Generic Attention-model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers

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

Transformers are increasingly dominating multi-modal reasoning tasks, such as visual question answering, achieving state-of-the-art results thanks to their ability to contextualize information using the self-attention and co-attention mechanisms. These attention modules also play a role in other computer vision tasks including object detection and image segmentation. Unlike Transformers that only use self-attention, Transformers with co-attention require to consider multiple attention maps in parallel in order to highlight the information that is relevant to the prediction in the model's input. In this work, we propose the first method to explain prediction by any Transformer-based architecture, including bi-modal Transformers and Transformers with co-attentions. We provide generic solutions and apply these to the three most commonly used of these architectures: (i) pure self-attention, (ii) self-attention combined with co-attention, and (iii) encoder-decoder attention. We show that our method is superior to all existing methods which are adapted from single modality explainability. Our code is available at: https://github.com/hila-chefer/Transformer-MM-Explainability.

1. Introduction

Multi-modal Transformers may change the way that computer vision is practiced. While the state of the art computer vision models are often trained as task-specific models that infer a fixed number of labels, Radford et al. [28] have demonstrated that by training an image-text model that employs Transformers for encoding each modality, tens of downstream tasks can be performed, without further training ("zero-shot"), at comparable accuracy to the state of the art. Subsequently, Ramesh et al. [30] used a bi-modal Transformer to generate images that match a given description in unseen domains with unprecedented performance.

These two contributions merge text and images differently. The first encodes the text with a Transformer [40], the image by either a ResNet [15] or a Transformer, and then applies a symmetric contrastive loss. The second concatenates the quantized image representation to the text tokens and then employs a Transformer model. There are also many other methods of combining text and images [38, 21, 19, 18]. What is common to all of these is that the mapping from the two inputs to the prediction contains interaction between the two modalities. These interactions often challenge the existing explainability methods that are aimed at attention-based models, since, as far as we can ascertain, all existing Transformer explainability methods (*e.g.*, [5, 1]) heavily rely on self-attention, and do not provide adaptations to any other form of attention, which is commonly used in multi-modal Transformers.

Another class of Transformer models that is not restricted to self-attention is that of Transformer encoder-decoders, *i.e.* generative models, in which the model typically receives an input from a single domain, and produces output from a different one. These models are used in an emerging class of object detection [4, 50] and image segmentation [43, 27, 42] methods, and are also widely used for various NLP tasks, such as machine translation [40, 17]. In these object detection methods, for example, embeddings of the position-specific and class-specific queries are crossed with the encoded image information.

We propose the first explainability method that is applicable to all Transformer architectures, and demonstrate its effectiveness on the three most commonly used Transformer architectures: (i) pure self-attention, (ii) self-attention combined with co-attention, and (iii) encoder-decoder attention. We use an exemplar model from each architecture, and prove our method's superiority over existing Transformer explainability methods, adapted from their single modality origin. Our explainability prescription is easier to implement than existing methods, such as [5], and can be readily applied to any attention-based architecture.

2. Related work

Explainability in computer vision Interpreting computer vision algorithms usually entails the synthesis of a heatmap that depicts the computed relevancy at each image location. This can be class-dependent (for every possible label), or class-agnostic, in which case it depends only on the input and the model. Unlike most methods below, our method is of the first type. There are multiple families of explainability methods, including saliency-based methods [8, 34, 23, 49, 45, 48], methods that consider activations [10] using the forward pass or the backprop [46], perturbation based methods [11, 12], and methods based on Shapley-values [22, 6]. The latter enjoy clear theoretical motivation. Theoretical justification is also given to attribution-based methods, through the theory of the Deep Taylor Decomposition [24]. Such methods assign relevancy recursively from the top layer, backward, such that the sum of relevancies remains fixed. The LRP method [3], is one such prominent method. Since LRP and most variants [25, 33, 22] are class agnostic [16], class-specific extensions were introduced [13, 16, 14].

Gradient-based methods directly consider the gradient of the loss with respect to the input of each layer, as computed through backpropagation. Examples include class agnostic methods [33, 37, 35, 36]. A related class-specific approach is the Grad-CAM method [32], which considers the input features with the class-dependent gradient at the top layers.

Explainability for Transformers Most attempts to explain Transformers directly employ the attention maps. This, however, neglects the intermediate attention scores, as well as the other components of the Transformers. As noted by Chefer et al [5], the computation in each attention head mixes queries, keys, and values and cannot be fully captured by considering only the inner products of queries and keys, which is what is referred to as attention.

LRP was applied to capture the relative importance of the attention heads within each Transformer block by Voita et al. [41]. This method, however, does not propagate the relevancy scores back to the input to produce a heatmap.

Abnar et al. [1] propose a way to combine the attention scores across multiple layers. Two methods are suggested: attention rollout and attention flow. The first combines attention linearly along alternative paths in the pairwise attention graph. [5] demonstrates that this method fails to distinguish between positive and negative contributions to the decision, leading to an accumulation of relevancy scores across the layers in cases where these should be cancelled out. The attention flow method is formulated as a max-flow problem on the same pairwise attention graph. While it was shown in [1] to somewhat outperform rollout in specific scenarios, it is too slow to support large-scale evaluations.

In contrast to these methods, Chefer et al. [5] provide a comprehensive treatment of the information propagation within all components of the Transformer model, which back-propagates the information through all layers from the decision back to the input. The solution is based on Layer-wise Relevance Propagation [3], with gradient integration for the self-attention layers, and is shown to be very effective for single modality Transformer encoders, such as [9]. This method, however, does not provide a solution for attention modules other than self-attention, thus can not provide explanations for all Transformer architectures. [44] presents a different approach to Transformer visualization using dictionary learning.

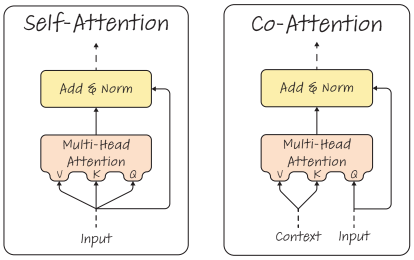
Transformers in computer vision Transformer technology has become increasingly prevalent for bi-modal tasks, such as image captioning and text-based image retrieval. We distinguish between networks that rely on self-attention, such as VisualBERT [18] and Oscar [19] and those that also employ co-attention modules, such as LXMERT [38] and ViLBERT [21]. Our method provides suitable visualization for both types.

Our method also provides the first complete solution, as far as we can ascertain, for Transformer encoder-decoders [40, 29, 17], which have been increasingly prevalent in computer vision. In the DETR Transformer-based detection method [4], the image is encoded by a Transformer encoder, and the obtained information is co-attended together with queries that are both positional and class-based. Our method can be also applied to encoder-based visual Transformers, such as those used for image recognition [7, 9, 39], and image segmentation with a CNN decoder [47]. However, in this case, existing Transformer explainability methods can also be applied.

3. Method

Our method uses the model's attention layers to produce relevancy maps for each of the interactions between the input modalities in the network. In this work, we focus on image and text interactions, and attention modules for generative models, *i.e.*, encoder-decoder attention. However, our method is easily applicable to any Transformer-based architecture, and can also be generalized to address more than two modalities. In the following, we discuss the method's propagation rules under the assumption of two modalities, *e.g.* text and image, followed by a detailed description of how to apply our method to each of the model types.

Let t, i be the number of text and image input tokens respectively. To simplify notation, we use the same symbols (t, i) to identify variables that are associated with the two domains. Multi-modal attention networks contain four types of interactions between the input tokens: $\mathbf{A}^{tt}$ and $\mathbf{A}^{ii}$ are the self-attention interactions for the text and image tokens, respectively. $\mathbf{A}^{ti}$, $\mathbf{A}^{it}$ are the multi-modal attention interactions, where $\mathbf{A}^{ti}$ represents the influence of the image tokens on each text token, and $\mathbf{A}^{it}$ represents the influ-



(a) (b)

Figure 1: (a) Self-attention and (b) co-attention modules.

ence of the text tokens on each image token.

In accordance with the attention interactions described, we construct a relevancy map per interaction, *i.e.* $\mathbf{R}^{tt}$, $\mathbf{R}^{ii}$ for self-attention, and $\mathbf{R}^{ti}$, $\mathbf{R}^{it}$ for bi-modal attention.

The method calculates the relevancy maps by a forward pass on the attention layers, with each layer contributing to the aggregated relevance matrices using the update rules we will describe in the following subsections.

Relevancy initialization Before the attention operations, each token is self-contained. Thus, self-attention interactions are initialized with the identity matrix. For bi-modal interactions, before the attention layers, each modality is separate and does not contain context from the other modality, therefore, the relevancy maps are initialized to zeros.

$$\mathbf{R}^{ii} = \mathbb{1}^{i\times i}, \quad \mathbf{R}^{tt} = \mathbb{1}^{t\times t} \tag{1}$$

$$\mathbf{R}^{it} = \mathbf{0}^{i\times t}, \quad \mathbf{R}^{ti} = \mathbf{0}^{t\times i} \tag{2}$$

Relevancy update rules As the attention layers contextualize the tokens, our method modifies the relevancy maps that are impacted by the mixture of token embeddings. Recall the attention mechanism presented in [40]:

$$\mathbf{A} = softmax(\frac{\mathbf{Q}\cdot\mathbf{K}^\top}{\sqrt{d_h}}) \tag{3}$$

$$\mathbf{O} = \mathbf{A}\cdot\mathbf{V} \tag{4}$$

where $(\cdot)$ denotes matrix multiplication, $\mathbf{O} \in \mathbb{R}^{h\times q\times d_h}$ is the output of the attention module, $\mathbf{Q} \in \mathbb{R}^{h\times q\times d_h}$ is the queries matrix, and $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{h\times k\times d_h}$ are the keys and values matrices. h is the number of heads, d_h is the embedding dimension, and $k, q \in \{i, t\}$ indicate the domains and the number of tokens in each domain, *i.e.*, the attention takes place between q query tokens and k key tokens. Note that, as can be seen in Fig. 1, for self-attention layers, it holds that $k = q$ and $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are all projections of the input to the attention unit, while in co-attention $\mathbf{Q}$ is a projection of the input, and $\mathbf{K}, \mathbf{V}$ are projections of the context input from the other modality. $\mathbf{A} \in \mathbb{R}^{h\times q\times k}$ is the attention map, which intuitively defines connections between each pair of tokens from q, k. Since the attention module is followed by a residual connection, as shown in Fig. 1, we accumulate the relevancies by adding each layer's contribution to the aggregated relevancies, similar to [1] in which the identity matrix is added to account for residual connections.

Our method uses the attention map $\mathbf{A}$ of each attention layer to update the relevancy maps. Since each such map is comprised of h heads, we follow [5] and use gradients to average across heads. Note that Voita et al. [41] show that attention heads differ in importance and relevance, thus a simple average across heads results in distorted relevancy maps. The final attention map $\bar{\mathbf{A}} \in \mathbb{R}^{q\times k}$ of our method is then defined as follows:

$$\bar{\mathbf{A}} = \mathbb{E}_h((\nabla\mathbf{A}\odot\mathbf{A})^+) \tag{5}$$

where $\odot$ is the Hadamard product, $\nabla\mathbf{A} := \frac{\partial y_t}{\partial \mathbf{A}}$ for y_t which is the model's output for the class we wish to visualize t, and $\mathbb{E}_h$ is the mean across the heads dimension. Following [5] we remove the negative contributions before averaging.

For self-attention layers that satisfy $\bar{\mathbf{A}} \in \mathbb{R}^{q\times q}$ the update rules for the affected aggregated relevancy scores are:

$$\mathbf{R}^{qq} \leftarrow \mathbf{R}^{qq} + \bar{\mathbf{A}}\cdot\mathbf{R}^{qq} \tag{6}$$

$$\mathbf{R}^{qk} \leftarrow \mathbf{R}^{qk} + \bar{\mathbf{A}}\cdot\mathbf{R}^{qk} \tag{7}$$

In Eq. 6 we account for the fact that the tokens were already contextualized in previous attention layers by applying matrix multiplication with the aggregated self-attention matrix $\mathbf{R^{qq}}$, as done in [1, 5]. For Eq. 7, notice that the previous bi-modal attention layers inserted context from k into q, therefore, when the self-attention mixes tokens from q, it also mixes the context k in each token from q. The previous layers' mixture of context is embodied by $\mathbf{R}^{qk}$. Thus, we calculate the added context from the self-attention process.

In the case of $\bar{\mathbf{A}} \in \mathbb{R}^{q\times k}$, where a bi-modal attention is applied, the update rules of the relevancy accumulators include normalization for the self-attention matrix $\mathbf{R}^{qq}$. Since we initialized $\mathbf{R}^{qq} = \mathbb{1}^{q\times q}$, and Eq. 6 accumulates the relevancy matrices at each layer, we can consider an aggregated self-attention matrix $\mathbf{R}^{qq}$ as a matrix comprised of two parts, the first is the identity matrix from the initialization, and the second, $\hat{\mathbf{R}}^{qq} = \mathbf{R}^{qq} - \mathbb{1}^{q\times q}$ is the matrix created by the aggregation of self-attention across the layers. Since Eq. 5 uses gradients to average across heads, the values of $\hat{\mathbf{R}}^{qq}$ are typically small due to the multiplication with the gradients. We wish to account equally both for the fact that each token influences itself and for the contextualization by the self-attention mechanism. Therefore, we normalize each row in $\hat{\mathbf{R}}^{qq}$ so that it sums to 1. Intuitively, row i in $\hat{\mathbf{R}}^{qq}$ disclosed the self-attention value of each token w.r.t. the i-th token, and the identity matrix $\mathbb{1}^{q\times q}$ sets that value for each token w.r.t. itself as 1. Thus we define:

$$\forall m, n \in q : \hat{\mathbf{S}}^{qq}_{m,n} = \sum_{l=1}^{|q|} \hat{\mathbf{R}}^{qq}_{m,l} \tag{8}$$

$$\bar{\mathbf{R}}^{qq} = \hat{\mathbf{R}}^{qq} / \hat{\mathbf{S}}^{qq} + \mathbb{1}^{q \times q}, \tag{9}$$

where $/$ stands for matrix division element by element. In the above, we normalize each row in $\hat{\mathbf{R}}^{qq}$ by dividing each element in the row by the sum of the row. Next, we define the following aggregation rules for bi-modal attention units:

$$\mathbf{R}^{qk} \leftarrow \mathbf{R}^{qk} + (\bar{\mathbf{R}}^{qq})^{\top} \cdot \bar{\mathbf{A}} \cdot \bar{\mathbf{R}}^{kk} \tag{10}$$

$$\mathbf{R}^{qq} \leftarrow \mathbf{R}^{qq} + \bar{\mathbf{A}} \cdot \mathbf{R}^{kq} \tag{11}$$

Eq. 10 accounts for the fact that the tokens of each modality were already contextualized in previous attention layers by applying matrix multiplication with the normalized aggregated self-attention matrices $\bar{\mathbf{R}}^{qq}, \bar{\mathbf{R}}^{kk}$.

For Eq. 11, notice that the previous bi-modal attention layers integrate the embeddings of the two modalities, thus when contextualizing q with k, k also contains information from q, embodied in $\mathbf{R}^{kq}$.

Note that the above rules are described w.r.t. input from modality $q \in \{i, t\}$, and context from modality $k \in \{i, t\}$ *i.e.* the rules are symmetrically applied to both modalities.

3.1. Obtaining classification relevancies

In order to make the final classification, Transformer-based models usually regard the `[CLS]` token, which is a token that is added to the input tokens and constructs a general representation of all the input tokens. To retrieve per-token relevancies for classification tasks, one can consider the row corresponding to the `[CLS]` token in the corresponding relevancy map. For instance, assuming the `[CLS]` token is the first token in the text modality, to extract relevancies per text token, one should consider the first row of $\mathbf{R}^{tt}$, and to extract the image token relevancies, consider the first row in $\mathbf{R}^{ti}$ which describes the connections between the `[CLS]` token and each image token.

3.2. Adaptation to attention types

In this work, we examine our method on three different types of attention mechanisms used in Transformer-based networks. The architectures and matching propagation rules are visualized in Fig. 2. The first architecture type is a multi-modal Transformer, where the two modalities are concatenated and separated by the `[SEP]` token [18, 19], as demonstrated in Fig. 2(a). Such networks only use self-attention to contextualize the modalities, *i.e.* only Eq. 6. Since the model is based on pure self-attention, we produce one relevancy map $\mathbf{R}^{(t+i, t+i)}$ which defines connections between the modalities, as well as within each modality. In order to visualize the tokens related to the classification, one should consider the row of $\mathbf{R}^{(t+i, t+i)}$ which corresponds to the token used for classification. This row ${\mathbf{R}_{\mathbf{cls}}}^{(t+i)}$ yields a relevancy score per image token and per text token.

The second type is a multi-modal attention network that incorporates co-attention modules that contextualize each modality with the other modality [38, 21], as can be seen in Fig. 2(b). Such networks require all propagation rules described above, for each modality. To produce relevancies for the classification, we simply follow the example in Sec 3.1, since as Fig. 2(b) depicts, the `[CLS]` token in this case is the first token of the text modality.

The third and last type is a generative model where there is one input modality, and the output is from a different domain [4, 50, 43, 27, 42, 40, 17], which is visualized in Fig. 2(c). Such networks contain an encoder that utilizes self-attention on the input and a decoder. The decoder has two types of inputs. The first is the encoded data, which remains unchanged, and the second are inputs from the decoder's domain. The decoder proceeds to utilize self-attention on the decoder domain's tokens, followed by a co-attention layer contextualizing them with the encoder's output. To clarify, in this case, the relevance update rules are as follows: notate by e the encoder's tokens, and by d the decoder's tokens. The relevancy matrices are: $\mathbf{R}^{ee}, \mathbf{R}^{dd}$ for the self-attention interactions, and $\mathbf{R}^{de}$ for the bi-modal interactions between the decoder's tokens and the encoder's tokens. Notice that since the encoder is not contextualized, we do not have a relevancy matrix $\mathbf{R}^{ed}$. The encoder's self-attention calculation for $\mathbf{R}^{ee}$ simply follows Eq. 6. For the decoder's self-attention, we apply Eq. 6, 7. For the bi-modal attention in the decoder, we follow Eq. 10 to account for self-attention in the encoder and the decoder. Notice that Eq. 11 is irrelevant since we do not have a relevancy map for $\mathbf{R}^{kq} = \mathbf{R}^{ed}$. In order to extract relevancies in this case, we consider the relevancy map $\mathbf{R}^{de}$. In this work, we use an object detection model as our exemplar encoder-decoder architecture. For such models, each token from d is a query representing an object in the input image. In order to produce relevancy for each of the image regions w.r.t. an object j that was detected, one should consider the j-th row of $\mathbf{R}^{de}$, which corresponds to the j-th detection. $\mathbf{R}^{de}_j$ contains a relevancy score per each encoder token, which is in this case an image region.

4. Baselines

We focus on methods that are both common in the explainability literature, and applicable to the extensive tests we report in this work. We present baselines of three classes, following [5]: attention map baselines, gradient baselines, and relevancy map baselines. Our attention map baselines are raw attention and rollout. Raw attention regards only the last layer's attention map as the relevancy map, *e.g.* $\mathbf{R}^{tt} = \mathbf{A}^{tt}$, where $\mathbf{A}^{tt}$ is the last text self-attention map. The second is rollout, which follows [1] for all the self-attention layers. Since the rollout baseline is based solely on self-attention, to distinguish from raw at-

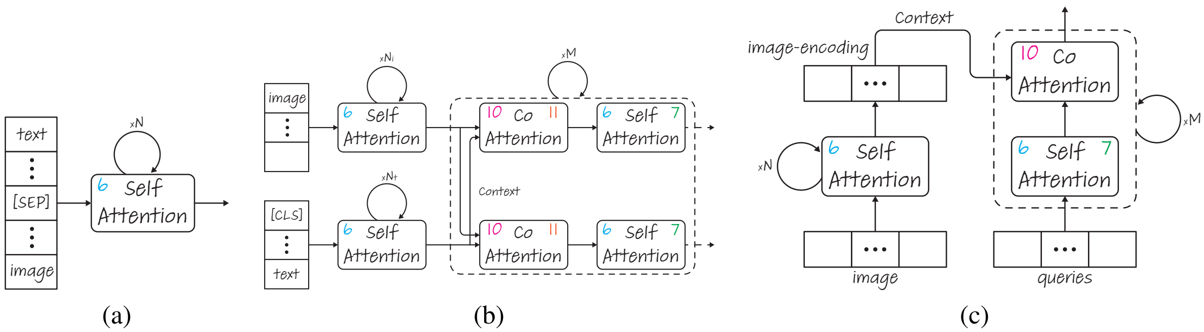


Figure 2: Illustration of the three architecture types presented in our work. The numbers in each attention module represent the Eq. number of the rule applied by our method on the module's forward pass. (a) VisualBERT: a pure self-attention architecture. (b) LXMERT: self-attention with co-attention encoder architecture. (c) DETR: encoder-decoder architecture.

tention, we employ the following for $\mathbf{R}^{qk}, q, k \in \{t, i\}$:

$$\mathbf{R}^{qk} = (\mathbf{R}^{qq})^\top \cdot \bar{\mathbf{A}} \cdot \mathbf{R}^{kk} \tag{12}$$

where $\mathbf{R}^{kk}, \mathbf{R}^{qq}$ are the self-attention relevancies computed by rollout, and $\bar{\mathbf{A}} \in \mathbb{R}^{\mathbf{q}\times\mathbf{k}}$ is the last bi-attention map. For our gradient baselines, we use the Grad-CAM [32] adaptation described in [5], *i.e.*, we examine the last attention layer, and perform Grad-CAM on the attention map's heads. Lastly, our relevancy map baselines include partial LRP, following [41], which uses the LRP relevancy values of the last attention layer to average across the heads, and the Transformer attribution method described in [5]. The method in [5] employs Eq. 5 for all attention layers in order to average across heads, in the following way:

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

where the only difference compared to Eq. 5, is that [5] uses the LRP [3] relevancy values of $\mathbf{A}$, *i.e.* $\mathbf{R}^{\mathbf{A}}$, instead of using the raw attention maps as done in Eq. 5. Additionally, [5] uses Eq. 6 for all self-attention layers. For non self-attention layers, our version of [5] takes the last attention map, and averages across heads using Eq. 13. Note that while applying our method only requires a few simple hooks for the attention modules, LRP requires a custom implementation of all network layers.

5. Experiments

Our experiments include three Transformer-based models, each representing one of the three types of architectures we refer to in this work. See Fig. 2 for illustrations of each of the architectures. In addition, to compare with previous work [5, 1] in the same setting for which these methods were conceived, we also consider ViT [9]. The relevancy propagation for each model follows Sec. 3.2.

The first model we examine is VisualBERT [18], which represents a self-attention based architecture, and the second model is LXMERT [38], which represents an architecture combining self-attention and co-attention in a Transformer encoder for two modalities.

For both models, we perform positive and negative perturbation tests on each modality separately to evaluate the quality of the relevancy matrices produced by the methods. We use the visual question answering [2] task in testing the explanations since this task requires the models to demonstrate an understanding of both input modalities and the connections between them.

The perturbation tests are performed as follows: first, a pre-trained network is used for extracting relevancy maps for $10,000$ randomly picked samples from the validation set of the VQA dataset. Second, we gradually remove the tokens of a given modality and measure the mean top-1 accuracy of the network. In positive perturbation, tokens are removed 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 removed tokens are important to the classification score. In negative perturbation, a good explanation would maintain the accuracy of the model while removing tokens that are not related to the classification. In both cases, we measure the area-under-the-curve (AUC), to evaluate the decrease in the model's accuracy.

We note that in all perturbation tests, the accuracy does not reach 0%, even when removing 100% of the tokens of each modality. This is since the input from the other modality remains intact therefore the models can rely on a single modality to provide a reasonable answer.

Notice that the LXMERT [38] image perturbation test results, which are depicted in Fig. 3(a,b), demonstrate a clear advantage to our method compared to other methods. For negative perturbation, the AUC using our method is the largest by a sizeable margin, and the accuracy is well-preserved even after removing more than 80% of the image tokens, and for positive perturbation, notice the very steep decrease in accuracy, and the low AUC.

As can be seen in Fig. 2(b), the `[CLS]` token for

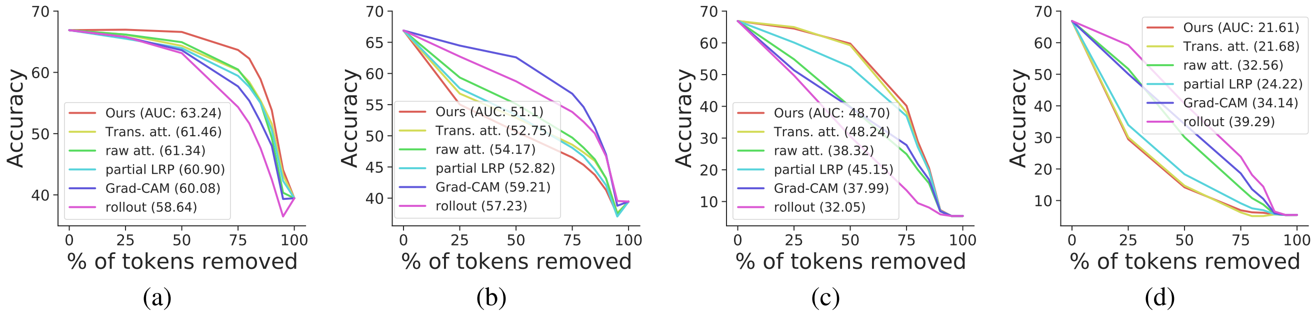


Figure 3: LXMERT perturbation test results. For negative perturbation, larger AUC is better; for positive perturbation, smaller AUC is better. (a) negative perturbation on image tokens, (b) positive perturbation on image tokens, (c) negative perturbation on text tokens, and (d) positive perturbation on text tokens.

Figure 4: A comparison between our method (top) and the method of [5] (bottom) for VQA with the LXMERT model. Relevancy for text is given as shades of red. Relevancy for images is given by multiplying each region by the relative relevancy. The results for the text part are similar. For the images, our method provides much more focused results. Both observations are aligned with the quantitative results. Answers (left to right): no, yes, yes, no.

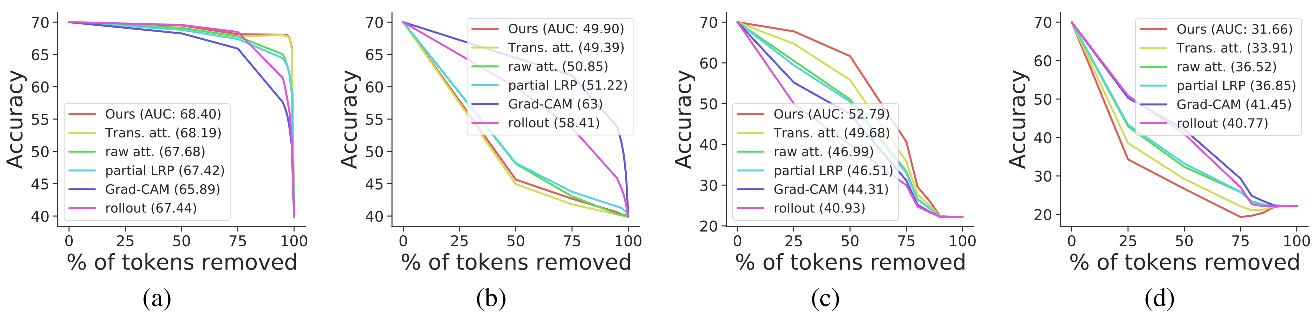


Figure 5: VisualBERT perturbation test results. For negative perturbation, larger AUC is better; positive perturbation, smaller AUC is better. (a) negative perturbation on image tokens, (b) positive perturbation on image tokens, (c) negative perturbation on text tokens, and (d) positive perturbation on text tokens.

	Supervised	Weakly supervised segmentation					
	detection	rollout [1]	raw attention	Grad-CAM [32]	partial LRP [41]	Trans. attribution [5]	Ours
AP	51.8	0.1	5.6	2.3	4.7	7.2	**13.1** (+5.9)
AP_{medium}	56.3	0.1	9.6	2.3	8.0	10.4	**14.4** (+4.0)
AP_{large}	67.6	0.2	6.9	4.7	5.1	12.4	**24.6** (+12.2)
AR	67.4	0.4	11.7	5.5	10.4	13.4	**19.3** (+5.9)
AR_{medium}	72.8	0.1	21.8	5.9	19.9	21.0	**23.9** (+2.1)
AR_{large}	85.1	0.9	10.8	10.7	8.0	19.4	**33.2** (+13.8)

Table 1: DETR [50]-based weakly supervised segmentation results on the MSCOCO [20] validation set, higher is better. AP=average precision, AR=average recall. The subscripts indicate benchmark subsets. The first column is the DETR [50] bounding boxes detection scores obtained for each category, while the rest of the columns are for segmentation maps.

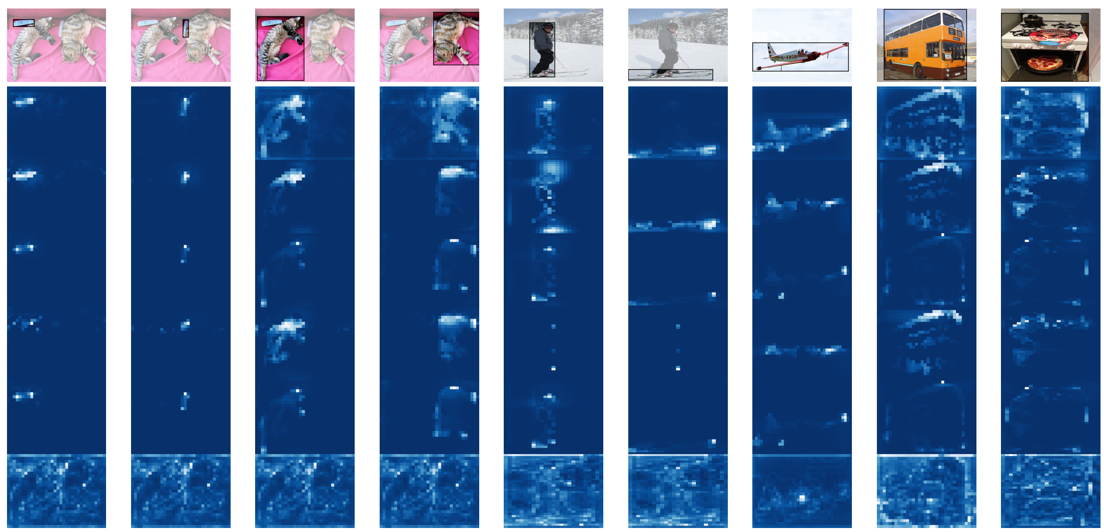


Figure 6: Sample segmentation masks for DETR [50]. Each row represents a method. Detections (from left to right): remote, remote, cat, cat, person, skis, airplane, bus, oven (in the last two samples, the bounding box is almost the entire frame). Our method produces the most accurate results, and the segmentations are consistent with the detections produced by DETR.

		rollout	raw att.	GCAM	LRP	T. Attr	Ours
N	Predicted	53.10	45.55	41.52	50.49	54.16	**54.61**
	Target	-	-	42.02	50.49	55.04	**55.67**
P	Predicted	20.05	23.99	34.06	19.64	**17.03**	17.32
	Target	-	-	33.56	19.64	**16.04**	16.72

Table 2: ViT [9] positive (P) and negative (N) perturbation AUC results for the predicted and target classes, on the ImageNet [31] validation set. For negative perturbation, larger AUC is better; positive perturbation, smaller AUC is better. GCAM=Grad-CAM; T. Attr = Transformer attribution [5].

LXMERT [38] is the first token of the text modality, thus following Sec. 3.2, $\mathbf{R}^{\mathbf{ti}}$ is the map used for extracting relevancies in the image perturbation case. Since $\mathbf{R}^{\mathbf{ti}}$ is a multi-modal relevancy map, the image perturbation tests best demonstrate the advantage of using our method over all existing methods, which fall short in evaluating relevancies from the co-attention modules.

For the LXMERT [38] text perturbation tests which are depicted in Fig. 3(c,d), notice that by Sec. 3.2, we visualize $\mathbf{R}^{\mathbf{tt}}$ which is a self-attention map, where the dominating update rule is Eq. 6. This rule is identical to the rule employed by the Transformer attribution [5] baseline, except for the head averaging in Eq. 13. Therefore, the main difference between our proposed method and the method described in [5] is the choice to use LRP [3] in the head averaging process. This results in very similar results for both methods. For completeness, we provide in the supplementary results for our method when adding LRP, as is done in Eq. 13. The rest of the methods fall far behind.

Fig. 4 presents typical results for our method and for

Transformer attribution [5]. The rest of the methods are not competitive and their matching samples are presented in the supplementary. As can be seen, the text results are similar, as predicted by the quantitative results. Our image attention results are much more focused on the relevant image parts than those of the baseline method.

Note that since VisualBERT [18] is based on pure self-attention, the difference between our method and the Transformer attribution [5] method stems from the choice of whether or not to use LRP [3] for head averaging in Eq. 5, similarly to the LXMERT [38] text (but not image) perturbation tests. As can be seen in Fig. 5, our method outperforms all methods and achieves very similar results to those of [5], and in some cases, such as the text perturbation test, even outperforms [5] by a sizeable margin. This demonstrates that the use of LRP [3] is unnecessary, even for pure self-attention architectures.

The third model we experiment on is DETR [4], which is an encoder-decoder model, as seen in Fig. 2(c). We use a pre-trained DETR model with the ImageNet pre-trained backbone ResNet-50, which is trained for object detection on the MSCOCO [20] dataset. Importantly, this model has only been trained for object detection, *i.e.*, producing bounding boxes and classifications for each object in the input image. To evaluate the different explainability methods, our test uses each of the methods on the $5,000$ samples of the MSCOCO [20] validation set to produce segmentation masks, *i.e.* we consider the output of each method to be a segmentation mask. We first filter the queries to include only ones where the classification probability is higher than 50% and then employ Otsu's thresholding method [26] to separate the foreground and the background of the segmentation. See supplementary for the full details.

Our generated segmentation masks visualize the bounding boxes predicted by DETR, therefore it should be noted that the produced masks are inherently dependent on the quality of the corresponding bounding boxes, *i.e.*, when the predicted bounding box is not sufficient, naturally, the mask produced for it will be at least equally inaccurate. In addition, since the explainability methods are not aimed at producing segmentation maps, they often do not output contiguous masks, and the Otsu threshold may also create "holes" in the produced masks. For all the reasons above, we decrease the minimal IoU used for MSCOCO evaluation from 0.5 to 0.2, which significantly benefits all the methods, and we present the results of the MSCOCO segmentation evaluation for the categories where the produced bounding boxes are good enough for the generation of segmentation masks, *e.g.*, we do not present results for small objects[1]. As can be seen in Tab. 1, our method outperforms all other methods by a very large margin, which indicates that our novel formulations are necessary for non self-attention architectures. Notice the correlation in Tab. 1 between the bounding box evaluation for DETR and our segmentation. See Fig. 6 for visualizations of the masks.

Lastly, in order to compare our method with existing single-modality baselines, we present the positive and negative perturbation tests on ViT-Base [9], as performed by [5]. As mentioned, since ViT-Base [9] is a single-modality Transformer encoder, the only difference between our method and the Transformer attribution method of [5] is the use of LRP [3] in Eq. 5, as shown in Eq. 13. Therefore, as can be seen in Tab. 2, the differences between our method and the method proposed in [5] are very mild, which is another indication that LRP [3] can be removed. Tab. 2 also shows improvement in performance when using the target class instead of the predicted class for gradient propagation in Eq. 5, which, as stated in [5], indicates that our method is able to produce class-specific visualizations.

Ablation study We present in the supplementary three variations of our method that demonstrate the effectiveness of our normalization (Eq. 8,9), the necessity of the aggregation in all our rules 6, 7, 10, 11, and the need for the self-attention updates to the bi-modal rule 10.

[1]We choose this working point since using a stricter threshold leads to baseline results that are slightly better than chance and our method outperforms but provides a score that is only 2-3 times better than chance.

6. Conclusions

Transformers play an increasingly dominant role in computer vision, with image-text Transformers and Transformers that perform tasks that have output domains that are more complex than the labels provided by a classifier, presenting groundbreaking results. In order to debug such models, as well as to support downstream tasks, and the increasing demand for model-interpretability, it is required to have complete and accurate explainability methods. However, the current explainability literature for Transformers is limited, overly focuses on pure attention maps, and lacks the methodology for treating co-attention maps.

Our method carefully tracks the evolution and mixing of the attention maps. It provides a generic prescription that is applicable to all attention models we are aware of. Empirically, it outperforms the existing methods across Transformer architectures and evaluation metrics. In some cases, when self-attention is prominent, the recent method by Chefer et al. [5] is the only method that can provide comparable results. However, in the majority of the experiments, our method leads over all methods by a very sizable margin.

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] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[3] 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.

[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] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. *arXiv preprint arXiv:2012.09838*, 2020.

[6] 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.

[7] 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.

[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] 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.

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

[11] 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.

[12] 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.

[13] 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.

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

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[16] 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.

[17] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[18] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[19] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

[26] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[27] Matthieu Paul, Martin Danelljan, Luc Van Gool, and Radu Timofte. Local memory attention for fast video semantic segmentation. *arXiv preprint arXiv:2101.01715*, 2021.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[30] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

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

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

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

[37] 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.

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

[39] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[40] 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.

[41] 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.

[42] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. *arXiv preprint arXiv:2012.00759*, 2020.

[43] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, 2020.

[44] Zeyu Yun, Yubei Chen, Bruno A Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. *arXiv preprint arXiv:2103.15949*, 2021.

[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] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020.

[48] 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.

[49] 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.

[50] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.