Visual Explanation Generation Based on Lambda Attention Branch Networks

Tsumugi Iida[1], Takumi Komatsu[1], Kanta Kaneda[1], Tsubasa Hirakawa[2], Takayoshi Yamashita[2], Hironobu Fujiyoshi[2], and Komei Sugiura[1]

[1] Keio University
{tiida,tak3k_1999,k.kaneda,komei.sugiura}@keio.jp
[2] Chubu University
{hirakawa,takayoshi,fujiyoshi}@isc.chubu.ac.jp

Abstract. Explanation generation for transformers enhances accountability for their predictions. However, there have been few studies on generating visual explanations for the transformers that use multidimensional context, such as LambdaNetworks. In this paper, we propose the Lambda Attention Branch Networks, which attend to important regions in detail and generate easily interpretable visual explanations. We also propose the Patch Insertion-Deletion score, an extension of the Insertion-Deletion score, as an effective evaluation metric for images with sparse important regions. Experimental results on two public datasets indicate that the proposed method successfully generates visual explanations.

Keywords: Lambda Networks · transformer · attention.

1 Introduction

Visual explanations for deep neural networks are important in terms of enhancing accountability about biomedical image processing and providing scientific insight to experts. Specifically, in the task of predicting solar flares, for which the theoretical background remains unclear, visual explanations using magnetograms can provide scientists with insights into underlying solar activities.

In this paper, we focus on the task of visualizing important regions in an image as a visual explanation of the model's decisions. In this task, pixels that contributed to the model's prediction should be attended.

Explanation generation for convolutional neural networks has been studied intensively in recent years [28, 34]. On the other hand, there have been few studies on generating visual explanations for transformers, especially those based on Lambda [3]. In addition, standard metrics for visual explanations (e.g. the Insertion-Deletion score [24]) are sometimes inappropriate for images with sparse important regions.

Given this background, we propose the Lambda Attention Branch Networks (LABN), which generates interpretable visual explanations for Lambda-based transformers. In addition, we introduce the loss used in saliency guided training [12] to reduce the importance of regions irrelevant to the prediction. Fig. 1 shows

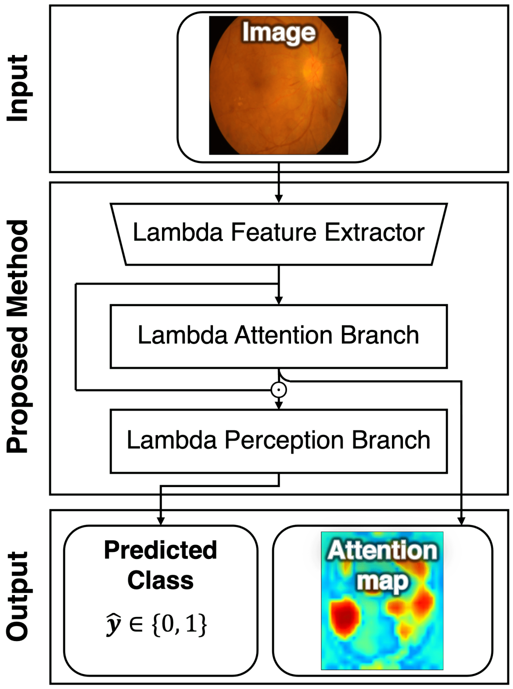


Fig. 1. Overview of our method.

an overview of our method. It is composed of three modules: the Lambda Feature Extractor (LFE), Lambda Attention Branch (LAB), and Lambda Perception Branch (LPB). Attention in the original Lambda layer [3] is sometimes not clear as visual explanation. On the one hand, we obtain a clear visual explanation by introducing a branch structure dedicated to visual explanation generation.

We also propose the Patch Insertion-Deletion (PID) score, an extension of the Insertion-Deletion score, as an effective evaluation metric for images with sparse important regions. Unlike the Insertion-Deletion score, the PID score evaluates visual explanations in a patch-wise manner.

The main contributions of this study are as follows:

- We propose the LABN, which has a parallel branching structure to obtain clear visual explanations than those provided by attention in the Lambda Layer.
- We propose the PID score, which is an extension of the Insertion-Deletion score, as an effective evaluation metric for images with sparse important regions.
- We introduce the loss used in saliency guided training to improve the quality of the visual explanations.

2 Related Work

There have been many studies in the field of explanation generation [4, 5, 10, 22, 24, 28, 34]. [6] is a comprehensive survey paper in this field that categorizes the methods according to their approach. [14] summarizes the characteristics of vision transformer models for various tasks. In the field of visual explanation generation, standard image classification datasets are used (e.g., ImageNet [7], CIFAR10, CIFAR100 IDRiD [25]).

Explanation generation methods can be classified into backpropagation methods, perturbation methods, and other methods. Backpropagation methods generate explanations by focusing on the gradient during backpropagation; Integrated Gradients [31], SmoothGrad [29], FullGrad [30], CAM [37], Grad-CAM [28], PatternNet [15], and LRP [4, 5] are typical backpropagation methods. For example, [31] is a method that satisfies the two axioms of Sensitivity and Implementation Invariance. It integrates the gradient to generate an explanation. The authors of [29] pointed out that gradient-based explanations are often noisy and proposed an averaging method to reduce the noise. Reference [12] reduces noise using saliency guided training, which brings the gradients of less important regions closer to zero. Reference [30] theoretically proved that the two axioms that the explanation should satisfy do not hold simultaneously. It then proposed a generation method that can balance both axioms.

By contrast, perturbation methods generate explanations from changes in the output when the input is perturbed; LIME [26], RISE [24] and Shapley Sampling [16] are typical perturbation methods. For example, [24] proposed a method to generate an explanation from the relationship between masked image and output.

The Attention Branch Network [10] (ABN), IA-CNN [36] and IA-RED^2 [22] are categorized as other methods. The authors of [22] argued that attention in the transformer layer is not always appropriate as an explanation. Reference [13] further showed that attention in the transformer layer can be controlled without changing the prediction. ABN, which uses a branch structure, has been extended as Multi-ABN [17], ABEN [21], and PonNet [18]. The authors of [36] proposed a method to generate the explanations for each key point by connecting parallel branching structures to the CNN.

Sanity Check [2], ROAR [11], and [9] are representative studies that evaluate visual explanations. Reference [2] evaluated an explanation by comparing the explanations generated by trained and randomized models. In [11], images are re-trained without the important regions and the difference in accuracy is calculated. Re-training can eliminate the effect of out-of-distribution data. Reference [9] evaluated robustness using the distance between explanations with and without samples in the training set.

The proposed method differs from other visual explanation generation methods (e.g., Attention Rollout [1]) in that it generates explanations using a branch structure rather than attention in the Lambda layer. The proposed method is also different from ABN in that it generates visual explanations and delete pixels simultaneously.

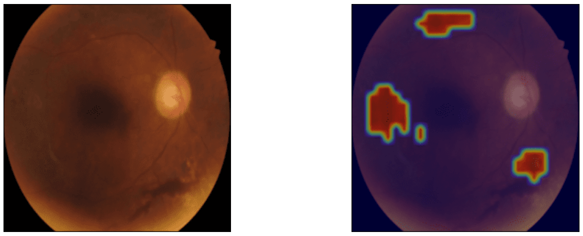


Fig. 2. Example data of visual explanation generation.

3 Problem Statement

In this paper, we focus on the task of visualizing important regions in an image as a visual explanation of the model's decisions. In particular, we focus on visual explanations of the Lambda-based transformer [3]. In this task, the pixels that contributed to the model's prediction should be attended. For example, Fig. 2 shows an example image from a standard dataset, Indian Diabetic Retinopathy Image Dataset (IDRiD) [25]. The left and right figures show the input image and the visual explanation, respectively.

The input and output of this task are defined as follows:

Input: Image $\boldsymbol{x} \in \mathbb{R}^{c_1 \times w_1 \times h_1}$
Output: Predicted probability for each class $\hat{\boldsymbol{y}} \in \mathbb{R}^{C}$

where C, c_1, w_1, and h_1 denote the number of classes, the number of channels, width and height of the input image, respectively. Additionally, the importance of each pixel is obtained as an attention map $\boldsymbol{\alpha} \in \mathbb{R}^{w_1 \times h_1}$, which is used as a visual explanation.

The Insertion-Deletion [24] and PID scores are used as an evaluation metrics (see Section 4 for details). Using the PID score, we can evaluate the match between the attention map and the region that contributed to the model's decision. In this paper, we assume that the model is based on a Lambda-based transformer. We also assume that the attention maps are not class specific. Furthermore, we focus on specific domains, such as medical care.

4 Proposed Method

Our method is inspired by transformer-based methods that capture the interactions among pixels, such as the Lambda ResNet [3]. Lambda ResNet can capture the relationship of the entire image with less computation than a typical self-attention mechanism used in simple vision transformers [8]. Since it does not assume patch partitioning, it is highly compatible with CNNs. Our method is also inspired by explanatory visualization methods, such as the Attention Branch Network [10], in the aspect of using a parallel branch structure. Attention Branch

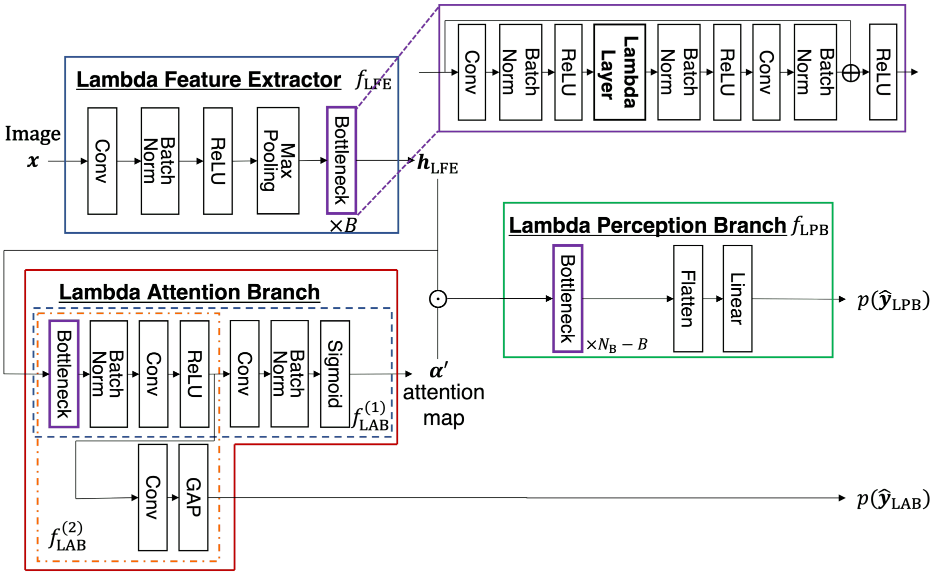


Fig. 3. Framework of LABN: our method consists of a Lambda Feature Extractor, Lambda Attention Branch and Lambda Perception Branch. Each module is explained in Section 4.1. "Conv" and "GAP" denote the convolution layer and global average pooling, respectively.

Network is a model that introduces a parallel branch structure to obtain an attention map. We use the parallel attention mechanism to obtain a clear visual explanation. This explanation is used to highlight important pixels for input.

The novelties of our approach are as follows:

- We introduce a structure that obtains an attention map as a parallel branch, which provides a clear explanation than a serial Lambda Layer.
- We propose the PID score, which is an extension of Insertion-Deletion score [24], as an effective explanation evaluation method for images with a large area of unimportant regions.

4.1 Structure

Fig. 3 shows the structure of our method. It is composed of three modules: the LFE, LAB, and LPB. We assume that the backbone network contains N_{B} bottleneck layers. First, we divide the backbone network into the LFE and LPB modules at the B-th bottleneck layer. Next, we introduce the LAB, which is placed in parallel between the LFE and LPB.

The input of the LFE is image $\boldsymbol{x}$. The LFE contains B bottleneck layers and a batch normalization layer to extract features from $\boldsymbol{x}$. The bottleneck layer consists of a Lambda layer and multiple convolutional layers, batch normalization layers, and ReLU activation functions. The Lambda layer is described later. The output of the LFE is denoted as $\boldsymbol{h}_{\mathrm{LFE}} \in \mathbb{R}^{c_2 \times w_2 \times h_2}$, where c_2, w_2 and h_2 denote the number of channels, width, and height of the output of the LFE, respectively.

The LAB is divided into two parts, $f_{\mathrm{LAB}}^{(1)}$ and $f_{\mathrm{LAB}}^{(2)}$. First, $f_{\mathrm{LAB}}^{(1)}$ generates an attention map. It contains a bottleneck layer and a global average pooling layer. The input and output of $f_{\mathrm{LAB}}^{(1)}$ are $\boldsymbol{h}_{\mathrm{LFE}}$ and $\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^{w_2 \times h_2}$. We upscale $\tilde{\boldsymbol{\alpha}}$ to obtain $\boldsymbol{\alpha} \in \mathbb{R}^{w_1 \times h_1}$ for the visual explanation. We obtain the final $\boldsymbol{\alpha}' \in \mathbb{R}^{w_2 \times h_2}$ by setting 0 to the values of $\tilde{\boldsymbol{\alpha}}$ that are below the value of $\theta_{\boldsymbol{\alpha}}$, where $\theta_{\boldsymbol{\alpha}}$ is a hyperparameter that represents the threshold of the attention map.

$$\tilde{\boldsymbol{\alpha}} = f_{\mathrm{LAB}}^{(1)} \left(\boldsymbol{h}_{\mathrm{LFE}}\right), \tag{1}$$

$$\boldsymbol{\alpha}'_{ij} = \begin{cases} \tilde{\boldsymbol{\alpha}}_{ij} & (\theta_{\boldsymbol{\alpha}} < \tilde{\boldsymbol{\alpha}}_{ij}), \\ 0 & (\text{otherwise}), \end{cases} \tag{2}$$

The reason for setting $\boldsymbol{\alpha}'_{ij} = 0$ for values less than $\theta_{\boldsymbol{\alpha}}$ is to use only the important regions for prediction. Because the input of LPB is $\boldsymbol{\alpha}' \odot \boldsymbol{h}_{\mathrm{LFE}}$, the value for importance less than $\theta_{\boldsymbol{\alpha}}$ is 0. In other words, regions of low importance are masked to 0. Therefore, regions essential for prediction will be given higher alpha than other regions. This prevents the importance of regions that contribute to the model's predictions from decreasing.

The input and output of $f_{\mathrm{LAB}}^{(2)}$ are $\boldsymbol{h}_{\mathrm{LFE}}$ and $p(\hat{\boldsymbol{y}}_{\mathrm{LAB}})$, respectively. Using $p(\hat{\boldsymbol{y}}_{\mathrm{LAB}})$ in the loss function, we can train LAB directly for classification tasks. As a result, we can generate attention maps associated strongly with the classification result.

Next, f_{LPB} performs classification based on the outputs of the LFE and LAB. It contains $(N_{\mathrm{B}} - B)$ bottleneck layers, flatten layers, and fully connected layers. The input and output of LPB are $\boldsymbol{\alpha}' \odot \boldsymbol{h}_{\mathrm{LFE}}$ and $p(\hat{\boldsymbol{y}}_{\mathrm{LPB}})$, respectively, where $\odot$ represents the Hadamard product.

4.2 Lambda Layer

We use the Lambda layer proposed in [3]. Self-attention [32] models relationships in a sequence by computing inner products. However, self-attention is computationally expensive and difficult to implement in images with long sequences. Unlike [8], Lambda Netowkrs [3] reduced computational complexity by performing dimensionality reduction followed by inner product calculation. This allows the relationship between pixels to be modeled without having to split them into patches.

Fig. 4 shows the structure of the Lambda layer. $\boldsymbol{h} \in \mathbb{R}^{c_3 \times w_3 \times h_3}$ denotes the input of the Lambda layer, where c_3, w_3 and h_3 denote the number of channels, width, and height of the input of the Lambda layer, respectively.

First, we generate the queries, keys and values as follows:

$$Q = V = \mathrm{Conv}(\boldsymbol{h}), \tag{3}$$

$$K = \mathrm{softmax}(\mathrm{Conv}(\boldsymbol{h})). \tag{4}$$

Next, we compute the content lambda $\boldsymbol{\lambda}_{\mathrm{c}}$ and position lambdas $\boldsymbol{\lambda}_{\mathrm{p}}$:

$$\boldsymbol{\lambda}_{\mathrm{c}} = K^{\top}V, \tag{5}$$

$$\boldsymbol{\lambda}_{\mathrm{p}} = \mathrm{Conv}(V). \tag{6}$$

The output of the Lambda Layer $\boldsymbol{h}_{\mathrm{L}} \in \mathbb{R}^{c_3 \times w_3 \times h_3}$ is computed as follows:

$$\boldsymbol{h}_{\mathrm{L}} = (\boldsymbol{\lambda}_{\mathrm{c}} + \boldsymbol{\lambda}_{\mathrm{p}})^{\top} Q. \tag{7}$$

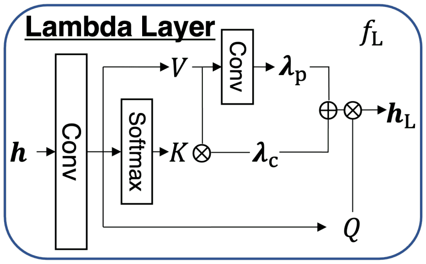

Lambda Layer
f_{L}
$\boldsymbol{h}$
Conv
Softmax
V
Conv
$\boldsymbol{\lambda}_{\mathrm{p}}$
K
$\boldsymbol{\lambda}_{\mathrm{c}}$
$\boldsymbol{h}_{\mathrm{L}}$
Q

Fig. 4. Structure of Lambda layer.

We use the following loss function:

$$\mathcal{L} = \mathcal{L}_{\mathrm{LPB}} + \lambda_1 \mathcal{L}_{\mathrm{LAB}} + \lambda_2 \mathcal{L}_{\mathrm{KL}}, \tag{8}$$

$$\mathcal{L}_{\mathrm{LPB}} = \mathrm{CE}(\hat{\boldsymbol{y}}^{(\boldsymbol{x})}_{\mathrm{LPB}}, \boldsymbol{y}), \tag{9}$$

$$\mathcal{L}_{\mathrm{LAB}} = \mathrm{CE}(\hat{\boldsymbol{y}}_{\mathrm{LAB}}, \boldsymbol{y}), \tag{10}$$

$$\mathcal{L}_{\mathrm{KL}} = D_{\mathrm{KL}}\left(\hat{\boldsymbol{y}}^{(\boldsymbol{x})}_{\mathrm{LPB}} || \hat{\boldsymbol{y}}^{(\tilde{\boldsymbol{x}})}_{\mathrm{LPB}}\right), \tag{11}$$

where $\boldsymbol{y}$, CE, D_{KL} and λ_1, λ_2 denote the ground truth label, cross-entropy loss function, Kullback–Leibler divergence and weights, respectively. In addition, $\tilde{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}^{(\boldsymbol{z})}_{\mathrm{LPB}}$ denote the image masked by the bias image and the output of LPB when $\boldsymbol{z}$ is input, respectively. $\tilde{\boldsymbol{x}}$ and the bias image $\boldsymbol{b}$ are computed as follows:

$$\tilde{\boldsymbol{x}}_{ij} = \begin{cases} \boldsymbol{x}_{ij} \ (\theta_{\boldsymbol{b}} < \tilde{\boldsymbol{\alpha}}_{ij}), \\ \boldsymbol{b}_{ij} \ (\text{otherwise}), \end{cases} \tag{12}$$

$$\boldsymbol{b}_{ij} = \frac{1}{N} \sum_{k=0}^{N} \boldsymbol{x}^{(k)}_{ij}, \tag{13}$$

where $\theta_{\boldsymbol{b}}$ and $\boldsymbol{x}^{(k)}$ denote the hyperparameter of the mask ratio and the k-th sample of the training sets. Note that it is often impossible to define an appropriate bias image on generic datasets (e.g., ImageNet and VOC).

4.3 PID Score

The Insertion-Deletion (ID) score is a standard evaluation metric for visual explanations [24]. It measures the change in the probability of a predicted class when pixels are inserted according to the importance given by a method. However, the ID score often overestimates coarse explanations. This is inappropriate for problems with sparse important regions.

Therefore, we propose the PID score, an extension of the ID score, as an effective evaluation metric for images with sparse important regions. the PID score uses the maximum importance in the patch. This increases the influence of the details in the fine-grained explanation and allows for proper evaluation as well as for normal images. The PID score can appropriately evaluate such images by inserting and deleting patches. The PID score is defined as follows:

$$\mathrm{PID} = \mathrm{AUC}(\text{patch-insertion}) - \mathrm{AUC}(\text{patch-deletion}), \tag{14}$$

where AUC denotes the area under the curve.

Patch-insertion and patch-deletion curves are obtained by the following procedure: First, we divide $\boldsymbol{x}$ into patches (submatrices) $\boldsymbol{p}_{ij} \in \mathbb{R}^{c_1 \times m^2}$, where m, i, and j denote the size of the patch, vertical indices, and horizontal indices, respectively. When $m = 1$ and $\boldsymbol{b}_{ij} = 0$ for any i and j, the PID score is the same as the ID score.

Next, we apply max-pooling to attention map $\boldsymbol{\alpha}$ to create the attention map $\boldsymbol{\alpha}_{\mathrm{p}} \in \mathbb{R}^{m^2}$ for each patch. The elements of $\boldsymbol{\alpha}_{\mathrm{p}}$ are denoted as $\boldsymbol{\alpha}_{i_1 j_1}, \boldsymbol{\alpha}_{i_2 j_2}, \cdots, \boldsymbol{\alpha}_{i_m j_m}$ in ascending order. We define A_n as follows:

$$A_n = \left\{ (i_k, j_k) | k \leqq n \right\}, \tag{15}$$

where n is the number of patches inserted or deleted.

Then, the inputs of patch-insertion $\boldsymbol{i}_n$ and patch-deletion $\boldsymbol{d}_n$ are represented using A_n as follows:

$$(\boldsymbol{i}_n, \boldsymbol{d}_n) = \begin{cases} (\boldsymbol{p}_{ij}, \boldsymbol{b}_{ij}) \ (i, j) \in A_n, \\ (\boldsymbol{b}_{ij}, \boldsymbol{p}_{ij}) \ (\text{otherwise}). \end{cases} \tag{16}$$

Finally, patch-insertion and patch-deletion curves are obtained by plotting n with $\boldsymbol{y}_{\mathrm{c}}^{(\mathrm{ins,n})}$ and $\boldsymbol{y}_{\mathrm{c}}^{(\mathrm{del},n)}$, respectively. Here, $\boldsymbol{y}^{(\mathrm{ins,n})}$, $\boldsymbol{y}^{(\mathrm{del},n)}$, and c respectively denote the outputs when $\boldsymbol{i}_n$ is input, outputs when $\boldsymbol{d}_n$ is input, and the class to which $\boldsymbol{x}$ belongs.

5 Experiments

5.1 Experimental Setup

IDRiD Dataset The Indian Diabetic Retinopathy Image Dataset (IDRiD) [25] and DeFN magnetograms dataset [20] were used for the experimental evaluation. The IDRiD is a dataset for detecting diabetic retinopathy from retinal fundus images. It was annotated by medical experts. The images were classified into separate groups ranging from 0 (No apparent) to 4 (Severe) according to the International Clinical Diabetic Retinopa [35]. The IDRiD contained 516 samples. Among these samples, 168 were negative and 348 were positive. The training, validation, and test sets consisted of 330, 83, and 103 samples, respectively. For the IDRiD, we assigned binary label to five lesions of retinopathy grades 0–4, by converting grade 0 to negative and grades 1–4 to positive. The size of each image were 4288 × 2848. The input images were resized to 224 × 224. We used the IDRiD because it is a standard dataset for visual explanation generation tasks [19]. In addition, we added a bias image to the training set as a negative class.

Table 1. Parameter settings.

		IDRiD	DeFN magnetograms
Optimizer		AdamW	AdamW
Learning rate	LAB, Linear	1.0×10^{-3}	1.0×10^{-3}
	LFE, LPB	1.0×10^{-4}	1.0×10^{-4}
Weight decay		0.09	0.09
Batch size		32	32
Loss weights	Negative	2	1
	Positive	1	1
$\theta_{\boldsymbol{\alpha}}$		0.75	0.5
$\theta_{\boldsymbol{b}}$		0.2	0.2
N_{B}		16	1
B		7	0

DeFN magnetograms Dataset The DeFN magnetograms dataset contained the hourly solar images taken by the Helioseismic and Magnetic Imager [27]. We collected magnetograms from the Solar Dynamic Observatory[3] [23] web archives. Because the mechanism underlying solar flares remains unclear, it is important to generate visual explanations that give insight into the related theory.

The label of each magnetogram was the maximum solar flare class that occurred within 24 hours. We assigned binary labels for the four solar flare classes of O, C, M, and X, by converting O and C classes to "$<$ M" and converting M and X to "$\geq$ M." The input images were resized to 512×512. The DeFN magnetograms dataset contained 61,315 samples, covering the period from June 2010 to December 2017. Among these samples, 56,078 images were labeled as "$<$ M" and 5,237 images were labeled as "$\geq$ M." The size of each image was 1024×1024. The training, validation, and test sets consisted of 45,530 samples from 2010–2015, 7,795 samples from 2016, and 7,990 samples from 2017, respectively. Similar to the IDRiD, we added a bias image to the training set as the "$<$ M" class. The training, validation, and test sets were used for parameter training, hyperparameter validation, and evaluation, respectively. The images were standardized for both datasets.

Hyperparameter Settings Table 1 shows the hyperparameter settings of the proposed method. Here, the loss weights denote the weight of each class in the loss function. When $N_{\mathrm{B}} = 1$ and $N_{\mathrm{B}} = 16$, the model had 8,490 and 22 million parameters, respectively. The parameters were trained on an RTX 2080 with 11GB of GPU memory and an Intel Core i9 processor. It took approximately 1 day and 40 minutes to train the model on the DeFN magnetograms dataset and IDRiD, respectively. The inference time was approximately 0.1 s. Warmup and cosine-decay were used to schedule the learning rate. We stopped the training when the loss on the validation set did not improve for six consecutive epochs.

[3] `https://sdo.gsfc.nasa.gov/data/`

Table 2. Quantitative results on the IDRiD (upper table) and DeFN magnetograms dataset (lower table).

Method		RISE [24]	Lambda attention [33]	Ours (LABN)
ID↑		0.319 ± 0.015	-0.101 ± 0.074	$\mathbf{0.431 \pm 0.213}$
PID↑	$m = 2$	0.179 ± 0.080	-0.105 ± 0.073	$\mathbf{0.458 \pm 0.198}$
	$m = 4$	0.130 ± 0.045	-0.116 ± 0.081	$\mathbf{0.473 \pm 0.178}$
	$m = 8$	0.136 ± 0.050	-0.123 ± 0.078	$\mathbf{0.470 \pm 0.178}$
	$m = 16$	0.101 ± 0.033	-0.093 ± 0.054	$\mathbf{0.455 \pm 0.181}$

Method		RISE [24]	Lambda attention [33]	Ours (LABN)
ID↑		0.235 ± 0.145	0.374 ± 0.080	$\mathbf{0.506 \pm 0.170}$
PID↑	$m = 2$	0.261 ± 0.217	0.414 ± 0.129	$\mathbf{0.748 \pm 0.102}$
	$m = 4$	0.296 ± 0.199	0.403 ± 0.138	$\mathbf{0.755 \pm 0.100}$
	$m = 8$	0.379 ± 0.172	0.378 ± 0.162	$\mathbf{0.757 \pm 0.094}$
	$m = 16$	0.461 ± 0.164	0.291 ± 0.216	$\mathbf{0.756 \pm 0.096}$

Table 3. Confusion matrix.

	IDRiD	DeFN magnetograms
TP	56	161
TN	25	7566
FP	9	243
FN	13	20

5.2 Quantitative Results

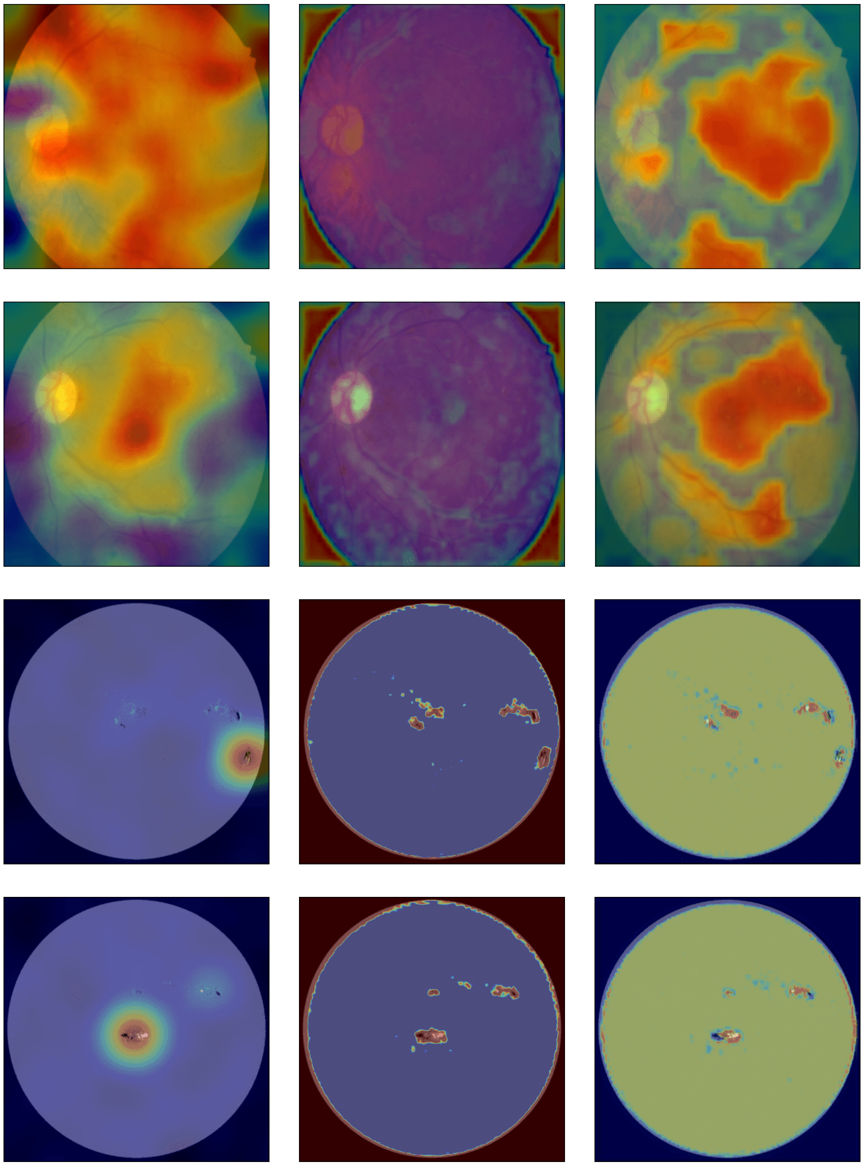
We used RISE [24] and Lambda attention [33] as the baseline methods. We obtained Lambda attention by computing the average of $\boldsymbol{\lambda}_{\mathrm{c}}^{\top} Q$ in the Lambda layer in the direction of channel dimension. Because there is no established explanation generation method for Lambda Networks, we constructed a baseline method based on standard explanation methods for transformers (e.g., attention rollout [33]). As a result, we selected the optimal $\boldsymbol{\lambda}_{\mathrm{c}}^{\top} Q$ and named it Lambda attention. RISE is also a standard method that can be applied to general models.

We used the ID and PID scores as the primary evaluation metrics. The ID score is a standard evaluation method for explanation generation. We used the PID score because the IDRiD and DeFN magnetograms dataset contain sparse images.

Table 2 shows the quantitative results. The upper and lower tables show the results on the IDRiD and DeFN magnetograms dataset, respectively. We conducted the experiment four times for each method, and the average and standard deviations of the scores are reported. For the IDRiD and DeFN magnetograms dataset, only the positive and "$\geq$ M" data were used to calculate the score, respectively. This is because the negative and "$<$ M" data do not contain regions that are appropriate for explanation.

Table 4. Quantitative results of ablation studies.

Condition		(i)	(ii)	(iii)	(iv) Ours
	$\mathcal{L}_{\mathrm{KL}}$			✓	✓
	bias image		✓		✓
ID↑		0.044	0.124	0.460	**0.506**
PID↑	$m = 2$	0.311	0.446	**0.774**	0.748
	$m = 4$	0.489	0.405	**0.792**	0.755
	$m = 8$	0.523	0.388	**0.808**	0.757
	$m = 16$	0.556	0.382	**0.807**	0.756



(a) RISE [24] (b) Lambda attention [33] (c) Ours

Fig. 5. Qualitative results. The top two and bottom two rows show the results on the IDRiD, DeFN magnetograms dataset, respectively. RISE focuses on areas that are too large, and Lambda attention focuses on the background. By contrast, the proposed method does not focus on inappropriate areas.

For the IDRiD, Table 2 shows that the PID score at $m = 16$ was 0.101, -0.093, and 0.455 points for RISE, Lambda attention and LABN, respectively. The PID score of LABN was better than that of RISE by 0.354 points. Similarly, the ID and PID scores at $m = 2, 4, 8$ improved by 0.112 and 0.279, 0.343, 0.334 points, respectively, when LABN was used. For the PID score ($m \geq 2$), these results suggest that the performance has improved ($p < 0.1$).

For the DeFN magnetograms dataset, the table shows that the ID and the PID scores were also improved. These results indicate that LABN successfully generates explanations for images with sparse important regions. In particular, for the PID score ($m \geq 2$), there was a statistically significant improvement ($p < 0.05$). On the other hand, in the ID score, these results suggest that there is no significant difference ($p > 0.1$).

Table 3 shows the confusion matrix of our method. On the IDRiD, 56, 25, 9 and 13 samples were classified as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), respectively. On the DeFN magnetograms dataset, 161, 7566, 243, and 20 samples were classified as TP, TN, FP, and FN, respectively.

LABN failed on 22 and 263 samples in the IDRiD and DeFN magnetograms dataset, respectively. LABN failed to generate explanations with a PID score < 0.2 for 32 and 1029 samples in the IDRiD and DeFN magnetograms dataset, respectively.

5.3 Ablation Studies

We set the following ablation conditions:

1. w/o $\mathcal{L}_{\mathrm{KL}}$
   We removed $\mathcal{L}_{\mathrm{KL}}$ to investigate the effect of saliency guided training on explanation generation performance.
2. w/o bias image
   We eliminated the bias image from the training set to investigate the effect on explanation generation performance.

Table 4 shows the quantitative results for the ablation studies. Under condition (ii), both ID and PID scores decreased significantly. Note that condition (iii) led to higher performance in terms of PID. These results indicate that $\mathcal{L}_{\mathrm{KL}}$ contributed more to the model performance.

5.4 Qualitative Results

The top two and bottom two rows of Fig. 5 show the qualitative results for the IDRiD and DeFN magnetograms dataset samples, respectively. The first and second rows in the left column of the image illustrate that RISE attended large areas and did not focus on the important regions. In the first and second rows in the middle column of image, most of the area attended by Lambda attention are background, which is inappropriate. By contrast, LABN successfully attended

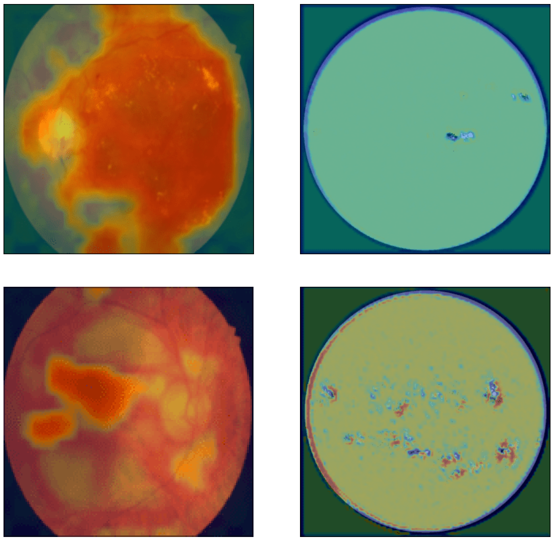


Fig. 6. Failure examples. Left and right figures are images from the IDRiD and DeFN magnetograms dataset, respectively.

the appropriate regions, which contributed to the accuracy, as shown in the first and second rows in the right column of image.

The third and forth rows in the left and middle columns of images demonstrate that RISE and Lambda attention attended unrelated regions such as the circumference and background. By contrast, the third and forth rows in the right column of image show that LABN appropriately attended the active sunspots, which are important for solar flare prediction.

Fig. 6 shows examples of failures (examples with a PID score less than 0.2), for the IDRiD and DeFN magnetograms dataset, respectively. In the left figure, a wide range of areas including the optic disk are highlighted. In the right figure, important regions were not appropriately highlighted, reducing the PID score.

For the IDRiD and DeFN magnetograms dataset, 32 and 100 samples, respectively were classed as failures. Note that we randomly selected 100 failure samples from the DeFN magnetograms dataset.

- IP (Incorrect Prediction)
  IP refers to a case in which the model predicts the incorrect class. The bottom left of Fig. 6 shows an example of an IP.
- OA (Over-Attended)
  OA refers to a case in which most of the image is highlighted. The top left of Fig. 6 shows an example of an OA image. In this example, the model paid attention to the entire image.
- IA (Insufficiently Attended)
  IA refers to a case in which the attended area is insufficient. The top right

Table 5. Error analysis

Error ID	Description	IDRiD	DeFN magnetograms
IP	Incorrect Prediction	12	58
OA	Over-Attended	16	3
IA	Insufficiently Attended	3	24
WA	Wrongly Attended	1	15

of Fig. 6 shows an example of an IA image. In this example, $||\boldsymbol{\alpha}||$ is small and the sunspots are not highlighted.

- WA (Wrongly Attended)
  WA refers to a case in which the pixels that do not contribute to the accuracy are given attention. The bottom right of Fig. 6 shows an example of a WA image. The PID score at $m = 16$ is 0.1, which indicates that non-important regions are attended.

5.5 Error Analysis

Table 5 shows that the bottleneck of the IDRiD and DeFN magnetograms dataset are OA and IP errors, respectively. Therefore, we expect that improving the model accuracy and regularizing $\boldsymbol{\alpha}$ each dataset will be an effective approach to generating appropriate visual explanations.

6 Conclusions

In this paper, we focused on the task of generating an attention map as a visual explanation for a model's decisions. The main contributions of this paper are as follows:

- We introduced a parallel branch structure to obtain explanations that are clearer than those obtained using Lambda attention.
- We proposed the PID score as an effective evaluation measure for explaining images with sparse important regions.
- We introduced the loss used in saliency guided training [12] to reduce the importance of irrelevant regions.
- LABN outperformed the baseline methods in terms of the ID and PID scores.

Acknowledgements This work was partially supported by JSPS KAKENHI Grant Number 20H04269 and NEDO.

References

1. Abnar, S., Zuidema, W.: Quantifying Attention Flow in Transformers. arXiv preprint arXiv:2005.00928 (2020)
2. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity Checks for Saliency Maps. In: NeurIPS. vol. 31 (2018)
3. Bello, I.: LambdaNetworks: Modeling Long-Range Interactions without Attention. In: ICLR (2021)
4. Binder, A., et al.: Layer-Wise Relevance Propagation for Neural Networks with Local Renormalization Layers. In: ICANN. pp. 63–71 (2016)
5. Chefer, H., Gur, S., Wolf, L.: Transformer Interpretability Beyond Attention Visualization. In: CVPR. pp. 782–791 (2021)
6. Das, A., Rad, P.: Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. arXiv preprint arXiv:2006.11371 (2020)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)
8. Dosovitskiy, A., Beyer, L., et al.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: ICLR (2021)
9. Fel, T., Vigouroux, D., Cadène, R., Serre, T.: How Good Is Your Explanation? Algorithmic Stability Measures To Assess the Quality of Explanations for Deep Neural Networks. In: WACV. pp. 720–730 (2022)
10. Fukui, H., Hirakawa, T., et al.: Attention Branch Network: Learning of Attention Mechanism for Visual Explanation. In: CVPR. pp. 10705–10714 (2019)
11. Hooker, S., Erhan, D., Kindermans, P.J., Kim, B.: A Benchmark for Interpretability Methods in Deep Neural Networks. In: NeurIPS. vol. 32 (2019)
12. Ismail, A.A., Corrada Bravo, H., Feizi, S.: Improving Deep Learning Interpretability by Saliency Guided Training. In: NeurIPS (2021)
13. Jain, S., Wallace, B.: Attention is not Explanation. In: NAACL. pp. 3543–3556 (2019)
14. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., et al.: Transformers in Vision: A Survey. arXiv preprint arXiv:2101.01169 (2021)
15. Li, H., Ellis, J., Zhang, L., Chang, S.F.: PatternNet: Visual Pattern Mining with Deep Neural Network. In: ICMR. pp. 291–299 (2018)
16. Lundberg, S., Lee, S.I.: A unified approach to interpreting model predictions. In: NeurIPS. pp. 4765–4774 (2017)
17. Magassouba, A., Sugiura, K., et al.: A Multimodal Classifier Generative Adversarial Network for Carry and Place Tasks from Ambiguous Language Instructions. RA-L **3**(4), 3113–3120 (2018)
18. Magassouba, A., Sugiura, K., et al.: Predicting and attending to damaging collisions for placing everyday objects in photo-realistic simulations. Advanced Robotics **35**(12), 787–799 (2021)
19. Mitsuhara, M., Fukui, H., Sakashita, Y., et al.: Embedding Human Knowledge into Deep Neural Network via Attention Map. In: VISAPP (2021)
20. Nishizuka, N., Sugiura, K., et al.: Deep Flare Net (DeFN) Model for Solar Flare Prediction. The Astrophysical Journal **858**(2), 113 (8pp) (2018)
21. Ogura, T., Magassouba, A., Sugiura, K., et al.: Alleviating the burden of labeling: Sentence generation by attention branch encoder-decoder network. RA-L **5**(4), 5945–5952 (2020)
22. Pan, B., Panda, R., Jiang, Y., et al.: IA-RED^2: Interpretability-aware redundancy reduction for vision transformers. In: NeurIPS (2021)

23. Pesnell, W., Thompson, B., Chamberlin, P.: The Solar Dynamics Observatory (SDO). Solar Physics **275**(1–2), 3–15 (2012)
24. Petsiuk, V., Das, A., Saenko, K.: RISE: Randomized input sampling for explanation of black-box models. In: BMVC. p. 151(13pp) (2018)
25. Porwal, P., et al.: IDRiD: Diabetic retinopathy – segmentation and grading challenge. Medical Image Analysis **59**(101561) (2020)
26. Ribeiro, M., Singh, S., et al.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: KDD. p. 1135–1144 (2016)
27. Scherrer, P., Schou, J., Bush, R., et al.: The helioseismic and magnetic imager (HMI) investigation for the solar dynamics observatory (SDO). Solar Physics **275**, 207–227 (2012)
28. Selvaraju, R., et al.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In: ICCV. pp. 618–626 (2017)
29. Smilkov, D., Thorat, N., Kim, B., Viégas, F.B., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
30. Srinivas, S., Fleuret, F.: Full-Gradient Representation for Neural Network Visualization. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
31. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic Attribution for Deep Networks. In: ICML. vol. 70, p. 3319–3328 (2017)
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., et al.: Attention is all you need. In: NeurIPS. pp. 6000–6010 (2017)
33. Vig, J.: A multiscale visualization of attention in the transformer model. In: ACL. pp. 37–42 (2019)
34. Wang, H., Wang, Z., Du, M., et al.: Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks. In: CVPR. pp. 24–25 (2020)
35. Wu, L., et al.: Classification of Diabetic Retinopathy and Diabetic Macular Edema. World Journal of Diabetes **4**(6), 290–294 (2013)
36. Zhang, Z., Chen, Y., Li, H., Zhang, Q.: IA-CNN: A generalised interpretable convolutional neural network with attention mechanism. In: IJCNN. pp. 1–8 (2021)
37. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., et al.: Learning deep features for discriminative localization. In: CVPR. pp. 2921–2929 (2016)