ViT-CX: Causal Explanation of Vision Transformers

Weiyan Xie[1], **Xiao-Hui Li**[2], **Caleb Chen Cao**[2], **Nevin L. Zhang**[1]

[1] The Hong Kong University of Science and Technology [2] Huawei Technologies Co., Ltd
[1] {wxieai,lzhang}@cse.ust.hk [2] {lixiaohui33,caleb.cao}@huawei.com

Abstract

Despite the popularity of Vision Transformers (ViTs) and eXplainable AI (XAI), only a few explanation methods have been proposed for ViTs thus far. They use attention weights of the classification token on patch embeddings and often produce unsatisfactory saliency maps. In this paper, we propose a novel method for explaining ViTs called *ViT-CX*. It is based on patch embeddings, rather than attentions paid to them, and their causal impacts on the model output. ViT-CX can be used to explain different ViT models. Empirical results show that, in comparison with previous methods, ViT-CX produces more meaningful saliency maps and does a better job at revealing all the important evidence for prediction. It is also significantly more faithful to the model as measured by deletion AUC and insertion AUC.

1 Introduction

The necessity of explaining the predictions by deep neural networks (DNNs) is now widely recognized (Miller 2019; Gunning and Aha 2019; Rebuffi et al. 2020). To practitioners, DNNs are black boxes whose inner workings are difficult to comprehend. Explaining why DNNs make specific predictions can help build user trust and ensure fairness.

Vision Transformers (ViTs) are a new class of deep learning models that rival or even surpass the performance of convolutional neural networks (CNNs) on various vision tasks (Dosovitskiy et al. 2020; Carion et al. 2020; Liu et al. 2021). A few methods for explaining ViTs have been proposed, e.g., CGW1 (Chefer, Gur, and Wolf 2021b), CGW2 (Chefer, Gur, and Wolf 2021a) and TAM (Yuan et al. 2021b). Those methods are based on attention weights of the classification token ($[CLS]$) on patch embeddings, or a combination of attention weights and class gradients. The use of attention weights for explaining NLP models has been extensively debated and the general conclusion seems to point to the negative side (Jain and Wallace 2019; Serrano and Smith 2019; Pruthi et al. 2020; Bastings and Filippova 2020). In ViT models, *attention weights are concerned with the importance of path embeddings to the $[CLS]$ token, but not the semantic contents of the embeddings*. Using attention weights for explanation is analogous to understanding a doctor's diagnosis by listing the body parts he/she examined (patches) rather than considering the information he/she got from the examinations (patch embeddings).

In this paper, we propose a novel explanation method for ViTs that is based on patch embeddings and their causal impacts on the output [1]. A quick comparison of our method, termed *ViT-CX*, with several previous methods is shown in Figure 1. The task here is to explain ViT-B/16 on several example images. We see that the explanations given by ViT-CX are visually more meaningful than those by previous methods. They are also more faithful to the model according to the deletion AUC and insertion AUC metrics (Petsiuk, Das, and Saenko 2018).

More specifically, all the patch embeddings at a self-attention layer can be arranged into a 3D tensor, with the (x, y)-coordinates indicating their spatial information and the z-coordinate representing their semantic contents. A frontal slice (with a fixed z value) of the tensor can be upsampled to the size of the input image and the resulting heatmap is known as a *ViT feature map*. Figure 2 (b.1 - b.5) show some example ViT feature maps. They are clearly more meaningful than the attention weight maps shown in (a.1 - 1.5). In general, ViT feature maps tend to be semantically meaningful even at early transformer blocks (Yuan et al. 2021a; Raghu et al. 2021). In ViT-CX, we treat the ViT feature maps from different stages of a ViT model as masks (*ViT masks*), apply them to the input image to obtain masked images (Figure 2 (c.1 - c.5)), and combine the masks using the class scores on the masked images (i.e., causal impacts on output) to produce a saliency map.

The first important technical issue of this paper concerns is what we call the *pixel coverage bias (PCB)*. It refers to the fact that some pixels are included in more masks than others and hence get unustifiably higher saliency values. This can lead to nonsensical explanations in the case of causal overdetermination, where the correct prediction can be made by any of many possible small patches, as often happens in ViTs (Naseer et al. 2021; Paul and Chen 2022). We analyze the phenomenon theoretically and propose a simple and elegant solution. The second issue is that making inferences on a large number of masked images is costly. We alleviate this problem by clustering the ViT masks. Mask clustering does not lead to severe loss of information since they tend to overlap substantially (Figure 2 (b.1 - b.5)). The third issue is

[1] Note that here we talk about causality in the inference process rather than in the data generation process.

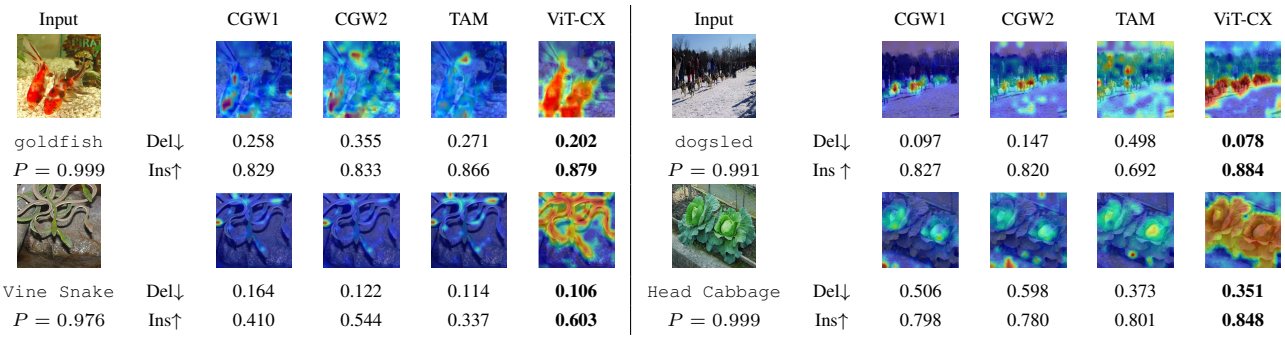


Figure 1: Explaining the predictions of ViT-B/16 on four images: The saliency maps produced by ViT-CX are clearly more meaningful than those by previous methods, as they highlight all the regions that are apparently important to predictions. They are also more faithful to the model as measured by the deletion (Del) and insertion (Ins) AUC metrics.

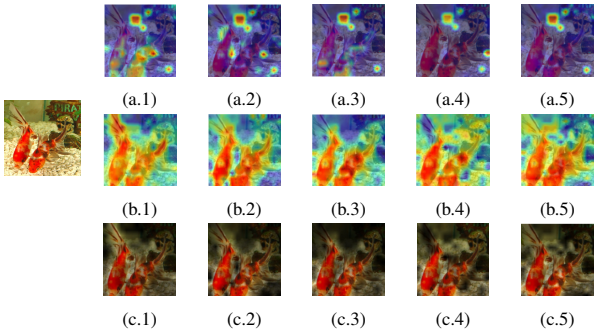


Figure 2: ViT feature maps (b.1 - b.5) are frontal slides of a 3D tensor made up of patch embedding vectors (as fibers). They are generally more meaningful than attention weight maps (a.1 - a.5), and they are used in ViT-CX as masks to create masked images (c.1 - c.5).

that applying a mask to an image might cause unintended artifacts (Fong and Vedaldi 2017; Fong, Patrick, and Vedaldi 2019). We resolve this issue by randomizing the values of the masked-out pixels.

The contributions of this work are listed below:

- We propose a new method for explaining ViT, and shows that it significantly outperforms previous methods (CGW1,CGW2,TAM) for the same task as evaluated by conventional protocols.
- Our ViT-CX is built upon on several previous ideas, namely mask-based explanation, creation of masks from feature maps and pixel coverage bias correction. We apply these ideas novelly to an important problem - ViT explanation.
- ViT-CX uses masks created from patch embeddings, whereas previous ViT explanation methods are based on attention weights paid to the path embeddings.
- We provide a detailed analysis of the impact of PCB in mask-based ViT explanation and theoretically show how it can be alleviated by the correction technique.
- An alternative way to correct PCB is to use a large number of random masks. We show that ViT-CX is computationally much more efficient than this brute-force method (384 vs. 5000 masks, a few seconds vs. 1 minute per image).

2 Related Work

Explanation Methods for Vision Transformers: The earliest methods for explaining ViT models are based on attention weights. All attention weights at an attention head can be arranged into an attention map, which can be upsampled to the input size to form a saliency map. Rollout (Abnar and Zuidema 2020) considers all heads from multiple layers, and combines the corresponding attention maps to form one saliency map. Partial LRP (Voita et al. 2019) is similar to Rollout, except that it assigns different weights to different heads, which are computed using Layer-wise Relevance Propagation (LRP). The saliency maps produced by Rollout and Partial LRP are not class-specific because the attention weights are class-agnostic. As such, those methods cannot be used to explain the reasons for particular output classes.

There are methods that aim to explain a particular output class. CGW1 (Chefer, Gur, and Wolf 2021b) is similar to Partial LRP, except that the gradients of the class score with respect to the heads are also considered, alongside LRP weights, when combining attention maps from different heads. In CGW2 (Chefer, Gur, and Wolf 2021a), the LRP weights are removed since they are found to be unnecessary. Transition Attention Map (TAM) (Yuan et al. 2021b) is similar to CGW2 except that simple gradients are replaced by integrated gradients (Sundararajan, Taly, and Yan 2017). Figure 1 shows several saliency maps produced by CGW1, CGW2 and TAM. They are apparently less satisfactory than those by ViT-CX. Moreover, attention weight-based methods cannot be used to explain the ViT models (Liu et al. 2021; Chu et al. 2021; Zhang et al. 2022) without $[CLS]$ token as they utilize the attention maps between the $[CLS]$ token and patch tokens. In contrast, ViT-CX does not rely on the $[CLS]$ token and can be adopted by more ViT variants.

Mask-based Explanation Methods: While there are only a few methods for explaining ViT models, a large number of methods have been proposed to explain CNN models. Mask-based explanation methods are one subclass. They de-

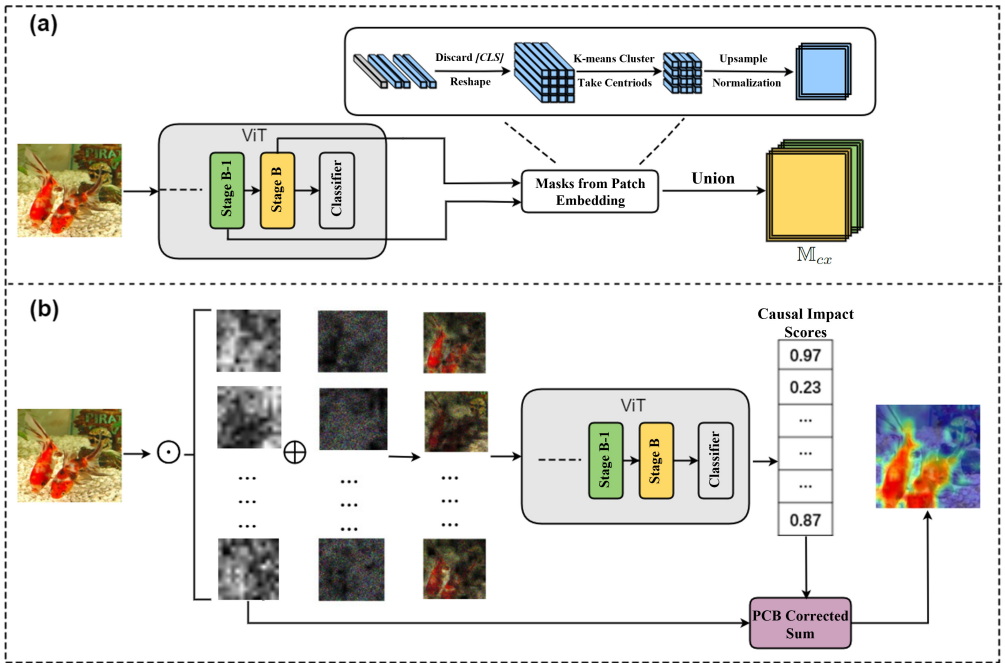


Figure 3: Overview of ViT-CX. (a) Mask Generation: A set of semantic masks is generated from the patch embeddings; (b) Mask Aggregation: A saliency map is created by combining the masks using the class scores of the masked images. Pixel coverage frequencies are used in the second step to correct for pixel coverage bias (PCB).

termine saliency values of pixels using a collection of masks $\mathbb{M} = \{\mathbf{M}_1, \cdots, \mathbf{M}_K\}$. Each mask $\mathbf{M}_i$ is of the same size as the input image $\mathbf{X}$, and its pixel values are between 0 and 1. It is combined with the input via element-wise product $\odot$ to yield a masked image $\mathbf{X} \odot \mathbf{M}_i$. Intuitively the pixels 'inside' the mask (with high values in $\mathbf{M}_i$) are kept and those 'outside' are erased. Each masked image is fed to the target model $f(\cdot)$ to get a score $f(\mathbf{X} \odot \mathbf{M}_i)$ for the target class, i.e., the output class being explained. Finally, a saliency map is created by combining the masks with the class scores:

$$S(x) = \sum_{i=1}^{K} f(\mathbf{X} \odot \mathbf{M}_i)\mathbf{M}_i(x). \tag{1}$$

For visualization, the saliency values are normalized to the interval $[0, 1]$ using $(S(x) - \min_x S(x))/(\max_x S(x) - \min_x S(x)$. Occlusion Map (Zeiler and Fergus 2014), Leave-One-Out (Li, Monroe, and Jurafsky 2016) and RISE (Petsiuk, Das, and Saenko 2018) are typical mask-based explanation methods. They differ in the way to generate masks.

Mask-based explanation is considered a causal approach because it perturbs the input image using masks, observes how the class score changes, and then builds a saliency map accordingly. Intuitively, we can think of a mask $\mathbf{M}_i$ as a 'team' of pixels, and the class score $f(\mathbf{X} \odot \mathbf{M}_i)$ as the *causal impact* of the 'team' on the output class. [2] The saliency value of a pixel is simply an aggregation of the causal impact scores of the 'teams' of which it is a member of. ViT-CX is a mask-based explanation method for ViT models. It has its own way of generating masks and calculating causal impact scores of masks. Being a causal method, ViT-CX is sensitive to the changes in model parameters.

Besides masked-based explanation, there are other causal explanation methods, e.g., LIME (Ribeiro, Singh, and Guestrin 2016), Kernel SHAP (Lundberg and Lee 2017). They are regression methods based on super-pixels. As such, they produce coarse-grained explanations.

Gradient-based Explanation Methods: Another branch of explanation methods for CNN models is gradient-based. Those methods can also be used to explain ViT models. As baselines to be compared with ViT-CX, we choose three popular methods, namely Grad-CAM (Selvaraju et al. 2017), Integrated-Grad (Sundararajan, Taly, and Yan 2017) and Smooth-Grad (Smilkov et al. 2017). Gradient-based methods are not causal and have known drawbacks. For instance, they can suffer from gradient saturation, which can lead to poor explanation results (Sundararajan, Taly, and Yan 2017; Shrikumar, Greenside, and Kundaje 2017).

3 Methodology

3.1 Preliminaries

In ViT models, an image $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ is split into $N = HW/p^2$ patches, with the i-th patch represented by a 2D vector $\mathbf{x}_i \in \mathbb{R}^{(p \times p) \times C}$, where (H, W, C) are the height, width, and the number of channels of the image, and (p, p) is

[2] Strictly speaking, the causal impact score (causal effect) should be the difference between the class score of a masked image and that of the blank image. However, the second term is usually ignored as it is the same for all masks.

the spatial resolution of each patch. The patches are mapped to embeddings with D dimensions via linear projection. In the vanilla ViT architecture (Dosovitskiy et al. 2020), the embeddings are fed into L transformer blocks. Each block includes two modules: A *Multi-Head Self-Attention (MHSA) module* and a *Multi-Layer Perceptron (MLP) module*. They yield new embeddings of the patches. In the process, the number and size of the patches remain the same and the dimension of the embeddings is kept constant.

In more recent hierarchical ViT architectures (Wang et al. 2021; Chu et al. 2021; Liu et al. 2021), the patches are gradually merged as the computation proceeds from layer to layer. As a consequence, the number of patches is gradually reduced, and the dimension of the embeddings is increased.

Hierarchical ViTs are divided into multiple (say B) stages. In the original ViT paper, there is no notion of the stages. Following Zheng et al. (2021), we divide the vanilla ViT into B stages for consistency. We denote the embedding output at the last attention module of stage i as $\mathbf{E}^{(i)} \in \mathbb{R}^{N_i \times D_i}$, where N_i is the number of patch tokens and D_i is the feature dimension.

3.2 Overview of ViT-CX

An overview of ViT-CX is shown in Figure 3. There are two phases. In the first phase, a collection of masks is generated from the patch embeddings in the target ViT model (Section 3.3). In the second phase, the masks are combined linearly to yield a saliency map. The weight for each mask is determined by the causal impact of the pixels 'inside' the mask on the class score (Section 3.4). Pixel coverage frequencies are used to correct the pixel coverage bias (Section 3.5).

3.3 Mask Generation

Different mask-based methods have their own ways of generating masks. Occlusion Map (Zeiler and Fergus 2014) uses a small sliding window and generates a binary mask at each location by setting the values of the pixels inside the window to 0 and the values of the pixels outside the window to 1. Leave-One-Out (Li, Monroe, and Jurafsky 2016) uses a special case of this strategy where the sliding window size is 1×1. RISE (Petsiuk, Das, and Saenko 2018) uses a random process to generate masks. Those methods lead to many masks, which implies high online computation costs.

Score-CAM (Wang et al. 2020) uses feature maps from a target CNN model as masks. The number of feature maps is relatively small, and they capture semantic information from the target model. We follow this strategy in ViT-CX to generate masks for a target ViT model from its patch embeddings.

We first create a collection of masks from the embedding output $\mathbf{E}^{(i)}$ at the last attention module of stage i. The embeddings are first reshaped into a 3D tensor of size $\sqrt{N_i} \times \sqrt{N_i} \times D_i$. Each fiber in the tensor corresponds to the embedding of a patch, and the (x, y)-coordinates of the fiber correspond to the spatial location of the patch in the input image. The frontal slices of the tensor are upsampled to the size of the input image, and the results are called *ViT feature maps*. [3] Several example ViT feature maps are shown in Figure 2 (b.1 - b.5). The feature maps are subsequently normalized to the interval $[0, 1]$ to get *ViT masks*.

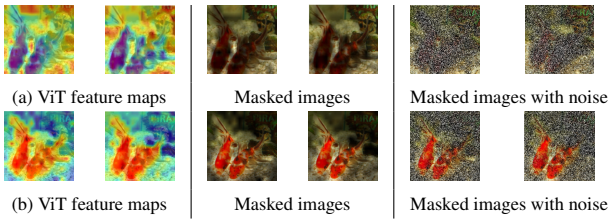


Figure 4: (a) Unintended artifacts of masking: The masked images are classified as `goldfish` with 94.8% and 95.1% though the fishes are masked out. After the random noises added, the prediction probabilities drop to 0.44% and 0.39%; (b) Examples for showing that the added random noises will not affect the preserved regions of the masked images.

The number of ViT masks created at stage i is D_i. We apply the K-means clustering algorithm to partition them into k clusters, where k is a hyperparameter, and then use the cluster centroids as the final masks. As the initial masks tend to overlap significantly (Figure 2: b.1 - b.5), clustering would not lead to much loss in information. However, it does improve the efficiency of online explanation. We denote the final set of masks for stage i as $\mathbb{M}^{(i)} = \{\mathbf{M}_1^{(i)}, \mathbf{M}_2^{(i)}, \cdots, \mathbf{M}_k^{(i)}\}$.

Feature maps from different stages of the model are different representations of the input image. To utilize the diversity of information, we use all the masks from the last few stages in ViT-CX:

$$\mathbb{M}_{cx} = \bigcup_{i=b}^{B} \mathbb{M}^{(i)},$$

where b is a hyperparameter, and the total number of masks K used in the explanation is $k \times (B - b + 1)$.

3.4 Causal Impact Score Revisited

In previous methods, masks are combined via Equation (1), where the $f(\mathbf{X} \odot \mathbf{M}_i)$ is the score of the target class on the masked image $\mathbf{X} \odot \mathbf{M}_i$. One issue here is that masking can lead to unintended artifacts, which gives misleading scores.

Consider the two examples in the Figure 4 (a). The ViT-feature maps focus on the background rather than the foreground, where the foreground pixels (the pixels of the goldfish) share the lowest values. When their corresponding masks are combined with the image, the foreground pixels are assigned nearly identical 'zero' pixel values. That 'erases' the detailed feature information of the goldfish, such as texture and color, but clearly leaves the shape of the goldfish in the masked images.

The two masked images with the shape of goldfish left are correctly classified as `goldfish` with high probabilities (94.8% and 95.1%). Evidently, those probabilities are

[3] If the value matrix in self-attention is the identity matrix, a frontal slice of the tensor plays a similar role as a feature map in a CNN model, in the sense that the 'pixel values' on it are aggregated using self-attention weights to compute activations for the next layer.

not a good measure of the causal impact of the pixels 'inside' the mask. To reduce the influence of such artifacts, we replace the term $f(\mathbf{X} \odot \mathbf{M}_i)$ in Equation (1) with the new definition of *causal impact score*:

$$s(\mathbf{X}, \mathbf{M}_i) = f(\mathbf{X} \odot \mathbf{M}_i + \mathbf{Rd} \odot (1 - \mathbf{M}_i)) \,, \tag{2}$$

$\mathbf{Rd} \in \mathbb{R}^{H \times W \times C}$ is a matrix of random numbers. Since we use soft masks with mask values in M_i between $[0, 1]$, most noises are added to masked-out pixels (pixels with mask values 0 in M_i), and noises are more or less added to other pixels based on their mask values. With the added noises, the preserved shape in the masked images is corrupted. In the examples of Figure 4 (a), the score of `goldfish` drops to 0.44% and 0.39% after the introduction of random noises. At the same time, the masked images with important regions to the prediction preserved will not be affected by the added noises as examples shown in Figure 4 (b). That is because fewest noises are added to the preserved pixels (with mask values closing to 1 in M_i) of the masked images.

3.5 Pixel Coverage Bias

Given a set of masks $\mathbb{M} = \{\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_K\}$, the *coverage frequency* of a pixel x is defined as:

$$\rho(x) = \frac{1}{K} \sum_{i=1}^{K} \mathbf{M}_i(x).$$

Pixel coverage bias (PCB) refers to the phenomenon that different pixels might have different coverage frequencies. It is a severe issue for mask-based explanation, and it has not received any attention. According to Equation (1), the saliency value of a pixel, before normalized to $[0, 1]$, is the sum of the causal impact scores of the 'teams' (masks) of which it is a member. Consequently, the more 'teams' a pixel in, the higher its saliency value. This is clearly not justified.

Adverse Effects of PCB: PCB can lead to nonsensical explanations. Figure 5 show two examples. For each example, a collection of masks is generated according to the procedure described in Section 3.3. The pixel coverage frequencies are shown in (a.1) and (b.1). The saliency maps, as computed using Equation (1), are given in (a.2) and (b.2). We see that they closely resemble the coverage frequency maps, and offer no meaningful explanations for the output labels.

Analysis: To understand why PCB causes the nonsensical explanations, we first note that, in both examples, the correct prediction can be made from various small patches of the input image. This phenomenon is called *causal overdetermination* (White et al. 2021). It is analogous to committee voting where any member's vote kills a proposal. Causal overdetermination is common with ViT models. In fact, it has been observed that the class scores in ViTs are more robust to deletions of small patches from the input image than many popular CNNs (Naseer et al. 2021).

Let $\mu = \frac{1}{K} \sum_{i=1}^{K} s(\mathbf{X}, \mathbf{M}_i)$ and $\beta_i = s(\mathbf{X}, \mathbf{M}_i) - \mu$. We divide the saliency score $S(x)$ of a pixel into two parts:

$$\begin{aligned} S(x) &= \sum_{i=1}^{K} \beta_i \mathbf{M}_i(x) + \sum_{i=1}^{K} \mu \mathbf{M}_i(x) && (3) \\ &= \sum_{i=1}^{K} \beta_i \mathbf{M}_i(x) + \mu N \rho(x). && (4) \end{aligned}$$

In the case of causal overdetermination, the impact score $s(\mathbf{X}, \mathbf{M}_i)$ of most masks are close to 1. Consequently, μ is also close to 1, but β_i is small for most i's. Thus the second term is much larger than the first term. When normalized to the interval $[0, 1]$, the first term basically vanishes. This explains why in Figure 5 the saliency maps closely resemble the coverage frequency maps.

Correction for PCB: A simple way to correct for PCB is to divide the saliency value $S(x)$ by the coverage frequency $\rho(x)$. This results in the *corrected saliency value*:

$$S^c(x) = \frac{S(x)}{\rho(x)} = \sum_{i=1}^{K} s(\mathbf{X}, \mathbf{M}_i) \frac{\mathbf{M}_i(x)}{\rho(x)} \,, \tag{5}$$

where $S^c(x) = 0$ by definition when $\rho(x) = 0$. Intuitively, the corrected saliency value of a pixel is the sum of the causal impact scores of the 'teams' of which it is a member, divided by the number of 'teams' it participates in. Similar to Equation (4), we decompose $S^c(x)$ into two parts:

$$S^c(x) = \sum_{i=1}^{K} \beta_i \frac{\mathbf{M}_i(x)}{\rho(x)} + \mu N. \tag{6}$$

The second term is still much larger than the first term. However, it is a constant and does not depend on the pixel. When the saliency values $S^c(x)$ are normalized to the interval $[0, 1]$ for visualization, the influence of the second term is completely eliminated.[4] This is why meaningful saliency maps emerge in Figure 5 after correcting for PCB.

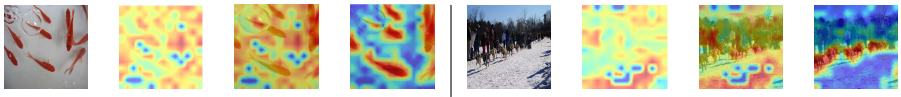
4 Experiments

4.1 Evaluation Metrics

We evaluate ViT-CX following a protocol similar to how previous ViT explanation methods (CGW1, CGW2, TAM) are evaluated, i.e., using deletion AUC, insertion AUC (Petsiuk, Das, and Saenko 2018), the Pointing Game (Zhang et al. 2018) and visual examples. This scheme is also commonly used to evaluate explanation methods for CNNs.

Deletion AUC and Insertion AUC: These two metrics are about the *faithfulness* of an explanation (saliency map) to the target model, i.e., whether pixels with high saliency values are really important to the prediction of the model (Petsiuk, Das, and Saenko 2018). Deletion AUC measures how fast the score of the target class drops when pixels are deleted from the input image in descending order of the saliency values. Insertion AUC measures how fast the score

[4] Recall that the normalization is done using $(S^c(x) - \min_x S^c(x))/(\max_x S^c(x) - \min_x S^c(x))$.



(a) (a.1) (a.2) (a.3) (b) (b.1) (b.2) (b.3)

Figure 5: Impact of PCB: The saliency maps (a.2, b.2) closely resemble the coverage frequency maps (a.1, b.1), and hence offer no meaningful explanations for the target classes (`goldfish` and `dogsled`). After correcting for PCB, the saliency maps (a.3, b.3) become more meaningful.

	ViT-B			DeiT-B			Swin-B		
	Del ↓	Ins ↑	PG Acc ↑	Del ↓	Ins ↑	PG Acc ↑	Del ↓	Ins ↑	PG Acc ↑
ViT-CX	**0.154**	**0.607**	**86.76%**	**0.209**	**0.806**	**87.51%**	**0.255**	**0.769**	**91.26%**
(a) Performance of Baselines									
Rollout	0.251	0.517	60.91%	0.406	0.642	35.70%	—	—	—
Partial LRP	0.239	0.499	66.52%	0.349	0.655	61.25%	—	—	—
CGW1	0.201	0.542	77.14%	0.286	0.717	70.54%	—	—	—
CGW2	0.209	0.549	70.94%	0.271	0.736	70.54%	—	—	—
TAM	0.180	0.556	77.87%	0.240	0.747	75.47%	—	—	—
Grad-CAM	0.212	0.456	50.45%	0.250	0.743	79.24 %	0.356	0.693	88.46%
Integrated-Grad	0.184	0.263	10.61%	0.259	0.362	10.74%	0.420	0.483	7.69%
Smooth-Grad	0.174	0.438	16.96%	0.231	0.528	31.05%	0.369	0.505	14.52%
LIME	0.207	0.572	64.78%	0.312	0.768	59.80%	0.388	0.692	63.53%
Occlusion	0.291	0.571	64.75%	0.380	0.801	59.51%	0.448	0.752	69.65%
RISE	0.234	0.581	73.30%	0.366	0.759	71.84%	0.416	0.727	75.07%
(b) Performance of RISE^c									
RISE^c	0.185	0.589	79.62%	0.279	0.797	80.61%	0.352	0.734	83.41%

Table 1: (a) Main results: **Boldface** and underline indicate best and second best performance respectively, and '—' means not applicable. ViT-CX significantly outperforms the baselines in terms of the faithfulness metrics deletion AUC (Del) and insertion AUC (Ins), and in terms of the interpretability metric Pointing Game Accuracy (PG Acc). (b): Performance of RISE^c. Under our mask aggregation approaches, the explanation quality of RISE^c is improved over RISE but still falls short of ViT-CX.

increases when pixels are inserted into an empty canvas in that order. Smaller deletion AUC and larger insertion AUC indicate a more faithful explanation. Let there be n steps of deletion or insertion, and $\mathbf{X}_r$ be the r-th image in the process. The metrics are defined as: $AUC = \frac{1}{n}\sum_{r=1}^{n} f(\mathbf{X}_r)$.

Pointing Game: This metric is about the *interpretability* of an explanation, i.e., whether it provides qualitative understanding between input and output (Ribeiro, Singh, and Guestrin 2016; Doshi-Velez and Kim 2017). In the Pointing Game, saliency maps from an explanation method are compared with human-annotated bounding boxes, which reflex how well the explanations match the qualitative understanding pre-defined by humans. For each pair of saliency map and bounding box, if the pixel with the highest saliency value falls inside the box, it is considered a hit. Otherwise it is considered a miss. The Pointing Game Accuracy is defined as: $Acc = \#Hits/(\#Hits + \#Misses)$. [5] As a supplement to quantitative metrics, visual examples are often used to demonstrate the interpretability of explanations.

[5]Following Wang et al. (2020), we use only those images where there is only a bounding box and it occupies no more than 50% of the image.

4.2 Experiment Settings

Models and Dataset: Three ViT variants are used in our experiments: (1) ViT-B/16 (Dosovitskiy et al. 2020), the vanilla ViT; (2) DeiT-B/16-Distill (Touvron et al. 2021), a improved version of the vanilla ViT with a distillation token added; (3) Swin-B (Liu et al. 2021), a hierarchical ViT. We use 5,000 images randomly selected from the ILSVRC2012 validation set (Deng et al. 2009). All experiments are run on an Intel Xeon E5-2620 CPU and an NVIDIA 2080 Ti GPU.

Hyper-parameters Setting: Swin-B is a hierarchical model with $B = 4$ stages. ViT-B and DeiT-B are also divided into 4 stages. Masks are generated from stage $b = 2$ to stage 4 and the number of mask for each stage is $k = 128$. The total number of masks is $K = 384$. The impact of b is investigated in **Appendix A**.

Baselines: We compare ViT-CX with three groups of baselines: (a) five attention-based methods, namely Rollout, Partial LRP, CGW1, CGW2 and TAM; (b) three gradient-based explanation methods, namely Grad-CAM, Integrated-Grad and Smooth-Grad; (c) Three other causal explanation methods, namely LIME, Occlusion Map and RISE, in which Occlusion map and RISE are also mask-based explanation methods.

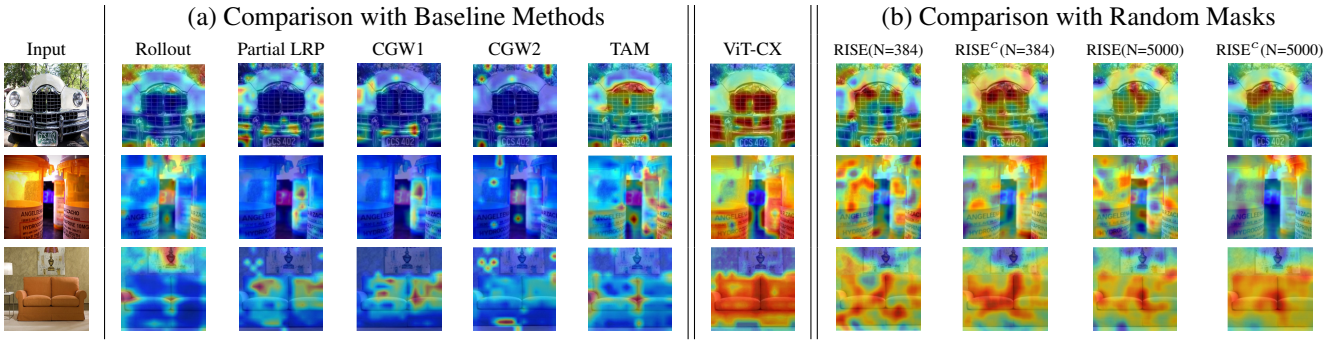


Figure 6: (a) Visual comparisons of ViT-CX and several baselines on explaining the predictions of ViT-B/16 on three examples. Clearly, the regions highlighted by ViT-CX better match what human considers as evidence for the classes; (b) Visual comparison of ViT-CX under ViT masks and RISE(without PCB correction)/RISE^c(with PCB correction) under random masks. It can be seen that even with 5,000 random masks, the explanation quality is still poorer than that of our ViT-CX with only 384 masks (Explained Labels: `Grile`, `Pill Bottle`, `Sofa`).

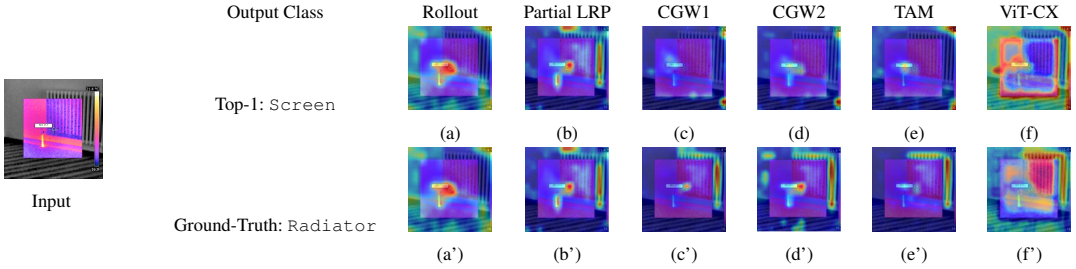


Figure 7: Visual comparisons of ViT-CX and several baselines on revealing what ViT-B/16 considers as the evidence for: (a-f) the predicted label `screen` (9.8%); (a'-f') the ground-truth label `radiator` (4.5%).

4.3 Results

The main results are in Table 1 (a).

Faithfulness: ViT-CX has the lowest deletion AUC values across the board, being 10% lower than the next best. ViT-CX also enjoys the highest insertion AUC values in all cases. Those indicate that ViT-CX is more faithful to the target models than the baselines.

Interpretability: ViT-CX enjoys significantly higher accuracy in the Pointing Game than the baselines in all cases. This implies that the explanation results of ViT-CX are more consistent with the human-annotated bounding boxes.

To supplement the quantitative comparisons, Figures 1 and 6 show several examples to demonstrate the interpretability of the saliency maps for ViT-B by ViT-CX in comparison with those by the baselines. More examples for other ViT models are given in **Appendix B**. In all cases, the regions highlighted by ViT-CX better match human's qualitative understanding of what information should be considered when labeling an image with a particular class.

Figure 7 shows an example that is misclassified. ViT-CX and other methods are used to explain the predicted and the ground-truth classes. Clearly, ViT-CX does a much better job than others in helping users understand why the model makes a mistake. Note that Rollout and Partial LRP yield identical saliency maps in the two cases. The reason is that those two methods are class-agnostic.

Sanity Check: As a causal method, ViT-CX is sensitive to the changes in model parameters and passes the sanity check (Adebayo et al. 2018). See **Appendix C** for details.

4.4 Ablation Study

ViT-CX consists of three key components: (1) A way to generate masks, (2) a method for avoiding unintended artifacts of masking, and (3) a correction for PCB during mask aggregation. In the following, we report ablation studies on the importance of each component.

Mask Generation: ViT-CX uses a set of masks $\mathbb{M}_{cx}$ (with number of masks $K = 384$) from patch embeddings. To compare the explanations under $\mathbb{M}_{cx}$ with those under the randomly generated masks $\mathbb{M}_{rd}$, we have a variant of RISE, termed RISE^c. RISE^c follows the setting of RISE to generate 5,000 masks randomly but uses our proposed Equation (2) to compute the causal impact scores and use Equation (5) to aggregate the masks with PCB correction. Thus, the only difference between ViT-CX and RISE^c is in the mask used - $\mathbb{M}_{cx}(K = 384)$ versus $\mathbb{M}_{rd}(K = 5000)$.

Table 1 (b) shows the scores of the explanations by RISE^c under various quality metrics, and Table 2 shows the average time needed for ViT-CX and RISE^c to explain one image. We see that the masks generated by ViT-CX yield significantly better explanation quality in terms of interpretability and faithfulness. ViT-CX is also much more computationally

	Masks	ViT-B	DeiT-B	Swin-B
ViT-CX	$\mathbb{M}_{cx}(K = 384)$	6.54	7.71	6.29
RISE^c	$\mathbb{M}_{rd}(K = 5,000)$	56.32	65.30	72.23

Table 2: Average time (secs) for explaining one image.

	Del ↓	Ins ↑	PG Acc ↑
ViT-CX	**0.154**	**0.607**	**86.76%**
w/o P	0.169	0.578	80.90%
w/o A	0.196	0.547	74.65%
w/o A&P	0.211	0.531	69.17%

Table 3: Ablation study on the mask aggregation of ViT-CX.

efficient, taking only a few seconds to explain one image, while RISE^c takes more than 1 minute.

Figure 6 (b) also provide visual comparison between explanations under ViT masks and those under the random masks, which indicates that ViT-CX can yield explanations in higher visual quality with much smaller amount of masks (384 ViT masks vs. 5,000 random masks).

Mask Aggregation: Table 3 compares different ablations of ViT-CX on ViT-B/16. In the names of the ablations, *w/o P* means without correcting for PCB, and *w/o A* means without accounting for an unintended artifact of masking. It is clear that both PCB correction and artifact avoidance are important for ViT-CX. Several visual examples are given in **Appendix D** to support this point further.

Besides, the results of RISE and RISE^c in Table 1 indicate that our mask aggregation approaches can also improve the explanation quality of RISE. In **Appendix E**, we show the importance of PCB correction when explaining CNNs.

5 Conclusion

A novel method for explaining ViT models named ViT-CX is proposed. While previous methods use attention weights on patch embeddings and class gradients, ViT-CX is based on patch embeddings themselves. It creates masks from the patch embeddings, determines their causal impact scores on a target output class, and combines the masks using those scores. Empirical evaluations show that ViT-CX does a noticeably better job highlighting the pixels that are important for model output. It also provides a better qualitative understanding of the relationship between input and output. ViT-CX can be used to explain different flavors of ViT models. It can be deployed with ViT model to boost user trust in them.

References

Abnar, S.; and Zuidema, W. 2020. Quantifying Attention Flow in Transformers. In *Annual Meeting of the Association for Computational Linguistics*, 4190–4197.

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, 9525–9536.

Bastings, J.; and Filippova, K. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 149–155.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Chefer, H.; Gur, S.; and Wolf, L. 2021a. Generic Attention-Model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 397–406.

Chefer, H.; Gur, S.; and Wolf, L. 2021b. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 782–791.

Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; and Shen, C. 2021. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34: 9355–9366.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 248–255. Ieee.

Doshi-Velez, F.; and Kim, B. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International conference on learning representations*.

Fong, R.; Patrick, M.; and Vedaldi, A. 2019. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2950–2958.

Fong, R. C.; and Vedaldi, A. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3429–3437.

Gunning, D.; and Aha, D. W. 2019. DARPA's explainable artificial intelligence program. *AI Magazine*, 40(2): 44–58.

Jain, S.; and Wallace, B. C. 2019. Attention is not Explanation. In *Proceedings of NA Annual Meeting of the Association for Computational Linguistics-HLT*, 3543–3556.

Li, J.; Monroe, W.; and Jurafsky, D. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, 4765–4774.

Miller, T. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267: 1–38.

Naseer, M.; Ranasinghe, K.; Khan, S.; Hayat, M.; Khan, F.; and Yang, M.-H. 2021. Intriguing Properties of Vision Transformers. In *Advances in Neural Information Processing Systems*.

Paul, S.; and Chen, P.-Y. 2022. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2071–2081.

Petsiuk, V.; Das, A.; and Saenko, K. 2018. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *Proceedings of the British Machine Vision Conference*.

Pruthi, D.; Gupta, M.; Dhingra, B.; Neubig, G.; and Lipton, Z. C. 2020. Learning to Deceive with Attention-Based Explanations. In *Annual Meeting of the Association for Computational Linguistics*, 4782–4793.

Raghu, M.; Unterthiner, T.; Kornblith, S.; Zhang, C.; and Dosovitskiy, A. 2021. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34: 12116–12128.

Rebuffi, S.-A.; Fong, R.; Ji, X.; and Vedaldi, A. 2020. There and back again: Revisiting backpropagation saliency methods. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8839–8848.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 618–626.

Serrano, S.; and Smith, N. A. 2019. Is Attention Interpretable? In *Annual Meeting of the Association for Computational Linguistics*, 2931–2951.

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*.

Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, 3319–3328. PMLR.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 10347–10357. PMLR.

Tuli, S.; Dasgupta, I.; Grant, E.; and Griffiths, T. 2021. Are Convolutional Neural Networks or Transformers more like human vision? In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43.

Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; and Titov, I. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Annual Meeting of the Association for Computational Linguistics*, 5797–5808. Annual Meeting of the Association for Computational Linguistics Anthology.

Wang, H.; Wang, Z.; Du, M.; Yang, F.; Zhang, Z.; Ding, S.; Mardziel, P.; and Hu, X. 2020. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshop*, 24–25.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 568–578.

White, A.; Ngan, K. H.; Phelan, J.; Afgeh, S. S.; Ryan, K.; Reyes-Aldasoro, C. C.; and Garcez, A. d. 2021. Contrastive Counterfactual Visual Explanations With Overdetermination. *arXiv preprint arXiv:2106.14556*.

Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.-H.; Tay, F. E.; Feng, J.; and Yan, S. 2021a. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 558–567.

Yuan, T.; Li, X.; Xiong, H.; Cao, H.; and Dou, D. 2021b. Explaining Information Flow Inside Vision Transformers Using Markov Chain. In *eXplainable AI approaches for debugging and diagnosis.*

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhang, J.; Bargal, S. A.; Lin, Z.; Brandt, J.; Shen, X.; and Sclaroff, S. 2018. Top-down neural attention by excitation backprop. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 126(10): 1084–1102.

Zhang, Z.; Zhang, H.; Zhao, L.; Chen, T.; Arik, S. Ö.; and Pfister, T. 2022. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 3417–3425.

Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6881–6890.