This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Frozen CLIP: A Strong Backbone for Weakly Supervised Semantic Segmentation

Bingfeng Zhang[1] Siyue Yu[2*] Yunchao Wei[3] Yao Zhao[3] Jimin Xiao[2*]
[1]China University of Petroleum (East China) [2]XJTLU [3]Beijing Jiaotong University
bingfeng.zhang@upc.edu.cn, {siyue.yu02, jimin.xiao}@xjtlu.edu.cn, yunchao.wei@bjtu.edu.cn

Abstract

Weakly supervised semantic segmentation has witnessed great achievements with image-level labels. Several recent approaches use the CLIP model to generate pseudo labels for training an individual segmentation model, while there is no attempt to apply the CLIP model as the backbone to directly segment objects with image-level labels. In this paper, we propose WeCLIP, a CLIP-based single-stage pipeline, for weakly supervised semantic segmentation. Specifically, the frozen CLIP model is applied as the backbone for semantic feature extraction, and a new decoder is designed to interpret extracted semantic features for final prediction. Meanwhile, we utilize the above frozen backbone to generate pseudo labels for training the decoder. Such labels cannot be optimized during training. We then propose a refinement module (RFM) to rectify them dynamically. Our architecture enforces the proposed decoder and RFM to benefit from each other to boost the final performance. Extensive experiments show that our approach significantly outperforms other approaches with less training cost. Additionally, our WeCLIP also obtains promising results for fully supervised settings. The code is available at https://github.com/zbf1991/WeCLIP.

1. Introduction

Weakly supervised semantic segmentation (WSSS) [2, 48, 56] aims to learn a pixel-level segmentation model from weak supervision so as to reduce the manual annotation efforts. The common weak supervision signals contain scribble [27], bounding-box [43], point [4] and image-level labels [1, 19, 47, 52]. Among these supervisions, image-level annotation is the most popular one, as such annotations can be easily obtained through web-crawling.

There are two training solutions for WSSS with image-level labels: multi-stage training and single-stage training. For existing single-stage approaches, their backbones rely on pre-training on ImageNet [11] and fine-tuning during training, as in Fig. 1(a). Such single-stage training [3, 55] focuses on using one model to directly segment objects with weak signals as supervision. The primary consideration of previous single-stage architectures is to online refine the Class Activation Map (CAM) [39] or to improve the segmentation branch [57, 61]. Due to the complicated architecture, single-stage approaches perform normally worse than multi-stage approaches.

On the other hand, multi-stage training attempts to utilize several individual models to form a training pipeline [23, 26, 47], where offline pixel-level pseudo labels are firstly generated from weak labels using CAM [63] and then a segmentation model is trained with such pseudo labels. Since CAM can only highlight discriminate regions, many previous approaches focus on improving the quality of CAM [14, 49, 54, 59] for better pseudo labels. Besides, some recent multi-stage approaches [25, 29, 51] attempt to introduce

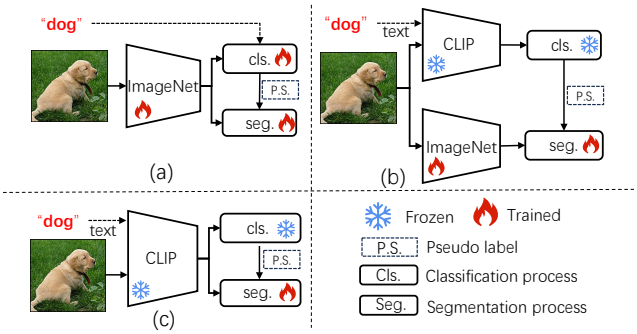

Figure 1. Comparisons between our approach and other single-stage or CLIP-based approaches. **(a) Previous single-stage approach,** which uses a trainable ImageNet [11] pre-trained backbone with trainable classification and segmentation process. **(b) Previous CLIP-based approach,** which is a multi-stage approach that uses the Frozen CLIP model to produce pseudo labels and trains an individual ImageNet pre-trained segmentation model. **(c) Our approach.** Our approach is a single-stage approach that uses a frozen CLIP model as the backbone with a trainable segmentation process, significantly reducing the training cost.

*Corresponding author.

Contrastive Language-Image Pre-training (CLIP) [36] for WSSS. Trained on 400 million image-text pairs, CLIP establishes a strong relationship between the image and text, demonstrating great ability to locate objects [18, 20, 51, 60, 65]. Based on this, existing approaches [29, 51] use CLIP to improve CAM, providing surprisingly high-quality pseudo labels. They follow the pipeline in Fig. 1(b). However, these methods only use the CLIP model to improve CAM for better pseudo labels. The potential of the CLIP model to be directly used as the backbone to extract strong semantic features for segmentation prediction is not explored.

In this paper, we propose a CLIP-based single-stage pipeline for weakly supervised semantic segmentation (**WeCLIP**) in which the CLIP model can be directly applied for segmentation prediction, as demonstrated in Fig. 1(c). Specifically, we adopt the frozen CLIP model as the backbone, followed by a newly designed light frozen CLIP feature decoder, where the CLIP backbone does not need any training or fine-tuning. Our decoder can successfully interpret the frozen CLIP features to conduct the segmentation task with a small number of learnable parameters.

We utilize the frozen CLIP backbone to generate CAMs for providing pixel-level pseudo labels to train our decoder. However, the frozen backbone can only provide static CAM, which means pseudo labels cannot be improved during training. The same errors in pseudo labels lead to uncorrectable optimization in the wrong directions. Thus, we propose a Frozen CLIP CAM Refinement module (RFM) to rectify the static CAM dynamically. Particularly, our RFM utilizes the dynamic features from our decoder and the prior features from the frozen CLIP backbone to establish high-quality pair-wise feature relationships to revise the initial CAM, leading to higher-quality pseudo labels. With such a design, our proposed two modules benefit from each other: refined pseudo labels provide more accurate supervision to train the decoder, and the trained decoder builds more reliable feature relationships for RFM to generate accurate pseudo labels.

Extensive experiments show that our approach achieves new state-of-the-art performances on both the PASCAL VOC 2012 and MS COCO datasets and significantly outperforms other approaches by a large margin. Further, our approach also achieves satisfactory performance for fully supervised semantic segmentation. More importantly, since WeCLIP has a frozen backbone, it only requires a small quantity of training cost, *i.e.*, 6.2GB GPU memory and less than 6M learnable parameters, much less than other weakly or fully supervised approaches.

Our contributions are summarized as:

- We find that the CLIP backbone can be directly used for weakly supervised semantic segmentation without fine-turning. With our designed decoder, the frozen CLIP feature is directly interpreted as semantic information to segment objects, building a strong single-stage solution.
- To overcome the drawback that the frozen backbone only provides static pseudo labels, we design a Frozen CLIP CAM Refinement module (RFM) to dynamically renew the initial CAM to provide better pseudo labels to train our model.
- With less training cost, our approach significantly outperforms previous approaches, reaching a new state-of-the-art performance for weakly supervised semantic segmentation (mIoU: 77.2% on VOC 2012 *test* set, 47.1% on COCO *val* set). Moreover, our approach also shows great potential for fully supervised semantic segmentation.

2. Related Work

2.1. Weakly Supervised Semantic Segmentation

Weakly supervised semantic segmentation with image-level supervision [1, 8, 51, 55] attracts more attention than other weak supervisions [27, 43] due to less human effort. There are two main solutions: multi-stage approaches [2, 19, 23, 25, 56] and single-stage approaches [39, 55].

The key to the multi-stage solution is to generate high-quality pseudo labels. For example, RIB [23] designed a margin loss in the classification network to reduce the information bottleneck, producing better pixel-level responses from image-level supervision. Du *et.al.* [14] proposed a pixel-to-prototype contrast strategy to impose feature semantic consistency to generate higher-quality pseudo labels. MCTformer [52] designed multi-class tokens in the transformer architecture to produce class-specific attention responses to generate refined CAM. Some recent multi-stage approaches attempted to introduce CLIP for this task. CLIMS [51] utilized the CLIP model to activate more complete object regions and suppress highly related background regions. CLIP-ES [29] proposed to use the softmax function in CLIP to compute the GradCAM [41]. With carefully designed text prompts, the GradCAM of CLIP provided reliable pseudo labels to train the segmentation model.

Previous single-stage solutions adopted the ImageNet [11] pre-train model as the backbone to concurrently learn the classification and segmentation tasks, and most of them focused on improving segmentation by providing more accurate supervision or constraining its learning. For example, RRM [55] proposed to select reliable pixels as supervision for the segmentation branch. 1Stage [3] designed a local consistency refinement module to directly generate semantic masks from image-level labels. AA&AR [61] proposed an adaptive affinity loss to enhance semantic propagation in the segmentation branch. AFA [39] designed an affinity branch to refine CAMs to generate better online pseudo labels. ToCo [40] proposed token contrast learning to mitigate over-smoothing in online CAM generation, thus providing better supervision for segmentation.

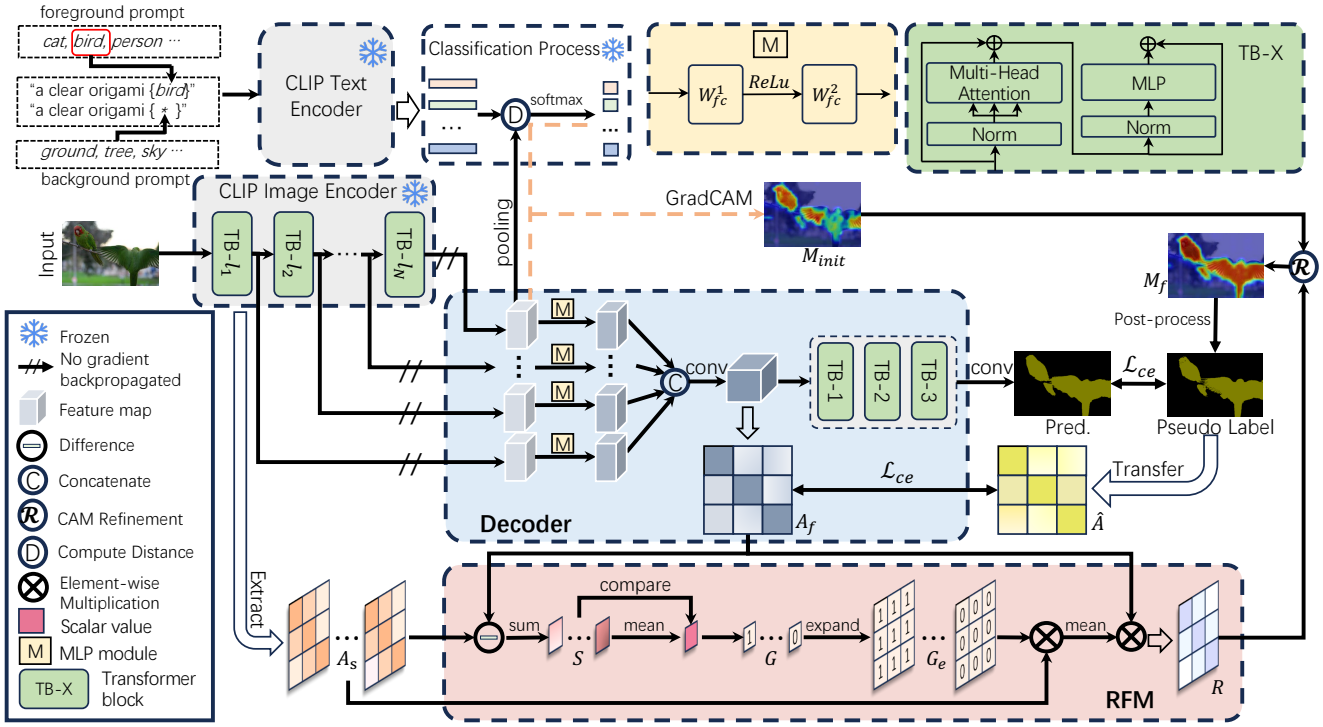

Figure 2. Framework of our WeCLIP. The image is input to the Frozen CLIP image encoder to generate the image features, and class labels are used to build text prompts and then input to the Frozen CLIP text encoder to generate the text features. The classification scores are generated based on the distance between the pooled image and text features. Using GradCAM, we can generate the initial CAM M_{init}. Then, the frozen image features from the last layer of each transformer block are input to our decoder to generate the final semantic segmentation predictions. Meanwhile, the affinity map A_f from our decoder and the multi-head attention maps A_s from CLIP are input to our RFM to establish refining maps R to refine M_{init} as M_f. After post-processing, it will be used as the supervision to train our decoder.

The CLIP model shows great effectiveness in the multi-stage solution, but using it as a single-stage solution, *i.e.*, directly learning to segment objects with image-level supervision, is not explored.

2.2. Fully Supervised Semantic Segmentation

Fully supervised semantic segmentation aims to segment objects using pixel-level labels as supervision. Most previous approaches are based on Fully Convolutional Network (FCN) [31] architecture, such as DeepLab [6], PSPNet [62] and UperNet [50]. Recent approaches introduced vision transformer [12] as the backbone to improve performance by building global relationships. For example, PVT [46] used a pyramid vision transformer for semantic segmentation. Swin [30] designed a window-based attention mechanism in the vision transformer to effectively improve attention computing. They added a UperNet head [50] for semantic segmentation. MaskFormer [9] and Mask2Former [10] proposed universal image segmentation architecture by combining the transformer decoder and pixel decoder. No matter whether fully or weakly supervised semantic segmentation, almost all segmentation models rely on the ImageNet [11] Pre-train models, and all the model parameters require to train or finetune, which requires a large number of computing costs, while we used a frozen CLIP model as the backbone, leading to much less resource on the computation.

3. Method

3.1. Overview

Fig. 2 shows the whole framework of our approach, including four main modules: a frozen CLIP backbone (image encoder and text encoder) to encode the image and text, a classification process to produce initial CAM, a decoder to generate segmentation predictions, a RFM to refine initial CAM to provide pseudo labels for training.

The training pipeline is divided into the following steps:

1. First of all, the image is input to the CLIP image encoder for image features. Besides, the foreground and background class labels are used to build text prompts and then input to the CLIP text encoder to generate the corresponding text features. Note here both image and text encoders are frozen during training.

2. Then, the classification scores are generated by computing distances between image features (after pooling) and text features. Based on classification scores, GradCAM [41] is utilized to generate the initial CAM.
3. Besides, image features from the last layer of each transformer block in the frozen CLIP image encoder are input to our proposed decoder for the final segmentation predictions.
4. Simultaneously, the intermediate feature maps from our decoder are used to generate an affinity map. Then, the affinity map is input to our proposed RFM with the multi-head attention maps from each block of the frozen CLIP image encoder.
5. Finally, RFM outputs a refining map to refine the initial CAM. After post-processing, the final converted pseudo label from refined CAM is used to supervise the training.

3.2. Frozen CLIP Feature Decoder

We use the frozen CLIP encoder with ViT-B as the backbone, which is not optimized during training. Therefore, how to design a decoder that interprets CLIP features to semantic features becomes a core challenge. We propose a light decoder based on the transformer architecture to conduct semantic segmentation using CLIP features as input.

Specifically, suppose the input image is $I \in \mathbb{R}^{3\times H\times W}$, H and W represent the height and width of the image, respectively. After passing the CLIP image encoder, we generate the initial feature maps $\{F^l_{\text{init}}\}^N_{l=1}$ from the output of each transformer block in the encoder, where l represents the index of the block. Then, for each feature map F^l_{init}, an individual MLP module is used to generate new corresponding feature maps F^l_{new}:

$$F^l_{\text{new}} = W^1_{\text{fc}}(\text{ReLU}(W^2_{\text{fc}}(F^l_{\text{init}}))), \tag{1}$$

where W^1_{fc} and W^2_{fc} are two different fully-connected layers. $\text{ReLU}(\cdot)$ is the ReLU activation function.

After that, all new feature maps $\{F^l_{\text{new}}\}^N_{l=1}$ are concatenated together, which are then processed by a convolution layer to generate a fused feature map F_u:

$$F_u = \text{Conv}(\text{Concat}[F^1_{\text{new}}, F^2_{\text{new}}, .., F^N_{\text{new}}]), \tag{2}$$

where $F_u \in \mathbb{R}^{d\times h\times w}$, where d, h, and w represent the channel dimension, height, and width of the feature map. $\text{Conv}(\cdot)$ is a convolutional layer, $\text{Concat}[\cdot]$ is the concatenation operation.

Finally, we design several sequential multi-head transformer layers to generate the final prediction P:

$$P = \text{Conv}(\phi(F_u)) \uparrow, \tag{3}$$

where $P \in \mathbb{R}^{C\times H\times W}$, C is the class number including background. ϕ represents the sequential multi-head transformer blocks [12], each block contains a multi-head self-attention module, a feed-forward network, and two normalization layers, as shown in the upper right corner of Fig. 2. $\uparrow$ is an upsample operation to align the prediction map size with the original image.

3.3. Frozen CLIP CAM Refinement

To provide supervision for the prediction P in Eq. (3), we generate the pixel-level pseudo label from the initial CAM of the frozen backbone. The frozen backbone can only provide static CAM, which means pseudo labels used as supervision cannot be improved during training. The same errors in pseudo labels lead to uncorrectable optimization in the wrong directions. Therefore, we design the Frozen CLIP CAM Refinement module (RFM) to dynamically update CAM to improve the quality of pseudo labels.

We first follow [29] to generate the initial CAM. For the given Image I with its class labels, I is input to the CLIP image encoder. The class labels are used to build text prompts and input to the CLIP text encoder. Then, the extracted image features (after pooling) and text features are used to compute the distance and further activated by the softmax function to get the classification scores. After that, we use GradCAM [41] to generate the initial CAM $M_{\text{init}} \in \mathbb{R}^{(|C_I|+1)\times h\times w}$, where $(|C_I|+1)$ indicates all class labels in the image I including the background. More details can be found in our supplementary material or [29].

To thoroughly utilize the prior knowledge of CLIP, the CLIP model is fixed. Although we find that such a frozen backbone can provide strong semantic features for the initial CAM with only image-level labels, as illustrated in Fig. 3(a), M_{init} cannot be optimized as it is generated from the frozen backbone, limiting the quality of pseudo labels. Therefore, how to rectify M_{init} during training becomes a key issue. Our intuition is to use feature relationships to rectify the initial CAM. However, we cannot directly use the attention maps from the CLIP image encoder as the feature relationship, as such attention maps are also fixed. Nevertheless, the decoder is constantly being optimized, and we attempt to use its features to establish feature relationships to guide the selection of attention values from the CLIP image encoder, keeping useful prior CLIP knowledge and removing noisy relationships. With more reliable feature relationships, the CAM quality can be dynamically enhanced.

In detail, we first generate an affinity map based on the feature map F_u in Eq. (2) from our decoder:

$$A_f = \text{Sigmoid}(F^T_u F_u), \tag{4}$$

where $F_u \in \mathbb{R}^{d\times h\times w}$ is first flattened to $\mathbb{R}^{d\times hw}$. $\text{Sigmoid}(\cdot)$ is the sigmoid function to guarantee the range of the output is from 0 to 1. $A_f \in \mathbb{R}^{hw\times hw}$ is the generated affinity map. T means matrix transpose.

Then we extract all the multi-head attention maps from the frozen CLIP image encoder, denoted as $\{A_s^l\}_{l=1}^N$ and each $A_s^l \in \mathbb{R}^{hw \times hw}$. For each A_s^l, we use A_f as a standard map to evaluate its quality:

$$S^l = \sum_{i=1}^{hw} \sum_{j=1}^{hw} \left| A_f(i,j) - A_s^l(i,j) \right|, \tag{5}$$

We use the above S^l to compute a filter for each attention map:

$$G^l = \begin{cases} 1, & \text{if } S^l < \frac{1}{N-N_0+1} \sum_{l=N_0}^{N} S^l \\ 0, & \text{else} \end{cases}, \tag{6}$$

where $G^l \in \mathbb{R}^{1\times 1}$, and it is expanded to $G_e^l \in \mathbb{R}^{hw \times hw}$ for further computation. We use the average value of all S^l as the threshold. If the current S^l is less than the threshold, it is more reliable, and we set its filter value as 1. Otherwise, we set the filter value as 0. Based on this rule, we keep high-quality attention maps and remove weak attention maps.

We then combine A_f and the above operation to build the refining map:

$$R = \frac{A_f}{N_m} \sum_{l=1}^{N} G_e^l A_s^l, \tag{7}$$

where N_m is the number of valid A_s^l, *i.e.*, $N_m = \sum_{l=N_0}^{N} G^l$.

Then, following the previous approaches [29], we generate the refined CAM:

$$M_f^c = \left(\frac{R_{\text{nor}} + R_{\text{nor}}^T}{2} \right)^{\alpha} \cdot M_{\text{init}}^c, \tag{8}$$

where c is the specific class, M_f^c is the refined CAM for class c, R_{nor} is obtained from R using row and column normalization (Sinkhorn normalization [42]). α is a hyper-parameter. This part passes a box mask indicator [29] to restrict the refining region. M_{init}^c is the CAM for class c after reshaping to $\mathbb{R}^{hw\times 1}$. Finally, M_f is input to the online post-processing module, *i.e.*, pixel adaptive refinement module proposed in [39], to generate final online pseudo labels $M_p \in \mathbb{R}^{h\times w}$.

In this way, our RFM uses the updated feature relationship in our decoder to assess the feature relationship in the frozen backbone to select reliable relationships. Then, higher-quality CAM can be generated with the help of more reliable feature relationships for each image. Fig. 3 shows the detailed comparison of generated CAM using different refinement methods. Our method generates more accurate responses than the static refinement method proposed in [29] and the initial CAM.

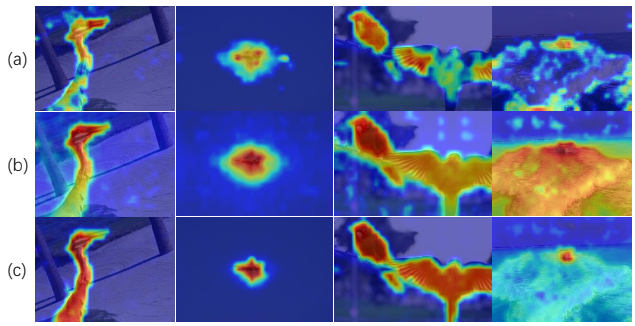

Figure 3. Qualitative comparison about the CAM. (a) Initial CAM. (b) Refined CAM by attention maps proposed in [29]. (c) Our refined CAM. Our method produces more accurate responses.

3.4. Loss Function

In our RFM, we use the affinity map A_f to select the attention map and build the final refining map. Therefore, the effectiveness of A_f directly determines the quality of the online pseudo labels. Considering A_f is generated using the feature map F_u in our decoder, and is a learnable module, we propose a learning process for A_f that uses the converted online pseudo label from M_p as supervision.

Specifically, M_p is first converted to the pixel-wise affinity label for each pair of pixels:

$$\hat{A} = O_h(M_p)^T O_h(M_p), \tag{9}$$

where $O_h(\cdot)$ is one-hot encoding and $O_h(M_p) \in \mathbb{R}^{C\times hw}$, $\hat{A} \in \mathbb{R}^{hw\times hw}$ is the affinity label. $\hat{A}(i,j) = 1$ means pixel i and j has the same label, otherwise, $\hat{A}(i,j) = 0$.

Based on the above label $\hat{A}$ and the online label M_p, the whole loss function of our WeCLIP is:

$$\mathcal{L} = \mathcal{L}_{ce}(P, M_p \uparrow) + \lambda \mathcal{L}_{ce}(A_f, \hat{A}), \tag{10}$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss, $M_p \uparrow \in \mathbb{R}^{H\times W}$, and λ is the weighting parameter. P is the prediction in Eq. (3). With Eq. (10), more accurate feature relationships are established for higher-quality pseudo labels. In turn, with better pseudo labels, more precise feature relationships are established. Thus, our decoder and RFM can benefit from each other to boost the training.

4. Experiment

4.1. Datasets

Following the setting in most previous weakly supervised semantic segmentation approaches [14, 19, 23], two datasets are used to evaluate our approach: PASCAL VOC 2012 [15] and MS COCO-2014 [28]. PASCAL VOC 2012 is appended with SBD [17] to expand the dataset, and the whole dataset contains 10,582 training images, 1,446 validation images, and 1,456 test images with 20 foreground

classes. The MS COCO-2014 dataset includes approximately 82,000 training images and 40,504 validation images with 80 foreground classes.

Mean Intersection-over-Union (mIoU) is applied as the evaluation criterion.

4.2. Implementation Details

We use the frozen CLIP backbone with the ViT-16-base architecture [13], N is a fixed number that equals 12. For training on the PASCAL VOC 2012 dataset, the batchsize is set as 4, and the maximum iteration is set as $30,000$. For training on the MS COCO-2014 dataset, we set batchsize as 8, and the maximum iteration as $80,000$.

All other settings adopt the same parameters for two datasets during training: We use AdamW [32] as the optimizer, the learning rate is $2e^{-3}$ with weight decay $1e^{-3}$, and all images are cropped to 320×320 during training. λ in Eq. (10) is set as 0.1, The dimension of the MLP module (Eq. (1)) in our decoder is set as 256. In ϕ of Eq. (3), three transformer encoder (the multi-head number is 8) layers are cascaded to generate the final feature map, and each layer's output dimension is 256. N_0 in Eq. (6) is set as 6. α is set as 2 in Eq. (8) following [29].

During inference, we use the multi-scale with $\{0.75, 1.0\}$. Following previous approaches [39, 40, 53], DenseCRF [21] is used as the post-processing method to refine the prediction.

4.3. Comparison with State-of-the-art Methods

In Tab. 1, we compare our approach with other state-of-the-art approaches on the PASCAL VOC 2012 dataset. It can be seen that our WeCLIP reaches 76.4% and 77.2% mIoU on *val* and *test* sets, both of which significantly outperform other single-stage approaches by a large margin. Specifically, compared to ToCo [40], the previous state-of-the-art single-stage approach, our WeCLIP brings 5.3% and 5.0% mIoU increase on *val* and *test* set, respectively. Besides, CLIP-ES [29] is the previous state-of-the-art multi-stage approach, and it is also a CLIP-based solution. Our approach performs much better than it, with 3.6% and 3.3% mIoU increase.

Tab. 2 shows the comparisons between our approach and previous state-of-the-art approaches on MS COCO-2014 *val* set. Our approach achieves new state-of-the-art performance, reaching 47.1% mIoU. Compared to other single-stage approaches, our WeCLIP brings more than 4.8% mIoU increase, which is a significant improvement. More importantly, our WeCLIP also outperforms other multi-stage approaches by a clear margin with fewer training steps. Considering our WeCLIP uses a frozen backbone, it shows great advantages to this task.

In Tab. 3, we compare the training cost between our approach and other state-of-the-art approaches on the PASCAL VOC 2012 dataset. It can be seen that our approach only needs 6.2G GPU memory, while other approaches require at least 12G GPU memory. ToCo [40] has less training time than us, but its GPU memory is much higher than our WeCLIP. More importantly, ToCo [40] spent 4 hours with 20,000 training iterations, while our WeCLIP spent 4.5 hours with 30,000 iterations, which also shows the high training efficiency of our approach.

In Fig. 4, we show some qualitative comparisons between our approach and other approaches on the PASCAL VOC 2012 and MS COCO-2014 *val* set. The visual results show that our WeCLIP generates more accurate object details than ToCo [40] for both the two datasets.

Table 1. Comparison of state-of-the-art approaches on the PASCAL VOC 2012 *val* and *test* dataset. mIoU (%) as the evaluation metric. I: image-level labels; S: saliency maps; L: language. mIoU as the evaluation metric. Without a specific description, results are reported with multi-scales and DenseCRF during inference.

Method	Backbone	Sup.	*val*	*test*
mutil-stage weakly supervised approaches				
RCA$_{\text{CVPR'22}}$ [64]	ResNet101	I+S	72.2	72.8
L2G$_{\text{CVPR'22}}$ [19]	ResNet101	I+S	72.1	71.7
Mat-label$_{\text{ICCV'23}}$ [45]	ResNet101	I+S	73.3	**74.0**
S-BCE$_{\text{ECCV'22}}$[49]	ResNet38	I+S	68.1	70.4
RIB$_{\text{NeurIPS'21}}$ [23]	ResNet38	I	68.3	68.6
W-OoD$_{\text{CVPR'22}}$ [24]	ResNet101	I	69.8	69.9
ESOL$_{\text{NeurIPS'22}}$ [25]	ResNet101	I	69.9	69.3
VML$_{\text{IJCV'22}}$ [38]	ResNet101	I	70.6	70.7
AETF$_{\text{ECCV'22}}$ [54]	ResNet38	I	70.9	71.7
MCTformer$_{\text{CVPR'22}}$ [52]	ViT+Res38	I	70.4	70.0
CDL$_{\text{IJCV'23}}$ [58]	ResNet101	I	72.4	72.2
ACR$_{\text{CVPR'23}}$ [22]	ViT	I	72.4	72.4
BECO$_{\text{CVPR'23}}$ [37]	MIT-B2	I	73.7	73.5
FPR$_{\text{ICCV'23}}$ [5]	ResNet101	I	70.0	70.6
USAGE$_{\text{ICCV'23}}$ [35]	ResNet38	I	71.9	72.8
CLIMS$_{\text{CVPR'22}}$ [51]	ViT+Res101	I+L	70.4	70.0
CLIP-ES$_{\text{CVPR'23}}$ [29]	ViT+Res101	I+L	**73.8**	73.9
single-stage weakly supervised approaches				
1Stage$_{\text{CVPR'20}}$ [3]	ResNet38	I	62.7	64.3
RRM$_{\text{AAAI'20}}$ [55]	ResNet38	I	62.6	62.9
AA&AR$_{\text{ACMMM'21}}$ [61]	ResNet38	I	63.9	64.8
SLRNet$_{\text{IJCV'22}}$ [34]	ResNet38	I	67.2	67.6
AFA$_{\text{CVPR'22}}$ [39]	MIT-B1	I	66.0	66.3
TSCD$_{\text{AAAI'23}}$ [53]	MIT-B1	I	67.3	67.5
ToCo$_{\text{CVPR'23}}$ [40]	ViT	I	71.1	72.2
ours-WeCLIP (w/o CRF)	ViT	I+L	74.9	75.2
ours-WeCLIP (w/ CRF)	ViT	I+L	**76.4**	**77.2**

4.4. Ablation Studies

We conduct ablation studies on the PASCAL VOC 2012 *val* set to evaluate the effectiveness of our approach. CRF is not used to refine the final prediction.

Tab. 4 shows the influence of our proposed decoder and RFM. As a single-stage approach, the decoder is necessary.

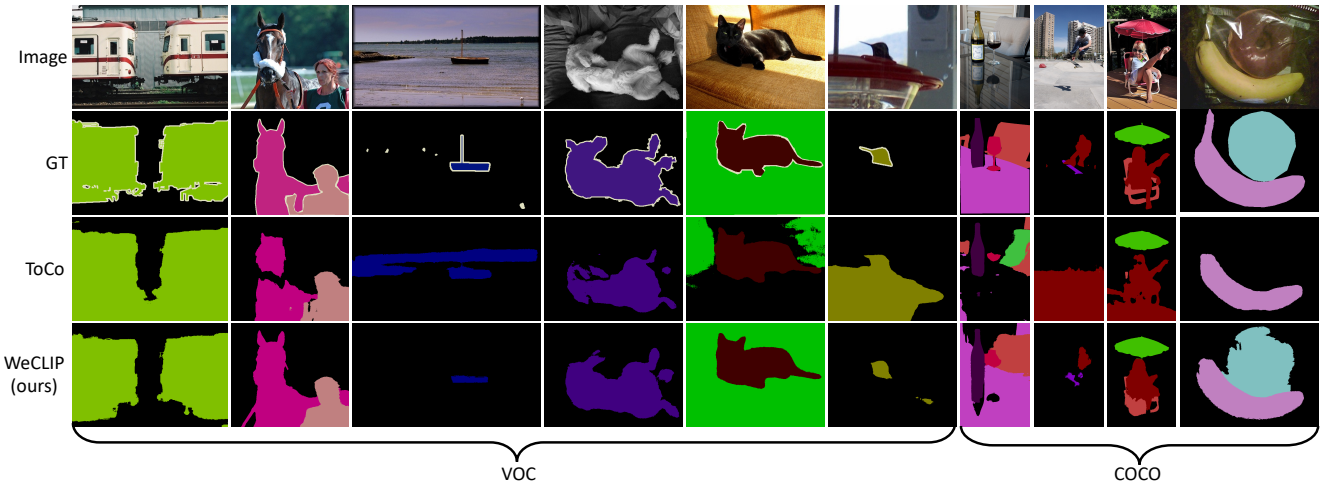

Figure 4. Qualitative comparisons bewteen our approach and ToCo [40] on PASCAL VOC 2012 and MS COCO-2014 *val* set. Our approach generates more detailed visual results.

Table 2. Comparison with other state-of-the-art methods on MS COCO-2014 *val* set.

Method	Backbone	Sup.	mIoU (%)
mutil-stage weakly supervised approaches			
$L2G_{CVPR'22}$ [19]	ResNet101	I+S	44.2
$RCA_{CVPR'22}$ [64]	ResNet101	I+S	36.8
$PMM_{ICCV'21}$ [26]	ResNet101	I	36.7
$RIB_{NeurIPS'21}$ [23]	ResNet101	I	43.8
$VWL_{IJCV'22}$ [38]	ResNet101	I	36.2
$MCTformer_{CVPR'22}$ [52]	ViT+Res38	I	42.0
$SIPE_{CVPR'22}$ [7]	ResNet38	I	43.6
$ESOL_{NeurIPS'22}$ [25]	ResNet101	I	42.6
$FPR_{ICCV'23}$ [5]	ResNet101	I	43.9
$USAGE_{ICCV'23}$ [35]	ResNet101	I	44.3
$CDL_{IJCV'23}$ [58]	ResNet101	I	**45.5**
$ACR_{CVPR'23}$ [22]	ResNet38	I	45.3
$BECO_{CVPR'23}$ [37]	ViT	I	45.1
$CLIP\text{-}ES_{CVPR'23}$ [29]	ViT+Res101	I+L	45.4
single-stage weakly supervised approaches			
$SLRNet_{IJCV'22}$ [34]	ResNet38	I	35.0
$AFA_{CVPR'22}$ [39]	MIT-B1	I	38.9
$TSCD_{AAAI'23}$ [53]	MIT-B1	I	40.1
$ToCo_{CVPR'23}$ [40]	ViT	I	42.3
ours-WeCLIP (w/o CRF)	ViT	I+L	46.4
ours-WeCLIP (w/ CRF)	ViT	I+L	**47.1**

Table 3. Training cost comparisons on PASCAL VOC 2012 dataset. All methods are run on NVIDIA RTX 3090 GPUs.

Method	Train time	Maximum GPU memory
MCTformer [52]	25h	12G
CLIP-ES [29]	7h	12G
ToCo [40]	**4h**	>24G
WeCLIP	4.5h	**6.2G**

Table 4. Ablation study of each component in our WeCLIP on PASCAL VOC 2012 *val* set.

Decoder	RFM	mIoU (%)
✓		68.7
✓	✓	**74.9**

Table 5. Ablation study about transformer layer numbers in ϕ of Eq. (3) on PASCAL VOC 2012 *val* set.

ϕ (Trans. Layer)	1	2	3	4	5
mIoU (%)	73.2	74.4	**74.9**	72.6	70.3

We cannot generate the prediction without it. Besides, introducing RFM brings a clear improvement, with a 6.2% mIoU increase. Since RFM is designed to improve the online pseudo labels, this increase also evaluates its effectiveness in generating higher quality pseudo labels.

Tab. 5 reports the influence of the number of transformer layers in our decoder, *i.e.*, ϕ in Eq. (3). The performance increases when the layer number increases to 3. This is because the limited size of the decoder cannot capture enough feature information, and it is easy to under-fit the features. With the increase of layer number, the decoder learns better feature representation. However, the performance drops if the layer number is larger than 3. One possible reason is that deeper decoder layers cause the over-fitting problem. Thus, it is reasonable that the performance drops after increasing to 4 or 5 for ϕ.

In Tab. 6, we evaluate the effectiveness of our refining map. When only A_s is used, it means that all attention maps are selected to refine the CAM, *i.e.*, the same process proposed in [29], it generates 71.8% mIoU score. Note that

Table 6. Ablation study of the dynamic refining map R in Eq. (7).

A_f	G_e	A_s	mIoU (%)
✓			65.7
		✓	71.8
	✓	✓	72.3
✓		✓	74.3
✓	✓	✓	**74.9**

such a process is a static operation, which is not optimized during training. Introducing G_e and A_f clearly improves it with 0.5% and 2.5% mIoU increase, respectively. Finally, combining A_f, G_e, and A_s using Eq. (7) generates much better results than others, showing the effectiveness of our refining method. More importantly, using the affinity map from our decoder provides a dynamic refinement strategy, making the refinement process optimized during training.

4.5. Performance on Fully-supervised Semantic Segmentation

We also use our WeCLIP to tackle fully-supervised semantic segmentation. For fully-supervised semantic segmentation, it provides accurate pixel-level labels, so we remove the frozen text encoder and our RFM, only keeping the frozen image encoder and our decoder. Besides, the loss function removes the part related to $\hat{A}$. The framework can be found in our supplementary material.

Table 7. Performance on PASCAL VOC 2012 *val* and *test* set for fully-supervised semantic segmentation. mIoU as the evaluation metric. "L. Params" means learnable parameters during training.

Method	Backbone	L. Params	*val*	*test*
DeepLabV3*	ResNet101	58M	79.9	79.8
Mask2former [10]*	ResNet50	44M	77.3	-
SegNeXt-S [16]	MSCAN-S	13.9M	-	**85.3**
WeCLIP	ViT-B	5.7M	**81.6**	81.1

* results are reproduced by [33].

In Tab. 7, we evaluate our approach on PASCAL VOC 2012 set for fully-supervised semantic segmentation. Since our approach utilizes a frozen backbone, it has less trainable parameters, but high-level segmentation performance is maintained, showing its great potential for fully-supervised semantic segmentation.

To illustrate why the vision feature from frozen CLIP can be directly used for semantic segmentation, we show some feature visualization results to compare the difference between the CLIP features and ImageNet features in Fig. 5. We randomly select 200 images from the PASCAL VOC 2012 *train* set. Without any training or finetune, we use ViT-B as the backbone and directly initialize it with frozen pre-train weights. It can be found that features belonging to the same class, pre-trained by CLIP, are denser and clustered, while features belonging to the same class, pre-trained by ImageNet, are more sparse and decentralized. Fig. 5 indicates that the extracted features from the CLIP model can better represent semantic information for different classes, making features belonging to different classes not confused. With such discriminative features, It is more convenient to conduct segmentation tasks.

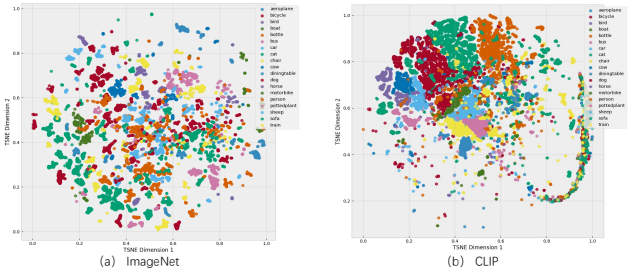

Figure 5. Feature visualization with T-SNE [44] to show why frozen CLIP can be used for semantic segmentation. Each color represents one specific category. (a) Frozen ImageNet pre-trained feature visualization of ViT-B. (b) Frozen CLIP pre-trained feature visualization of VIT-B. It can be seen that without any retraining, the features belonging to the same class from the frozen CLIP are more compact compared with that in (a). Best viewed in color.

5. Conclusion

We propose WeCLIP, a single-stage pipeline based on the frozen CLIP backbone for weakly supervised semantic segmentation. To interpret the frozen features for semantic prediction, we design a frozen CLIP feature decoder based on the transformer architecture. Meanwhile, we propose a frozen CLIP CAM refinement module, which uses the learnable feature relationship from our decoder to refine CAM, thus clearly improving the quality of pseudo labels. Our approach achieves better performance with less training cost, showing great advantages to tackle this task. We also evaluate the effectiveness of our approach to fully-supervised semantic segmentation. Our solution offers a different perspective from traditional approaches that the training of the backbone is unnecessary. We believe the proposed approach can further boost research in this direction.

Acknowledge: This work was supported by the National Key R&D Program of China (No.2022YFE0200300), the National Natural Science Foundation of China (No. 62301613 & No. 62301451)), the Taishan Scholar Program of Shandong (No. tsqn202306130), the Suzhou Basic Research Program (SYG202316), Shandong Natural Science Foundation (No. ZR2023QF046), Qingdao Postdoctoral Applied Research Project (No. QDBSH20230102091), and Independent Innovation Research Project of UPC (No. 22CX06060A).

References

[1] 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. 1, 2

[2] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, pages 2209–2218, 2019. 1, 2

[3] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *CVPR*, pages 4253–4262, 2020. 1, 2, 6

[4] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, pages 549–565, 2016. 1

[5] Liyi Chen, Chenyang Lei, Ruihuang Li, Shuai Li, Zhaoxiang Zhang, and Lei Zhang. Fpr: False positive rectification for weakly supervised semantic segmentation. In *ICCV*, 2023. 6, 7

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017. 3

[7] Qi Chen, Lingxiao Yang, Jian-Huang Lai, and Xiaohua Xie. Self-supervised image-specific prototype exploration for weakly supervised semantic segmentation. In *CVPR*, pages 4288–4298, 2022. 7

[8] Zhaozheng Chen, Tan Wang, Xiongwei Wu, Xian-Sheng Hua, Hanwang Zhang, and Qianru Sun. Class re-activation maps for weakly-supervised semantic segmentation. In *CVPR*, pages 969–978, 2022. 2

[9] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, pages 17864–17875, 2021. 3

[10] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 3, 8

[11] 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. 1, 2, 3

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, pages 1–21, 2021. 3, 4

[13] 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*, 2021. 6

[14] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *CVPR*, pages 4320–4329, 2022. 1, 2, 5

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 5

[16] Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. In *NeurIPS*, pages 1140–1156, 2022. 8

[17] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, pages 991–998, 2011. 5

[18] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *ICCV*, pages 22157–22167, 2023. 2

[19] Peng-Tao Jiang, Yuqi Yang, Qibin Hou, and Yunchao Wei. L2g: A simple local-to-global knowledge transfer framework for weakly supervised semantic segmentation. In *CVPR*, pages 16886–16896, 2022. 1, 2, 5, 6, 7

[20] Siyu Jiao, Yunchao Wei, Yaowei Wang, Yao Zhao, and Humphrey Shi. Learning mask-aware clip representations for zero-shot segmentation. pages 35631–35653, 2023. 2

[21] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In *ICML*, pages 513–521, 2013. 6

[22] Hyeokjun Kweon, Sung-Hoon Yoon, and Kuk-Jin Yoon. Weakly supervised semantic segmentation via adversarial learning of classifier and reconstructor. In *CVPR*, pages 11329–11339, 2023. 6, 7

[23] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. In *NeurIPS*, pages 27408–27421, 2021. 1, 2, 5, 6, 7

[24] Jungbeom Lee, Seong Joon Oh, Sangdoo Yun, Junsuk Choe, Eunji Kim, and Sungroh Yoon. Weakly supervised semantic segmentation using out-of-distribution data. In *CVPR*, pages 16897–16906, 2022. 6

[25] Jinlong Li, Zequn Jie, Xu Wang, Xiaolin Wei, and Lin Ma. Expansion and shrinkage of localization for weakly-supervised semantic segmentation. In *NeurIPS*, pages 16037–16051, 2022. 1, 2, 6, 7

[26] Yi Li, Zhanghui Kuang, Liyang Liu, Yimin Chen, and Wayne Zhang. Pseudo-mask matters in weakly-supervised semantic segmentation. In *ICCV*, pages 6964–6973, 2021. 1, 7

[27] Zhiyuan Liang, Tiancai Wang, Xiangyu Zhang, Jian Sun, and Jianbing Shen. Tree energy loss: Towards sparsely annotated semantic segmentation. In *CVPR*, pages 16907–16916, 2022. 1, 2

[28] 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, 2014. 5

[29] Yuqi Lin, Minghao Chen, Wenxiao Wang, Boxi Wu, Ke Li, Binbin Lin, Haifeng Liu, and Xiaofei He. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In *CVPR*, pages 15305–15314, 2023. 1, 2, 4, 5, 6, 7

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *CVPR*, pages 10012–10022, 2021. 3

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 3

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2017. 6

[33] Quang Nguyen, Truong Vu, Anh Tran, and Khoi Nguyen. Dataset diffusion: Diffusion-based synthetic dataset generation for pixel-level semantic segmentation. In *NeurIPS*, 2023. 8

[34] Junwen Pan, Pengfei Zhu, Kaihua Zhang, Bing Cao, Yu Wang, Dingwen Zhang, Junwei Han, and Qinghua Hu. Learning self-supervised low-rank network for single-stage weakly and semi-supervised semantic segmentation. *IJCV*, 130(5):1181–1195, 2022. 6, 7

[35] Zelin Peng, Guanchun Wang, Lingxi Xie, Dongsheng Jiang, Wei Shen, and Qi Tian. Usage: A unified seed area generation paradigm for weakly supervised semantic segmentation. In *ICCV*, 2023. 6, 7

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 2

[37] Shenghai Rong, Bohai Tu, Zilei Wang, and Junjie Li. Boundary-enhanced co-training for weakly supervised semantic segmentation. In *CVPR*, pages 19574–19584, 2023. 6, 7

[38] Lixiang Ru, Bo Du, Yibing Zhan, and Chen Wu. Weakly-supervised semantic segmentation with visual words learning and hybrid pooling. *IJCV*, 130(4):1127–1144, 2022. 6, 7

[39] Lixiang Ru, Yibing Zhan, Baosheng Yu, and Bo Du. Learning affinity from attention: end-to-end weakly-supervised semantic segmentation with transformers. In *CVPR*, pages 16846–16855, 2022. 1, 2, 5, 6, 7

[40] Lixiang Ru, Heliang Zheng, Yibing Zhan, and Bo Du. Token contrast for weakly-supervised semantic segmentation. In *CVPR*, pages 3093–3102, 2023. 2, 6, 7

[41] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017. 2, 4

[42] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964. 5

[43] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *CVPR*, pages 3136–3145, 2019. 1, 2

[44] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008. 8

[45] Changwei Wang, Rongtao Xu, Shibiao Xu, Weiliang Meng, and Xiaopeng Zhang. Treating pseudo-labels generation as image matting for weakly supervised semantic segmentation. In *ICCV*, pages 755–765, 2023. 6

[46] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 3

[47] 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. 1

[48] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Zequn Jie, Yanhui Xiao, Yao Zhao, and Shuicheng Yan. Learning to segment with image-level annotations. *PR*, 59:234–244, 2016. 1

[49] Tong Wu, Guangyu Gao, Junshi Huang, Xiaolin Wei, Xiaoming Wei, and Chi Harold Liu. Adaptive spatial-bce loss for weakly supervised semantic segmentation. In *ECCV*, pages 199–216, 2022. 1, 6

[50] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 3

[51] Jinheng Xie, Xianxu Hou, Kai Ye, and Linlin Shen. Clims: cross language image matching for weakly supervised semantic segmentation. In *CVPR*, pages 4483–4492, 2022. 1, 2, 6

[52] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *CVPR*, pages 4310–4319, 2022. 1, 2, 6, 7

[53] Rongtao Xu, Changwei Wang, Jiaxi Sun, Shibiao Xu, Weiliang Meng, and Xiaopeng Zhang. Self correspondence distillation for end-to-end weakly-supervised semantic segmentation. In *AAAI*, 2023. 6, 7

[54] Sung-Hoon Yoon, Hyeokjun Kweon, Jegyeong Cho, Shinjeong Kim, and Kuk-Jin Yoon. Adversarial erasing framework via triplet with gated pyramid pooling layer for weakly supervised semantic segmentation. In *ECCV*, pages 326–344, 2022. 1, 6

[55] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Mingjie Sun, and Kaizhu Huang. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In *AAAI*, pages 12765–12772, 2020. 1, 2, 6

[56] Bingfeng Zhang, Jimin Xiao, Jianbo Jiao, Yunchao Wei, and Yao Zhao. Affinity attention graph neural network for weakly supervised semantic segmentation. *IEEE TPAMI*, 44 (11):8082–8096, 2021. 1, 2

[57] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Kaizhu Huang, Shan Luo, and Yao Zhao. End-to-end weakly supervised semantic segmentation with reliable region mining. *PR*, 128: 108663, 2022. 1

[58] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, and Yao Zhao. Credible dual-expert learning for weakly supervised semantic segmentation. *IJCV*, 131:1892–1908, 2023. 6, 7

[59] Fei Zhang, Chaochen Gu, Chenyue Zhang, and Yuchao Dai. Complementary patch for weakly supervised semantic segmentation. In *ICCV*, pages 7242–7251, 2021. 1

[60] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier align-

ment for continual learning on a pre-trained model. In *ICCV*, pages 19148–19158, 2023. 2

[61] Xiangrong Zhang, Zelin Peng, Peng Zhu, Tianyang Zhang, Chen Li, Huiyu Zhou, and Licheng Jiao. Adaptive affinity loss and erroneous pseudo-label refinement for weakly supervised semantic segmentation. In *ACM MM*, pages 5463–5472, 2021. 1, 2, 6

[62] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 3

[63] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 1

[64] Tianfei Zhou, Meijie Zhang, Fang Zhao, and Jianwu Li. Regional semantic contrast and aggregation for weakly supervised semantic segmentation. In *CVPR*, pages 4299–4309, 2022. 6, 7

[65] Hongguang Zhu, Yunchao Wei, Xiaodan Liang, Chunjie Zhang, and Yao Zhao. Ctp: Towards vision-language continual pretraining via compatible momentum contrast and topology preservation. In *ICCV*, pages 22257–22267, 2023. 2