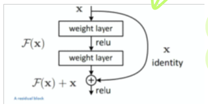
Transformer Interpretability Beyond Attention Visualization

Hila Chefer[1] Shir Gur[1] Lior Wolf[1,2]
[1]The School of Computer Science, Tel Aviv University
[2]Facebook AI Research (FAIR)

Abstract

Self-attention techniques, and specifically Transformers, are dominating the field of text processing and are becoming increasingly popular in computer vision classification tasks. In order to visualize the parts of the image that led to a certain classification, existing methods either rely on the obtained attention maps or employ heuristic propagation along the attention graph. In this work, we propose a novel way to compute relevancy for Transformer networks. The method assigns local relevance based on the Deep Taylor Decomposition principle and then propagates these relevancy scores through the layers. This propagation involves attention layers and skip connections, which challenge existing methods. Our solution is based on a specific formulation that is shown to maintain the total relevancy across layers. We benchmark our method on very recent visual Transformer networks, as well as on a text classification problem, and demonstrate a clear advantage over the existing explainability methods. Our code is available at: `https://github.com/hila-chefer/Transformer-Explainability`.


x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu
A residual block

1. Introduction

Transformers and derived methods [41, 9, 22, 30] are currently the state-of-the-art methods in almost all NLP benchmarks. The power of these methods has led to their adoption in the field of language and vision [23, 40, 38]. More recently, Transformers have become a leading tool in traditional computer vision tasks, such as object detection [4] and image recognition [6, 11]. The importance of Transformer networks necessitates tools for the visualization of their decision process. Such a visualization can aid in debugging the models, help verify that the models are fair and unbiased, and enable downstream tasks.

The main building block of Transformer networks are self-attention layers [29, 7], which assign a pairwise attention value between every two tokens. In NLP, a token is typically a word or a word part. In vision, each token can be associated with a patch [11, 4]. A common practice when trying to visualize Transformer models is, therefore, to consider these attentions as a relevancy score [41, 43, 4]. This is usually done for a single attention layer. Another option is to combine multiple layers. Simply averaging the attentions obtained for each token, would lead to blurring of the signal and would not consider the different roles of the layers: deeper layers are more semantic, but each token accumulates additional context each time self-attention is applied. The rollout method [1] is an alternative, which reassigns all attention scores by considering the pairwise attentions and assuming that attentions are combined linearly into subsequent contexts. The method seems to improve results over the utilization of a single attention layer. However, as we show, by relying on simplistic assumptions, irrelevant tokens often become highlighted.

In this work, we follow the line of work that assigns relevancy and propagates it, such that the sum of relevancy is maintained throughout the layers [27]. While the application of such methods to Transformers has been attempted [42], this was done in a partial way that does not propagate attention throughout all layers.

Transformer networks heavily rely on skip connection and attention operators, both involving the mixing of two activation maps, and each leading to unique challenges. Moreover, Transformers apply non-linearities other than ReLU, which result in both positive and negative features. Because of the non-positive values, skip connections lead, if not carefully handled, to numerical instabilities. Methods such as LRP [3] for example, tend to fail in such cases. Self-attention layers form a challenge since a naive propagation through these would not maintain the total amount of relevancy.

We handle these challenges by first introducing a relevancy propagation rule that is applicable to both positive and negative attributions. Second, we present a normalization term for non-parametric layers, such as "add" (*e.g.* skip-connection) and matrix multiplication. Third, we integrate the attention and the relevancy scores, and combine the integrated results for multiple attention blocks.

Many of the interpretability methods used in computer vision are not class-specific in practice, *i.e.*, return the same visualization regardless of the class one tries to visualize, even for images that contain multiple objects. The class-specific signal, especially for methods that propagate all the way to the input, is often blurred by the salient regions of the image. Some methods avoid this by not propagating to the lower layers [32], while other methods contrast different classes to emphasize the differences [15]. Our method provides the class-based separation by design and it is the only Transformer visualization method, as far as we can ascertain, that presents this property.

Explainability, interpretability, and relevance are not uniformly defined in the literature [26]. For example, it is not clear if one would expect the resulting image to contain all of the pixels of the identified object, which would lead to better downstream tasks [21] and for favorable human impressions, or to identify the sparse image locations that cause the predicted label to dominate. While some methods offer a clear theoretical framework [24], these rely on specific assumptions and often do not lead to better performance on real data. Our approach is a mechanistic one and avoids controversial issues. Our goal is to improve the performance on the acceptable benchmarks of the field. This goal is achieved on a diverse and complementary set of computer vision benchmarks, representing multiple approaches to explainability.

These benchmarks include image segmentation on a subset of the ImageNet dataset, as well as positive and negative perturbations on the ImageNet validation set. In NLP, we consider a public NLP explainability benchmark [10]. In this benchmark, the task is to identify the excerpt that was marked by humans as leading to a decision.

2. Related Work

Explainability in computer vision Many methods were suggested for generating a heatmap that indicates local relevancy, given an input image and a CNN. Most of these methods belong to one of two classes: gradient methods and attribution methods.

Gradient based methods are based on the gradients with respect to the input of each layer, as computed through backpropagation. The gradient is often multiplied by the input activations, which was first done in the Gradient*Input method [34]. Integrated Gradients [39] also compute the multiplication of the inputs with their derivatives. However, this computation is done on the average gradient and a linear interpolation of the input. SmoothGrad [36], visualizes the mean gradients of the input, and performs smoothing by adding to the input image a random Gaussian noise at each iteration. The FullGrad method [37] offers a more complete modeling of the gradient by also considering the gradient with respect to the bias term, and not just with respect to the input. We observe that these methods are all class-agnostic: at least in practice, similar outputs are obtained, regardless of the class used to compute the gradient that is being propagated.

The GradCAM method [32] is a class-specific approach, which combines both the input features and the gradients of a network's layer. Being class-specific, and providing consistent results, this method is used by downstream applications, such as weakly-supervised semantic segmentation [21]. However, the method's computation is based only on the gradients of the deepest layers. The result, obtained by upsampling these low-spatial resolution layers, is coarse.

A second class of methods, the *Attribution propagation* methods, are justified theoretically by the Deep Taylor Decomposition (DTD) framework [27]. Such methods decompose, in a recursive manner, the decision made by the network, into the contributions of the previous layers, all the way to the elements of the network's input. The Layer-wise Relevance Propagation (LRP) method [2], propagates relevance from the predicated class, backward, to the input image based on the DTD principle. This assumes that the rectified linear unit (ReLU) non-linearity is used. Since Transformers typically rely on other types of applications, our method has to apply DTD differently. Other variants of attribution methods include RAP [28], AGF [17], DeepLIFT [33], and DeepSHAP [24]. A disadvantage of some of these methods is the class-agnostic behavior observed in practice [20]. Class-specific behavior is obtained by Contrastive-LRP (CLRP) [15] and Softmax-Gradient-LRP (SGLRP) [20]. In both cases, the LRP propagation results of the class to be visualized are contrasted with the results of all other classes, to emphasize the differences and produce a class-dependent heatmap. Our method is class-specific by construction and not by adding additional contrasting stages.

Methods that do not fall into these two main categories include saliency based methods [8, 35, 25, 48, 45, 47], Activation Maximization [12] and Excitation Backprop [46]. Perturbation methods [13, 14] consider the change to the decision of the network, as small changes are applied to the input. Such methods are intuitive and applicable to black-box models (no need to inspect either the activations or the gradients). However, the process of generating the heatmap is computationally expensive. In the context of Transformers, it is not clear how to apply these correctly to discrete tokens, such as in text. Shapley-value methods [24] have a solid theoretical justification. However, such methods suffer from a large computational complexity and their accuracy is often not as high as other methods. Several variants have been proposed, which improve both aspects [5].

Explainability for Transformers There are not many contributions that explore the field of visualization for Transformers and, as mentioned, many contributions employ the attention scores themselves. This practice ignores

$$A(K,Q) = \text{softmax}\left(QK^{T} / \sqrt{d_{K}}\right)$$

most of the attention components, as well as the parts of the networks that perform other types of computation. A self-attention head involves the computation of queries, keys, and values. Reducing it only to the obtained attention scores (inner products of queries and keys) is myopic. Other layers are not even considered. Our method, in contrast, propagates through all layers from the decision back to the input.

LRP was applied for Transformers based on the premise that considering mean attention heads is not optimal due to different relevance of the attention heads in each layer [42]. However, this was done in a limiting way, in which no relevance scores were propagated back to the input, thus providing partial information on the relevance of each head. We note that the relevancy scores were not directly evaluated, only used for visualization of the relative importance and for pruning less relevant attention heads.

The main challenge in assigning attributions based on attentions is that attentions are combining non-linearly from one layer to the next. ✓ The rollout method [1] assumes that attentions are combined linearly ✗ and considers paths along the pairwise attention graph. 1) We observe that this method often leads to an emphasis on irrelevant tokens since even average attention scores can be attenuated. 2) The method also fails to distinguish between positive and negative contributions to the decision. Without such a distinction, one can mix between the two and obtain high relevancy scores, when the contributions should have cancelled out. Despite these shortcomings, the method was already applied by others [11] to obtain integrated attention maps.

Abnar et al. [1] present, in addition to rollout, a second method called attention flow. The latter considers the max-flow problem along the pair-wise attention graph. It is shown to be sometimes more correlated than the rollout method with relevance scores that are obtained by applying masking, or with gradients with respect to the input. This method is much slower and we did not evaluate it in our experiments for computational reasons.

We note this concurrent work [1] did not perform an evaluation on benchmarks (for either rollout or attention-flow) in which relevancy is assigned in a way that is independent of the BERT [9] network, for which the methods were employed. There was also no comparison to relevancy assignment methods, other than the raw attention scores.

3. Method

The method 1) employs LRP-based relevance to compute scores for each attention head in each layer of a Transformer model [41]. It then 2) integrates these scores throughout the attention graph, by incorporating both relevancy and gradient information, in a way that iteratively removes the negative contributions. The result is a class-specific visualization for self-attention models.

3.1. Relevance and gradients

Let C be the number of classes in the classification head, and $t \in 1 \dots |C|$ the class to be visualized. We propagate relevance and gradients with respect to class t, which is not necessarily the predicted class. Following literature convention, we denote $x^{(n)}$ as the input of layer $L^{(n)}$, where $n \in [1 \dots N]$ is the layer index in a network that consists of N layers, $x^{(N)}$ is the input to the network, and $x^{(1)}$ is the output of the network.

Recalling the chain-rule, we propagate gradients with respect to the classifier's output y, at class t, namely y_t:

backprop. alg. in comp. graph

$$\nabla x_j^{(n)} := \frac{\partial y_t}{\partial x_j^{(n)}} = \sum_i \underbrace{\frac{\partial y_t}{\partial x_i^{(n-1)}}}_{\text{incoming grad.}} \underbrace{\frac{\partial x_i^{(n-1)}}{\partial x_j^{(n)}}}_{\text{local grad.}} \tag{1}$$

where the index j corresponds to elements in $x^{(n)}$, and i corresponds to elements in $x^{(n-1)}$.

We denote by $L^{(n)}(\mathbf{X}, \mathbf{Y})$ the layer's operation on two tensors $\mathbf{X}$ and $\mathbf{Y}$. Typically, the two tensors are the input feature map and weights for layer n. Relevance propagation follows the generic Deep Taylor Decomposition [27]:

$$\begin{aligned} R_j^{(n)} &= \mathcal{G}(\mathbf{X}, \mathbf{Y}, R^{(n-1)}) \\ &= \sum_i \mathbf{X}_j \frac{\partial L_i^{(n)}(\mathbf{X}, \mathbf{Y})}{\partial \mathbf{X}_j} \frac{R_i^{(n-1)}}{L_i^{(n)}(\mathbf{X}, \mathbf{Y})}, \end{aligned} \tag{2}$$

where, similarly to Eq. 1, the index j corresponds to elements in $R^{(n)}$, and i corresponds to elements in $R^{(n-1)}$. Eq. 2 satisfies the conservation rule [27], *i.e.*:

$$\sum_j R_j^{(n)} = \sum_i R_i^{(n-1)} \tag{3}$$

LRP [2] assumes ReLU non-linearity activations, resulting in non-negative feature maps, where the relevance propagation rule can be defined as follows:

$$R_j^{(n)} = \mathcal{G}(x^+, w^+, R^{(n-1)}) = \sum_i \frac{x_j^+ w_{ji}^+}{\sum_{j'} x_{j'}^+ w_{j'i}^+} R_i^{(n-1)} \tag{4}$$

$$\text{ReLU}(x) = \max\{x, 0\},\ x \in \mathbb{R}$$

where $\mathbf{X} = x$ and $\mathbf{Y} = w$ are the layer's input and weights. The superscript denotes the operation $\max(0, v)$ as v^+.

Non-linearities other that ReLU, such as GELU [18], output both positive and negative values. To address this, LRP propagation in Eq. 4 can be modified by constructing a subset of indices $q = \{(i,j) | x_j w_{ji} \geq 0\}$, resulting in the following relevance propagation:

$$\begin{aligned} R_j^{(n)} &= \mathcal{G}_q(x, w, q, R^{(n-1)}) \\ &= \sum_{\{i|(i,j)\in q\}} \frac{x_j w_{ji}}{\sum_{\{j'|(j',i)\in q\}} x_{j'} w_{j'i}} R_i^{(n-1)} \end{aligned} \tag{5}$$

same expression but sums become conditional

In other words, we consider only the elements that have a positive weighed relevance.

To initialize the relevance propagation, we set $R^{(0)} = \mathbb{1}_t$, where $\mathbb{1}_t$ is a one-hot indicating the target class t.

3.2. Non parametric relevance propagation:

There are two operators in Transformer models that involve mixing of two feature map tensors (as opposed to a feature map with a learned tensor): skip connections and matrix multiplications (*e.g.* in attention modules). The two operators require the propagation of relevance through both input tensors. Note that the two tensors may be of different shapes in the case of matrix multiplication.

Given two tensors u and v, we compute the relevance propagation of these binary operators (*i.e.*, operators that process two operands), as follows:

$$R_j^{u^{(n)}} = \mathcal{G}(u, v, R^{(n-1)}), \quad R_k^{v^{(n)}} = \mathcal{G}(v, u, R^{(n-1)}) \quad (6)$$

where $R_j^{u^{(n)}}$ and $R_k^{v^{(n)}}$ are the relevances for u and v respectively. These operations yield both positive and negative values.

The following lemma shows that for the case of addition, the conservation rule is preserved, *i.e.*,

$$\sum_j R_j^{u^{(n)}} + \sum_k R_k^{v^{(n)}} = \sum_i R_i^{(n-1)}. \quad (7)$$

However, this is not the case for matrix multiplication.

Lemma 1. *Given two tensors u and v, consider the relevances that are computed according to Eq. 6. Then, (i) if layer $L^{(n)}$ adds the two tensors,* i.e., *$L^{(n)}(u, v) = u + v$ then the conservation rule of Eq. 7 is maintained. (ii) if the layer performs matrix multiplication $L^{(n)}(u, v) = uv$, then Eq. 7 does not hold in general.*

Proof. (i) and (ii) are obtained from the output derivative of $L^{(n)}$ with respect to $\mathbf{X}$. In an add layer, u and v are independent of each other, while in matrix multiplication they are connected. A detailed proof of Lemma 1 is available in the supplementary. □

When propagating relevance of skip connections, we encounter numerical instabilities. This arises despite the fact that, by the conservation rule of the addition operator, the sum of relevance scores is constant. The underlying reason is that the relevance scores tend to obtain large absolute values, due to the way they are computed (Eq. 2). To see this, consider the following example:

$$u = \begin{pmatrix} e^a \\ e^b \end{pmatrix} \quad , v = \begin{pmatrix} 1 - e^a \\ 1 - e^b \end{pmatrix}, \quad R = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (8)$$

$$R^u = \begin{pmatrix} \frac{e^a}{e^a - e^a + 1} 1 \\ \frac{e^b}{e^b - e^b + 1} 1 \end{pmatrix} = \begin{pmatrix} e^a \\ e^b \end{pmatrix}, \quad R^v = \begin{pmatrix} 1 - e^a \\ 1 - e^b \end{pmatrix} \quad (9)$$

where a and b are large positive numbers. It is easy to verify that $\sum R^u + \sum R^v = e^a + 1 - e^a + e^b + 1 - e^b = \sum R$. As can be seen, while the conservation rule is preserved, the relevance scores of u and v may explode. See supplementary for a step by step computation.

To address the lack of conservation in the attention mechanism due to matrix multiplication, and the numerical issues of the skip connections, our method applies a normalization to $R_j^{u^{(n)}}$ and $R_k^{v^{(n)}}$:

$$\bar{R}_j^{u^{(n)}} = R_j^{u^{(n)}} \frac{\left|\sum_j R_j^{u^{(n)}}\right|}{\left|\sum_j R_j^{u^{(n)}}\right| + \left|\sum_k R_k^{v^{(n)}}\right|} \cdot \frac{\sum_i R_i^{(n-1)}}{\sum_j R_j^{u^{(n)}}}$$

$$\bar{R}_k^{v^{(n)}} = R_k^{v^{(n)}} \frac{\left|\sum_k R_k^{v^{(n)}}\right|}{\left|\sum_j R_j^{u^{(n)}}\right| + \left|\sum_k R_k^{v^{(n)}}\right|} \cdot \frac{\sum_i R_i^{(n-1)}}{\sum_k R_k^{v^{(n)}}}$$

Following the conservation rule (Eq. 3), and the initial relevance, we obtain $\sum_i R_i^{(n)} = 1$ for each layer n.

The following lemma presents the properties of the normalized relevancy scores.

Lemma 2. *The normalization technique upholds the following properties: (i) it maintains the conservation rule, i.e.: $\sum_j \bar{R}_j^{u^{(n)}} + \sum_k \bar{R}_k^{v^{(n)}} = \sum_i R_i^{(n-1)}$, (ii) it bounds the relevance sum of each tensor such that:*

$$0 \leq \sum_j \bar{R}_j^{u^{(n)}}, \sum_k \bar{R}_k^{v^{(n)}} \leq \sum_i R_i^{(n-1)} \quad (10)$$

Proof. See supplementary. □

3.3. Relevance and gradient diffusion

Let M be a Transformer model consisting of B blocks, where each block b is composed of self-attention, skip connections, and additional linear and normalization layers in a certain assembly. The model takes as an input a sequence of s tokens, each of dimension d, with a special token for classification, commonly identified as the token `[CLS]`. M outputs a classification probability vector y of length C, computed using the classification token. The self-attention module operates on a small sub-space d_h of the embedding dimension d, where h is the number of "heads", such that $hd_h = d$. The self-attention module is defined as follows:

$$\mathbf{A}^{(b)} = softmax\left(\frac{\mathbf{Q}^{(b)} \cdot \mathbf{K}^{(b)T}}{\sqrt{d_h}}\right) \quad (11)$$

$$\mathbf{O}^{(b)} = \mathbf{A}^{(b)} \cdot \mathbf{V}^{(b)} \quad (12)$$

where $(\cdot)$ denotes matrix multiplication, $\mathbf{O}^{(b)} \in \mathbb{R}^{h \times s \times d_h}$ is the output of the attention module in block b, $\mathbf{Q}^{(b)}, \mathbf{K}^{(b)}, \mathbf{V}^{(b)} \in \mathbb{R}^{h \times s \times d_h}$ are the query key and value inputs in block b, namely, different projections of an input $x^{(n)}$ for a self-attention module. $\mathbf{A}^{(b)} \in \mathbb{R}^{h \times s \times s}$ is the attention map of block b, where row i represents the attention coefficients of each token in the input with respect to the token i. The $softmax$ in Eq. 11 is applied, such that the sum of each row in each attention head of $\mathbf{A}^{(b)}$ is one.

Following the propagation procedure of relevance and gradients, each attention map $\mathbf{A}^{(b)}$ has its gradients $\nabla\mathbf{A}^{(b)}$, and relevance $R^{(n_b)}$, with respect to a target class t, where n_b is the layer that corresponds to the $softmax$ operation in Eq. 11 of block b, and $R^{(n_b)}$ is the layer's relevance.

The final output $\mathbf{C} \in \mathbb{R}^{s \times s}$ of our method is then defined by the weighted attention relevance:

$$\bar{\mathbf{A}}^{(b)} = I + \mathbb{E}_h(\nabla\mathbf{A}^{(b)} \odot R^{(n_b)})^+ \tag{13}$$

$$\mathbf{C} = \bar{\mathbf{A}}^{(1)} \cdot \bar{\mathbf{A}}^{(2)} \cdot \ldots \cdot \bar{\mathbf{A}}^{(B)} \tag{14}$$

where $\odot$ is the Hadamard product, and $\mathbb{E}_h$ is the mean across the "heads" dimension. In order to compute the weighted attention relevance, we consider only the positive values of the gradients-relevance multiplication, resembling positive relevance. To account for the skip connections in the Transformer block, we add the identity matrix to avoid self inhibition for each token.

For comparison, using the same notation, the rollout [1] method is given by:

$$\hat{\mathbf{A}}^{(b)} = I + \mathbb{E}_h\mathbf{A}^{(b)} \tag{15}$$

$$\text{rollout} = \hat{\mathbf{A}}^{(1)} \cdot \hat{\mathbf{A}}^{(2)} \cdot \ldots \cdot \hat{\mathbf{A}}^{(B)} \tag{16}$$

We can observe that the result of rollout is fixed given an input sample, regardless of the target class to be visualized. In addition, it does not consider any signal, except for the pairwise attention scores.

3.4. Obtaining the image relevance map

The resulting explanation of our method is a matrix $\mathbf{C}$ of size $s \times s$, where s represents the sequence length of the input fed to the Transformer. Each row corresponds to a relevance map for each token given the other tokens - following the attention computation convention in Eq. 14, 11. Since this work focuses on classification models, only the `[CLS]` token, which encapsulates the explanation of the classification, is considered. The relevance map is, therefore, derived from the row $\mathbf{C}_{\texttt{[CLS]}} \in \mathbb{R}^s$ that corresponds to the `[CLS]` token. This row contains a score evaluating each token's influence on the classification token.

We consider only the tokens that correspond to the actual input, without special tokens, such as the `[CLS]` token and other separators. In vision models, such as ViT [11], the content tokens represent image patches. To obtain the final relevance map, we reshape the sequence to the patches grid size, *e.g.* for a square image, the patch grid size is $\sqrt{s-1} \times \sqrt{s-1}$. This map is upsampled back to the size of the original image using bilinear interpolation.

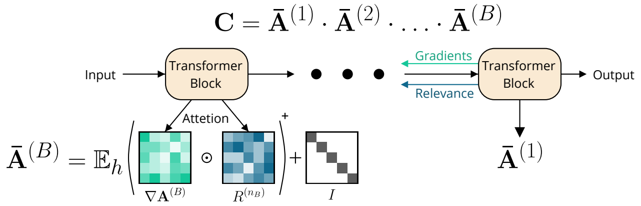

$\mathbf{C} = \bar{\mathbf{A}}^{(1)} \cdot \bar{\mathbf{A}}^{(2)} \cdot \ldots \cdot \bar{\mathbf{A}}^{(B)}$
Input
Transformer Block
Gradients
Relevance
Transformer Block
Output
Attetion
$\bar{\mathbf{A}}^{(B)} = \mathbb{E}_h$
$\nabla\mathbf{A}^{(B)}$
$R^{(n_B)}$
I
$\bar{\mathbf{A}}^{(1)}$

Figure 1: Illustration of our method. Gradients and relevancies are propagated through the network, and integrated to produce the final relevancy maps, as described in Eq. 13, 14.

4. Experiments

For the linguistic classification task, we experiment with the BERT-base [9] model as our classifier, assuming a maximum of 512 tokens, and a classification token `[CLS]` that is used as the input to the classification head.

For the visual classification task, we experiment with the pretrained ViT-base [11] model, which consists of a BERT-like model. The input is a sequence of all non-overlapping patches of size 16×16 of the input image, followed by flattening and linear layers, to produce a sequence of vectors. Similar to BERT, a classification token `[CLS]` is appended at the beginning of the sequence and used for classification.

The **baselines** are divided into three classes: attention-maps, relevance, and gradient-based methods. Each has different properties and assumptions over the architecture and propagation of information in the network. To best reflect the performance of different baselines, we focus on methods that are both common in the explainability literature, and applicable to the extensive tests we report in this section, *e.g.* Black-box methods, such as Perturbation and Shapely based methods, are computationally too expensive and inherently different from the proposed method. We briefly describe each baseline in the following section and the different experiments for each domain.

The attention-map baselines include rollout [1], following Eq. 16, which produces an explanation that takes into account all the attention-maps computed along the forward-pass. A more straightforward method is raw attention, *i.e.* using the attention map of block 1 to extract the relevance scores. These methods are class-agnostic by definition.

Unlike attention-map based methods, the relevance propagation methods consider the information flow through the entire network, and not just the attention maps. These baselines include Eq. 4 and the partial application of LRP that follows [42]. As we show in our experiments, the different variants of the LRP method are practically class-agnostic, meaning the visualization remains approximately the same for different target classes.

A common class-specific explanation method is Grad-CAM [32], which computes a weighted gradient-feature-map to the last convolution layer in a CNN model. The best way we found to apply GradCAM was to treat the last attention layer's `[CLS]` token as the designated feature map, without considering the `[CLS]` token itself. We note that the last output of a Transformer model (before the classification head), is a tensor $v \in \mathbb{R}^{s \times d}$, where the first dimension relates to different input tokens, and only the `[CLS]` token is fed to the classification head. Thus, performing Grad-CAM on v will impose a sparse gradients tensor ∇v, with zeros for all tokens, except `[CLS]`.

Evaluation settings For the visual domain, we follow the convention of reporting results for negative and positive perturbations, as well as showing results for segmentation, which can be seen as a general case of "The Pointing-Game" [19]. The dataset used is the validation set of ImageNet [31] (ILSVRC) 2012, consisting of 50K images from 1000 classes, and an annotated subset of ImageNet called ImageNet-Segmentation [16], containing 4,276 images from 445 categories. For the linguistic domain, we follow ERASER [10] and evaluate the reasoning for the Movies Reviews [44] dataset, which consists of 1600/200/200 reviews for train/val/test. This task is a binary sentiment analysis task. Providing explanations for question answering and entailment tasks of the other datasets in ERASER, which require input sizes of more than 512 tokens (the limit of our BERT model), is left for future work.

The positive and negative perturbation tests follow a two-stage setting. First, a pre-trained network is used for extracting visualizations for the validation set of ImageNet. Second, we gradually mask out the pixels of the input image and measure the mean top-1 accuracy of the network. In positive perturbation, pixels are masked from the highest relevance to the lowest, while in the negative version, from lowest to highest. In positive perturbation, one expects to see a steep decrease in performance, which indicates that the masked pixels are important to the classification score. In negative perturbation, a good explanation would maintain the accuracy of the model, while removing pixels that are not related to the class. In both cases, we measure the area-under-the-curve (AUC), for erasing between $10\% - 90\%$ of the pixels.

The two tests can be applied to the predicted or the ground-truth class. Class-specific methods are expected to gain performance in the latter case, while class-agnostic methods would present similar performance in both tests.

The segmentation tests consider each visualization as a soft-segmentation of the image, and compare it to the ground truth segmentation of the ImageNet-Segmentation dataset. Performance is measured by (i) pixel-accuracy, obtained after thresholding each visualization by the mean value, (ii) mean-intersection-over-union (mIoU), and (iii) mean-Average-Precision (mAP), which uses the soft-segmentation to obtain a score that is threshold-agnostic.

The NLP benchmark follows the evaluation setting of ERASER [10] for rationales extraction, where the goal is to extract parts of the input that support the (ground truth) classification. The BERT model is first fine-tuned on the training set of the Movie Reviews Dataset and the various evaluation methods are applied to its results on the test set. We report the token-F1 score, which is best suited for per-token explanation (in contrast to explanations that extract an excerpt). To best illustrate the performance of each method, we consider a token to be part of the "rationale" if it is part of the top-k tokens, and show results for $k = 10 \ldots 80$ in steps of 10 tokens. This way, we do not employ thresholding that may benefit some methods over others.

4.1. Results

Qualitative evaluation Fig.2 presents a visual comparison between our method and the various baselines. As can be seen, the baseline methods produce inconsistent performance, while our method results in a much clearer and consistent visualization.

In order to show that our method is class-specific, we show in Fig. 3 images with two objects, each from a different class. As can be seen, all methods, except Grad-CAM, produce similar visualization for each class, while our method provides two different and accurate visualizations.

Perturbation tests Tab. 1 presents the AUC obtained for both negative and positive perturbation tests, for both the predicted and the target class. As can be seen, our method achieves better performance by a large margin in both tests. Notice that because rollout and raw attention produce constant visualization given an input image, we omit their scores in the target-class test.

Segmentation The segmentation metrics (pixel-accuracy, mAP, and mIoU) on ImageNet-segmentation are shown in Tab. 2. As can be seen, our method outperforms all baselines by a significant margin.

Language reasoning Fig. 4 depicts the performance on the Movie Reviews "rationales" experiment, evaluating for top-K tokens, ranging from 10 to 80. As can be seen, while all methods benefit from increasing the amount tokens, our method consistently outperforms the baselines. See supplementary for a depiction of the obtained visualization.

Ablation study. We consider three variants of our method and present their performance on the segmentation and predicted class perturbation experiments. (i) Ours w/o $\nabla \mathbf{A}^{(b)}$, which modifies Eq. 13 s.t. we use $\mathbf{A}^{(b)}$ instead of $\nabla \mathbf{A}^{(b)}$, (ii) $\nabla \mathbf{A}^{(1)} \mathbf{R}^{(n_1)}$, *i.e.* disregarding rollout in Eq. 14, and using our method only on block 1, which is the block closest to the output, and (iii) $\nabla \mathbf{A}^{(B-1)} \mathbf{R}^{(n_{B-1})}$ which similar to (ii), only for block $B-1$ which is closer to the input.

As can be seen in Tab. 3 the ablation $\nabla \mathbf{A}^{(1)} \mathbf{R}^{(n_1)}$ in which one removes the rollout component, *i.e.*, Eq. 14,

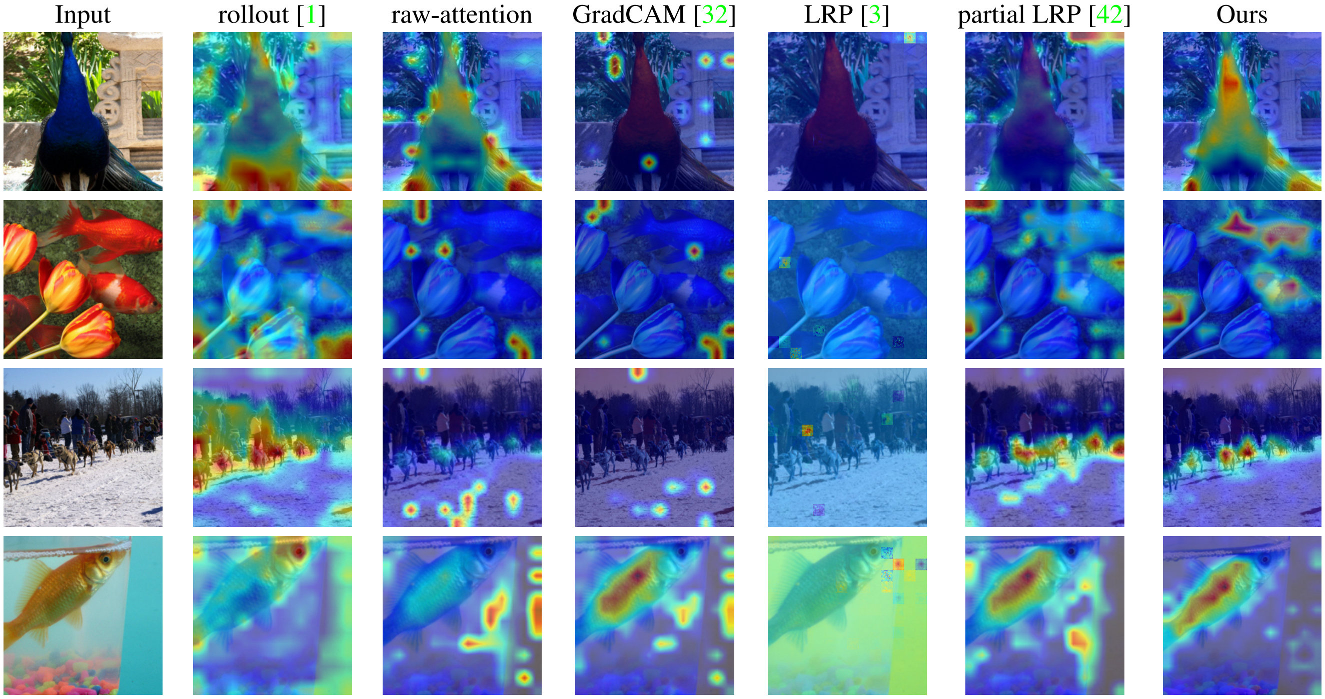


Figure 2: Sample results. As can be seen, our method produces more accurate visualizations.

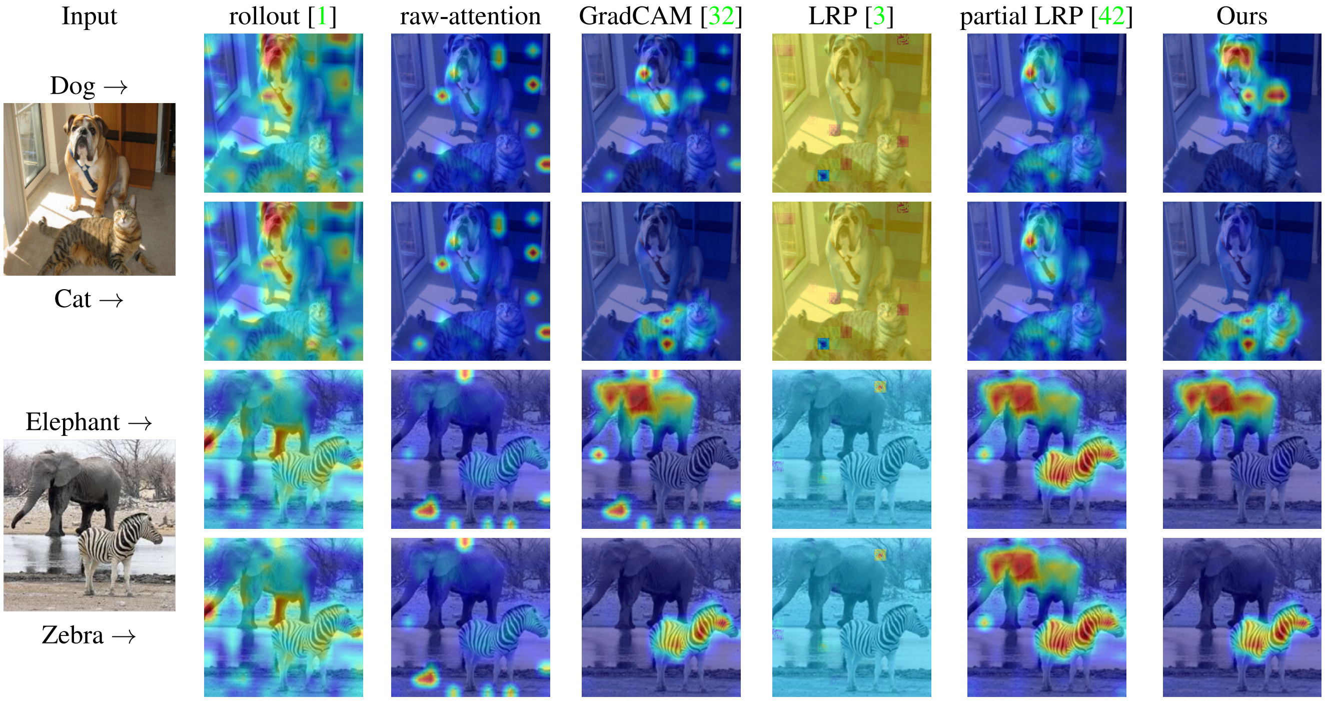


Figure 3: Class-specific visualizations. For each image we present results for two different classes. GradCam is the only method to generate different maps. However, its results are not convincing.

		rollout [1]	raw attention	GradCAM [32]	LRP [3]	partial LRP [42]	Ours
Negative	Predicted	53.1	45.55	41.52	43.49	50.49	**54.16**
	Target	-	-	42.02	43.49	50.49	**55.04**
Positive	Predicted	20.05	23.99	34.06	41.94	19.64	**17.03**
	Target	-	-	33.56	41.93	19.64	**16.04**

Table 1: Positive and Negative perturbation AUC results (percents) for the predicted and target classes, on the ImageNet [31] validation set. For positive perturbation lower is better, and for negative perturbation higher is better.

	rollout [1]	raw attention	GradCAM [32]	LRP [3]	partial LRP [42]	Ours
pixel accuracy	73.54	67.84	64.44	51.09	76.31	**79.70**
mAP	84.76	80.24	71.60	55.68	84.67	**86.03**
mIoU	55.42	46.37	40.82	32.89	57.94	**61.95**

Table 2: Segmentation performance on the ImageNet-segmentation [16] dataset (percent). Higher is better.

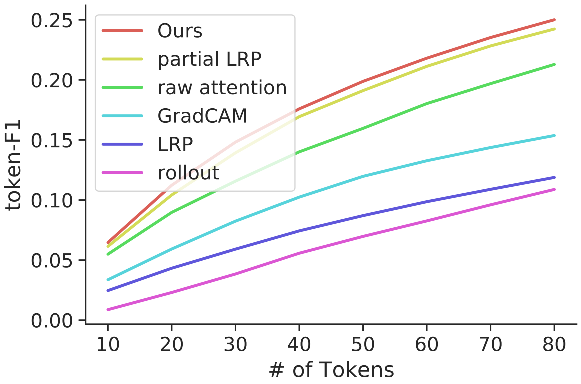


Figure 4: token-F1 scores on the Movie Reviews reasoning task.

	Segmentation			Perturbations	
	Acc.	mAP	mIoU	Pos.	Neg.
Ours w/o $\nabla\mathbf{A}^{(b)}$	77.66	85.66	59.88	18.23	52.88
$\nabla\mathbf{A}^{(1)}\mathbf{R}^{(n_1)}$	78.32	85.25	59.93	18.01	52.43
$\nabla\mathbf{A}^{(B-1)}\mathbf{R}^{(n_{B-1})}$	60.30	73.63	39.06	27.33	37.42
Ours	**79.70**	**86.03**	**61.95**	**17.03**	**54.16**

Table 3: Performance of different variants of our method.

while keeping the relevance and gradient integration, and only considering the last attention layer, leads to a moderate drop in performance. Out of the two single block visualizations ((ii), and (iii)), the combined attention gradient and relevancy at the $b = 1$ block, which is the closest to the output, is more informative than the block closest to the input. This is the same block that is being used for the raw-attention, partial LRP, and the GradCAM methods. The ablation that considers only this block outperforms these methods, indicating that the advantage of our method stems mostly from the combination of relevancy as we compute it and attention-map gradients.

5. Conclusions

The self-attention mechanism links each of the tokens to the `[CLS]` token. The strength of this attention link can be intuitively considered as an indicator of the contribution of each token to the classification. While this is intuitive, given the term “attention”, the attention values reflect only one aspect of the Transformer network or even of the self-attention head. As we demonstrate, both when using a fine-tuned BERT model for NLP and with the ViT model, attentions lead to fragmented and non-competitive explanations.

Despite this shortcoming and the importance of Transformer models, the literature with regards to interpretability of Transformers is sparse. In comparison to CNNs, multiple factors prevent methods developed for other forms of neural networks (not including the slower black-box methods) from being applied. These include the use of non-positive activation functions, the frequent use of skip connections, and the challenge of modeling the matrix multiplication that is used in self-attention.

Our method provides specific solutions to each of these challenges and obtains state-of-the-art results when compared to the methods of the Transformer literature, the LRP method, and the GradCam method, which can be applied directly to Transformers.

Acknowledgment

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant ERC CoG 725974).

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):e0130140, 2015.

[3] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer, 2016.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.

[5] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. In *International Conference on Learning Representations*, 2019.

[6] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of the 37th International Conference on Machine Learning*, volume 1, 2020.

[7] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, 2016.

[8] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pages 6970–6979, 2017.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*, 2019.

[11] 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.

[12] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[13] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2950–2958, 2019.

[14] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.

[15] Jindong Gu, Yinchong Yang, and Volker Tresp. Understanding individual decisions of cnns via contrastive backpropagation. In *Asian Conference on Computer Vision*, pages 119–134. Springer, 2018.

[16] Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari. Imagenet auto-annotation with segmentation propagation. *International Journal of Computer Vision*, 110(3):328–348, 2014.

[17] Shir Gur, Ameen Ali, and Lior Wolf. Visualization of supervised and self-supervised neural networks via attribution guided factorization. In *AAAI*, 2021.

[18] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[19] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 9737–9748, 2019.

[20] Brian Kenji Iwana, Ryohei Kuroki, and Seiichi Uchida. Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation. *arXiv preprint arXiv:1908.04351*, 2019.

[21] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9215–9223, 2018.

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[23] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019.

[24] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[25] Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3):233–255, 2016.

[26] Brent Mittelstadt, Chris Russell, and Sandra Wachter. Explaining explanations in ai. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 279–288, 2019.

[27] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

[28] Woo-Jeoung Nam, Shir Gur, Jaesik Choi, Lior Wolf, and Seong-Whan Lee. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. *arXiv preprint arXiv:1904.00605*, 2019.

[29] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, 2016.

[30] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[32] 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 *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[33] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org, 2017.

[34] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[35] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

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

[37] Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. In *Advances in Neural Information Processing Systems*, pages 4126–4135, 2019.

[38] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[39] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.

[40] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[42] 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. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, 2019.

[43] 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, 2015.

[44] Omar Zaidan and Jason Eisner. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 conference on Empirical methods in natural language processing*, pages 31–40, 2008.

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

[46] 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.

[47] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[48] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.