Explaining Information Flow Inside Vision Transformers Using Markov Chain

Tingyi Yuan*
Department of Industrial Automation,
Xi'an Jiaotong University
Xi'an, China
qq842195980@stu.xjtu.edu.cn

Xuhong Li
Baidu Research,
Baidu Inc.
Beijing, China
lixuhong@baidu.com

Haoyi Xiong
Baidu Research,
Baidu Inc.
Beijing, China
xionghaoyi@baidu.com

Hui Cao
Department of Industrial Automation
Xi'an Jiaotong University
Xi'an, China
huicao@mail.xjtu.edu.cn

Dejing Dou
Baidu Research,
Baidu Inc.
Beijing, China
doudejing@baidu.com

Abstract

Transformer-based models are receiving increasingly popularity in the field of computer vision, however, the corresponding interpretability study is less. As the simplest explainability method, visualization of attention weights exerts poor performance because of lacking association between the input and model decisions. In this study, we propose a method, named *Transition Attention Maps*, to generate the saliency map concerning a specific target category. The proposed approach connects the idea of the Markov chain, to investigate the information flow across layers of the Transformer and combine the integrated gradients to compute the relevance of input tokens for the model decisions. We compare with other explainability methods using Vision Transformer as a benchmark and demonstrate that our method achieves better performance in various aspects. We open source the implementation of our approach at https://github.com/PaddlePaddle/InterpretDL.

1 Introduction

Self-attention-based architectures, specifically Transformers [1–3] are dominating the field of natural language processing (NLP). More recently, Transformers have become an alternative architecture against convolutional neural networks (CNN). Nevertheless, as other *deep* architectures, these models are uninterpretable black boxes. The reasoning behind the prediction and decision of the model is unseen. Although the Transformers can precisely classify images with very high accuracy, it is unknown whether the correct and proper features have been learned, or the desired information has been extracted, to recognize the visual objects. To address this issue, visualization of the deep model decision process is an appropriate method to explain the model and gain insight into its internals.

In this paper, we set the Vision Transformer (ViT) [4], which is of the pure Transformer architecture without CNN components, as a benchmark. Tokens for ViT are a sequence of image patches, treated the same way as words in NLP tasks. Embeddings of tokens in ViT layers are the representations of the image content and global position information. Similar to NLP tasks, an additional [class] token is added in ViT for specifically the classification problem.

*Work done as an intern at Baidu Research.

1st Workshop on eXplainable AI approaches for debugging and diagnosis (XAI4Debugging@NeurIPS2021).

To explain Transformers, the simplest and most widely used method is to investigate and visualize the attention weights in the Transformer blocks [1, 5, 6]. Visualizing the relations between the [class] token and other tokens seems direct and reasonable. However, attention weights are not convincingly used as explanations for general attention-based models, proposed by some recent works. To cite a few, Pruthi et al. [7], Jain and Wallace [8] illustrated that there are limitations in using attentions as explanations, in terms of being incapable to provide fully faithful explanations concerning the model decisions. In this work, we do not directly equate attentions with explanations but utilize it to construct the information flow between the higher-level semantic information and the lower-level image information.

The simplest method to use attention weights for interpretability is to use raw attention scores [1, 5, 6, 9, 10] of the class token over input tokens and restore to the original size of the image as a saliency map. Abnar and Zuidema [11] raised the issue that the contextual information from tokens gets more similar as going deeper into the model, leading to unreliable explanations using raw attentions. To cope with this issue, the rollout method [11] was proposed to reassign the attribution scores to the tokens through the linear combination of attention weights along with the layers of the Transformer, to model the information flow in the deep model. Based on the idea of the rollout method, the attribution method [12] computed the relevance scores of the tokens with the layer-wise relevance propagation (LRP) [13], to visualize ViT's decision process.

Following the lines of the above methods, we continue to explore the information flow in the Transformer. In this work, we propose an approach named *Transition Attention Maps*, that relates the information flow in the Transformer to the Markov process, using the hidden state for tokens at each layer. These states start from an initial value and are recursively changing along with layers according to the Transformer's processing for tokens' representations. When a data sample passes through the self-attention-based blocks of the Transformer, traits are left in the computed attention scores. To track the traits and investigate the information flow, our approach considers these attention weights as *transition matrices* within the context of *the Markov chain* as they are naturally row-stochastic matrices, with each row summing to 1. Therefore, the proposed approach propagates the information from top to down and computes the relevance between high-level semantics and input features using transitions of states. Furthermore, to exhibit the class discriminative ability, the proposed approach combines the idea of attentions being transition matrices with Integrated Gradient [14] and Grad-CAM [15], to assign the importance scores w.r.t. the predicted category to input features.

Extensive experiments are conducted to demonstrate the trustworthiness and localization ability of our approach compared to other methods, using evaluations of quantitative metrics and visualizations. Three use cases based on the model interpretability with our approach are proposed to better understand the decision process of Vision Transformers.

2 Related work

We review the works that are related to the explanations and interpretations of deep models. Although there are many other directions to interpret the deep model, we mainly focus on the methods that generate heatmap or saliency maps highlighting the important part of input features on which the deep model mainly relies.

Some *gradient-based* methods are usually used to highlight the areas in the input related to the classification, while the vanilla gradients contain noises due to the saturation and artifacts of gradients in activation functions. Smilkov et al. [16] proposed to remove the noises by adding noises on the input and averaging the gradients w.r.t. noised inputs. Integrated Gradient [14] expects the contribution of non-zero gradients in the non-saturated region to the decision importance through integrating the gradients along different paths. More methods based on gradients w.r.t. inputs include DeepLIFT [17], GradientSHAP [18], etc.

To compute the importance scores of input features, several popular approaches were proposed without using gradients. LIME [19], Local Interpretable Model-agnostic Explanations, was proposed to fit an interpretable model with samples generated in the vicinity of the original data point, which will be used to explain the behaviors of the deep model in this local region. Layer-wise Relevance Propagation (LRP) methods [13, 20, 21] propagate the relevance scores from the output layer to the input features. Through using a set of purposely designed propagation rules, the contribution scores of input features for the final decision will be obtained eventually.

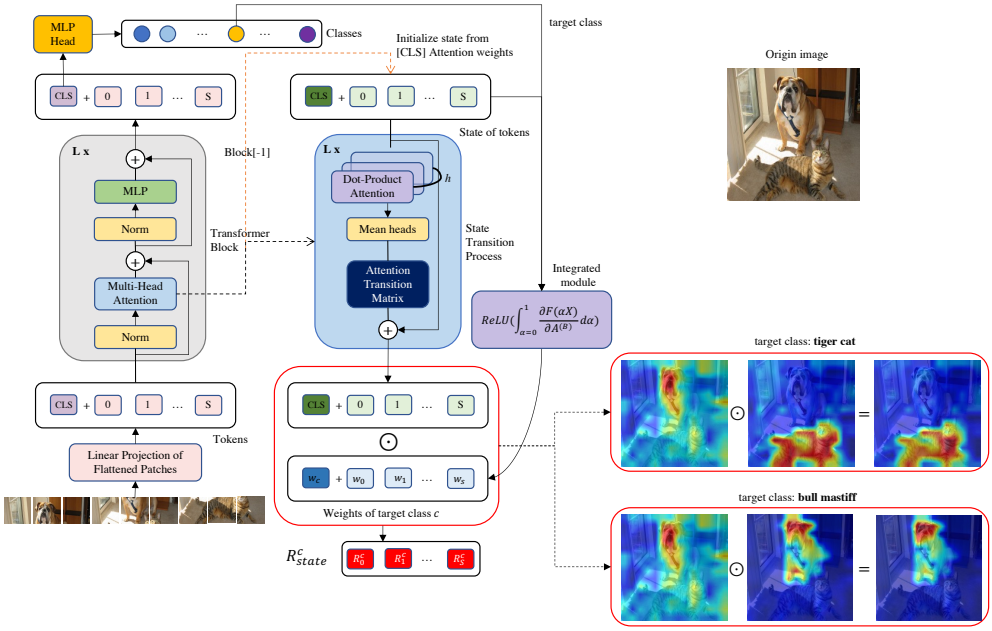


Figure 1: The pipeline of our proposed method *Transition Attention Maps*.

Some algorithms target intermediate features (or feature maps for CNN models). Class Activation Maps (CAM) [22] was proposed to highlight the important areas that contribute the most of model decisions. Based on CAM, many variants have been established to improve faithfulness and trustworthiness, such as Grad-CAM [15], Score-CAM [23], and others [24–27].

In terms of the Transformer interpretability, although some gradient-based and model-agnostic ones can be applied, there are limited approaches that are specifically designed for them. Voita et al. [28] identified the most important heads in each encoder layer using stochastic gates and a differentiable relaxation of the L_0 penalty to prune the least important heads. Hao et al. [29] proposed a self-attention attribution method to interpret the information interactions inside Transformer through constructing interaction trees. Abnar and Zuidema [11] proposed the rollout method that reassigns all attention scores by considering the pairwise attentions and assuming that attentions are combined linearly into subsequent contexts. However, we found that some common irrelevant features will also be highlighted through this approach. The Transformer attribution method [12] assigns local relevance scores based on LRP and propagates the relevance scores mixed with gradients through layers.

Attention modules are one of the most important blocks in the Transformer architecture, both for NLP and CV tasks. Investigations of attentions and explanations are capable of revealing much information about the decision process. In this work, we propose to explore the information flow inside ViT using Markov chains with the approach *Transition Attention Maps*.

3 Transition Attention Maps: Proposed Approach

In this section, we introduce the proposed explainability approach for Vision Transformers, to better explain the decision process of the model.

3.1 Markov Chain in Transformer

A Markov chain is a stochastic process describing a sequence of random variables X_l in which the probability of moving to the next state depends only on the current state. The single-step transition probability (transition matrix) between random variables in a Markov chain can be defined in the following form:

$$p_{ij}^{l,l+1} = P[X_{l+1} = j | X_l = i], \tag{1}$$

where the subscripts i,j are the indices of states, the l, $l+1$ indicates the step of the transition, and specifically, P is the transition matrix.

The ViT model divides an image into hundreds of patches and considers each patch as a token for input, with self-attention blocks to compute the representations of tokens. Considering the representations of output tokens at each block as states, which only depend on the input tokens, we can connect the decision process of ViT with the Markov process. We build the connection between the input and output tokens through computing the transitions of states using attention weights, which are the row-stochastic matrices:

$$\forall i, j : A_{i,j} > 0, \quad \forall i : \sum_j A_{i,j} = 1. \tag{2}$$

With attention weights as transition matrices for tokens, we can build an information flow from top to bottom or any intermediate hidden layer and establish a link between the abstract semantic features extracted by the model and the figurative information from the image, which can generate more comprehensible explanations.

3.2 Transitions of States Using Attention Weight Matrices

Classification tasks are performed by Transformers through the embeddings of an additional [class] token. Some works use the [class] attention weights of the last block to produce explanations, which show poor interpretability. We think it is the semantic gap between the higher and lower layers that causes this phenomenon. Therefore, we can leverage the state transition process mentioned above to get the relevance between model decisions and the image patches. Based on this motivation, we initialize the states of the Markov Chain $states^{(0)}$ to the relevance scores corresponding to the attention weights of the [class] token in the last block,

$$states^{(0)} = E_h[A_h^{(B)}](class) \in \mathbb{R}^{1\times s}, \tag{3}$$

where E_h is the expectation across multiple *heads* in the attention module, we treat the multiple *heads* equally, that is, averaging for multiple *heads*. $(class)$ is an index representation of the row corresponding to the [class] token, B is the index of the last block and s is the total number of tokens.

Attention weights in Transformer blocks indicate the correlation between tokens, where the tokens include both image patches and the [class] token. We consider attention weights in each Transformer block $A^{(b)}$ as the state transition matrix for the Markov chain.

Residual connections in Transformer: However, the residual connections in Transformer are not presented by attention weights. To incorporate with the residuals, we add an identity transition (i.e., no transition) as residuals to the transition matrix, with the following recursive formula:

$$states^{(i)} = \begin{cases} E_h[A_h^{(B)}](class) & \text{if } i = 0 \\ \frac{1}{2} states^{(i-1)} + \frac{1}{2} states^{(i-1)} \cdot \hat{A}^{(B-i)} & \text{otherwise} \end{cases}, \tag{4}$$

where $\hat{A}^{(B-i)} = E_h[A_h^{(B-i)}]$ is the expectation across heads, indicating the equal treatment in multiple heads, $B-i \in \{l_{end}, ..., B\}$. We remark that each block in Transformer has a corresponding $state^{(i)}$ describing the information flow propagated to the current layer. Note that we set a hyper-parameter l_{end} to early stop the transitions for the reasons that the first layers are supposed to extract local features, which contribute less to locate the high-level semantics for predictions. Without early stopping using l_{end}, our proposed approach produces better explanations than the rollout [11] and attribution [12] methods. We have ablation studies and provide thorough analyses on this hyper-parameter in Section 4.1.

3.3 Integrated Gradient for Patch-Level Importance

Since some noise and irrelevant features will be introduced during the transition process, we need to use some methods to eliminate or diminish them. Simonyan et al. [30] illustrated that gradients are a natural analog of the model parameters for deep networks, and Chefer et al. [12] proposed a better explanation algorithm using the product of gradients and feature attributions. Furthermore, Sundararajan et al. [14] suggested that an attribution method should satisfy the Sensitivity Axiom because the lack of sensitivity causes gradients to focus on irrelevant features practically. Inspired

by their work, we use the Integrated Gradient to get the relevant features from the gradients of the attention module, and translate them with the weights for the states of the Markov chain to get the class discriminative explanations. The reason for using integrated gradients rather than gradients is that the integration process effectively retains the relevant parts and reduces the gradient self-induced noise. By weighting the states, some noise in the model amplified by applying the transition process will also be reduced or eliminated.

Following [14], we set the baseline to $\mathbf{0}$, i.e., a black image, and assume the path from the baseline to input image X is a straight line. We compute the gradient of the output with respect to the last attention module and get the attribution scores by accumulating these gradients from baseline to input. Finally, the integration is discretized by the Riemann approximation:

$$W^c_{states} = ReLU(\frac{1}{m}\sum_{k=1}^{m}\frac{\partial \mathrm{F}^c(\frac{k}{m}X)}{\partial A^{(B)}}) \rightarrow ReLU(\int_{\alpha=0}^{1}\frac{\partial \mathrm{F}^c(\alpha X)}{\partial A^{(B)}}d\alpha), \tag{5}$$

where W^c_{states} represents the weights for the states of the category c, $\mathrm{F}^c(\cdot)$ is the objective function when c is the target class, and $\frac{\partial \mathrm{F}^c(\alpha X)}{\partial A^{(B)}}$ is the gradient of model $\mathrm{F}^c(\cdot)$ w.r.t. $A^{(B)}$. As α gradually increases, the gradients corresponding to the truly relevant and significant features will accumulate to large values. Note that only the positive part of the attributions is taken for removing noises, following [14, 15], and that m is set to 20, which works well in practice.

3.4 Transition Attention Maps for Transformer Interpretability and Visualizations

In summary, we propose the approach named *Transition Attention Maps*, to explain the information flow inside the Vision Transformers. This approach considers the information flow inside ViT as a Markov process and uses the states of tokens to denote the information at each layer. The transitions of states can be described by the matrices of attention weights, which satisfy the definition of matrices of transitions. This approach thus tracks the information flow using the attention weights and computes the attributions scores of tokens for the model's decisions. For obtaining class discriminative explanations, we further combine the gradient information using Integrated Gradients with the states of tokens. This approach exploits the information flow through the attentions and is able to explain the contributions of tokens with respect to the predictions. The pipeline of the proposed approach is illustrated in Figure 1 with the pseudocode described in Algorithm 1. More implementation details are referred to the repository: https://github.com/XianrenYty/Transition_Attention_Maps.git.

Algorithm 1: Transition Attention Maps

Input: Input Image X
Output: R^c_{state}
Parameters: l_{end}, m(steps) ;
Initialization: ;
$states \leftarrow E_h[A_h^{(B)}](class) \in \mathbb{R}^{1\times s}$;
$i \leftarrow 1$;
$l \leftarrow B - i$; // $l \in [1, B]$:block_index
while $l > l_{end}$ **do**
  $states_{res} \leftarrow states$;
  $\hat{A}^{(l)} \leftarrow E_h[A_h^{(l)}]$;
  $states \leftarrow states \cdot \hat{A}^{(B-i)}$;
  $states \leftarrow \frac{1}{2}states + \frac{1}{2}states_{res}$;
  $i \leftarrow i + 1$;
  $l \leftarrow B - i$;
end
$W^c_{states} \leftarrow ReLU(\frac{1}{m}\sum_{k=1}^{m}\frac{\partial \mathrm{F}^c(\frac{k}{m}X)}{\partial A^{(B)}})$;
$R^c_{state} \leftarrow W^c_{states} \odot states$;
Return R^c_{state}

4 Trustworthiness Evaluations

To assess the trustworthiness of explainability algorithms, we quantitatively evaluate the proposed approach with comparisons to the state-of-the-art algorithms through the commonly-used evaluation metrics. For implementation details, we experiment with the pretrained ViT-B/16 model [4], which means "Base" variant (Layers:12, Hidden size:768, MLP size:3072, Heads:12) with non-overlaping 16×16 input patch size. The input images are resized to 224×224. The evaluation results for more ViT variations are shown in the Appendix F. The code for evaluations is available at: https://github.com/XianrenYty/Transition_Attention_Maps.git.

Table 1: Evaluation results of ablation-study experiments.

	Faithfulness Evaluation				Localization Evaluation		
	Deletion & Insertion		Perturbation test		Segmentation		
Variants	Del.	Ins.	Pos.	Neg.	Acc.	mIoU	mAP
Original (l_{end}=0)	**13.07**	**61.92**	**20.30**	**59.94**	**76.21**	**58.54**	**85.29**
w/o Integrated	13.96	60.63	21.36	57.97	71.32	53.08	83.06
w/o transition	15.91	57.92	22.37	56.30	77.29	52.50	84.44
different l_{end}	13.10(1)	62.79(1)	20.03(1)	60.83(1)	78.71(1)	61.40(1)	86.18(1)
	12.95(2)	62.96(2)	19.83(2)	61.20(2)	80.19(2)	62.61(2)	86.29(2)
	12.91(3)	63.00(3)	19.73(3)	61.35(3)	81.02(3)	63.05(3)	86.19(3)
	12.77(4)	**63.23(4)**	**19.67(4)**	**61.53(4)**	**81.57(4)**	**63.37(4)**	**86.30(4)**
	12.74(5)	63.00(5)	19.77(5)	61.30(5)	81.40(5)	62.72(5)	85.96(5)
	12.46(6)	62.40(6)	19.96(6)	60.93(6)	80.13(6)	59.89(6)	84.69(6)
	12.72(7)	62.79(7)	20.31(7)	59.76(7)	76.61(7)	58.44(7)	83.71(7)
	13.31(8)	61.36(8)	20.88(8)	58.56(8)	78.13(8)	55.43(8)	83.25(8)
	13.69(9)	60.52(9)	21.28(9)	58.56(9)	77.09(9)	52.63(9)	83.28(9)
	15.14(10)	59.11(10)	22.23(10)	57.14(10)	75.87(10)	49.21(10)	83.20(10)
	16.24(11)	57.74(11)	22.58(11)	56.59(11)	77.29(11)	52.50(11)	84.44(11)

Deletion & Insertion metrics [31]. The *deletion* metric measures a decrease in the probability of the predicted class as important pixels are sequentially removed, where the importance score is obtained from the explanations. A sharp drop and thus a low area under the probability curve (AUC) score indicates a good explanation. The *insertion* metric, on the other hand, takes a complementary approach. It measures the increase in probability as important pixels are sequentially introduced, with higher AUC scores indicate a better explanation.

Perturbation Tests [32, 33]. The idea of *perturbation* metric is similar to the *deletion* and *insertion* metrics. After obtaining the explanation results, we gradually mask out the pixels of the input image and measure the mean top-1 accuracy. In the positive perturbation, pixels are masked from the highest relevance to the lowest, and one expects to see a steep decrease in performance, which indicates that the masked pixels are important to the classification score. In the negative version, pixels are removed from lowest to highest, and 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 AUC scores, for erasing between $0\% - 90\%$ of the pixels.

Segmentation. The segmentation tests consider each visualization as a semantic segmentation of the image and compare it to the ground truth segmentation of the ImageNet-Segmentation dataset [34]. Performance is measured by (i) pixel accuracy, (ii) mean-intersection-over-union (mIoU), and (iii) mean-Average-Precision (mAP). Note that (i) and (ii) are obtained after binarizing each visualization, which depends on the pre-set threshold (30% of the max value is used in practice [12]), while (iii) is threshold-free.

Energy-based Pointing Game. Localization evaluations using bounding-box-based annotations [23, 35] may be less accurate than segmentation tests, because the bounding box does not describe the outline of the objects. To give a complete comparison, we have conducted such experiments and report the performance of our proposed approach in comparison with baselines in the Appendix A.

4.1 Ablation Study

We provide the ablation-study experiments to validate the effectiveness of the proposed approach from three aspects: (i) integrated gradients; (ii) the transitions of states; (iii) the affects of l_{end} in the proposed approach.

(i) Gradients. Instead of vanilla gradients, we adopt the integrated gradients for removing the spurs and noises in vanilla gradients [14] in our proposed approach. With the metrics previously introduced, we conduct the comparison experiments using these two gradients separately and obtain the results in the first two rows of Table 1, which show the significant improvement for both faithfulness and localization evaluations.

(ii) Transitions of States. Our proposed approach takes the states of tokens in each block of the Transformer and computes their transitions using attention weights, for eventually explaining the

Table 2: Comparison in terms of deletion (lower is better) and insertion (higher is better) scores.

Methods	raw attention	rollout[11]	attribution[12]	Ours	Ours(l_{end}=4)
Deletion	21.83	16.57	14.23	13.08	**12.77**
Insertion	50.82	59.47	60.02	61.92	**63.23**

Table 3: Comparison in terms of Positive (lower is better) and Negative (higher is better) perturbation AUC scores.

Methods	raw attention	rollout[11]	attribution[12]	Ours	Ours(l_{end}=4)
Positive	28.04	24.10	21.06	20.30	**19.67**
Negative	49.72	57.79	58.73	59.94	**61.53**

decision processing. We design the experiments that do not use the transitions of states to compare with the original approach. As shown in Table 1, the third row takes the same initial state as our approach but does not perform the transition process. We compare this setting with our approach to show the effectiveness of using transitions of states to present the information flow in the Transformer. The experiment results support our claims.

(iii) l_{end}**.** The rest rows in Table 1 investigate the affect of l_{end} on the states of tokens. The hyper-parameter l_{end}, as mentioned previously, controls an early stopping for the transition and propagation. Note that $l_{end} = 0$ indicates no early stopping. Intuitively, the more layers of propagation the richer the set of information. However, the algorithm works best in most cases when $l_{end} = 4$. The reason for this may be due to the fact that the information near the input is mainly the information of common features, contributing little to the trustworthiness assessment experiment.

4.2 Evaluation Results: Comparisons with SOTA

The proposed approach produces better explanations than currently state-of-the-art algorithms [11, 12]. We conduct experiments for evaluating the trustworthiness of generated explanations through transitions of states with attention weights and comparing with the SOTA approaches. The evaluations are quantitatively measured by the metrics presented previously.

Deletion & Insertion [31]. Table 2 shows results of the *deletion* and *insertion* scores obtained by the baselines and our approaches. Our approach significantly outperforms the previous approaches. Moreover, with setting $l_{end} = 4$ for an early stopping of transitions, the performance can be further improved. Deletion & Insertion evaluations measure the trustworthiness of explanation results to the model. These results demonstrate that our approach produces better results.

Perturbation Tests [32, 33]. Table 3 shows the results of the *perturbation* metric. The experiments' result is similar to Table 2, which quantitatively demonstrates the effectiveness of our proposed method.

Segmentation [34]. Segmentation tests measure the alignment between the segmentation ground truth and the explanation results. This is a simple way to visualize the important input features and explore the reasons behind the decision-making process of the Transformer-based models. Table 4 shows the results of segmentation tests. Our approach gets the best mIoU scores compared to other baselines, and comparative performance in the other two metrics. However, with $l_{end} = 4$, our approach achieves the best performance in all segmentation tests.

5 Use Cases

5.1 Visualizing Model Decisions

Selvaraju et al. [15] proposed that a *good* visual explanation of the model should be (a) class-discriminative (i.e. localize the category in the image) and (b) high-resolution (capture fine-grained details). The most direct application of explainability algorithms is to determine whether the basis for the model decision is reasonable. Many samples contain more than one category, the explainability algorithms should be able to highlight the basis on which the model classifies that image as a specific class. As shown in Figure 2, the rollout method [11] is not class-discriminative but highlights

Table 4: Comparison concerning Segmentation performance (higher is better) on the ImageNet-segmentation dataset [34].

Methods	raw attention	rollout[11]	attribution[12]	Ours	Ours(l_{end}=4)
Pixel accuracy	67.42	74.95	79.12	76.21	**81.57**
mIoU	38.16	57.17	55.70	58.54	**63.37**
mAP	80.24	84.76	86.03	85.29	**86.30**

fine-grained details in the images. The attribution method [12] can extract category-related features and part of the target region but is not comprehensive and complete. Our approach can discriminate multiple categories, not only highlights the entire region of the target category but also highlight the stripes and something important for predicting the target category. More samples can be found in the Appendix B.

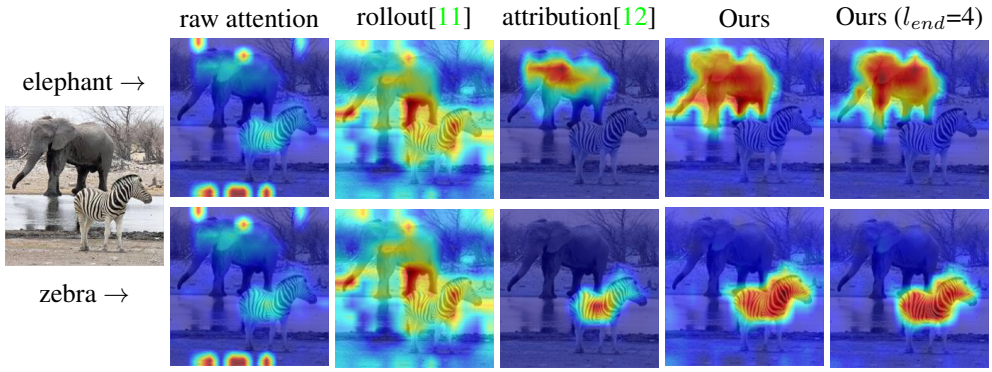


Figure 2: Illustration of class discriminative ability by different methods.

On the other hand, the algorithm should be able to highlight the complete region and fine-grained features of the specific category, regardless of whether there are single or multiple target objects in the image, as Figure 3 shows. More samples can be found in the Appendix C.

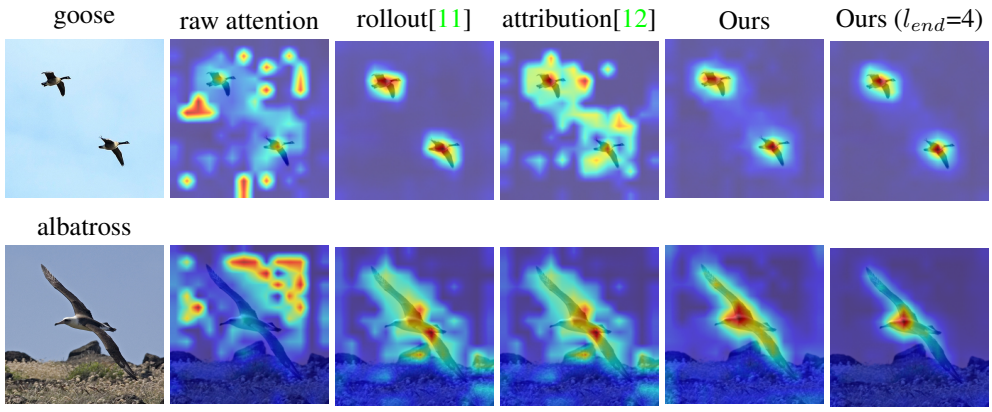
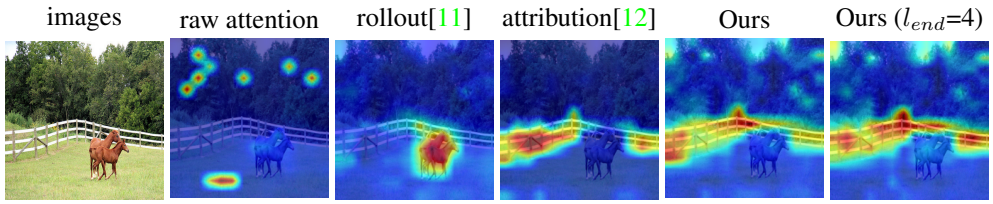


Figure 3: Localization of fine-grained features and object regions for a single-class image by different methods.

5.2 Explaining the Misclassification and Debugging the Model

One main objective for the research in model interpretability is to understand/debug why the model makes unexpected decisions and to improve the model with explanations. In the image classification task, we can figure out why the model misclassifies the samples and find out whether the problem is caused by the confusing labeling of the sample or the model itself. Figure 4 shows us one misclassification sample, visualized by different methods.




```
Target label: 339
Top 5 classes:
    912 : worm fence, snake fence, snake-rail fence, Virginia fence     value = 12.663  prob = 93.7%
    339 : sorrel                                                        value = 9.824  prob = 5.5%
    716 : picket fence, paling                                          value = 6.884  prob = 0.3%
    425 : barn                                                          value = 5.168  prob = 0.1%
    355 : llama                                                         value = 5.155  prob = 0.1%
```

Figure 4: Illustration of explaining the model misclassification by different methods.

Below the images, we can see the target label, the top 5 predictions and probabilities given by the model. The target label is provided by ImageNet [36], which is 'sorrel' while the predicted label is 'worm fence' for the given image. We can see that both the attribution method [12] and ours can localize the 'worm fence' part well, explaining why the model made this decision. The rollout [11] method only highlights some fine-grained details in the image but is not relative to the model's decision. More samples in the Appendix D show the same conclusion.

5.3 Weakly-Supervised Semantic Segmentation

In the weakly-supervised setting, the dataset consists of images and corresponding annotations that are relatively easy to obtain, such as tags/labels of objects present in the image. Weakly supervised semantic segmentation (WSSS) with image-level labels has been widely studied because image-level labels are much less costly than pixel-level labels. A good visual explanation should have high-resolution (capture fine-grained detail) which embodies the localization ability of the networks, which is also a great benefit for WSSS.

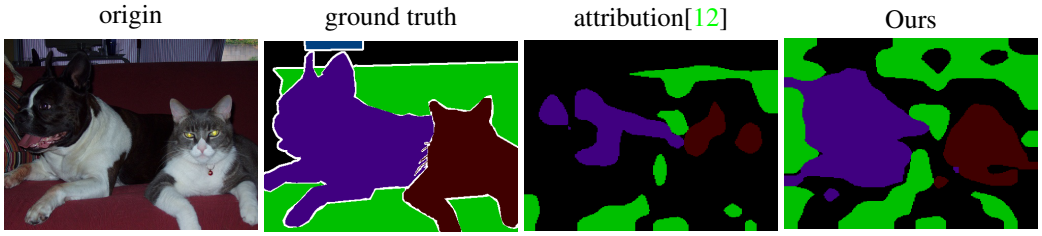
Most WSSS methods are CNN-based and use CAMs [22] to obtain segmentation masks using image-level supervision. The study on WSSS based on Transformer explainability methods is scarce. In our work, we follow the setting in [37] to train the ViT-B/16-224 [4] classifier with image-level labels on the PASCAL VOC dataset [38], which contains 20 semantic classes and the background, and is split into 1464 training images, 1449 validation images, and 1456 test images. We leverage our explainability approach to generate the segmentation masks of each class and set a threshold to generate the background mask. The segmentation map is obtained from the probability of these masks.

Since the raw attention and rollout [11] methods are not class discriminative, we only use the attribution method [12] as a comparison. The visualizations of the segmentation maps and qualities in IoU score are shown in Figure 5. The performance of our method in terms of mIoU scores on the PASCAL VOC *val* (*test*) set is 49.58% (51.28%) better than 31.92% (32.77%) of the attribution method [12].

6 Conclusions

In this work, we propose a novel approach, *Transition Attention Maps*, to explain the information flow through Vision Transformers. Instead of directly visualizing the attention weights, this approach addresses the semantic gap between the top and bottom layers. Further, we obtain more accurate category regions by accumulating the gradient information of the target categories.

To assess the trustworthiness of explainability methods, we performed quantitative evaluations of faithfulness and localization on the ImageNet validation set. As shown by the experimental results, our approach can better display the features, contours, and multi-category information of the objects of the specified category compared with state-of-the-art explainability approaches.



IoU(**dog**)	→	27.93	**66.07**
IoU(**cat**)	→	16.71	**51.01**
IoU(**sofa**)	→	34.79	**37.25**

Figure 5: Visualization of predicted weakly-supervised semantic segmentation results from the attribution approach and ours.

Our approach does not require any model changes and can be directly applied to the Transformer-based models. The interpretability provided by our approach gives an efficient and effective way to debug and improve the Transformer-based models. Furthermore, we show some interpretability use cases with examples of our approach. Based on these applications, we can explore more aspects of Transformer-based models in the field of computer vision.

References

[1] 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.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[4] 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.

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

[6] Zijie J Wang, Robert Turko, and Duen Horng Chau. Dodrio: Exploring transformer models with interactive visualization. *arXiv preprint arXiv:2103.14625*, 2021.

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

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

[9] Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*, 2019.

[10] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

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

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

[13] 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.

[14] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[15] 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.

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

[17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the International Conference on Machine Learning*, 2017.

[18] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. 2017.

[19] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[20] 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.

[21] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019.

[22] 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.

[23] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020.

[24] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *Proceedings of the Winter Conference on Applications of Computer Vision*, 2018.

[25] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 2021.

[26] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldermariam. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv preprint arXiv:1908.01224*, 2019.

[27] Desai Ramaswamy and Guruprasad Harish. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the Winter Conference on Applications of Computer Vision*, 2020.

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

[29] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. *arXiv preprint arXiv:2004.11207*, 2020.

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

[31] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[32] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

[33] Minh N. Vu, Truc D. T. Nguyen, N. Phan, Ralucca Gera, and M. Thai. Evaluating explainers via perturbation. *ArXiv*, abs/1906.02032, 2019.

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

[35] 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.

[36] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[37] Sanghyun Jo and In-Jae Yu. Puzzle-cam: Improved localization via matching partial and full features. *arXiv preprint arXiv:2101.11253*, 2021.

[38] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 2010.

[39] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of International Conference on Machine Learning*, 2021.

A Results of the Energy-based Pointing Game

Energy-based Pointing game [35] extracts maximum point in the saliency map to see whether the maximum falls into the object bounding box. Instead of using only the maximum point, they compute the quantity of energy concerning the saliency map that falls into the target object bounding box. The metrics for the energy-based pointing game can be denoted as $Precision = \frac{\sum L^c_{(i,j)\in bbox}}{\sum L^c_{(i,j)\in bbox} + \sum L^c_{(i,j)\notin bbox}}$. We complete this evaluation by adding recall ($Recall = \frac{\sum L^c_{(i,j)\in bbox}}{\sum P_{(i,j)\in bbox}}$) and F1 score. Evaluations are done on the ImageNet validation set.

Table 5 shows that the rollout method [11] leads to a higher recall due to a smoother distribution, while the attribution method [12] highlights significant areas, resulting in a low recall. Our approach gets results that are comparable to the attribution method in terms of precision, but with a higher recall score.

Table 5: Comparison in terms of Energy-based Pointing Game (higher is better) on the ImageNet validation set [36].

Methods	raw attention	rollout[11]	attribution[12]	Ours	Ours(l_{end}=4)
Precision	53.09	54.07	**55.83**	55.26	55.76
Recall	3.11	**20.17**	1.26	9.43	5.62
F1 score	2.96	**25.17**	3.12	14.47	9.35

B Visualization of Class Discriminative Performance for different methods

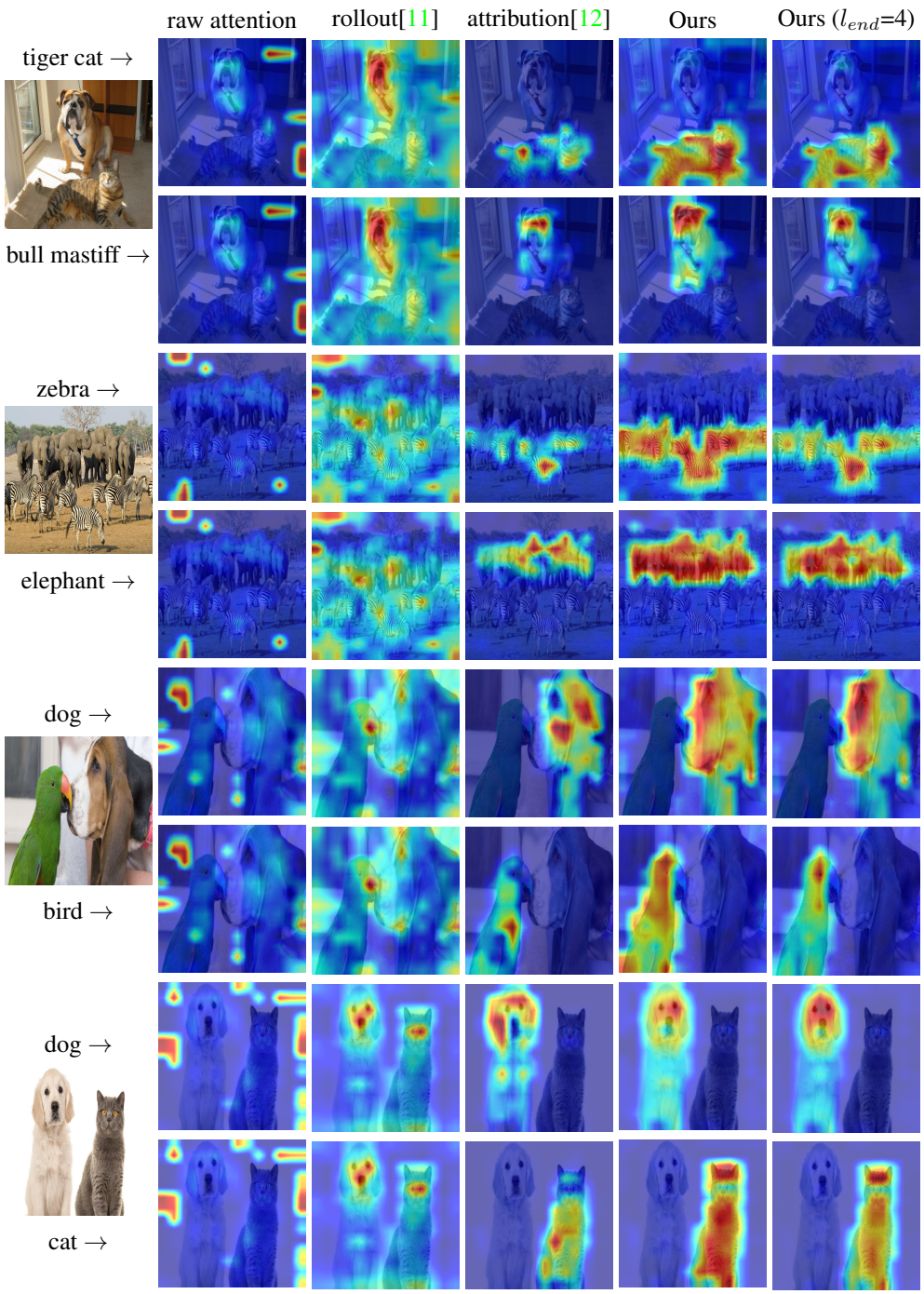


Figure 6: Performance of Class Discriminative Ability for different methods

C Visualization of Localization Performance for different methods

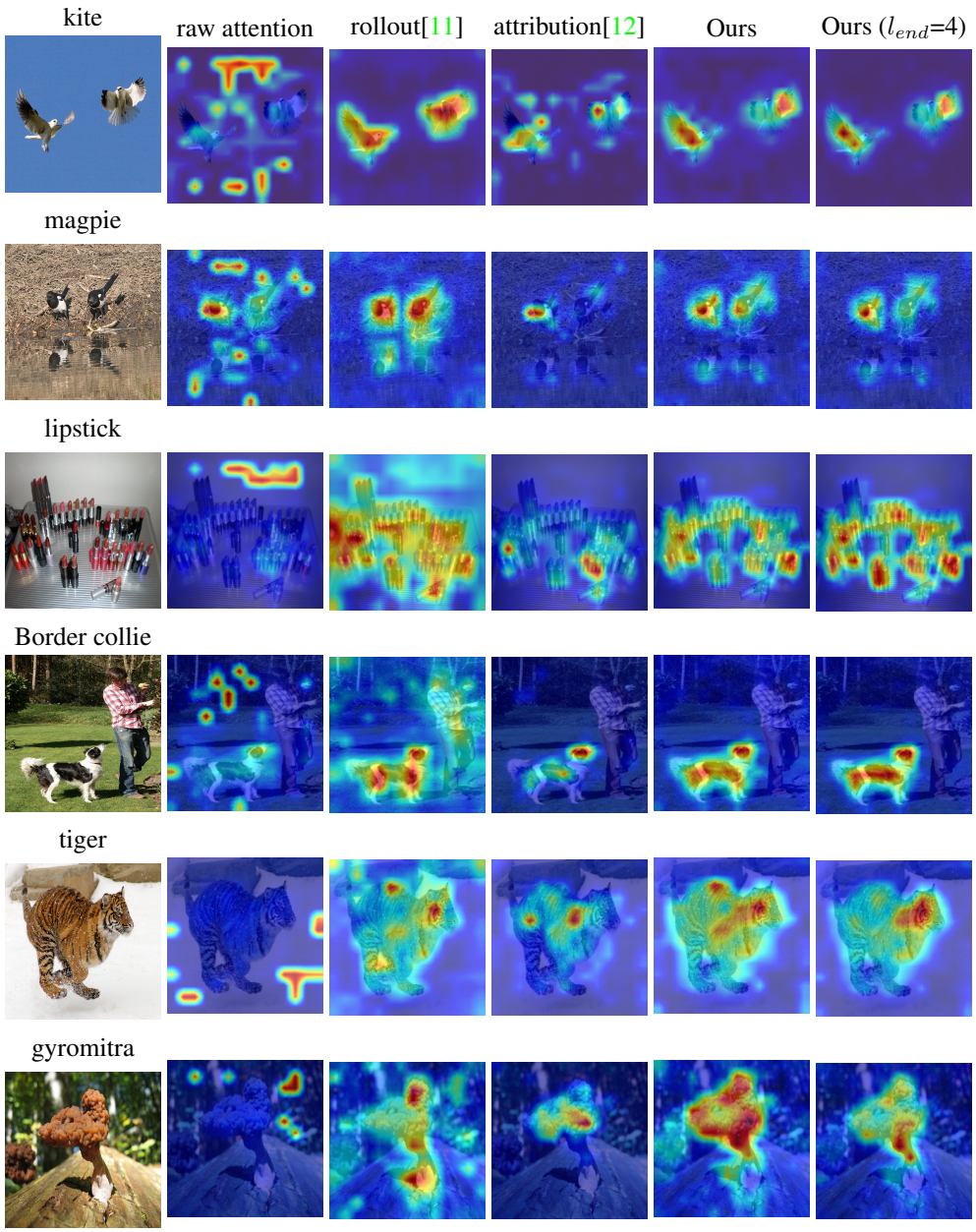
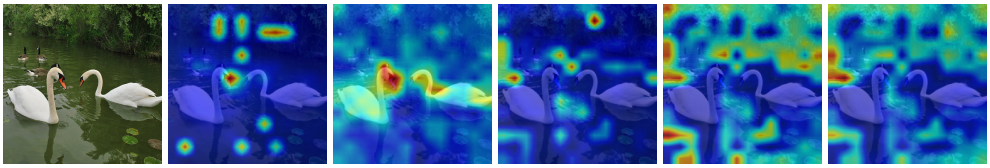
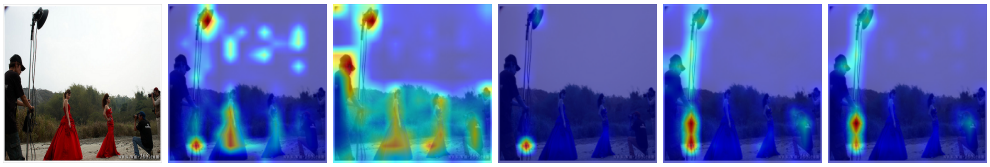


Figure 7: Localization of Fine-grained feature and Object Region of a single class for different methods

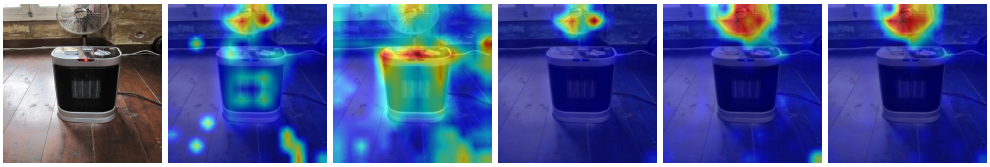
D Visualization of Explanation of misclassification for different methods



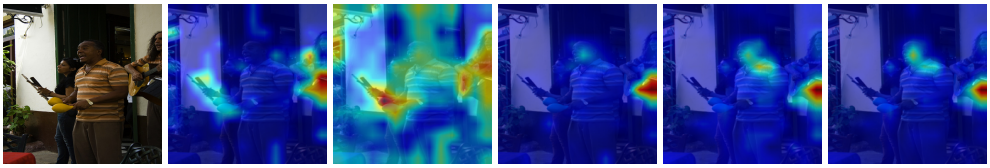
```
Target label: 99
Top 5 classes:
    975 : lakeside, lakeshore                                   value = 9.200  prob = 50.3%
    100 : black swan, Cygnus atratus                            value = 8.768  prob = 32.7%
    99 : goose                                                  value = 7.360  prob = 8.0%
    130 : flamingo                                              value = 4.868  prob = 0.7%
    449 : boathouse                                             value = 4.053  prob = 0.3%
```



```
Target label: 578
Top 5 classes:
    872 : tripod                                                value = 8.563  prob = 44.8%
    578 : gown                                                  value = 7.549  prob = 16.2%
    862 : torch                                                 value = 5.799  prob = 2.8%
    447 : binoculars, field glasses, opera glasses              value = 5.543  prob = 2.2%
    843 : swing                                                 value = 5.404  prob = 1.9%
```



```
Target label: 811
Top 5 classes:
    545 : electric fan, blower                                  value = 14.275  prob = 87.9%
    811 : space heater                                          value = 12.259  prob = 11.7%
    827 : stove                                                 value = 7.756  prob = 0.1%
    753 : radiator                                              value = 6.547  prob = 0.0%
    556 : fire screen, fireguard                                value = 5.082  prob = 0.0%
```



```
Target label: 641
Top 5 classes:
    822 : steel drum                                            value = 7.103  prob = 21.9%
    641 : maraca                                                value = 6.016  prob = 7.4%
    762 : restaurant, eating house, eating place, eatery        value = 5.652  prob = 5.1%
    860 : tobacco shop, tobacconist shop, tobacconist           value = 4.946  prob = 2.5%
    577 : gong, tam-tam                                         value = 4.887  prob = 2.4%
```

Figure 8: Performance of Explaining Model Misclassification for different methods

E Visualization of Segmentation results for different methods

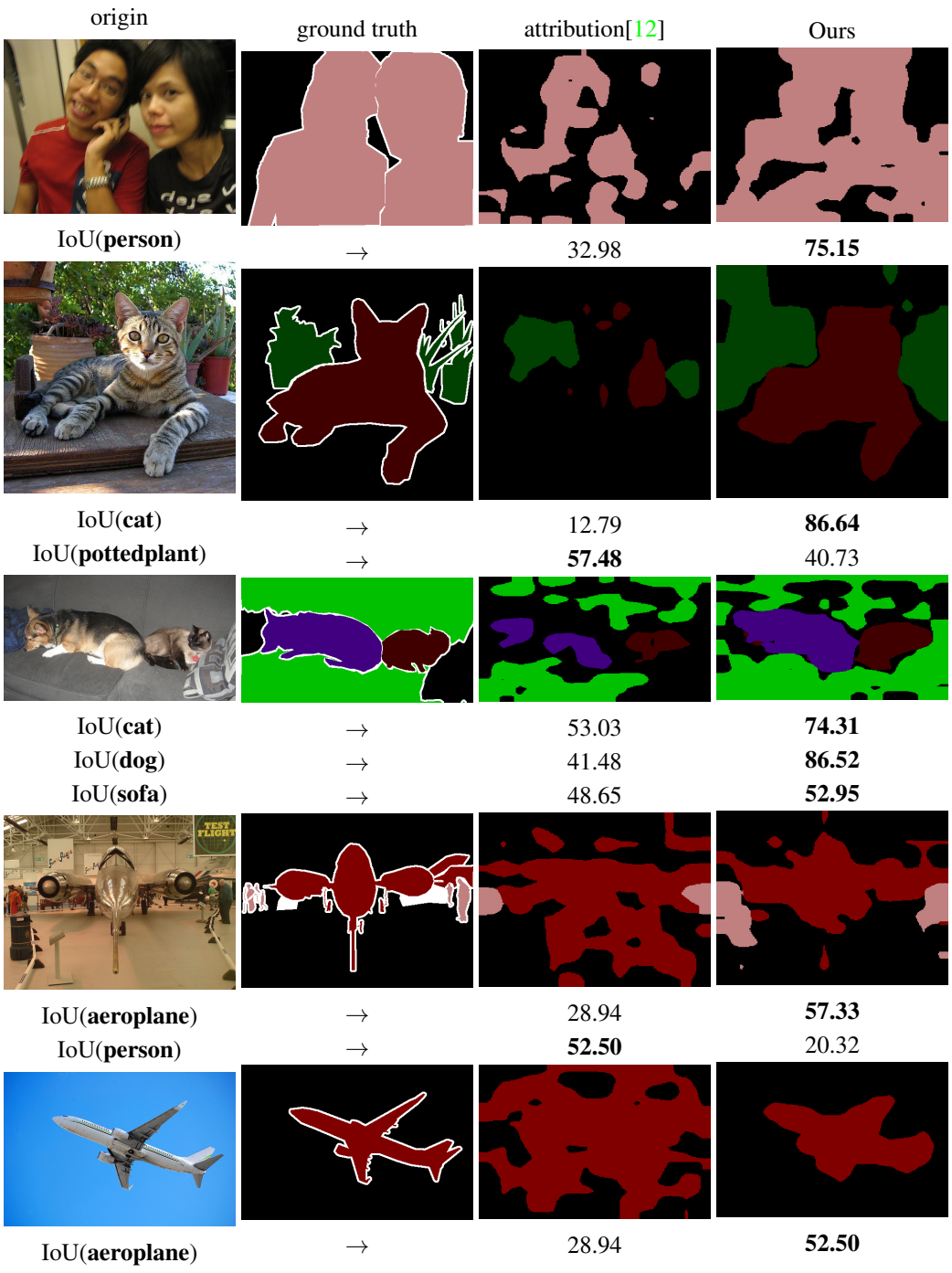


Figure 9: Visualization of predicted weakly-supervised semantic segmentation results from the attribution approach and ours.

F Results on ViT variations

We furthermore evaluate our approach on other variants of ViT (ViT-L/16-224, ViT-B/16-384) [4] and Data-efficient image Transformer (DeiT) [39], compared with SOTA approaches. The details of Vision Transformer model variants are shown in Tabel 6.

The experiments on deletion & insertion metrics and segmentation tests are conducted to assess the trustworthiness and localization ability of explainability algorithms respectively. Results in Tables 7 and 8 show that our approach outperforms others.

Table 6: Details of ViT variations

Model	Layers	Hidden size D	MLP size	Heads	Patch_size	Img_size
ViT-B/16-224	12	768	3072	12	16	224
DeiT-B/16-224	12	768	3072	12	16	224
ViT-L/16-224	24	1024	4096	16	16	224
ViT-B/16-384	12	768	3072	12	16	384

Table 7: Comparison in terms of deletion (lower is better) and insertion (higher is better) scores.

(i) DeiT-B/16-224

Methods	raw attention	rollout[11]	attribution[12]	Ours(l_{end}=4)
Deletion	21.05	26.39	9.38	**8.82**
Insertion	25.48	25.44	36.14	**36.95**

(ii) ViT-L/16-224

Methods	raw attention	rollout[11]	attribution[12]	Ours(l_{end}=4)
Deletion	27.14	20.97	19.37	**15.39**
Insertion	48.80	58.01	58.30	**62.71**

(iii) ViT-B/16-384

Methods	raw attention	rollout[11]	attribution[12]	Ours(l_{end}=4)
Deletion	25.59	17.69	16.32	**14.28**
Insertion	60.50	69.34	69.79	**71.54**

Table 8: Comparison concerning Segmentation performance (higher is better) on the ImageNet-segmentation dataset [34].

(i) DeiT-B/16-224

Methods	raw attention	rollout[11]	attribution[12]	Ours(l_{end}=4)
Pixel accuracy	65.77	46.43	**80.09**	75.89
mIoU	34.78	29.48	57.81	**58.89**
mAP	76.51	67.98	**86.07**	85.98

(ii) ViT-L/16-224

Methods	raw attention	rollout[11]	attribution[12]	Ours(l_{end}=4)
Pixel accuracy	67.37	54.26	74.60	**76.81**
mIoU	37.68	37.18	49.33	**59.00**
mAP	74.75	79.18	81.24	**84.42**

(iii) ViT-B/16-384

Methods	raw attention	rollout[11]	attribution[12]	Ours(l_{end}=4)
Pixel accuracy	67.75	68.63	80.19	**82.02**
mIoU	40.80	50.56	58.17	**65.14**
mAP	80.13	81.92	**85.97**	85.79