Attention-based Class Activation Diffusion for Weakly-Supervised Semantic Segmentation

Jianqiang Huang, Jian Wang, Qianru Sun and Hanwang Zhang

1 Nanyang Technological University, Singapore
jianqiang.jqh@gmail.com, hanwangzhang@ntu.edu.sg
2 DAMO Academy, Alibaba Group, China
xuefei.wj@alibaba-inc.com
3 Singapore Management University, Singapore
qianrusun@smu.edu.sg

Abstract. Extracting class activation maps (CAM) [49] is a key step for weakly-supervised semantic segmentation (WSSS). The CAM of convolution neural networks fails to capture long-range feature dependency on the image and result in the coverage on only foreground object parts, i.e., a lot of false negatives. An intuitive solution is "coupling" the CAM with the long-range attention matrix of visual transformers (ViT) [9]. We find that the direct "coupling", e.g., pixel-wise multiplication of attention and activation [11], achieves a more global coverage (on the foreground), but unfortunately goes with a great increase of false positives, i.e., background pixels are mistakenly included. This paper aims to tackle this issue. It proposes a new method to couple CAM and Attention matrix in a probabilistic Diffusion way, and dub it AD-CAM. Intuitively, it integrates ViT attention and CAM activation in a conservative and convincing way. Conservative is achieved by refining the attention between a pair of pixels based on their respective attentions to common neighbors, where the intuition is two pixels having very different neighborhoods are rarely dependent, i.e., their attention should be reduced. Convincing is achieved by diffusing a pixel’s activation to its neighbors (on the CAM) in proportion to the corresponding attentions (on the AM). In experiments, our results on two challenging WSSS benchmarks PASCAL VOC and MS COCO show that AD-CAM as pseudo labels can yield stronger WSSS models than the state-of-the-art variants of CAM.

Keywords: semantic segmentation, class activation map, visual transformer, self-attention

1 Introduction

Semantic segmentation aims to infer the object label on each pixel of an image. Its fully-supervised labels are expensive to obtain. A lot of recent research is thus focused on the training settings with only coarse or weak labels such as bounding boxes [7,34], scribbles [30,36] and image-level class labels [19,1,42,45,28,29,6]. This paper aims to tackle the last setting which is the most challenging one, but
Fig. 1. CAM heatmaps (i.e., seeds of WSSS) generated by different methods: 1) CNN column is by ResNet-50 [49]; 2) ViT column shows the model attention from visual transformer [9] without distinguishing classes; 3) TS-CAM column combines attention and activation on a cascaded CNN-ViT architecture [11] which is a close related work to ours; and 4) our AD-CAM column shows our method results based on a double-branch CNN-ViT architecture [33]. For each individual class, we display the statistics of false positive (FP) and false negative (FN) pixels on 0-1 masks (generated by applying the optimal threshold of each method on the heatmaps). In experiments, we also compare TS-CAM and AD-CAM on the same network architecture.

calls the task simply as weakly-supervised semantic segmentation (WSSS). The prevailing pipeline of WSSS using image labels includes a few steps: 1) loading a large-scale classification model pre-trained e.g. on ImageNet [8]; 2) fine-tuning the model on the WSSS dataset where each sample has an image and a multi-hot label that represents co-occurring classes on the image; 3) extracting a class activation map (CAM) [49] for each occurring class, and then hard-thresholding it to a 0-1 mask (note that it is optional to refine the mask via erosion and expansion techniques [11,23]); and 4) taking such masks as pseudo labels to train the semantic segmentation model in a standard fully-supervised pipeline [3,4,41]. The quality of the segmentation model thus depends on two major factors: the multi-label classification model and the extraction method of CAM. In this work, we inspect the problems in existing CNN- and ViT-based classification models when they are used to extract CAM.

Problems. Figure 1 demonstrates example CAM heatmaps generated by using ResNet-50, ViT [9], TS-CAM [11] on a cascaded CNN-ViT architecture, and our AD-CAM on a two-branch CNN-ViT architecture [33], respectively. For each individual class, we also show the statistics of the false positive (FP) and false negative (FN) pixels on 0-1 masks (generated by applying the optimal hard threshold of each method on the heatmaps). We have several observations from Figure 1. 1) CNN has local activations and its 0-1 mask suffers from a great number of FN pixels, e.g., its activations on dog are mainly on the dog head and its FP rate is significantly higher than others. 2) ViT based on long-range
attention sheds light on activating more regions on the image (than CNN). However, its effectiveness of reducing FN goes along with the increase of FP and it does not distinguish classes. 3) TS-CAM, using a basic operation (i.e., dot product) to couple the activation maps of CNN and ViT, does not solve the key problem. Thanks to the coupling with CNN, TS-CAM can distinguish classes. Yet, it still has to trade off FN and FP. For example, its FP rates increase greatly (over CNN) for tv monitor, car and bicycle, and slightly for dog, when it reduces FN. The visual examples show TS-CAM has a lower FP than ViT.

Motivations. From Figure 1, we further inspect the activated pixels of ViT that contain all semantics on the image without distinguishing classes. We find a large number of long-range but spurious correlations either between labeled objects such as dog and tv monitor and non-labeled objects such as the slipper beside the dog and the keyboard beside tv monitor, or between co-occurring labeled objects such as person (especially the helmet) and bicycle in the bottom example. In many cases, they even dominate the coupling results if the coupling operation is straight-forward. For example, after TS-CAM, the activation on the tv monitor image is almost overtaken by the keyboard, and the bicycle case is overtaken by the helmet. We think the crux behind has two aspects. 1) ViT classifiers tend to use every cue of the object image, no matter how far it is, to make the prediction even through some of cues are from the context. 2) Using basic coupling operations, e.g., dot product, can not prevent the spurious and dominating cues from merging into CAM.

Solutions. To solve these issues, we introduce two novel modules in our AD-CAM method, accordingly. First, we propose a new measurement on the confidence of ViT attention, and call it co-neighbor similarity. It indicates the attention between two pixels is confident only when their respective neighborhoods are similar, i.e., sharing a great number of nearest neighbors. This makes AD-CAM (when using this attention) more conservative and prudent against FP attentions. Second, we propose a more convincing operation (than dot product) to couple the attention of ViT to the CAM of CNN. We diffuse each pixel’s class activation value to its nearest neighbors on the CAM, in proportional to the corresponding values on the attention matrix. Intuitively, each pixel computes its diffusion probabilities to nearest neighbors based on attentions. Each probability determines how much activation to diffuse from the pixel to this neighbor. After n steps of diffusion, where n is a hyperparameter, we obtain the final class activation maps of AD-CAM. Then, we proceed the following steps of WSSS: refining seed masks; training semantic segmentation models; and evaluating them.

4 Please note that the activation map by ViT models is generated by calculating the attention matrix between classification and all tokens and then upsampling the attention matrix ($14 \times 14$ in our case) to the image size ($H \times W$).

5 Please note that our demonstration results in Figure 1 are all based on the optimal thresholds that are carefully chosen. We leave the implementation details in supplementary materials.

6 AD-CAM is the name of method as well as the name of the produced class activation maps using the method.
In summary, our technical contributions are two folds. 1) A new co-neighbor similarity to measure the confidence of attention produced by ViT. This helps to alleviate false positive issues. 2) A new activation diffusion operation to incorporate the attention of ViT to expand the CAM of CNN (note that here the attention indicates the refined version by using co-neighbor similarity). It solves the issue of false negative in CAM and maintains a low false positive rate when incorporating long-range attentions. It yields more convincing results than using doc product.

2 Related Work

The training of multi-label classification and semantic segmentation models are almost uniform in the related works of WSSS [44,23,21]. Below, we focus on the introduction of their particular steps of generating or refining CAM.

CAM Generation and Refinement. Vanilla CAM [49] is the pioneer work of generating the activation map for a specific class. It uses the weight of the last FC layer (i.e., the classifier in an end-to-end CNN model) to average the feature maps and then generate a mask through spatial-wise normalization and hard thresholding (see Section 3). Such masks are usually taken as seeds, and a lot of follow-up works are focused on seed refinement. GAIN [26] applies the CAM on original images to generate masked images, and minimizes the model prediction scores on them, forcing the model to capture the features in other regions (outside the current CAM) in the new training. A similar idea was used in erasing-based methods [39,20,15,47]. Score-CAM [37] proposed to replace the FC weights used in vanilla CAM with a new set of scores predicted from the images masked by channel-wise (not class-specific) activation maps. EDAM [40] and ReCAM [6] proposed to improve the CAM by adding an extra classifier. One category of refinement methods [2,1,42,5] propagate the object regions in the seed to semantically similar pixels in the neighborhood. It is achieved by the random walk [32] on a transition matrix where each element is an affinity score. The related methods have different designs of this matrix. PSA [2] is an AffinityNet to predict semantic affinities between adjacent pixels. IRN [1] is an inter-pixel relation network to estimate class boundary maps based on which it computes affinities. Another method is BES [5] that learns to predict boundary maps by using CAM as pseudo ground truth. Our AD-CAM also apply a diffusion strategy, but we use the distilled attention matrix to capture the pixel-pixel affinity without introducing any additional networks. The CAM generated by CAM can be further refined by existing refinement methods. Another category of refinement methods [43,16,27,21,46,18] utilize saliency maps [14,48]. EPS [25] proposed a joint training strategy to combine CAM and saliency maps. EDAM [40] introduced a post-processing method to integrate the confident areas in the saliency map into CAM. A more recent category of methods leverage iterative post-processing to refine CAM. OOA [17] ensembles the CAM generated in multiple training iterations. CONTA [44] iterated through the whole process of WSSS including a sequence of model training and inference. AdvCAM [23]
used the gradients with respect to the input image to perturb the image, and iteratively find newly activated pixels.

**CAM Generation based on Attention (of ViT).** It is a known problem that using CNN to extract CAM fails to mask the complete object regions, i.e., suffers from a great number of false negative pixels in the results. ViT is different with CNN and can capture long-range dependencies on the input image [9]. A pioneer method of combining the self-attention of ViT with CNN (to extract CAM) is called TS-CAM [11]. Its key operation is multiplying CAM by the class-wise attention maps (extracted by using class tokens). This naive pixel-wise operation however results a lot of false positive in the results. Our AD-CAM also uses the attention of ViT to improve CAM. We highlight two differences with TS-CAM. 1) We propose a new method of using the direct output of ViT, i.e., self-attention matrix, in AD-CAM. While TS-CAM needs to compute an additional “attention activation map” using class token and image patch token. Therefore, ours is more generic to different ViT models. 2) Our AD-CAM aims for tackling WSSS tasks. While TS-CAM was for weakly-supervised object localization (WSOL) tasks which use single object images and the result models can not be adapted for WSSS (because the class token in multi-label ViT do not distinguish co-occurring classes on the image). A more recent work GETAM [35] is the first ViT-based approach for WSSS. It improves CAM with better object shapes by coupling the “attention activation map” and its class-wise gradient maps. Its method of using self-attention is different with ours. We compare to it in experiments.

### 3 Preliminaries: A Pipeline of Training WSSS Models

**Seed Generation.** Extracting CAM [49] is based on a trained multi-label classification model. The first step is thus to train this model. It is an end-to-end training on the network consisted of a feature extractor (backbone), a global average pooling (GAP) layer, and a class prediction layer (e.g., an FC layer on ResNet-50 [13]). The loss used for training is usually a binary cross-entropy (BCE) loss:

\[
\mathcal{L}_{bce} = -\frac{1}{K} \sum_{k=1}^{K} y[k] \log \sigma(\hat{y}[k]) + (1 - y[k]) \log [1 - \sigma(\hat{y}[k])],
\]  

where \(\hat{y}[k]\) denotes the prediction logit for the \(k\)-th class, \(\sigma(\cdot)\) is the Sigmoid function, and \(K\) represents the total number of object classes on WSSS dataset. \(y[k] \in \{0, 1\}\) is the image-level label for the \(k\)-th object class, where 1 denotes the class is present in the image and 0 otherwise.

When the model is trained, each training image \(x\) is again fed into it to extract CAM for the \(k\)-th class:

\[
{\text{CAM}}_k(x) = \frac{\text{ReLU}({F}_k)}{\max(\text{ReLU}({F}_k))}, \quad F_k = w_k^T f(x),
\]
Our training pipeline is consisted of three steps. (a) Pre-training the backbone of two branches: one for CNN and one for ViT. We directly load the trained weights published in [33]. Please note we apply this backbone also for related works to keep a fair comparison. (b) Training a multi-label classification model on the WSSS datasets such as MS COCO [31]. (c) Extracting AD-CAM, i.e., seed masks for WSSS. We highlight our proposed modules in the orange box. “AM” stands for attention matrix.

where $w_k$ denotes the classification weights of the $k$-th class, and $f(x)$ denotes the feature maps before GAP. Please note that, for simplicity, we assume the classification head of the model is always a single FC layer, and use $w$ to denote its learned weights.

**Pseudo Mask Generation.** Taking CAM as seed, there are mainly two options to produce pseudo masks: 1) thresholding CAM to be 0-1 masks, and 2) refining CAM with IRNet [1] and then thresholding the refined CAM to be 0-1 masks.

**Semantic Segmentation.** This is the last step of WSSS. The generated pseudo masks are used to train a standard semantic segmentation model, e.g., any DeepLab variants [3, 4], in a fully-supervised fashion. The objective function is:

$$L_{ss} = - \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{k=1}^{K+1} \frac{1}{k+1} y_{i,j}[k] \log \frac{\exp(\hat{y}_{i,j}[k])}{\sum_{k} \exp(\hat{y}_{i,j}[k])},$$

where $y_{i,j}$ and $\hat{y}_{i,j}$ denote the label and the prediction logit at pixel $(i, j)$, respectively. $y_{i,j}[k]$ and $\hat{y}_{i,j}[k]$ denote the $k$-th element of $y_{i,j}$ and $\hat{y}_{i,j}$, respectively. $h$ and $w$ are the height and width of the image. $K$ is the total number of classes in WSSS dataset. $K+1$ means taking the background as one more “class”.

Fig. 2. Our training pipeline is consisted of three steps. (a) Pre-training the backbone of two branches: one for CNN and one for ViT. We directly load the trained weights published in [33]. Please note we apply this backbone also for related works to keep a fair comparison. (b) Training a multi-label classification model on the WSSS datasets such as MS COCO [31]. (c) Extracting AD-CAM, i.e., seed masks for WSSS. We highlight our proposed modules in the orange box. “AM” stands for attention matrix.
4 AD-CAM

The overview of our approach is shown in Figure 2. It does not include the steps of seed refinement and semantic segmentation model training as they are standard as in related works [6,24]. The architecture of AD-CAM is based on CNN and ViT, and is called Conformer [33]. The procedure to deriving AD-CAM has three steps. Step 1: Pre-training the backbone of Conformer using a standard multi-class classification pipeline on ImageNet [8]. To save the computational cost, we directly load the trained weights published in [33]. Please note we apply this stronger backbone also for related works, such as CAM [49] and TS-CAM [11], to keep a fair comparison. Step 2: Training a multi-label classification model on WSSS datasets, as formulated in Section 3, based on the pre-trained Conformer. Step 3: Extracting AD-CAM for all training images. In Figure 2 (c), we highlight our two new modules in the orange box. “AM” stands for the attention matrix of ViT. In the following, we elaborate Step 2 and Step 3 in three subsections.

4.1 Training Multi-Label Classifiers

**Feature Extraction.** Given an input image \( x \) and its multi-hot class label \( y \in \{0,1\}^K \), \( K \) is the total number of foreground classes in the dataset. We denote the output of CNN feature encoder as \( f_{\text{CNN}}(x) \in \mathbb{R}^{h \times w \times C} \) and the classification token of ViT as \( f_{\text{ViT}}(x) \in \mathbb{R}^{1 \times C} \). \( C \) denotes the number of channels. \( h \) and \( w \) are height and width of feature map, respectively.

**Prediction.** The prediction on the CNN branch is formulated as:

\[
\hat{y}_{\text{CNN}} = FC_1(\text{GAP}(f_{\text{CNN}}(x))),
\]

(4)

The prediction on the ViT branch is formulated as:

\[
\hat{y}_{\text{ViT}} = FC_2(f_{\text{ViT}}(x)).
\]

(5)

where \( FC_1 \) and \( FC_2 \) denote classification heads.

**Optimization.** We use these predictions to compute the binary cross entropy (BCE) losses, as in Eq. (1), for two branches respectively. We highlight that when backpropagating gradients, we detach the interaction between two branches, as shown by a red cross in Figure 2 (b). Our aim is to enlarge the “difference” between two branches that may compensate each other in the final coupling step. We evaluate this in Section 5.3.

4.2 Extracting CAM and AM

As shown in Figure 2 (c), we need to extract CAM from CNN branch and AM from ViT branch, before the orange box.

**CAM.** This is similar to the case in an independent convolutional backbone. Following Eq. (2), we extract \( \text{CAM}_k(x) \in \mathbb{R}^{h \times w} \) for each class \( k \) given the feature \( f_{\text{CNN}}(x) \) and the classifier weights \( w_k \) on the CNN branch. For brevity, we denote the CAM\(_k(x)\) as \( M_k \) omitting the notation of input image.
For the \(i\)-th transformer module, we use \(A_i \in \mathbb{R}^{(hw) \times (hw)}\) to represent the average results over its multi-head attentions. Each value on this matrix denotes the correlation between two feature pixels (or more precisely, two image patches). We aggregate the AMs of all transformer modules as \(A = \frac{1}{L} \sum_{i=1}^{L} A_i\), where \(L\) is the total number of transformer modules on the ViT branch.

### 4.3 Generating AD-CAM

This step is illustrated in the orange box of Figure 2 (c). We have two novel modules to generate AD-CAM.

**Co-Neighbor AM Refinement.** As inspected in Figure 1 and mentioned in Section 1, the AM often contain false dependencies, and these dependencies may corrupt vanilla CAM especially when the coupling method (of AM and CAM) is improper. To alleviate the false dependencies, we introduce a new measurement on the confidence of attention, called co-neighbor similarity. It indicates the attention between two pixels is confident only when their respective neighborhoods are similar, i.e., sharing a great number of nearest neighbors. In the following, we first elaborate the definition of this similarity and then introduce how to use it to refine AM.

For a pair of feature pixels \(i\) and \(j\), their co-neighbor similarity is defined as:

\[
S_{ij} = \sum_k \sqrt{A_{ik} A_{jk}},
\]

In \(A\), the sum on each row equals to 1. Thus, we have \(S_{ij} \in [0, 1]\) by Cauchy-Schwarz inequality and \(S_{ij} = 1\) if and only if \(A_{ik} = A_{jk}\) for any \(k\).

**Theorem 1.** If pixel \(i\) and pixel \(j\) has \(n\) neighbors with equal attention \(\frac{1}{n}\), then we have

\[
S_{ij} = \frac{2 \text{Jaccard}(i, j)}{1 + \text{Jaccard}(i, j)},
\]

where \(\text{Jaccard}(i, j)\) is the Jaccard similarity between \(i\) and \(j\).

By theorem 1 we can see that \(S_{ij}\) is a generalization of Jaccard similarity, which implies that \(S_{ij}\) is a valid metric by considering neighboring information.

Then, based on \(S\), we conduct the co-neighbor AM refinement as follows,

\[
\tilde{A} = \text{NORM}(\text{TOP}(S, k) \otimes A),
\]

where \(\text{TOP}(S, k)\) is an operation to search the top-\(k\) nearest neighbor pixels for each pixel, based on the similarities in \(S\). \(\otimes\) denotes Hadamard product. \(\text{NORM}\) is to make each row of \(\tilde{A}\) sum up to 1. In \(\tilde{A}\), each value is refined to be more confident (than that in \(A\)) based on its neighborhood distribution.

**Attention-based CAM Diffusion.** Each row of \(\tilde{A}\) sum up to 1. We can build a diffusion process based on \(\tilde{A}\). Our aim is to diffuse the activation of each pixel
Neighbors

\[ M_k(i)^{(t+1)} = \sum_j a_{ji} M_k(i)^{(t)}, \quad (8) \]

where \( M_k^{(0)} \) is the initial (vanilla) CAM of the \( k \)-th class. During diffusion, each pixel \( i \) diffuses its own activation to neighbors and also receives activation from the pixels which take \( i \) as nearest neighbors in \( \hat{A} \). After \( T \) steps of diffusion, we obtain the AD-CAM \( M'_k \) where \( k = 1, 2, \cdots, K \) and \( T \) is a hyperparameter.

### 4.4 Refining AD-CAM (Optional)

In recent works of WSSS, it is almost a standard step to leverage IRNet \(^1\) to refine the mask before training any segmentation models. In our case, we can take AD-CAM \( M' \) as the input and train an IRNet to estimate the class boundary maps \( B \). We omit the training details of IRNet for brevity. Then as in \(^1\), we apply a random walk to refine AD-CAM with \( B \) and a transition probability matrix \( T \):

\[ \text{vec} \left( \hat{M}_k \right) = T^l \cdot \text{vec} \left( M'_k \otimes (1 - B) \right), \quad (9) \]

where \( l \) is the number of iterations and \( \text{vec}(\cdot) \) denotes vectorization. Finally, we use \( \{\hat{M}_k\} \), where \( k \) denotes every positive class in the image, as pseudo masks to train semantic segmentation models in a supervised way.

## 5 Experiments

### 5.1 Datasets and Settings

**Datasets.** We use two commonly used datasets to evaluate our method: PASCAL VOC 2012 \(^10\) and MS COCO 2014 \(^31\). VOC contains 20 foreground object classes and 1 background class. It has 1,464, 1,449, and 1,456 samples
in train, val, and test sets, respectively. In our estimation, there are around 59.1% single-label images, i.e., with one-hot labels, on this dataset. Following our related works [1,44,23,6], we exploit an extra training set with 10,582 training images that are provided by Harlharen et al. [12]. MS COCO has 80 object classes and 1 background class. Its train and val sets contains 80k and 40k samples, respectively. Compared to VOC, this dataset is more challenging: 1) it has more classes in total; and 2) there are only 20.3% single-label images, i.e., with one-hot labels, and the average object number per image is 2.9. Please note that on both datasets, we use the samples with only image-level labels for training, i.e., the most challenging setting of WSSS.

Evaluation Metrics. We have mainly two evaluation steps. Mask Generation: We generate pseudo masks, including Seed Masks and final Pseudo Masks (also called Refined Masks by IRNet [1]), for the images in the train set and use corresponding ground truth masks to compute mIoU. Semantic Segmentation: We train the segmentation model using Pseudo Masks, use the resultant model to predict masks for the images in val or test sets, and compute mIoU based on ground truth masks. We provide the results of F1 and pixel accuracy in the supplementary materials.

Network Architectures. In the step of seed mask generation, we use Conformer [33] as backbone, and drop the last Transformer and Conv block to make its output feature map size to be $\frac{1}{16}$ of image size. This is to match the feature resolution as in conventional CNN-based CAM variants [13]. In the step of semantic segmentation, we employ ResNet-101 based DeepLabV2, pre-trained on ImageNet, following the common practice in related works [1,47,24].

Implementation Details. As shown in Figure 2 (b), we fine-tune the Conformer on multi-label classification tasks, e.g., on VOC and MS COCO. In this step, we use AdamW optimizer with 1) a constant learning rate of 2e-4 on the ViT branch, and 2) a polynomial decay strategy from the initial learning rate of 5e-4 on the CNN branch. It takes in total 5 epochs in this step (much shorter than in the pre-training step). We use the same techniques of image pre-processing and augmentation technique as in IRN [1]. The only difference is that we crop each input image $384 \times 384$ pixels to adapt the pre-trained model.

As shown in Figure 2 (c), it is the step of generating AD-CAM. We have two key hyperparameters. 1) For the top-$k$ nearest neighbors in co-neighbor AM refinement, we set $k$ as 50. 2) For the attention-based CAM diffusion, we set the number of diffusion steps $T$ as 2.

Please note that we put the implementation details of training semantic segmentation models in the supplementary materials.

5.2 Results and Analyses

Evaluating Seeds and Pseudo Masks. Table 1 compares our initial seeds and pseudo labels with that obtained by state-of-the-art methods. Among them, AdvCAM [23], RIB [21] and ReCAM [6] are based on ResNet-50, TS-CAM [11] and GETAM [35] are based on ViT. Both seeds and pseudo labels are generated from the training images of the PASCAL VOC and MS COCO datasets. For
Table 1. Comparing our method with baselines in terms of seed, pseudo label (P.L.) and the final segmentation result mIoU (%) on VOC dataset (the results of MS COCO are in supplementary file). * denotes the backbone for segmentation is transformer. ** denotes the backbone of DeepLabV2 for VOC has been pre-trained on MS COCO, which applies to “Seg(val)” and “Seg(test)” columns.

| Methods               | Seed P.L. | Seg(val) | Seg(test) |
|-----------------------|-----------|----------|-----------|
| CAM(ResNet-50)        | 4913      | 48.8     | 66.3      | 64.8      |
| AdvCAM [23]           |           | 55.6     | 69.9      | 68.1      | 68.0      |
| RIB [21]              |           | 56.5     | 70.6      | 68.3      | 68.6      |
| ReCAM [6]             |           | 54.8     | 70.9      | 68.7      | 68.5      |
| CAM(Conformer)        | 4933      | 54.7     | 67.3      | 65.7      | 66.6      |
| AD-CAM                |           | 63.5     | 71.7      | 69.0      | 69.4      |
| AD-CAM**              |           | 63.5     | 71.7      | 70.7      | 70.7      |

the initial seed, we report the best results by applying a range of thresholds to separate the foreground and background in the CAM \( \hat{M}_k \), as following to SEAM [35]. It is clear to see that the initial seeds of our AD-CAM is much better than the CAM generated from the convolution head of Conformer, e.g., 8.4% higher on VOC, which implies that our attention diffusion procedure is essential. And AD-CAM also outperforms a wide range of state-of-the-art methods. For all of the methods, the pseudo labels (P.L.) are obtained by a general process of IRN [1]. Without doubts, AD-CAM achieves the best performance.

Evaluating WSSS Models on VOC. In the middle columns of Table 1, we demonstrate the val and test results of WSSS models trained on pseudo masks. Please note that we report only the results on VOC in the main paper, and put the results on MS COCO in the supplementary materials.

Using our AD-CAM achieves 69.0% and 69.4% for PASCAL VOC val and test, respectively. Both are clearly This is significantly better than the top baselines ReCAM and RIB. Note that GETAM adopted DeepLabV2 that was pre-trained on MS COCO, we also apply the pre-trained DeepLabV2 to AD-CAM, it can be seen that AD-CAM is still better than GETAM.

Efficiency of AD-CAM.

Table 2 shows the consumption time of compared methods on VOC. We have the following two observations: 1) Compared with ResNet-50 based methods, AD-CAM uses a slight larger backbone Conformer-S (i.e., 25.6M vs 37.7M), but yields similar computa-

| Methods       | Backbone | Params | time(ut) |
|----------------|----------|--------|----------|
| CAM [1]       | ResNet-50| 25.6M  | 1.0      |
| AdvCAM [23]   | ResNet-50| 25.6M  | 316.3    |
| ReCAM [6]     | ResNet-50| 25.6M  | 1.9      |
| GETAM [35]    | ViT-base | 86M    | 23.4     |
| AD-CAM        | Conformer-S| 37.7M  | 2.2      |
tion time compared with ReCAM, i.e., 2.2ut vs. And AD-CAM achieves significantly better seeds (63.5 vs 54.8), a small reduction in efficiency is worthwhile. 2) Compared with ViT-based method, GETAM uses a stronger ViT-base model, which has 86M parameters, but AD-CAM generates better seeds. This also demonstrates the superiority of AD-CAM.

5.3 Ablation Study

We perform extensive ablation study on VOC dataset to investigate the effects of different modules of AD-CAM. **Effect of Co-neighbor AM Refinement.** As seen in Table 3, it achieves even worse seeds than standard CAM if we use attention-based CAM diffusion only (54.6 vs 54.7). It can result in more false positive pixels since the initial AM contains many spurious attentions.

**Effect of Stop-gradient.** In multi-label training, the back propagations between CNN branch and ViT branch are switched off to achieve initial CAM with fewer false positive pixels. To test the effect of this stop-gradient strategy, we also conduct a standard version AD-CAM*. We can see that AD-CAM* failed to generate comparable CAM, since the crosstalk between two branches will make CNN branch focusing on more global information, thus introduce more false positive pixels.

| Methods   | mIoU | FP  | FN  |
|-----------|------|-----|-----|
| CAM       | 54.7 | 29.5| 17.8|
| TS-CAM    | 55.2 | 27.4| 18.4|
| Att-Diffu Only | 54.6 | 26.4| 20.3|
| AD-CAM*   | 51.7 | 28.8| 20.8|
| AD-CAM    | 63.5 | 20.9| 16.6|

5.4 Sensitivity Analysis

The number of selected top-k attentions based on co-neighbor similarity and diffusion steps T are two important hyper-parameters in our method. We perform sensitivity analysis on VOC dataset by varying k from 20 to 100 and T from 1 to 8. The mIoU, FP and FN are provided in Figure 4. (1) The mIoU increases rapidly over k with small value, since more and more long-range attentions are included that leads to a significant drop of false negatives (see green line). The performance is relatively stable when $k \geq 50$, so we choose $k$ as 50. (2) The mIoU reaches maximum when $T = 2$, and decreases rapidly when the diffusion process goes further since more false positive pixels are introduced (see orange line).
Fig. 4. Sensitivity analysis. mIoU, FP and FN are provided. (a) Number of selected top-k attentions based on co-neighbor similarity. (b) Diffusion steps $T$.

Fig. 5. Qualitative Results. CAM heatmaps (i.e., seeds of WSSS) generated by different methods: 1) CNN column is by Conformer conv branch; 2) TS-CAM on Conformer; 3) AD-CAM* means attn diffusion only; and 4) our AD-CAM column shows our method results based on a double-branch CNN-ViT architecture [33]. For each individual class, we display the statistics of false positive (FP) and false negative (FN) pixels on 0-1 seed masks (generated by applying the optimal threshold of each method on the heatmaps).

6 Conclusion

Recent works proposed to couple CAM with the long-range attention matrix of visual transformers (ViT) to reduce the false negatives. We find that the attention map of ViT usually contains spurious dependencies, thus leads to a great increase of false positives. To address this issue, we propose AD-CAM to
integrate the ViT attention and CAM activation in a probabilistic diffusion way. It first refines the initial attention map by introducing co-neighbor similarity, which take the neighboring information into consideration. Then the attention-based CAM diffusion is achieved by diffusing a pixel’s activation to its neighbors in proportion to the corresponding attentions. Extensive experiments on two challenging WSSS benchmarks PASCAL VOC and MS COCO demonstrates the superiority of AD-CAM.
References

1. Ahn, J., Cho, S., Kwak, S.: Weakly supervised learning of instance segmentation with inter-pixel relations. In: CVPR. pp. 2209–2218 (2019)
2. Ahn, J., Kwak, S.: Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In: CVPR. pp. 4981–4990 (2018)
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI 40(4), 834–848 (2017)
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV. pp. 801–818 (2018)
5. Chen, L., Wu, W., Fu, C., Han, X., Zhang, Y.: Weakly supervised semantic segmentation with boundary exploration. In: ECCV. pp. 347–362 (2020)
6. Chen, Z., Wang, T., Wu, X., Hua, X.S., Zhang, H., Sun, Q.: Class re-activation maps for weakly-supervised semantic segmentation. In: CVPR (2022)
7. Dai, J., He, K., Sun, J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: ICCV. pp. 1635–1643 (2015)
8. 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)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
10. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV 88(2), 303–338 (2010)
11. Gao, W., Wan, F., Pan, X., Peng, Z., Tian, Q., Han, Z., Zhou, B., Ye, Q.: Ts-cam: Token semantic coupled attention map for weakly supervised object localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2886–2895 (2021)
12. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV. pp. 991–998. IEEE (2011)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
14. Hou, Q., Cheng, M.M., Hu, X., Borji, A., Tu, Z., Torr, P.H.: Deeply supervised salient object detection with short connections. In: CVPR. pp. 3203–3212 (2017)
15. Hou, Q., Jiang, P.T., Wei, Y., Cheng, M.M.: Self-erasing network for integral object attention. In: NeurIPS. pp. 549–559 (2018)
16. Huang, Z., Wang, X., Wang, J., Liu, W., Wang, J.: Weakly-supervised semantic segmentation network with deep seeded region growing. In: CVPR. pp. 7014–7023 (2018)
17. Jiang, P.T., Hou, Q., Cao, Y., Cheng, M.M., Wei, Y., Xiong, H.K.: Integral object mining via online attention accumulation. In: ICCV. pp. 2070–2079 (2019)
18. Kim, B., Han, S., Kim, J.: Discriminative region suppression for weakly-supervised semantic segmentation. In: AAAI. vol. 35, pp. 1754–1761 (2021)
19. Kolesnikov, A., Lampert, C.H.: Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In: ECCV. pp. 695–711 (2016)
20. Kweon, H., Yoon, S.H., Kim, H., Park, D., Yoon, K.J.: Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In: ICCV. pp. 6994–7003 (2021)
21. Lee, J., Choi, J., Mok, J., Yoon, S.: Reducing information bottleneck for weakly supervised semantic segmentation. In: NeurIPS (2021)
22. Lee, J., Kim, E., Lee, S., Lee, J., Yoon, S.: Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In: CVPR. pp. 5267–5276 (2019)
23. Lee, J., Kim, E., Yoon, S.: Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In: CVPR. pp. 4071–4080 (2021)
24. Lee, J., Kim, E., Yoon, S.: Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In: CVPR. pp. 4071–4080 (2021)
25. Lee, S., Lee, M., Lee, J., Shim, H.: Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In: CVPR. pp. 5495–5505 (2021)
26. Li, K., Wu, Z., Peng, K.C., Ernst, J., Fu, Y.: Tell me where to look: Guided attention inference network. In: CVPR. pp. 9215–9223 (2018)
27. Li, K., Zhang, Y., Li, K., Li, Y., Fu, Y.: Attention bridging network for knowledge transfer. In: ICCV. pp. 5198–5207 (2019)
28. Li, X., Zhou, T., Li, J., Zhou, Y., Zhang, Z.: Group-wise semantic mining for weakly supervised semantic segmentation. In: AAAI. vol. 35, pp. 1984–1992 (2021)
29. Li, Y., Kuang, Z., Liu, L., Chen, Y., Zhang, W.: Pseudo-mask matters in weakly-supervised semantic segmentation. In: ICCV. pp. 6964–6973 (2021)
30. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In: CVPR. pp. 3159–3167 (2016)
31. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755 (2014)
32. Lovász, L.: Random walks on graphs. Combinatorics, Paul erdos is eighty 2(1–46), 4 (1993)
33. Peng, Z., Huang, W., Gu, S., Xie, L., Wang, Y., Jiao, J., Ye, Q.: Conformer: Local features coupling global representations for visual recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 367–376 (2021)
34. Song, C., Huang, Y., Ouyang, W., Wang, L.: Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In: CVPR. pp. 3136–3145 (2019)
35. Sun, W., Zhang, J., Liu, Z., Zhong, Y., Barnes, N.: Getam: Gradient-weighted element-wise transformer attention map for weakly-supervised semantic segmentation. arXiv preprint arXiv:2112.02841 (2021)
36. Vernaza, P., Chandraker, M.: Learning random-walk label propagation for weakly-supervised semantic segmentation. In: CVPR. pp. 7158–7166 (2017)
37. Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., Hu, X.: Score-cam: Score-weighted visual explanations for convolutional neural networks. In: CVPRW. pp. 24–25 (2020)
38. Wang, Y., Zhang, J., Kan, M., Shan, S., Chen, X.: Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In: CVPR. pp. 12275–12284 (2020)
39. Wei, Y., Feng, J., Liang, X., Cheng, M.M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In: CVPR. pp. 1568–1576 (2017)
40. Wu, T., Huang, J., Gao, G., Wei, X., Wei, X., Luo, X., Liu, C.H.: Embedded discriminative attention mechanism for weakly supervised semantic segmentation. In: CVPR. pp. 16765–16774 (2021)
41. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: ECCV. pp. 418–434 (2018)
42. Xu, L., Ouyang, W., Bennamoun, M., Boussaid, F., Sohel, F., Xu, D.: Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In: ICCV. pp. 6984–6993 (2021)
43. Yao, Y., Chen, T., Xie, G.S., Zhang, C., Shen, F., Wu, Q., Tang, Z., Zhang, J.: Non-salient region object mining for weakly supervised semantic segmentation. In: CVPR. pp. 2623–2632 (2021)
44. Zhang, D., Zhang, H., Tang, J., Hua, X., Sun, Q.: Causal intervention for weakly-supervised semantic segmentation. In: NeurIPS. pp. 655–666 (2020)
45. Zhang, F., Gu, C., Zhang, C., Dai, Y.: Complementary patch for weakly supervised semantic segmentation. In: ICCV. pp. 7242–7251 (2021)
46. Zhang, T., Lin, G., Liu, W., Cai, J., Kot, A.: Splitting vs. merging: Mining object regions with discrepancy and intersection loss for weakly supervised semantic segmentation. In: ECCV. pp. 663–679. Springer (2020)
47. Zhang, X., Wei, Y., Feng, J., Yang, Y., Huang, T.S.: Adversarial complementary learning for weakly supervised object localization. In: CVPR. pp. 1325–1334 (2018)
48. Zhao, T., Wu, X.: Pyramid feature attention network for saliency detection. In: CVPR. pp. 3085–3094 (2019)
49. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR. pp. 2921–2929 (2016)