbb{R}^{N_t \times L.D}$ with L being the number of layers of encoders in the transformer. Since the computational requirements increase with the forward passes computed for each mask and many maps are redundant, we use a clustering process to reduce the number of masks, similarly to ViT-CX [32]. A K-Means clustering is used on the columns of A to produce a smaller matrix $K \in \mathbb{R}^{N_t \times N_m}$ with N_m being the number of masks. The number of centroids of K-Means N_m is a parameter of our method. The choice of N_m is evaluated in the Supplementary material and set to 1024 in the remaining of the paper.

$$K = KMeans(A, N_m) \tag{1}$$

Unlike previous works based on masks generated from the activation maps with continuous values [32, 30, 8], we propose to binarize the masks so that each value in the matrix means whether we will keep the corresponding token or not when computing the class score. We thus produce a binary matrix $M \in \{0,1\}^{N_t \times N_m}$.

Formally,

$$M_{ij} = \begin{cases} 1 & \text{if } K_{ij} \in \text{topk}(K_{.j}, N_k) \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

with N_k being the number of tokens to sample.

We obtain N_m sequences of sampled tokens. The j^{th} sequence $F_j \in \mathbb{R}^{N_k \times D}$, is associated to the mask $M_{.j}$ in M, and is defined as follows,

$$F_j = \{T_i | M_{ij} = 1\} \tag{3}$$

3.4. Mask Scoring and Saliency Map

The class-specific relevance score $w_{j,c}$ of each mask M.j is obtained by passing its corresponding set of tokens F_j

in the transformer and retrieving the model output for the target class c. Formally,

$$w_{j,c} = transformer_c(F_j), \quad \text{for } 1 \leq j \leq N_m \tag{4}$$

Since each token is related to a patch in the input image, the more a particular token is relevant for a given model output, the more the corresponding patch is also relevant. Therefore, it becomes relevant to compute a saliency map as the sum of the masks weighted by the score obtained by the corresponding sampled tokens. This sum can be improved by dividing by the sum of the masks to account for possible token frequency bias, similar to the pixel coverage bias addressed in ViT-CX[32]. Hence,

$$TIS_c = \sum_{j=1}^{N_m} w_{j,c} M_{.j} \oslash \sum_{j=1}^{N_m} M_{.j} \tag{5}$$

In the following, the resulting saliency maps are bilinearly upsampled to the resolution of the input image.

4. Experimental Setup

This section describes the experimental setup used to benchmark the proposed method in comparison to previous works. For our proposed TIS method, we employ a token masking ratio of 0.5, translating to 98 tokens over 196 (formally, $n_k = 98$ in Equation 3.3) and 1024 masks ($N_m = 1024$ in Equation 3.3). This set of parameters is discussed in supplementary material. Good results are obtained with values ranging from 128 to 1024 masks, with little gain beyond 1024. The methods used for comparison are ViT-CX [32], Transition Attention Maps (TAM) [35], the two methods from Chefer [4, 5], Attention Rollout [1], the token (BT-T) and head (BT-H) methods from Bidirectional Transformers [6], RISE [16], Integrated Gradient [26] and SmoothGrad [24]. The parameters used are 20 steps for TAM, 4000 masks for RISE, 50 interpolations for Integrated Gradient, and 50 perturbations for SmoothGrad. We used the released codes from the authors for ViT-CX[2], RISE[3], Chefer[4], TAM[5] and Bidirectional Transformers[6]. We applied Captum [13] implementations for SmoothGrad and Integrated Gradient.

4.1. Transformer Models

The two models used in the experiments are ViT and DeiT, typically used to solve computer vision tasks such as image classification. The Vision Transformer (ViT) [7] is an encoder-only transformer architecture. In particular, each image is divided into N non-overlapping patches which are then projected into the embedding space as a sequence of tokens that serve as input to the transformer backbone. In addition, a learned classification token (CLS token) is prepended to this sequence. After the final encoder layer, the representation of the CLS token depicts a global embedding of the image and is classically used as input to a head trained for downstream tasks such as classification. DeiT [27] derives from ViT, but in addition to the CLS token it also has a distillation token that is combined with a second classification head dedicated to learning by distillation from the predictions of a teacher network. In the following experiments, the ViT model denotes the ViT-Base variant [7], and the DeiT denotes the DeiT-Base variant [27]. We utilized the implementations from the timm library [31] using ImageNet 21k pretraining with ImageNet 1k finetuning weights for both models.

4.2. Explainability Metrics

In the XAI domain, explainability metrics are used to evaluate the performance of explainability methods and to avoid the subjectivity of human judgment [20]. However, the evaluation is complex due to the lack of ground-truth explanations. Consequently, explainability metrics evaluate XAI methods according to different concepts or properties. In this paper, in an effort to make a broader and fairer comparison with respect to the different properties evaluated in the state-of-the-art of explainability metrics, we report the results with respect to the following metrics: Insertion and Deletion [16] (faithfulness metric), Pointing Game [37] (localization metric), Max-Sensitivity [34] (robustness metric) and Sparseness [3] (complexity metric). The choice of metrics was made according to two criteria: their current use to report the results of the state-of-the-art explainability methods (e.g., Insertion, Deletion, Pointing Game), as well as the diversity of the represented properties (e.g., Sparseness, Max-Sensitivity). For the Insertion and Deletion metrics, 224 steps were used in the iterative computation and each metric was computed using four baselines (blur, random, black, and mean). Regarding the Pointing Game metric, we excluded images where the bounding box covered more than 50% of the image, thereby following the recommendations in [30, 32]. This results in 2892 images excluded and 2108 images included for this metric. For Max sensitivity, we used Captum's implementation with a number of perturbed samples set to 10 and a perturbation radius set to 0.02. For Sparseness, in the case of negative values, we shifted the minimum value to zero before applying the metric[7]. Since this metric serves as an additional indicator (concise explanations) rather than a ranking, the corresponding results are presented in the Supplementary material.

[2]https://github.com/vaynexie/CausalX-ViT

[3]https://github.com/eclique/RISE

[4]https://github.com/hila-chefer/Transformer-Explainability

[5]https://github.com/XianrenYty/Transition_Attention_Maps

[6]https://github.com/jiaminchen-1031/transformerinterp

[7]https://github.com/oliviaguest/gini

4.3. Assessment Protocol

Given the evaluation metrics, the assessment adopts the protocol used in previous works [16, 4, 5, 32, 1] on explainable AI applied to convolutional neural networks and vision transformers. It consists in evaluating the saliency maps generated with the different methods on a random subset of the ImageNet validation set [19]. We set the size of this subset to 5000 images [32, 6].

5. Experimental Results

This section analyzes the results obtained by our method and compares them to previous works from a qualitative and quantitative point of view.

5.1. Qualitative Assessment

5.1.1 General Comparison

In the field of explainable AI, metrics primarily represent approximations of isolated properties, unable to fully quantify the relevance and quality of saliency maps. Consequently, visualizing the generated maps is also crucial. In Figure 2 we observe that maps generated by our TIS method are generally more expressive, often highlighting the whole object with a variable range of intensity, for example with the Maltese dog where the head of the dog is the most highlighted, followed by the dog's body with intermediate intensity, and then the background with low intensity. In general, other methods tend to be more categorical with a generally very localized high signal and most of the remaining of the map being low signal. ViT-CX and TAM are the only methods that seem to also display this behavior, while ViT-CX often highlight more background information and TAM is sparser. In contrast, the Integrated Gradient and the SmoothGrad methods produce maps with a lot of isolated peaky points, related to the importance of the gradient at the input. They are not always class specific and tend to be noisy and hard to interpret.

5.1.2 Class Disagreement

When generating the saliency maps for both the target class from the ImageNet Dataset and the model predicted class, we noticed that major disagreements between the ground truth and the model can lead to bad saliency maps for the target class, and good saliency maps for the model predicted class. An example is provided in Figure 3 where a bird with a target class of "Kite" is present, the model top prediction is "Bald Eagle" with a confidence level of 0.998, while the confidence of the target class is 0.0004. The saliency map for "Bald Eagle", the predicted class, clearly highlights the bird, while the saliency map for "Kite", the target class, highlights the background. We observed this behavior for multiple images, the stronger the disagreement between the model and the target, the stronger this phenomenon. Through our experiments, we discovered that highlighting the target class can be forced by removing the softmax at the end of the model. However, this comes at the price of class specificity. This behavior is thus strongly related to the class specificity of the method, leading us to interpret it as proof of our method's strong class specificity.

5.2. Quantitative Assessment

Faithfulness Results for Insertion and Deletion metrics are provided in Table 1 and Table 2 for ViT and DeiT, respectively. Our proposed method performed best on the Insertion for all baselines, except the blur baseline where it finished second behind Integrated Gradient by a thin margin. Interestingly, it's worth noting that Integrated Gradient had the worst performance among all methods for the other Insertion baselines. Concerning the Deletion metric, our method performed second, just behind Integrated Gradient. This is not surprising since the gradient on which Integrated Gradient is based corresponds to the pixels with the highest influences on the output. When balancing the two metrics by the subtraction of the Deletion metric from the Insertion metric, our method appears to surpass other methods by a wide margin for all baselines, except for the blur baseline where it finishes second.

Localisation The results for the Pointing Game metric can be found in Table 3. Our method performed best in comparison to the other methods for DeiT on this metric and fell just behind the BT methods for ViT. Furthermore, TIS is the only method that achieves a score over 0.8 on both models (0.825 and 0.823). Our proposed method is thus competitive in terms of the localization property.

Robustness In Table 4, we show the results related to the Max Sensitivity metric. Two groups emerge from these results. The first group contains RISE, TAM, BT-H, Chefer2 Rollout, TIS and ViT-CX (ranked by lowest sensitivity score respectively) and has good robustness when small perturbations are inputted to an image (Max Sensitivity $\leq$ 0.2). On the contrary, Integrated Gradients and Chefer1 in the second group are at the other end of the range (Max Sensitivity ≥ 0.8), being very sensitive to perturbations. TIS has appropriate scores with respect to the metric (not being too sensitive) but is not the best method in terms of robustness.

5.2.1 Deletion for TiS and Integrated Gradients

Based on the results indicating that Integrated Gradients may outperform TIS in terms of the Deletion metric, we explored the results obtained by both methods when applying

[7]The best result is in bold, and the second best result is underlined

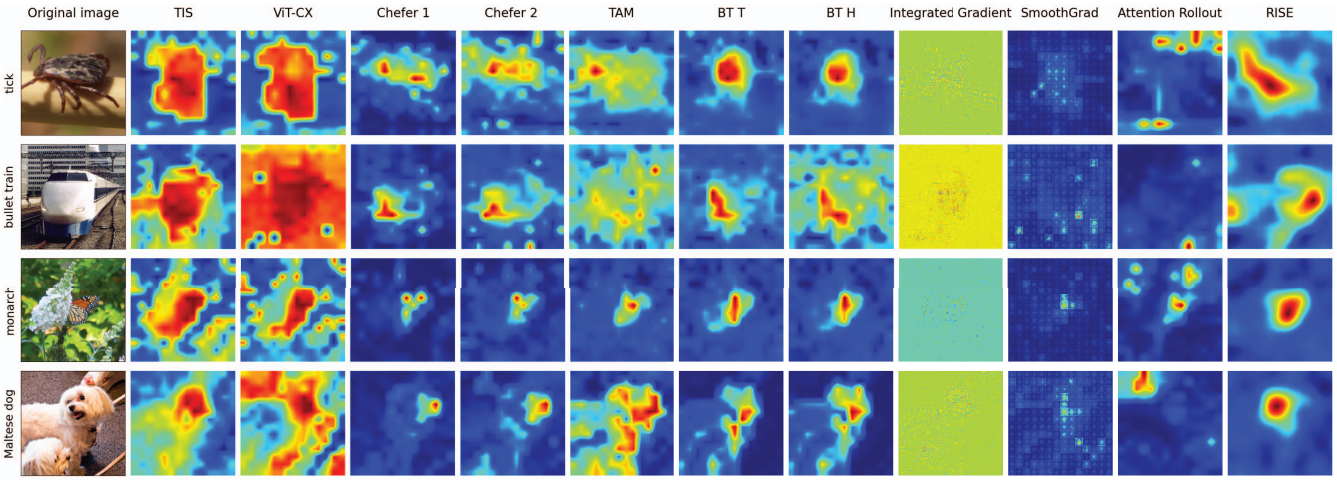


Figure 2: Comparison of the explainability methods for the ViT-Base model [7] on four random images from the ImageNet Validation set [19].

	Insertion ↑				Deletion ↓				Insertion - Deletion ↑			
Method	Mean	Blur	Black	Rand	Mean	Blur	Black	Rand	Mean	Blur	Black	Rand
TIS	**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	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	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	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	0.43	0.61	0.41	0.39	0.15	0.44	0.14	0.13	0.28	0.17	0.27	0.26
Att. Rollout	0.31	0.55	0.30	0.29	0.29	0.52	0.28	0.27	0.02	0.03	0.02	0.02
BT H	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	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	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	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	0.37	0.59	0.36	0.35	0.10	0.45	0.10	0.09	0.27	0.14	0.26	0.26

Table 1: Results of the Insertion and Deletion metrics and their difference (Insertion - Deletion) for ViT-Base [7]. [7]

the deletion metric to an image (Figure 4). Integrated Gradients exhibit a faster drop in the metric and achieve a better overall result. However, upon examining the perturbed image at intermediate steps, it became apparent that Integrated Gradient significantly affects the model by removing target pixels everywhere in the image, while the overall shape of the bird remains distinguishable to a human observer. In contrast, TIS effectively masks the object.

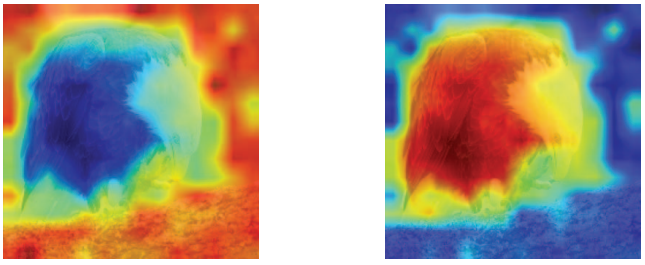
6. Discussion and Conclusion

In this paper, we introduced a method to explain vision transformers using token sampling guided by the model embeddings. This is an alternative to methods based on attention and gradients to explain transformers. The main contribution of our method in comparison to other perturbation methods, such as RISE or VIT-CX, is to provide a more versatile and complete ablation of masked input information instead of masking in input space. Even if the absence of a real ground truth metric in the explainability field makes the evaluation difficult, we showed the competitiveness of our method amongst all metrics with current explainability method. A common downside of perturbation-based methods is the requirement for more computing power, as multiple forward passes must be performed.This limits the application in use cases such as low-power or embedded devices. TIS shows good performances with as few as 128 samples and half of the tokens, significantly reducing the inference time. Although this work has only explored vision transformers, our method also has the advantage of being potentially applicable to any type of transformer using conventional encoding and/or decoding layers. Although, on the other hand, it is not directly applicable to modified transformers with hierarchical mechanisms such as a Swim transformer [15]. Since TIS is not limited by design to vision transformers, future works should explore the adaptation of the token sampling to transformers working with

Method	Insertion ↑				Deletion ↓				Insertion - Deletion ↑			
	Mean	Blur	Black	Rand	Mean	Blur	Black	Rand	Mean	Blur	Black	Rand
TIS	**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	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	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	0.51	0.60	0.51	0.48	0.22	0.45 0.51	0.22	0.18	0.29	0.15	0.29	0.29
Chefer2	0.50	0.60	0.50	0.47	0.23	0.45	0.23	0.19	0.28	0.14	0.27	0.28
Att. Rollout	0.37	0.54	0.37	0.34	0.41	0.53	0.41	0.37	-0.04	0.01	-0.05	-0.03
BT H	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	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	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	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	0.45	0.62	0.43	0.43	**0.14**	0.44	0.14	**0.13**	0.31	0.18	0.30	0.31

Table 2: Results of the Insertion and Deletion metrics and their difference (Insertion - Deletion) for DeiT-Base [27]. [7]

(a) Original image (b) Kite (c) Bald Eagle



(d) TIS for "Kite" (e) TIS for "Bald Eagle"

Figure 3: Class mismatch between the target and predicted class. 3a is the original image. The dataset target class is "Kite" while the model predicts "Bald Eagle". For illustration purposes, 3b and 3c display other images of a Kite and a Bald Eagle, respectively. 3d is the saliency map produced by TIS for class "Kite" (dataset target) and 3e is the TIS saliency map for the model predicted class "Bald Eagle".

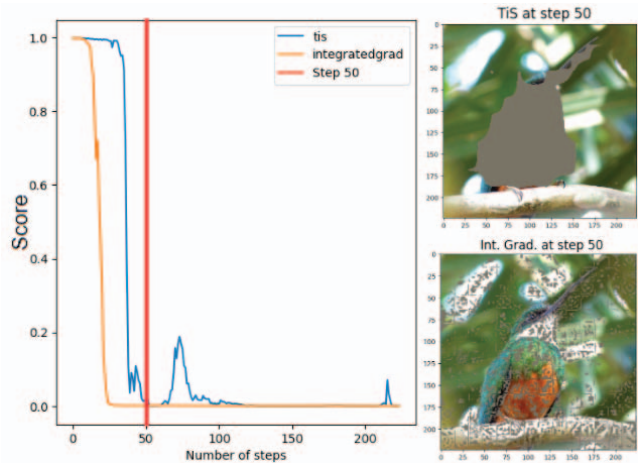


Figure 4: Comparison of TIS and Integrated Gradients results after 50 steps of the deletion metric. The target class is a jacamar (bird). Integrated Gradient perturbates the image diffusely, resulting in a better metric while still keeping the bird visible. On the other hand, TIS masks the bird itself, even though it may take more steps to reduce the target score.

other modalities and/or multi-modal transformers.

7. Acknowledgments

The Research Foundation for Industry and Agriculture, National Scientific Research Foundation (FRIA-FNRS) funded this research as grants attributed to Alexandre Englebert, consisting in Ph.D. financing.
Sédrick Stassin thanks the support of the E-origin project funded by the Walloon Region within the pole of logistics in Wallonia. Christophe De Vleeschouwer is funded by the FNRS (National Scientific Research Foundation).

Computational resources have been provided by the supercomputing facilities of the Université Catholique de Louvain (CISM/UCL) and the Consortium des Equipements de Calcul Intensif en Fédération Wallonie Bruxelles (CECI) funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region.

Method	DeiT	ViT
TIS	**0.825**	0.823
ViT-CX	0.700	0.700
TAM	0.635	0.737
Chefer1	0.748	0.768
Chefer2	0.654	0.727
Attention Rollout	0.118	0.127
BT H	0.775	**0.855**
BT T	0.755	0.846
RISE	0.766	0.753
Integrated Gradient	0.297	0.633
SmoothGrad	0.742	0.499

Table 3: Results of the Pointing Game metric [37] for the ViT [7] and DeiT model [27] [7]

Method	DeiT	ViT
TIS	0.162	0.156
ViT-CX	0.173	0.172
TAM	0.085	0.060
Chefer1	1.017	0.752
Chefer2	0.087	0.082
Attention Rollout	0.143	0.144
BT H	0.088	0.620
BT T	0.086	0.062
RISE	**0.011**	**0.009**
Integrated Gradient	0.827	0.891
SmoothGrad	0.218	0.412

Table 4: Results of the Max Sensitivity metric [34] for the ViT [7] and DeiT model [27]. [7]

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. arXiv preprint arXiv:2005.00928, 2020.

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one, 10(7):1–46, 2015.

[3] Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. Concise explanations of neural networks using adversarial training. In ICML, 2020.

[4] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 397–406, 2021.

[5] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 782–791, 2021.

[6] Jiamin Chen, Xuhong Li, Lei Yu, Dejing Dou, and Haoyi Xiong. Beyond intuition: Rethinking token attributions inside transformers. Transactions on Machine Learning Research, 2022.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[8] Alexandre Englebert, Olivier Cornu, and Christophe de Vleeschouwer. Backward recursive class activation map refinement for high resolution saliency map. In 2022 26th International Conference on Pattern Recognition (ICPR). IEEE, 2021.

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16000–16009, 2022.

[10] Sarthak Jain and Byron C Wallace. Attention is not explanation. arXiv preprint arXiv:1902.10186, 2019.

[11] Mohammad AAK Jalwana, Naveed Akhtar, Mohammed Bennamoun, and Ajmal Mian. Cameras: Enhanced resolution and sanity preserving class activation mapping for image saliency. In CVPR, 2021.

[12] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. Layercam: Exploring hierarchical class activation maps for localization. IEEE Trans. Image Process., 30:5875–5888, 2021.

[13] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. arXiv preprint arXiv:2009.07896, 2020.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012.

[15] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using

shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10012–10022, 2021.

[16] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In BMC, 2018.

[17] Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C Lipton. Learning to deceive with attention-based explanations. arXiv preprint arXiv:1909.07913, 2019.

[18] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? Advances in Neural Information Processing Systems, 34:12116–12128, 2021.

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. IJCV, 115(3):211–252, 2015.

[20] Wojciech Samek, Alexander Binder, Gregoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. IEEE Trans. Neural Netw. Learn. Syst., 28:2660–2673, 2017.

[21] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In ICCV, 2017.

[22] Sofia Serrano and Noah A Smith. Is attention interpretable? arXiv preprint arXiv:1906.03731, 2019.

[23] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. ArXiv:1312.6034 [Cs], 2014.

[24] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825, 2017.

[25] Sédrick Stassin, Alexandre Englebert, Géraldin Nanfack, Julien Albert, Nassim Versbraegen, Gilles Peiffer, Miriam Doh, Nicolas Riche, Benoît Frenay, and Christophe De Vleeschouwer. An experimental investigation into the evaluation of explainability methods. arXiv preprint arXiv:2305.16361, 2023.

[26] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In ICML, 2017.

[27] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In International Conference on Machine Learning, pages 10347–10357. PMLR, 2021.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[29] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. arXiv preprint arXiv:1905.09418, 2019.

[30] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, workshop on Fair, Data Efficient and Trusted Computer Vision, 2020.

[31] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[32] Weiyan Xie, Xiao-Hui Li, Caleb Chen Cao, and Nevin L Zhang. Vit-cx: Causal explanation of vision transformers. arXiv preprint arXiv:2211.03064, 2022.

[33] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning, pages 2048–2057. PMLR, 2015.

[34] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. Advances in Neural Information Processing Systems, 32, 2019.

[35] Tingyi Yuan, Xuhong Li, Haoyi Xiong, Hui Cao, and Dejing Dou. Explaining information flow inside vision transformers using markov chain. In eXplainable AI approaches for debugging and diagnosis., 2021.

[36] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In European conference on computer vision, pages 818–833. Springer, 2014.

[37] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. International Journal of Computer Vision, 126(10):1084–1102, 2018.