Learning Affinity from Attention: End-to-End Weakly-Supervised Semantic Segmentation with Transformers

Lixiang Ru\textsuperscript{1} Yibing Zhan\textsuperscript{2} Baosheng Yu\textsuperscript{3} Bo Du\textsuperscript{1}\textsuperscript{*}

\textsuperscript{1} National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence, School of Computer Science and Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, China.
\textsuperscript{2} JD Explore Academy, China
\textsuperscript{3} The University of Sydney, Australia

{rulixiang, dubo}@whu.edu.cn zhanyibing@jd.com baosheng.yu@sydney.edu.au

Abstract

Weakly-supervised semantic segmentation (WSSS) with image-level labels is an important and challenging task. Due to the high training efficiency, end-to-end solutions for WSSS have received increasing attention from the community. However, current methods are mainly based on convolutional neural networks and fail to explore the global information properly, thus usually resulting in incomplete object regions. In this paper, to address the aforementioned problem, we introduce Transformers, which naturally integrate global information, to generate more integral initial pseudo labels for end-to-end WSSS. Motivated by the inherent consistency between the self-attention in Transformers and the semantic affinity, we propose an Affinity from Attention (AFA) module to learn semantic affinity from the multi-head self-attention (MHSA) in Transformers. The learned affinity is then leveraged to refine the initial pseudo labels for segmentation. In addition, to efficiently derive reliable affinity labels for supervising AFA and ensure the local consistency of pseudo labels, we devise a Pixel-Adaptive Refinement module that incorporates low-level image appearance information to refine the pseudo labels. We perform extensive experiments and our method achieves 66.0% and 38.9% mIoU on the PASCAL VOC 2012 and MS COCO 2014 datasets, respectively, significantly outperforming recent end-to-end methods and several multi-stage competitors. Code is available at https://github.com/rulixiang/afa.

1. Introduction

Semantic segmentation, aiming at labeling each pixel in an image, is a fundamental task in vision. In the past decade, deep neural networks have achieved great success in semantic segmentation. However, due to the data-hungry nature of deep neural networks, fully-supervised semantic segmentation models usually require a large amount of data with labour intensive pixel-level annotations. To settle this problem, some recent methods seek to devise semantic segmentation models using weak/cheap labels, such as image-level labels [2, 25, 47, 23, 50, 27, 35], points [3], scribbles [28, 54, 52], and bounding boxes [24]. Our method falls into the category of weakly-supervised semantic segmentation (WSSS) using only image-level labels, which is the most challenging one in all WSSS scenarios.

Prevailing WSSS methods with image-level labels commonly adopt a multi-stage framework [35, 23, 22]. Specifically, these methods firstly train a classification model and...
then generate Class Activation Maps (CAM) [59] as the pseudo labels. After refinement, the pseudo labels are leveraged to train a standalone semantic segmentation network as the final model. This multi-stage framework needs to train multiple models for different purposes, thus obviously complicating the training streamline and slowing down the efficiency. To avoid this problem, several end-to-end solutions have been recently proposed for WSSS [4, 52, 53, 3]. However, these methods are commonly based on convolutional neural networks and fail to explore the global feature relations properly, which turns out to be crucial for activation neural networks and fail to explore the global feature relations. We further exploit the inherent virtue of Transformer and devise an Affinity from Attention (AFA) module. AFA learns reliable semantic affinity from MHSA and propagates the pseudo labels with the learned affinity.

We propose a Pixel-Adaptive Refinement (PAR) module, which incorporates the RGB and position information of local pixels for label refinement.

2. Related Work

2.1. Weakly-Supervised Semantic Segmentation

Multi-stage Methods. Most WSSS methods with image-level labels are accomplished in a multi-stage process. Commonly, these approaches train a classification network to produce the initial pseudo pixel-level labels with CAM. To address the drawback of incomplete object activation of CAM, [46, 56, 40] utilize Erasing strategy to erase the most discriminative regions and thus discover more complete object regions. Inspired by the observation that the classification network tends to focus on different object regions at different training stages, [16, 51, 18] accumulate the activated regions in the training process. [26, 39, 47] propose to mine semantic regions from multiple input images, discovering similar semantic regions. A prevailing group of WSSS trains classification network with auxiliary tasks to ensure integral object discovery [45, 7, 35, 36]. Some recent researches interpret CAM generation from novel perspectives, such as causal inference [55], information bottleneck theory [22], and anti-adversarial attack [23].

End-to-End Methods. Due to the extremely limited supervision, training an end-to-end model for WSSS with favorable performance is difficult. [31] proposes an adap-
The proposed end-to-end framework for WSSS. We use a Transformer backbone as the encoder to extract feature maps. The initial pseudo labels are generated with CAM [59] and then refined with the proposed PAR. In the AFA module, we derive the semantic affinity from MHSA in Transformer blocks. AFA is supervised with the pseudo affinity labels derived from the refined label. Next, we employ the learned affinity to revise the pseudo labels via random walk propagation [2, 1]. The propagated labels are finally refined with PAR as the pseudo labels for the segmentation branch.

Figure 3. The proposed end-to-end framework for WSSS. We use a Transformer backbone as the encoder to extract feature maps. The initial pseudo labels are generated with CAM [59] and then refined with the proposed PAR. In the AFA module, we derive the semantic affinity from MHSA in Transformer blocks. AFA is supervised with the pseudo affinity labels derived from the refined label. Next, we employ the learned affinity to revise the pseudo labels via random walk propagation [2, 1]. The propagated labels are finally refined with PAR as the pseudo labels for the segmentation branch.

### 3. Methodology

In this section, we first introduce the Transformer backbone and CAM to generate initial pseudo labels. We then present the Affinity from Attention (AFA) module to learn reliable semantic affinity and propagate the initial pseudo labels with the learned affinity. Afterward, we introduce a Pixel-Adaptive Refinement (PAR) module to ensure the local consistency of pseudo labels. The overall loss function for optimization is presented in Section 3.5.

#### 3.1. Transformer Backbone

As shown in Fig. 3, our framework uses the Transformers as the backbone. An input image is firstly split into \( h \times w \) patches, where each patch is flattened and linearly projected to form \( h \times w \) tokens. In each Transformer block, the multi-head self-attention (MHSA) is used to capture global feature dependencies. Specifically, for the \( i^{th} \) head, patch tokens are projected with Multi-Layer Perceptron (MLP) layers and construct queries \( Q_i \in \mathbb{R}^{h \times w \times d_k} \), keys \( K_i \in \mathbb{R}^{h \times w \times d_k} \), and values \( V_i \in \mathbb{R}^{h \times w \times d_v} \). \( d_k \) is feature dimension of queries and keys, and \( d_v \) denotes the feature dimension of values. The class token \( c \in \mathbb{R}^{d_v} \) is the output of the embedding to drive the network to assign correct pixel labels. Incorporating nGWF pooling, pixel-adaptive mask refinement, and stochastic low-level information transfer, 1Stage [4] achieves comparable performance with multi-stage models.

In [53], RRM takes CAM as the initial pseudo labels and employs CRF [20] to produce the refined label as supervision for segmentation. RRM also introduces an auxiliary regularization loss [41] to ensure the consistency between segmentation map and low-level image appearance. [57] introduces the adaptive affinity field [17] with weighted affinity kernels and feature-to-prototype alignment loss to ensure the semantic fidelity. The above methods commonly adopt CNN and raise the inherent drawback of convolution, i.e., failing to capture the global information, leading to the incomplete activation of objects [13]. In this work, we explore Transformers for end-to-end WSSS to address this issue.

#### 2.2. Transformer in Vision

In [11], Dosovitskiy et al. proposes Vision Transformer (ViT), the first work to apply pure Transformer architecture for visual recognition tasks, achieving astonishing performance on visual classification benchmarks. Later variants show ViT also benefits downstream vision tasks, such as semantic segmentation [49, 9, 58], depth estimation [33], and video understanding [5]. In [13], Gao et al. proposes the first Transformers-based method (TS-CAM) for weakly-supervised object localization (WSOL). Close to WSSS, WSOL aims at localizing objects with only image-level supervision. TS-CAM trains a ViT model with image-level supervision, generates semantic-aware CAM, and couples the generated CAM with semantic-agnostic attention maps. The semantic-agnostic attention maps are derived from the attention of class token against other patch tokens. Nevertheless, TS-CAM didn’t exploit the intrinsic semantic affinity in MHSA to promote the localization results. In this work, we propose to learn the reliable semantic affinity from MHSA and propagate CAM with the learned affinity.
dimension of values. Based on $Q_i$, $K_i$ and $V_i$, the self-attention matrix $S_i$ and outputs $X_i$ are

$$S_i = \frac{Q_i K_i^T}{\sqrt{d_k}}, \quad X_i = \text{softmax}(S_i) V_i. \quad (1)$$

The final output $X_o$ of the Transformer block is constructed by feeding $(X_1||X_2||...||X_n)$ into feed-forward layers (FFN), i.e., $X_o = \text{FFN}(X_1||X_2||...||X_n)$, where FFN(·) consists of Layer Normalization [6] and MLP layers. $(·||·)$ denotes the concatenation operation. By stacking multiple Transformer blocks, the backbone produces feature maps for the subsequent modules.

3.2. CAM Generation

Considering the simplicity and inference efficiency, we adopt class activation maps (CAM) [59] as the initial pseudo labels. For the extracted feature maps $F \in \mathbb{R}^{h \times w \times d}$ and a given class $c$, the activation map $M^c$ is generated via weighting the feature maps $F$ with their contribution to class $c$, i.e., the weight matrix $W^c$ in the classification layer,

$$M^c = \text{ReLU}(\sum_{i=1}^d W^{i,c} F^i), \quad (2)$$

where ReLu function is used to remove the negative activations. Min–Max normalization is applied to scale $M^c$ to $[0, 1]$. A background score $\beta$ ($0 < \beta < 1$) is then used to differentiate foreground and background regions.

3.3. Affinity from Attention

As shown in Fig. 1, we notice the consistency between MHSA in Transformers and the semantic-level affinity, which motivates us to use MHSA to discover the object regions. However, since no explicit constraints are imposed on self-attention matrices during the training process, the learned affinity in MHSA is typically coarse and inaccurate, which means directly applying MHSA as affinity to refine the initial labels does not work well (Fig 2 (c)). Therefore, we propose the Affinity from Attention module (AFA) to counter this problem.

Assuming MHSA in a Transformer block is denoted as $S \in \mathbb{R}^{h \times w \times n}$, where $h \times w$ is the flattened spatial size and $n$ is the number of attention heads. In our AFA module, we directly produce the semantic affinity by linearly combining the multi-head attention, i.e., using an MLP layer. Essentially, the self-attention mechanism is a kind of directed graphical model [43], while the affinity matrix should be symmetric since nodes sharing the same semantics are supposed to be equal. To perform such transformation, we simply add $S$ and its transpose. The predicted semantic affinity matrix $A \in \mathbb{R}^{h \times w \times n}$ is thus denoted as

$$A = \text{MLP}(S + S^\top). \quad (3)$$

Here, we use the matrix transpose operator $^\top$ to denote the transpose of each self-attention matrix in tensor $S$.

**Pseudo Affinity Label Generation.** To learn the favorable semantic affinity $A$, a key step is to derive a reliable pseudo affinity label $Y_{aff}$ as supervision. As shown in Fig. 3, we derive $Y_{aff}$ from the refined pseudo labels (the refinement module will be introduced later).

We first use two background scores $\beta_l$ and $\beta_h$, where $0 < \beta_l < \beta_h < 1$, to filter the refined pseudo labels to reliable foreground, background, and uncertain regions. Formally, given $\text{CAM} M \in \mathbb{R}^{h \times w \times C}$, the pseudo label $Y_p$ is constructed as

$$Y_{p,i,j} = \begin{cases} \text{argmax}(M^{i,j,:}), & \text{if } \max(M^{i,j,:}) \geq \beta_h, \\ 0, & \text{if } \max(M^{i,j,:}) \leq \beta_l, \\ 255, & \text{otherwise}, \end{cases} \quad (4)$$

where 0 and 255 denote the index of the background class and the ignored regions, respectively. argmax(·) extracts the semantic class with the maximum activation value.

The pseudo affinity label $Y_{aff} \in \mathbb{R}^{h \times w \times h \times w}$ is then derived from $Y_p$. Specifically, for $Y_p$, if the pixel $(i, j)$ and $(k, l)$ share the same semantic, we set their affinity as positive; otherwise, their affinity is set as negative. Note that if pixels $(i, j)$ or $(k, l)$ are sampled from the ignored regions, their affinity will also be ignored. Besides, we only consider the situation that pixel $(i, j)$ and $(k, l)$ are in the same local window, and disregard the affinity of distant pixel pairs.

**Affinity Loss.** The generated pseudo affinity label $Y_{aff}$ is then used to supervise the predicted affinity $A$. The affinity loss term $\mathcal{L}_{aff}$ is constructed as

$$\mathcal{L}_{aff} = \frac{1}{N^-} \sum_{(i,j) \in \mathcal{R}^-} (1 - \text{sigmoid}(A^{i,j})) + \frac{1}{N^+} \sum_{(i,j) \in \mathcal{R}^+} \text{sigmoid}(A^{i,j}), \quad (5)$$

where $\mathcal{R}^+$ and $\mathcal{R}^−$ denote the set of positive and negative samples in $Y_{aff}$, respectively. $N^+$ and $N^−$ count the number of $\mathcal{R}^+$ and $\mathcal{R}^−$. Intuitively, Eq. 5 enforces the network to learn highly confident semantic affinity relations from MHSA. On the other hand, since the affinity prediction $A$ is the linear combination of MHSA, Eq. 5 also benefits the learning of self-attention and further helps to discover the integral object regions.

**Propagation with Affinity.** The learned reliable semantic affinity could be used to revise the initial CAM. Following [2, 1], we fulfill this process via random walk [44]. For the learned semantic affinity matrix $A$, the semantic transition matrix $T$ is derived as

$$T = D^{-1} A^\alpha, \quad \text{with } D^{ij} = \sum_k A^{ik\alpha}, \quad (6)$$

16828
where \( \alpha > 1 \) is a hyper-parameter to ignore trivial affinity values in \( A \), and \( D \) is a diagonal matrix to normalize \( A \) row-wise. The random walk propagation for the initial CAM \( M \in \mathbb{R}^{h \times w \times C} \) is accomplished as

\[
M_{aff} = T \ast \text{vec}(M),
\]

where \( \text{vec}(\cdot) \) vectorizes \( M \). This propagation process diffuses the semantic regions with high affinity and damps the wrongly activated regions so that the activation maps align better with semantic boundaries.

### 3.4. Pixel-Adaptive Refinement

As shown in Fig. 3, the pseudo affinity label \( Y_{aff} \) is derived from the initial pseudo labels. However, the initial pseudo labels are typically coarse and locally inconsistent, \( i.e., \) neighbor pixels with similar low-level image appearance may not share the same semantic. To ensure the local consistency, \( [19, 53, 57] \) adopt dense CRF \([20]\) to refine the initial pseudo labels. However, CRF is not a favorable choice in end-to-end framework since it remarkably slows down the training efficiency. Inspired by \([4]\), which utilizes the pixel-adaptive convolution \([37]\) to extract local RGB information for refinement, we incorporate the RGB and spatial information to define the low-level pairwise affinity and construct our Pixel-Adaptive Refinement module (PAR).

Given the input image \( I \in \mathbb{R}^{h \times w \times 3} \), for the pixel at position \((i, j)\) and \((k, l)\), the RGB and spatial pairwise terms are defined as:

\[
\kappa_{rgb}^{ij,kl} = \frac{(I_{ij} - I_{kl})^2}{w_1 \sigma_{rgb}^2}, \quad \kappa_{pos}^{ij,kl} = \frac{(P_{ij} - P_{kl})^2}{w_2 \sigma_{pos}^2},
\]

where \( I_{ij} \) and \( P_{ij} \) denote the RGB information and the spatial location of pixel \((i, j)\), respectively. In practice, we use the \( x, y \) coordinates as the spatial location. In Eq. 8, \( \sigma_{rgb} \) and \( \sigma_{pos} \) denote the standard deviation of RGB and position difference, respectively. \( w_1 \) and \( w_2 \) control the smoothness of \( \kappa_{rgb} \) and \( \kappa_{pos} \), respectively. The affinity kernel for PAR is then constructed by normalizing \( \kappa_{rgb} \) and \( \kappa_{pos} \) with softmax and adding them together, \( i.e., \)

\[
\kappa^{ij,kl} = \frac{\exp(\kappa_{rgb}^{ij,kl})}{\sum_{(x,y)} \exp(\kappa_{rgb}^{x,y})} + w_3 \frac{\exp(\kappa_{pos}^{ij,kl})}{\sum_{(x,y)} \exp(\kappa_{pos}^{x,y})},
\]

where \((x, y)\) is sampled from the neighbor set of \((i, j)\), \( i.e., \mathcal{N}(i, j) \), and \( w_3 \) adjusts the importance of the position term. Based on the constructed affinity kernel, we refine both the initial CAM and the propagated CAM. The refinement is conducted for multiple iterations. For CAM \( M \in \mathbb{R}^{h \times w \times C} \), in iteration \( t \), we have

\[
M_t^{ij,lc} = \sum_{(k,l) \in \mathcal{N}(i,j)} \kappa^{ij,kl} M_{t-1}^{k,l,c}.
\]

For the neighbor pixel sets \( \mathcal{N}(\cdot) \), we follow \([4]\) and define it as the \( 8 \)-way neighbors with multiple dilation rates.

Such design ensures the training efficiency, since the dilated neighbors of a given pixel can be easily extracted using \( 3 \times 3 \) dilated convolutions.

### 3.5. Network Training

As shown in Fig. 3, our framework consists of three loss terms, \( i.e., \) a classification loss \( L_{cls} \), a segmentation loss \( L_{seg} \), and an affinity loss \( L_{aff} \).

For the classification loss, following the common practice, we feed the aggregated features into a classification layer to compute the class probability vector \( p_{cls} \), then employ the multi-label soft margin loss as the classification function.

\[
L_{cls} = \frac{1}{C} \sum_{c=1}^{C} (y^c \log(p_{cls}^c) + (1 - y^c) \log(1 - p_{cls}^c)), \tag{11}
\]

where \( C \) is the total number of classes, and \( y \) is the ground truth image-level label.

For the segmentation loss \( L_{seg} \), we adopt the commonly-used cross-entropy loss. As shown in Fig. 3, the supervision for the segmentation branch is the revised label with affinity propagation. In order to obtain better alignment with low-level image appearance, we use the proposed PAR to further refine the propagated labels. The affinity loss \( L_{aff} \) for affinity learning is previously described in Eq. 5.

The overall loss is the weighted sum of \( L_{cls}, L_{aff}, \) and \( L_{seg} \). In addition, to further promote the performance, we also employ the regularization loss \( L_{reg} \) used in \([41, 57, 54, 53]\), which ensures the local consistency of the segmentation predictions. The overall loss is finally formulated as

\[
L = L_{cls} + \lambda_1 L_{seg} + \lambda_2 L_{aff} + \lambda_3 L_{reg}, \tag{12}
\]

where \( \lambda_1, \lambda_2, \) and \( \lambda_3 \) balance the contributions of different losses.

### 4. Experiments

#### 4.1. Setup

**Datasets.** We conduct experiments on PASCAL VOC 2012 and MS COCO 2014 datasets. PASCAL VOC 2012 dataset [12] contains 21 semantic classes (including the background class). This dataset is usually augmented with the SBD dataset [14]. The augmented dataset includes 10,582, 1,449, and 1,464 images for training, validation, and testing, respectively. MS COCO 2014 dataset [29] contains 81 classes and includes 82,081 images for training and 40,137 images for validation. The images in \textit{train} sets of PASCAL VOC and MS COCO are annotated with image-level labels only. By default, we report mean Intersection-Over-Union (mIoU) as the evaluation criteria.
Network Configuration. For the Transformer backbone, we use the Mix Transformer (MiT) proposed in Segformer [49], which is a more friendly backbone for image segmentation tasks than the vanilla ViT [58]. In brief, MiT uses overlapped patch embedding to keep local consistency, spatial-reductive self-attention to accelerate computation, and FFN with convolutions to safely replace position embedding. For the segmentation decoder, we use the MLP decoder head [49], which fuses multi-level feature maps for prediction with simple MLP layers. The backbone parameters are initialized with ImageNet-1k [10] pre-trained weights, while other parameters are randomly initialized.

Implementation Details. We use an AdamW optimizer [30] to train our network. For the backbone parameters, the initial learning rate is set as $6 \times 10^{-5}$ and decays every iteration with a polynomial scheduler. The learning rate for other parameters is 10 times the learning rate of backbone parameters. The weight decay factor is set as 0.01. For data augmentation, random rescaling with a range of $[0.5, 2.0]$, random horizontally flipping, and random cropping with a cropping size of $512 \times 512$ are adopted. The batch size is set as 8. For the experiments on the PASCAL VOC dataset, we train the network for 20,000 iterations. To ensure the initial pseudo labels are favorable, we warm-up the classification module for the first 2,000 iterations and the affinity branch for the next 4,000 iterations. For experiments on the MS COCO dataset, the number of total iterations is 80,000. Accordingly, the number of warm-up iteration for the classification and affinity branch are 5,000 and 15,000, respectively.

The default hyper-parameters are set as follows. For pseudo label generation, the background thresholds $(\beta_h, \beta_l)$ are $(0.55, 0.35)$. In PAR, same as [4], the dilation rates for extracting neighbor pixels are $[1, 2, 4, 8, 12, 24]$. We set the weight factors $(w_1, w_2, w_3)$ as $(0.3, 0.3, 0.01)$. When computing the affinity loss, the radius of the local window to ignore distant affinity pairs is set as 8. In Eq. 6, we set the power factor $\alpha$ as 2. The weight factors in Eq. 12 are 0.1, 0.1, and 0.01, respectively. The detailed investigation of the hyper-parameters is reported in the supplementary material.

4.2. Initial Pseudo Label Generation.

In this work, we use the popular CAM to generate the initial pseudo labels. Empirically, for a CNN-based classification network, the choice of pooling method notably affects the quality of CAM. Specifically, global max-pooling (gmp) tends to underestimate the object size, while global average-pooling (gap) typically overestimates the object regions [19, 59]. Here, we investigate the favorable pooling method for Transformer-based classification network. We first generalize gmp and gap with top-k pooling, i.e., averaging the top k% values in each feature map. In this situation, gmp and gap are two special cases of top-k pooling, i.e., top-100% and top-1 pooling. We present the impact of top-k pooling with different k in Tab. 1.

4.3. Ablation Study and Analysis

The quantitative results of ablation analysis are reported in Tab. 2. Tab. 2 shows that our baseline model based on Transformers achieves 46.7% mIoU on the PASCAL VOC val set. The proposed PAR and AFA further significantly improve the mIoU to 56.2% and 62.6%, respectively. With the auxiliary regularization loss $\mathcal{L}_\text{reg}$, the proposed framework achieves 63.8% mIoU. The CRF post-processing brings further 2.2% mIoU improvements, promoting the final performance to 66.0% mIoU. In short, the quantitative results in Tab. 2 demonstrate our proposed modules are remarkably effective.

AFA. The motivation of AFA is to learn reliable semantic affinity from MHSA and revise the pseudo labels with the learned affinity. In Fig. 4, we present some example images of the self-attention maps (extracted from the last Transformer block) and the learned affinity maps. Fig. 4 shows that our AFA could effectively learn reliable semantic affinity from the inaccurate MHSA. The affinity loss in the AFA module also encourages the MHSA to model the semantic relations well. In Fig. 4, we also present the pseudo la-
Figure 4. Visualization of the MHSA maps, learned affinity maps, and generated pseudo labels for segmentation. “★” denotes the query point to visualize the attention and affinity maps.

Figure 5. Examples of PAR’s improvements on the pseudo labels. The pseudo labels are generated with CAM and Transformer baseline.

Table 4. Comparison of refinement methods for CAM.

| Method  | \(\kappa_{rgb}\) | \(\kappa_{pos}\) | train |
|---------|------------------|------------------|-------|
| CAM     | ✓                |                  | 48.2  |
| PAMR [4]| ✓                | ✓                | 51.4  |
| PAR     | ✓                | ✓                | 51.7  |

4.4. Comparison to State-of-the-art

PASCAL VOC 2012. We report the semantic segmentation performance on PASCAL VOC 2012 val and test set in Tab. 5. R101 and WR38 denote the method uses ResNet101 [15] and WideResNet38 [48] as backbone, respectively. Tab. 5 shows that the proposed model clearly surpasses previous state-of-the-art end-to-end methods. Our method achieves 83.8% of its fully-supervised counterpart, i.e., Segformer [49], while 1Stage [4] and AA&LR [57] only achieve 77.6% and 79.1% of WideResNet38, respectively. Our method is also competitive with some recent multi-
Table 5. Semantic segmentation results on PASCAL VOC 2012 dataset. *Sup.* denotes supervision type. *F*: full supervision; *I*: image-level labels; *S*: saliency maps. † denotes our implementation.

| Method          | Sup. | Backbone | val    | test   |
|-----------------|------|----------|--------|--------|
| **Fully-supervised models.** |     |          |        |        |
| DeepLab [8]     | F    | R101     | 77.6   | 79.7   |
| WideResNet38 [48] | F    | WR38     | 80.8   | 82.5   |
| Segformer † [49] | F    | MiT-B1   | 78.7   | –      |
| **Multi-Stage weakly-supervised models.** |     |          |        |        |
| OAA+ [16] ICCV'2019 | I + S | R101     | 65.2   | 66.4   |
| MCIS [39] ICCV'2020 | I    | R101     | 66.2   | 66.9   |
| AuxSegNet [50] ICCV'2021 | I    | WR38     | 69.0   | 68.6   |
| NSROM [51] CVPR'2021 | I    | R101     | 70.4   | 70.2   |
| EPS [25] CVPR'2021 | I    | R101     | 70.9   | 70.8   |
| SEAM [45] CVPR'2020 | I    | WR38     | 64.5   | 65.7   |
| SC-CAM [7] CVPR'2020 | I    | R101     | 66.1   | 65.9   |
| CDA [38] ICCV'2021 | I    | WR38     | 66.1   | 66.8   |
| AdvCAM [23] CVPR'2021 | I    | R101     | 68.1   | 68.0   |
| CPN [56] ICCV'2021 | I    | R101     | 67.8   | 68.5   |
| RIB [22] NonIPS'2021 | I    | R101     | 68.3   | 68.6   |
| **End-to-End weakly-supervised models.** |     |          |        |        |
| EM [31] ICCV'2015 | I    | VGG16     | 38.2   | 39.6   |
| MIL [32] CVPR'2015 | I    | –        | 42.0   | 40.6   |
| CRF-RNN [34] CVPR'2017 | I    | VGG16     | 52.8   | 53.7   |
| RRM [53] AAS'2020 | I    | WR38     | 62.6   | 62.9   |
| RRM † [53] AAS'2020 | I    | MiT-B1   | 63.5   | –      |
| 1Stage [4] CVPR'2020 | I    | WR38     | 62.7   | 64.3   |
| AA&LR [57] ACM MM'2021 | I    | WR38     | 63.9   | 64.8   |
| Ours MiT-B1 | I    | MiT-B1   | 66.0   | 66.3   |

Table 6. Semantic segmentation results on MS COCO dataset.

| Method          | Sup. | Backbone | val    |
|-----------------|------|----------|--------|
| **Multi-Stage weakly-supervised models.** |     |          |        |
| EPS [25] CVPR'2021 | I + S | R101     | 35.7   |
| AuxSegNet [50] ICCV'2021 | I    | WR38     | 33.9   |
| SEAM [45] CVPR'2020 | I    | WR38     | 31.9   |
| CONTA [55] NonIPS'2020 | I    | WR38     | 32.8   |
| CDA [38] ICCV'2021 | I    | WR38     | 31.7   |
| CGNet [21] ICCV'2021 | I    | WR38     | 36.4   |
| RIB [22] NonIPS'2021 | I    | R101     | 43.8   |
| **End-to-End weakly-supervised models.** |     |          |        |
| Ours MiT-B1 | I    | MiT-B1   | 38.0   |
| Ours + CRF | I    | MiT-B1   | 38.9   |

stage WSSSS methods, such as OAA+ [16], SEAM [45], SC-CAM [7], and CDA [38]. It’s also noted that our method also outperforms RRM with MiT-B1 as backbone, which demonstrates the efficacy of the proposed AFA and PAR.

MS COCO 2014. We present the semantic segmentation performance on the challenging MS COCO 2014 dataset in Tab. 6. Our end-to-end method could achieve 38.9% mIoU on the *val* set, which remarkably outperforms most recent multi-stage methods (except RIB [22]).

Figure 6. Segmentation results of 1Stage [4] and our method on VOC and COCO *val* set.

**Qualitative Results.** In Fig. 6, we present the qualitative results of our method on the PASCAL VOC and MS COCO *val* set. On the PASCAL VOC dataset, visually, our method could outperform 1Stage [4] and produce segmentation results that align finely with object boundaries. The qualitative results on MS COCO dataset are also comparable with the ground truth.

5. Conclusion

In this work, we explore the intrinsic virtue of Transformer architecture for WSSSSS tasks. Specifically, we use a Transformer-based backbone to generate CAM as the initial pseudo labels, avoiding the inherent flaw of CNN. Besides, we note the consistency between the MHSA and semantic affinity, and thus propose the AFA module. AFA derives reliable affinity labels from pseudo labels, imposes the affinity labels to supervise the MHSA, and produces reliable affinity predictions. The learned affinity is used to revise the initial pseudo labels via random walk propagation. On PASCAL VOC and MS COCO datasets, our method achieves new state-of-the-art performance for end-to-end WSSS. In a broader view, the proposed method also shows a novel perspective for vision transformers, i.e. guiding the self-attention with semantic relation to ensure better feature aggregation.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 62141112, 41871243, and 6202090, the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170, the Major Science and Technology Innovation 2030 "New Generation Artificial Intelligence" key project (No. 2021ZD0111700). Dr. Baosheng Yu is supported by ARC project FL-170100117.
References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In CVPR, pages 2209–2218, 2019.

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In CVPR, pages 4981–4990, 2018.

[3] Peri Akiva and Kristin Dana. Towards single stage weakly supervised semantic segmentation. arXiv preprint arXiv:2106.10309, 2021.

[4] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In CVPR, pages 4253–4262, 2020.

[5] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. ICCV, 2021.

[6] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

[7] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via sub-category exploration. In CVPR, pages 8991–9000, 2020.

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI, 40(4):834–848, 2017.

[9] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. arXiv preprint arXiv:2107.06278, 2021.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, pages 248–255, 2009.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In ICLR, 2020.

[12] Mark Everingham, Luc Van Gool, ChristopherKI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. IJCV, 88(2):303–338, 2010.

[13] Wei Gao, Fang Wan, Xingjia Pan, Zhiliang Peng, Qi Tian, Zhenjun Han, Bolei Zhou, and Qixiang Ye. Ts-cam: Token semantic coupled attention map for weakly supervised object localization. In ICCV, pages 2886–2895, October 2021.

[14] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In 2011 International Conference on Computer Vision, pages 991–998, 2011.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016.

[16] Peng-Tao Jiang, Qibin Hou, Yang Cao, Ming-Ming Cheng, Yunchao Wei, and Hong-Kai Xiong. Integral object mining via online attention accumulation. In ICCV, pages 2070–2079, 2019.

[17] Tsung-Wei Ke, Jyh-Jing Hwang, Ziwei Liu, and Stella X Yu. Adaptive affinity fields for semantic segmentation. In ECCV, pages 587–602, 2018.

[18] Beomyoung Kim, Sangeun Han, and Junmo Kim. Discriminative region suppression for weakly-supervised semantic segmentation. In AAAI, volume 35, pages 1754–1761, 2021.

[19] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In ECCV, pages 695–711. Springer, 2016.

[20] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. NeurIPS, 24:109–117, 2011.

[21] Hyeokjun Kweon, Sung-Hoon Yoon, Hyeonsoong Kim, Dahee Park, and Ku-Jin Yoon. Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In ICCV, pages 6994–7003, 2021.

[22] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. NeurIPS, 34, 2021.

[23] Jungbeom Lee, Eunji Kim, and Sungroh Yoon. Anti-adversarily manipulated attributions for weakly and semi-supervised semantic segmentation. In CVPR, pages 4071–4080, 2021.

[24] Jungbeom Lee, Jihun Yi, Chaehun Shin, and Sungroh Yoon. Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation. In CVPR, pages 2643–2652, 2021.

[25] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel-supervision for weakly supervised semantic segmentation. In CVPR, pages 5495–5505, 2021.

[26] Xueyi Li, Tianfei Zhou, Jianwu Li, Yi Zhou, and Zhaoxiang Zhang. Group-wise semantic mining for weakly supervised semantic segmentation. In AAAI, volume 35, pages 1984–1992, 2021.

[27] Yi Li, Zitnick. Microsoft coco: Common objects in context. In CVPR, 2014.

[28] Yi Li, Zhanghui Kuang, Liyang Liu, Yimin Chen, and Wayne Zhang. Pseudo-mask matters in weakly-supervised semantic segmentation. In CVPR, pages 695–711. Springer, 2016.

[29] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In CVPR, pages 3159–3167, 2016.

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

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In ICLR, 2019.

[32] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In ICCV, pages 1742–1750, 2015.

[33] Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In CVPR, pages 1713–1721, 2015.
[33] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In ICCV, pages 12179–12188, 2021.

[34] Anirban Roy and Sinisa Todorovic. Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. In CVPR, pages 3529–3538, 2017.

[35] Lixiang Ru, Bo Du, and Chen Wu. Learning visual words for weakly-supervised semantic segmentation. In IJCAI, pages 982–988, 8 2021.

[36] Lixiang Ru, Bo Du, Yibing Zhan, and Chen Wu. Weakly-supervised semantic segmentation with visual words learning and hybrid pooling. IJCV, 2022.

[37] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In CVPR, pages 11166–11175, 2019.

[38] Yukun Su, Ruizhou Sun, Guosheng Lin, and Qingyao Wu. Context decoupling augmentation for weakly supervised semantic segmentation. pages 7004–7014, October 2021.

[39] Guolei Sun, Wenguang Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In ECCV, pages 347–365, Springer, 2020.

[40] Kunyang Sun, Haoqing Shi, Zhengming Zhang, and Yongming Huang. Ecs-net: Improving weakly supervised semantic segmentation by using connections between class activation maps. In IJCV, pages 7283–7292, 2021.

[41] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised cnn segmentation. In ECCV, pages 507–522, 2018.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NeurIPS, pages 5998–6008, 2017.

[43] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In ICLR, 2018.

[44] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In CVPR, pages 7158–7166, 2017.

[45] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In CVPR, pages 12275–12284, 2020.

[46] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In CVPR, pages 1568–1576, 2017.

[47] Tong Wu, Junshi Huang, Guangyu Gao, Xiaoming Wei, Xiaolin Wei, Xuan Luo, and Chi Harold Liu. Embedded discriminative attention mechanism for weakly supervised semantic segmentation. In CVPR, pages 16765–16774, 2021.

[48] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. PR, 90:119–133, 2019.

[49] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. arXiv preprint arXiv:2105.15203, 2021.