CLIP is Also an Efficient Segmenter: A Text-Driven Approach for Weakly Supervised Semantic Segmentation

Yuqi Lin[1*] Minghao Chen[1*†] Wenxiao Wang[2] Boxi Wu[2] Ke Li[3]
Binbin Lin[2] Haifeng Liu[1] Xiaofei He[1]
[1]State Key Lab of CAD&CG, College of Computer Science, Zhejiang University
[2]School of Software Technology, Zhejiang University [3]Fullong Technology
{linyq5566, minghaochen01}@gmail.com

Abstract

Weakly supervised semantic segmentation (WSSS) with image-level labels is a challenging task. Mainstream approaches follow a multi-stage framework and suffer from high training costs. In this paper, we explore the potential of Contrastive Language-Image Pre-training models (CLIP) to localize different categories with only image-level labels and without further training. To efficiently generate high-quality segmentation masks from CLIP, we propose a novel WSSS framework called CLIP-ES. Our framework improves all three stages of WSSS with special designs for CLIP: 1) We introduce the softmax function into GradCAM and exploit the zero-shot ability of CLIP to suppress the confusion caused by non-target classes and backgrounds. Meanwhile, to take full advantage of CLIP, we re-explore text inputs under the WSSS setting and customize two text-driven strategies: sharpness-based prompt selection and synonym fusion. 2) To simplify the stage of CAM refinement, we propose a real-time class-aware attention-based affinity (CAA) module based on the inherent multi-head self-attention (MHSA) in CLIP-ViTs. 3) When training the final segmentation model with the masks generated by CLIP, we introduced a confidence-guided loss (CGL) focus on confident regions. Our CLIP-ES achieves SOTA performance on Pascal VOC 2012 and MS COCO 2014 while only taking 10% time of previous methods for the pseudo mask generation. Code is available at https://github.com/linyq2117/CLIP-ES.

1. Introduction

Semantic segmentation [7,40] aims to predict pixel-level labels but requires labor-intensive pixel-level annotations. Weakly supervised semantic segmentation (WSSS) is proposed to reduce the annotation cost. WSSS only requires weak supervision, *e.g.*, image-level labels [2], bounding boxes [10, 33], points [4] or scribbles [31, 42]. The most commonly used one is WSSS with image-level annotations, which is the focus of our paper.

Previous WSSS approaches [24, 43, 46, 48] with image-level labels typically follow a three-stage framework. First, a classification model is trained on the specific dataset to generate initial CAMs (Class Activation Maps). Then, the initial CAMs are refined by the pixel affinity network [1, 2] or extra saliency maps [18, 41]. At last, the refined CAMs serve as the pseudo masks to train a semantic segmentation model. Obviously, this multi-stage framework is compli-

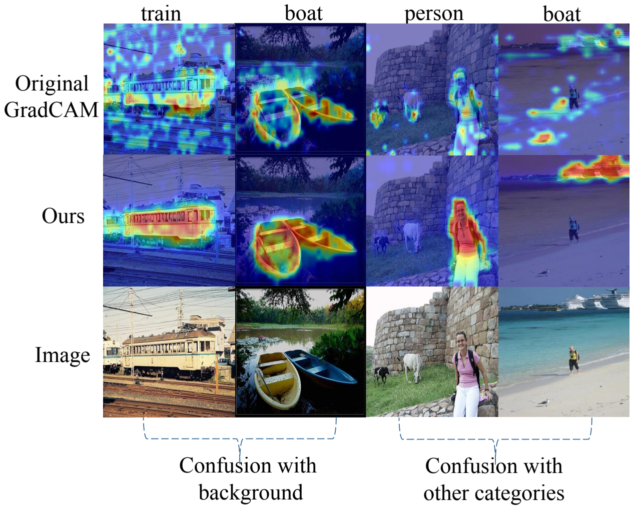

Figure 1. Effect of the softmax function on GradCAM of CLIP. The original GradCAM uses the logit (before the softmax) of the target class to compute gradient. We propose to compute gradient based on the probability (after the softmax). It can avoid confusion between the target class and background (the first two columns) and other object classes in the dataset (the last two columns).

[*] Equal contribution.
[†] Corresponding author.

cated as it needs to train multiple models at different stages, especially the separate classification model and affinity network in the first two stages. Although some end-to-end methods [3, 50] are proposed to improve efficiency, they tend to achieve poor performance compared to multi-stage methods. Therefore, it is a challenge to simplify the procedure of WSSS while maintaining its high performance.

Recently, the Contrastive Language-Image Pre-training (CLIP) [34], a model pre-trained on 400 million image-text pairs from the Internet to predict if an image and a text snippet are matched, has shown great success in the zero-shot classification. This dataset-agnostic model could transfer to unseen datasets directly. Besides, the powerful text-to-image generation ability of CLIP, *i.e.*, DALL-E2 [35], indicates the strong relation between texts and corresponding components in the image. On the other hand, multi-head self-attention (MHSA) in ViT [12] reflects semantic affinity among patches and has the potential to substitute for affinity network. Motivated by these, we believe CLIP with ViT architecture could simplify the procedure of WSSS and localize categories in the image through well-designed texts.

This paper proposes a new framework, CLIP-ES, to improve each stage in terms of efficiency and accuracy for WSSS. In the first stage, the generated CAMs are usually redundant and incomplete. Most methods [43,45] are based on binary cross-entropy for multi-label classification. The loss is not mutually exclusive, so the generated CAMs suffer from confusion between foreground and non-target foreground categories, *e.g.*, person and cow, or foreground and background categories, *e.g.*, boat and water, as shown in Fig. 1. The incompleteness stems from the gap between the classification and localization tasks, causing CAMs only focus on discriminative regions. To solve the confusion problems above, we introduce the softmax function into GradCAM to make categories mutually exclusive and define a background set to realize class-related background suppression. To get more complete CAMs and fully enjoy the merits inherited from CLIP, we investigate the effect of text inputs in the setting of WSSS and design two task-specific text-driven strategies: sharpness-based prompt selection and synonym fusion.

In the second stage, instead of training an affinity network as in previous works, we leverage the attention obtained from the vision transformer. However, the attention map is class-agnostic, while the CAM is class-wise. To bridge this gap, we propose a class-aware attention-based affinity (CAA) module to refine the initial CAMs in real-time, which can be integrated into the first stage. Without fine-tuning CLIP on downstream datasets, our method retains CLIP's generalization ability and is flexible to generate pseudo labels for new classes and new datasets.

In the last stage, the pseudo masks from the refined CAMs are viewed as ground truth to train a segmentation model in a fully supervised manner. However, the pseudo mask may be noisy and directly applied to training may mislead the optimization process. We proposed a confidence-guided loss (CGL) for training the final segmentation model by ignoring the noise in pseudo masks.

Our contributions are summarized as follows:

- We propose a simple yet effective framework for WSSS based on frozen CLIP. We reveal that given only image-level labels, CLIP can perform remarkable semantic segmentation without further training. Our method can induce this potential of localizing objects that exists in CLIP.
- We introduce the softmax function into GradCAM and design a class-related background set to overcome category confusion problems. To get better CAMs, some text-driven strategies inherited from CLIP are explored and specially redesigned for WSSS.
- We present a class-aware attention-based affinity module (CAA) to refine the initial CAMs in real time, and introduce confidence-guided loss (CGL) to mitigate the noise in pseudo masks when training the final segmentation model.
- Experiment results demonstrate that our framework can achieve SOTA performance and is 10x efficient than other methods when generating pseudo masks.

2. Related Work

2.1. Weakly Supervised Semantic Segmentation

Most existing approaches for WSSS train a classification network and extract localization maps from CNNs based on Class Activation Maps (CAMs) [51]. However, the initial CAMs are usually incomplete or redundant. Several methods are proposed to improve the quality of CAMs and the final segmentation at different stages.

Generating Initial CAM Stage. In this stage, to address the incompleteness problem, some works train classification networks with auxiliary tasks, and additional losses are designed to guide the model to discover more object regions [5, 36, 43, 47]. “Erasing” is another strategy that erases an image's or feature map's discriminative parts to force the network to discover more regions [17, 22, 44]. Some works accumulate multiple activations in the training process [18, 20, 49] and other works are from perspective of cross-image mining [16, 28, 41, 45], self-supervised mechanism [8,43] and anti-adversarial attack [24]. To solve the redundancy problem, previous works use softmax cross entropy as an additional loss to reactivate the model [9] or introduce extra out-of-distribution(OoD) data [25]. Recently, some transformer-based methods [37, 48] appear in the WSSS task and achieve competitive performance.

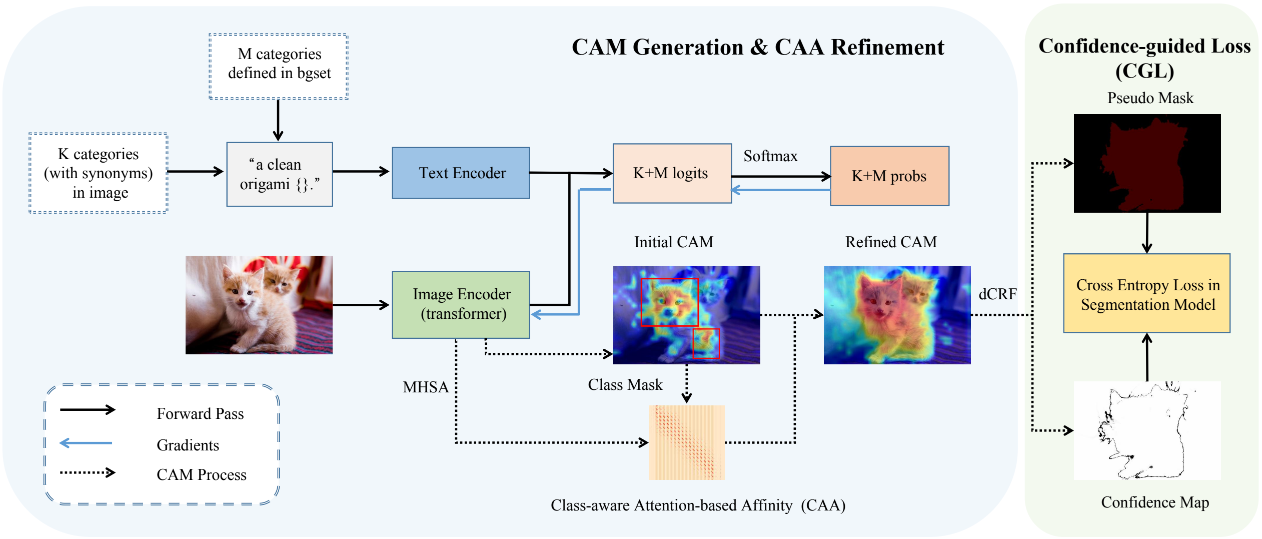
CAM Generation & CAA Refinement
M categories defined in bgset
K categories (with synonyms) in image
"a clean origami {}."
Text Encoder
K+M logits
Softmax
K+M probs
Initial CAM
Refined CAM
Image Encoder (transformer)
dCRF
MHSA
Class Mask
Forward Pass
Gradients
CAM Process
Class-aware Attention-based Affinity (CAA)
Confidence-guided Loss (CGL)
Pseudo Mask
Cross Entropy Loss in Segmentation Model
Confidence Map

Figure 2. An overview of our proposed framework. We introduce the softmax function into GradCAM and define a class-related background set to make categories mutually exclusive. K and M represent the number of categories in an image and background set, respectively. The initial CAMs are generated by Grad-CAM with well-designed texts (*e.g.*, prompt selection, synonym fusion). CAA module is proposed based on intrinsic MHSA in the transformer to refine the initial CAMs in real time. The whole CAM generation process is training-free. CGL ignores noisy positions when computing loss based on the confidence map.

Refining Initial CAM Stage. In this stage, pairwise semantic affinity is typically learned to refine CAM maps. PSA [2] trains a network to learn pixel affinity and propagate the semantics of strong responses in attention maps to semantically similar pixels. IRNet [1] and BES [6] synthesize class boundaries and expand the object coverage until boundaries. Another approach exploits additional saliency maps to obtain precise background or distinguish co-occurring objects [15, 19, 27].

Training Segmentation Model Stage. Traditional methods [19, 27] generate pseudo masks from CAMs by applying a global threshold, which can't fully utilize CAMs due to ignorance of confidence information. Only a few works attempt to suppress the noise at this stage. PMM [30] proposes the pretended under-fitting strategy to reweight losses of potential noise pixels. URN [29] scales the prediction map multiple times for uncertainty estimation. However, the former is merely operated on the loss level and doesn't use confidence while the latter is time-consuming for multiple dense CRF processes.

2.2. Contrastive Language-Image Pretraining

Contrastive Language-Image Pretraining (CLIP) [34] consists of an image encoder and a text encoder. It learns corresponding embeddings and measures the similarity between images and texts. Benefiting from this flexible framework, CLIP can be trained on super-large datasets and is widely used on the downstream zero-shot task. CLIMS [46] first introduced CLIP into WSSS to activate more complete object regions and suppress background regions. However, in CLIMS, CLIP is just a tool to evaluate the existence of objects and another CNN model is used to generate CAMs. In this paper, we directly use CLIP to generate CAMs and thoroughly explore the relationship between the text and objects in the image, which is more simple and more efficient.

3. Method

In this section, we propose our CLIP-ES framework, which is depicted in Fig. 2. We first review GradCAM and CLIP, and demonstrate the effect of the softmax function on GradCAM with the corresponding class-related background suppression strategy. Then, we introduce two text-driven strategies proposed for CLIP in the WSSS setting: sharpness-based prompt selection and synonym fusion. Finally, we present class-aware attention-based affinity (CAA) and confidence-guided loss (CGL) in detail.

3.1. Softmax-GradCAM

Class Activation Mapping (CAM) [51] is widely used to identify the discriminative regions for the target class by the weighted combination of feature maps. However, it is only applicable to specific CNN architectures, *e.g.*, models with a global average pooling (GAP) layer immediately after the feature maps. GradCAM [38] uses the gradient information to combine feature maps and thus there is no requirement for network architecture. For original GradCAM, the class feature weights can be calculated as Eq. (1):

$$w_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \tag{1}$$

where w_k^c is the weight corresponding to *c-th* class for *k-th* feature map, Z is the number of pixels in the feature map, Y^c is the logit score for *c-th* class and A_{ij}^k represents the activation value for *k-th* feature map at location (i, j). Then the CAM map of class c at spatial location (i, j) can be obtained by Eq. (2). *ReLU* is adopted to ignore features that negatively influence the target class.

$$CAM_{ij}^c = ReLU\left(\sum_k w_k^c A_{ij}^k\right) \tag{2}$$

Pretrained CLIP models include two architectures, *e.g*., ResNet-based and ViT-based. Note that Grad-CAM is not only applicable to CNN-based architecture but also works on the vision transformer. In this paper, we leverage the ViT-based CLIP model because the CNN-based model fails to explore the global context and suffers from the discriminative part domain heavily. The comparison between these two architectures can be found in Appendix.

Our work adapts GradCAM to CLIP. In vanilla Grad-CAM [38], the final score is the logits before the softmax function. Due to the multi-label setting of WSSS, the classification network often employs the binary cross entropy loss [43, 45], thus lacking competition among different classes. CLIP is trained by cross-entropy loss with softmax, but it still suffers from the category confusion problem in our experiment. We assume it is because the training data of CLIP are image-text pairs rather than a fixed set of separate categories. For an image, the corresponding text snippet could contain visual concepts of several classes, which can't compete with each other through softmax either. This paper introduces the softmax function into GradCAM to make different categories mutually exclusive. Specifically, the final score is computed by softmax as follows:

$$s^c = \frac{\exp(Y^c)}{\sum_{c'=1}^{C}\exp(Y^{c'})} \tag{3}$$

s^c is the score for *c-th* class after softmax. The processed scores are then used to compute the gradient, and the class feature weights can be calculated as:

$$\begin{aligned} w_k^c &= \frac{1}{Z}\sum_i\sum_j\sum_{c'}\frac{\partial Y^{c'}}{\partial A_{ij}^k} * \frac{\partial s^c}{\partial Y^{c'}} \\ &= \frac{1}{Z}\sum_i\sum_j\frac{\partial Y^{c}}{\partial A_{ij}^k} * s^c(1-s^c) \\ &+ \frac{1}{Z}\sum_i\sum_j\sum_{c'\neq c}\frac{\partial Y^{c'}}{\partial A_{ij}^k} * s^c(-s^{c'}) \end{aligned} \tag{4}$$

Eq. (4) indicates that the weight of the target feature map will be suppressed by non-target classes. So the corresponding CAMs of the target class can be revised by the remaining classes. However, the competition is only limited to categories defined in the dataset. To disentangle pixels of the target class from background classes, we propose a class-related background suppression method. We define a background category set containing M common class-related categories for classes defined in datasets. In this way, pixels of background categories will be suppressed. Thanks to the zero-shot capability of CLIP, we only need to revise input texts rather than retrain the classification network for background categories like previous training-based methods.

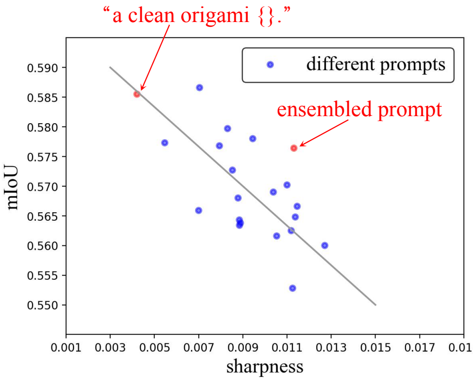

Figure 3. Relations between $sharpness$ and $mIoU$ using different prompts on PASCAL VOC 2012 train set. "a clean origami {}." is the prompt we finally adopted in our paper.

3.2. Text-driven Strategies

For CLIP, the text encoder acts as a linear classifier weight generator based on the text specifying the visual concepts the classes represent. Our framework could enjoy multiple merits inherited from CLIP by designing specific text inputs. In this part, we re-explore the effect of text inputs under the WSSS setting and propose sharpness-based prompt selection and synonyms fusion to boost the CAM generation process.

3.2.1 Sharpness-based Prompt Selection

We find that the performance of prompt ensembling differs between the classification task and the WSSS task. Specifically, prompt ensembling can outperform every single prompt by a large margin for the classification task on ImageNet [11], while it is not the optimal choice when performing WSSS on PASCAL VOC [14]. We suspect this difference is primarily due to the varying amount of labels per image. The classification dataset, *e.g*., ImageNet, is single-labeled, while the segmentation dataset, *e.g*., PASCAL VOC, is multi-labeled. The former aims to assign a maximum score for the unique target class, while the latter need to consider all target classes in an image. We claim that prompt ensembling will make the target class with the top score more prominent. But for multi-labeled images, a prominent target class will suppress scores of other target classes. This affects subsequent gradient computing for

GradCAM and leads to poor segmentation performance.

To verify our conjecture, we design a metric, namely *sharpness*, to measure the distribution of target class scores for multi-label images using different prompts. This metric is inspired by *Coefficient of Variation*, a metric widely used in statistics. Assume there are n images in the dataset and $k(k >= 1)$ classes in an image, the *sharpness* based on a specific prompt can be calculated as follows:

$$sharpness(\text{prompt}) = \frac{\sum_i^n var(s_{i1}, ..., s_{ik})}{\sum_i^n mean(s_{i1}, ..., s_{ik})} \quad (5)$$

s_{ij} represents scores for *j-th* class after softmax in *i-th* image. Since *Coefficient of Variation* is unstable when *mean* is close to 0, we use variance instead of standard deviation to highlight the effect of dispersion.

In Fig. 3, we compare *sharpness* and corresponding segmentation results among 20 prompts randomly selected from the ImageNet prompts used in CLIP[1] on Pascal VOC 2012 train set. As the result demonstrates, our proposed metric is roughly negatively correlated to segmentation performance. Consequently, ***sharpness* can serve as convenient guidance for prompt choice, and only image-level labels are needed.** After trial and error, we find that some abstract descriptions, *e.g.*, *"origami"* and *"rendering"*, and some adjectives, *e.g.*, *"clean"*, *"large"* and *"weird"*, have a positive effect on segmentation performance. We finally select *"a clean origami* {}*."* as our prompt, which has the lowest *sharpness*.

3.2.2 Synonym Fusion

Since the category names provided in the datasets are limited, we use synonyms to enrich semantics and disambiguate. There are various strategies to merge semantics of different synonyms, *e.g.*, sentence-level, feature-level, or CAM-level. We provide a detailed comparison of the three strategies in the Appendix. In this paper, we merge synonyms at the sentence level. Specially, we put different synonyms into one sentence, *e.g.*, "A clean origami of person, people, human". This can disambiguate when facing polysemous words and is time-efficient as other methods require multiple forward passes. The synonyms can easily be obtained from WordNet or the nearest Glove word embedding. In addition, the performance of some classes can be further improved by customizing specific words. For example, CAMs of "person" tend to focus on faces, while the ground truth segmentation masks cover the whole body. It is likely that "person" and "clothes" are treated as two different categories in CLIP. By replacing "person" with "person with clothes", this problem can be alleviated to some extent.

3.3. Class-aware Attention-based Affinity (CAA)

Recently, some works [37, 48] use attention obtained from the transformer as semantic-level affinity to refine initial CAMs. But the improvement is limited and they still require an additional network [48] or extra layers [37] to further refine CAMs. It is because the original multi-head self-attention (MHSA) is class-agnostic, while the CAM is class-wise. Leveraging MHSA directly may amplify noise by propagating noisy pixels to its semantically similar regions during refinement, as is shown in Fig. 5.

We propose class-aware attention-based affinity (CAA) to improve vanilla MHSA. Given an image, we can get the class-wise CAM map $M_c \in R^{h\times w}$ for each target class c and the attention weight $W^{attn} \in R^{hw\times hw}$ from MHSA. For the attention weight, which is asymmetric because of the different projection layers used by the query and key, we leverage Sinkhorn normalization [39] (alternately applying row-normalization and column-normalization) to convert it to a doubly stochastic matrix D, and the symmetric affinity matrix A can be obtained as follows:

$$A = \frac{D + D^T}{2}, where D = Sinkhorn(W^{attn}). \quad (6)$$

For the CAM map $M_c \in R^{h\times w}$, we can obtain a mask map for each target class c by thresholding the CAM of this class with λ. We find connected regions on the mask map and use the minimum rectangle bounding boxes covering those connected regions. These boxes mask the affinity weight A, and then each pixel can be refined based on the masked affinity weight by its semantically similar pixels. We employ the bounding box mask rather than the pixel mask to cover more regions of the objects for the extreme incompleteness of initial CAMs. We repeat this refinement multiple times, and this process can be formalized as follows.

$$M_c^{aff} = B_c \odot A^t \cdot vec(M_c) \quad (7)$$

where $B_c \in R^{1\times hw}$ is box mask obtained from CAM of class c, $\odot$ is Hadamard product, t denotes the number of refining iterations and $vec(\cdot)$ means vectorization of a matrix. Note that we extract the attention map and CAM with the same forward pass. Hence, CAA refinement is real-time and requires no additional stage like previous works.

3.4. Confidence-guided Loss (CGL)

Each pixel in the CAM indicates the confidence of this position belonging to the target class. Most methods generate pseudo masks from CAMs by simply setting a threshold to distinguish target objects and backgrounds. It may bring noise into pseudo masks because those positions with low confidence are too uncertain to belong to the correct class. Thus, we attempt to ignore those unconfident positions and propose a confidence-guided loss (CGL) to make

[1] https://github.com/openai/CLIP/blob/main/notebooks

full use of generated CAMs. Specifically, given CAM maps $X \in R^{h \times w \times c}$ of an image with c target classes, the confidence map can be obtained as:

$$Conf(i,j) = \max(1 - \max_c(X(i,j,c)), \max_c(X(i,j,c))) \tag{8}$$

and the final loss is defined as Eq. (9):

$$\hat{L}(i,j) = \begin{cases} L(i,j), & Conf(i,j) >= \mu \\ 0, & Conf(i,j) < \mu \end{cases} \tag{9}$$

where $L(i,j)$ is the cross entropy loss between the prediction of the semantic segmentation model and the pseudo mask for pixel (i,j), and μ is a hyper-parameter to ignore pixels with low confidence.

4. Experiments

4.1. Experimental Setup

Datasets and Evaluation Metric. We evaluate our proposed framework on PASCAL VOC 2012 [14] and MS COCO 2014 [32] datasets. PASCAL VOC 2012 contains 21 categories (one background category). An augmented set with 10,582 images is used for training following [24, 27]. MS COCO 2014 contains 80 object classes and one background class. It includes 82,081 images for training and 40,137 images for validation. We only used image-level ground-truth labels during CAM generation. The mean Intersection over Union (mIoU) is adopted as the evaluation metric for all experiments.

Implementation Details. For CAM generation, we adopt CLIP pre-trained model ViT-B-16 [34]. The feature map used to generate CAM is the one before the last self-attention layer in ViT. We replace the class token with the average of remaining tokens to compute final logits, which can significantly boost the performance. Detailed analysis is discussed in Appendix. Input images remain their original size, and we do not use the multi-scale strategy during inference. λ used in the CAA module is set to 0.4 and 0.7 for VOC and COCO, respectively. The generated CAMs are further post-processed by dense CRF [21] to generate final pseudo masks. For final segmentation, we use ResNet101-based DeepLabV2 following prior works [24, 27, 46], and more details are provided in Appendix.

4.2. Experimental Results

Quality of Generated CAMs. Tab. 1 shows the quality of our generated CAMs. Our framework outperforms all previous methods by a large margin on initial seeds. CRF could further boosts the performance to 75.0%, which even outperforms previous methods with extra affinity networks. The result is accurate enough, hence the stage of training an affinity network is omitted. We show qualitative results of our framework and another language-guided method CLIMS [46] in Fig. 4. Our framework can produce accurate and complete segmentation masks. The bad cases mainly stem from occlusion and small objects, which are challenging even in a fully supervised setting. In addition, it is a common practice to aggregate the prediction results from multi-scale images during inference in previous works. In Tab. 2, we compare CAM quality generated by single-scale and multi-scale strategies (denoted with *). The multi-scale inference has no improvement with the CAA module and dense CRF postprocessing and thus single-scale inference is adopted in our experiments.

Method	Seed	dCRF	RW
IRN [1]	48.8	54.3	66.3
SC-CAM [5]	50.9	55.3	63.4
SEAM [43]	55.4	56.8	63.6
AdvCAM [24]	55.6	62.1	68.0
CLIMS [46]	56.6	62.4	70.5
RIB [23]	56.5	62.9	70.6
OoD [25]	59.1	65.5	72.1
MCTfomer [48]	61.7	64.5	69.1
Ours	**70.8**	**75.0**	-

Table 1. mIoU of generated CAMs on PASCAL VOC 2012 train set. dCRF denotes using dense CRF [21] to post-process CAMs. RW represents training affinity networks to refine CAMs.

Method	mIoU
Initial	58.6 / 62.4*
Initial + MHSA	68.2 / 67.0*
Initial + CAA	70.8 / 70.5*
Initial + MHSA + dCRF	72.1 / 70.1*
Initial + CAA + dCRF	**75.0** / 74.1*

Table 2. mIoU of initial CAMs, CAA refined CAMs, and vanilla MHSA refined CAMs on PASCAL VOC 2012 train set. * means adopting the multi-scale strategy during inference.

Time and Memory Efficiency. In Tab. 3, we compare our time and memory costs with some related works. Benefiting from the pre-trained CLIP model, our method requires no classification training on specific datasets. The CAA module intrinsic in ViT is integrated into the first stage that generates initial CAMs. Thus, our framework can refine CAMs in real time and requires no additional refinement stage by training affinity networks, *e.g.*, PSA [2] and IRN [1]. The maximum memory occurs during the affinity network training for previous works, which is about 18GB for both PSA and IRN. As a result, our method is more than 10x efficient than other works in terms of time and memory. Meanwhile, inference speed is ensured by adopting the single-scale strategy, which is competitive with the multi-scale strategy in our approach (Tab. 2).

Method	Classification Time		dCRF	Affinity	Total Time	Memory Cost
	Train	Inference				
AdvCAM [24]	-	70.5	0.2	6.5	77.2	18G
CLIMS [46]	2.1	0.3	0.2	6.5	9.1	18G
MCTformer [48]	0.5	2.5	-	3.0	6.0	18G
Ours	-	0.4	0.2	-	**0.6**	**2G**

Table 3. Time and memory cost of different methods to generate pseudo masks on PASCAL VOC train aug set (containing 10582 images in total). The time unit is **hour** and the memory unit is **GB**. Note that the inference and dCRF processes are combined in MCTformer.

Method	Backbone	Seg.	Val	Test
Image-level supervision + Saliency maps.				
OAA+ [18]	R101	V1‡	65.2	66.4
MCIS [41]	R101	V1‡	66.2	66.9
ICD [15]	R101	V1‡	67.8	68.0
NSROM [49]	R101	V2‡	70.4	70.2
DRS [20]	R101	V2‡	71.2	71.4
EPS [27]	R101	V2‡	70.9	70.8
EDAM [45]	R101	V1‡	70.9	70.6
RIB [23]	R101	V2	70.2	70.0
L2G [19]	R101	V2‡	72.1	71.7
RCA [52]	R101	V2‡	72.2	72.8
PPC+EPS [13]	R101	V2	72.6	73.6
Image-level supervision only.				
PSA [2]	WR38	V1	61.7	63.7
IRN [1]	R50	V2	63.5	64.8
ICD [15]	R101	V1‡	64.1	64.3
SEAM [43]	WR38	V1	64.5	65.7
SC-CAM [5]	R101	V2‡	66.1	65.9
BES [6]	R101	V2‡	65.7	66.6
AdvCAM [24]	R101	V2	68.1	68.0
SIPE [8]	R101	V2‡	68.8	69.7
RIB [23]	R101	V2	68.3	68.6
ReCAM [9]	R101	V2	68.5	68.4
AMN [26]	R101	V2‡	70.7	70.6
MCTformer [48]	WR38	V1†	71.9	71.6
Image-level supervision + Language supervision.				
CLIMS [46]	R101	V2	69.3	68.7
CLIMS [46]	R101	V2‡	70.4	70.0
Ours	R101	V2	**71.1**	**71.4**
Ours	R101	V2‡	**73.8**	**73.9**

Table 4. Evaluation results on PASCAL VOC 2012 validation and test sets. The best results are in **bold**. Seg. denotes segmentation network. [†] and [‡] represents adopting VOC and MS COCO pre-trained model, respectively.

Method	Backbone	Seg.	Sup.	Val
EPS [27]	VGG16	V2	I+S	35.7
L2G [19]	R101	V2	I+S	44.2
IRN [1]	R50	V2	I	32.6
IRN [1]	R101	V2	I	41.4
URN [29]	R101	PSPnet	I	40.7
SIPE [8]	R101	V2	I	40.6
RIB [23]	R101	V2	I	43.8
AMN [26]	R101	V2	I	44.7
Ours	R101	V2	I+L	**45.4**

Table 5. Evaluation results on MS COCO 2014 validation set. The best results are shown in **bold**. Seg. denotes segmentation network, and Sup. denotes the weak supervision type.

Method	total	boat	train
w/o softmax	49.4* / 49.4	24.1	43.8
with softmax	53.3* / 58.6	46.9	57.5

Table 6. Ablation study of softmax function on VOC train set. [*] denotes only 20 categories defined in the dataset are used. Results are based on the initial CAMs and not refined by the CAA module.

Segmentation Performance. To further evaluate the quality of pseudo masks, we train the segmentation model based on DeepLabV2 with ResNet-101 following [6, 8, 46]. In Tab. 4, we compare our framework with related methods on PASCAL VOC 2012. Our method outperforms all previous works, even those with saliency maps as auxiliary supervision. Our CLIP-ES achieves 73.8% and 73.9% mIoU on the validation and test set, respectively, which is a new state-of-the-art. The evaluation results on MS COCO 2014 are reported in Tab. 5. Our method also achieves the best performance, with 45.4% mIoU on the validation set.

4.3. Ablation Study

Effect of Softmax Function. We introduce the softmax function into GradCAM to make categories mutually exclusive. First, 20 classes defined in VOC with and without softmax are compared. Results in Tab. 6 (denoted with *) show that softmax-based GradCAM can boost the performance remarkably (from 49.4% to 53.3%). Afterwards, to evaluate the effectiveness of the class-related background set we defined, we report results of “boat” (usually confused with “water”) and “train” (usually confused with “railway”) following [27, 46]. As Tab. 6 shown, $mIoU$ can be improved

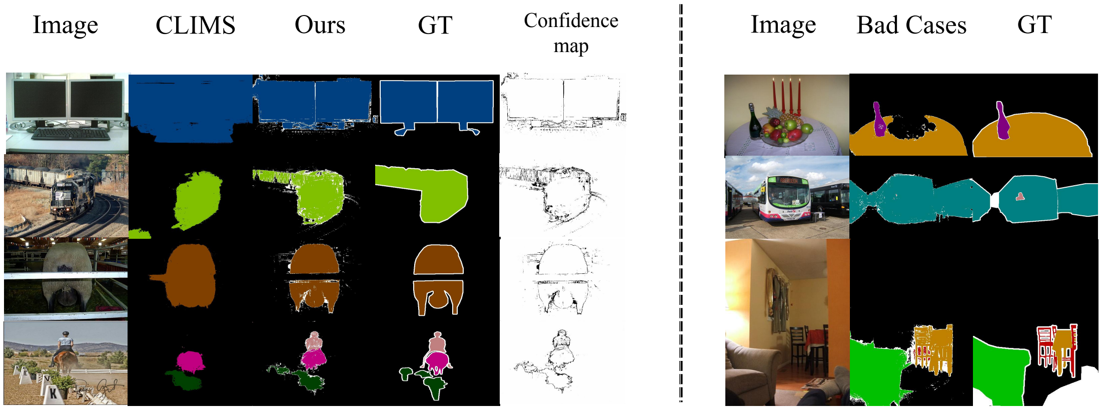

Figure 4. **Left:** Visualization of the pseudo masks generated by our framework and CLIMS. **Right:** Visualization of some bad cases.

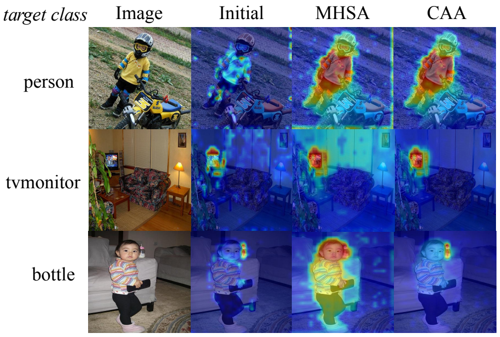

Figure 5. The initial CAMs generated by our proposed framework and comparison between CAA and MHSA refinement.

by 22.8% and 13.7% for boat and train, respectively. The overall performance improves by 9.2% among all classes. The results above suggest that softmax could solve the categories confusion problem efficiently.

Effect of CAA. In Tab. 2, we provide mIoU of the initial and CAA refined CAMs and compare our CAA module with vanilla MHSA in ViT. Results demonstrate that our CAA module can improve MHSA remarkably by introducing the class-aware mask. Fig. 5 shows the visual comparison of different refinement strategies. Our CAA module could make object activations of the initial CAMs complete and mitigate the effect of falsely activated regions.

Effect of CGL. In Tab. 7, we compare CGL with the original Cross Entropy Loss. Results show that CGL can further boost performance. Note that it requires no additional information and merely fully utilizes the confidential information in CAMs. Visualization of the confidence map is shown in Fig. 4. We can find that unconfident pixels mainly focus on object boundaries, which is reasonable because boundaries tend to be semantically murky regions.

Effect of Synonym Fusion. In Tab. 8, we compare performance on some classes with/without synonyms. The result can be improved a lot by applying synonyms, especially for category *"person"*, which we use *"person with clothes, people, human"* to replace.

Model	Cross Entropy	CGL
VOC	70.6	71.1
VOC‡	73.3	73.8
COCO	45.1	45.4

Table 7. Ablation study of Confidence-Guided Loss(CGL).‡ denotes using MS COCO pre-trained model.

Category	bird	chair	person	tvmonitor
Original name	62.9	40.7	43.6	37.0
Synonym fusion	63.9	44.1	51.6	40.3

Table 8. Ablation study of synonym fusion on PASCAL VOC 2012 train set. The results above are based on the initial CAMs and not refined by CAA.

5. Conclusion

This paper explores the potential of CLIP to localize different categories with image-level labels and proposes a simple yet effective framework, CLIP-ES, for WSSS. We present several improvement strategies for each stage to obtain high-quality CAMs and reduce the training cost. The novel framework is text-driven and can efficiently generate pseudo masks for semantic segmentation without further training. Our framework achieves state-of-the-art performance on PASCAL VOC 2012 and COCO 2014 and is potential to generate segmentation masks for new classes.

Acknowledgments This work was supported in part by The National Nature Science Foundation of China (Grant Nos: 62273303, U1909203, 61973271, U1909203), in part by S&T Plan of Zhejiang Province (No. 202218).

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2019. 1, 3, 6, 7

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018. 1, 3, 6, 7

[3] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *CVPR*, June 2020. 2

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

[5] 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*, 2020. 2, 6, 7

[6] Liyin Chen, Weiwei Wu, Chenchen Fu, Xiao Han, and Yuntao Zhang. Weakly supervised semantic segmentation with boundary exploration. In *ECCV*, 2020. 3, 7

[7] 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. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[8] Qi Chen, Lingxiao Yang, Jian-Huang Lai, and Xiaohua Xie. Self-supervised image-specific prototype exploration for weakly supervised semantic segmentation. In *CVPR*, June 2022. 2, 7

[9] 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*, 2022. 2, 7

[10] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 1

[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*. Ieee, 2009. 4

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[13] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *CVPR*, 2022. 7

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 4, 6

[15] Junsong Fan, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *CVPR*, 2020. 3, 7

[16] Junsong Fan, Zhaoxiang Zhang, and Tieniu Tan. Cian: Cross-image affinity net for weakly supervised semantic segmentation. In *AAAI*, 2020. 2

[17] Qibin Hou, Peng-Tao Jiang, Yunchao Wei, and Ming-Ming Cheng. Self-erasing network for integral object attention. In *NeurIPS*, 2018. 2

[18] Peng-Tao Jiang, Qibin Hou, Yang Cao, Ming-Ming Cheng, Yunchao Wei, and Hongkai Xiong. Integral object mining via online attention accumulation. In *ICCV*, 2019. 1, 2, 7

[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*, 2022. 3, 7

[20] Beomyoung Kim, Sangeun Han, and Junmo Kim. Discriminative region suppression for weakly-supervised semantic segmentation. In *AAAI*, 2021. 2, 7

[21] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. 6

[22] Hyeok Ryool Kweon, Sung-Hoon Yoon, Hyeonseong Kim, Dae-Soon Park, and Kuk-Jin Yoon. Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In *ICCV*, 2021. 2

[23] Jungbeom Lee, Jooyoung Choi, Ji-Yoon Choi Ji-Hyeok Moon Young-Ilc Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. In *NeurIPS*, 2021. 6, 7

[24] Jungbeom Lee, Eunji Kim, and Sungroh Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In *CVPR*, 2021. 1, 2, 6, 7

[25] 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*, 2022. 2, 6

[26] Minhyun Lee, Dongseob Kim, and Hyunjung Shim. Threshold matters in wsss: Manipulating the activation for the robust and accurate segmentation model against thresholds. In *CVPR*, 2022. 7

[27] 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*, 2021. 3, 6, 7

[28] Xueyi Li, Tianfei Zhou, Jianwu Li, Yi Zhou, and Zhaoxiang Zhang. Group-wise semantic mining for weakly supervised semantic segmentation. In *AAAI*, 2021. 2

[29] Yi Li, Yiqun Duan, Zhanghui Kuang, Yimin Chen, Wayne Zhang, and Xiaomeng Li. Uncertainty estimation via response scaling for pseudo-mask noise mitigation in weakly-supervised semantic segmentation. In *AAAI*, 2022. 3, 7

[30] Yi Li, Zhanghui Kuang, Liyang Liu, Yimin Chen, and Wayne Zhang. Pseudo-mask matters in weakly-supervised semantic segmentation. In *ICCV*, 2021. 3

[31] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 1

[32] 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*, 2014. 6

[33] George Papandreou, Liang-Chieh Chen, Kevin P. Murphy, and Alan Loddon Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, 2015. 1

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 3, 6

[35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[36] Lixiang Ru, Bo Du, and Chen Wu. Learning visual words for weakly-supervised semantic segmentation. In *IJCAI*, 2021. 2

[37] Lixiang Ru, Yibing Zhan, Baosheng Yu, and Bo Du. Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers. In *CVPR*, 2022. 2, 5

[38] 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*, 2017. 3, 4

[39] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Annals of Mathematical Statistics*, 35:876–879, 1964. 5

[40] Robin Strudel, Ricardo Garcia Pinel, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 1

[41] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*, 2020. 1, 2, 7

[42] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *CVPR*, 2017. 1

[43] Yude Wang, Jie Zhang, Meina Kan, S. Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020. 1, 2, 4, 6, 7

[44] 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*, 2017. 2

[45] 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*, 2021. 2, 4, 7

[46] Jinheng Xie, Xianxu Hou, Kai Ye, and Linlin Shen. CLIMS: Cross language image matching for weakly supervised semantic segmentation. In *CVPR*, June 2022. 1, 3, 6, 7

[47] Lian Xu, Wanli Ouyang, Bennamoun, Farid Boussaid, Ferdous Sohel, and Dan Xu. Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In *ICCV*, 2021. 2

[48] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *CVPR*, 2022. 1, 2, 5, 6, 7

[49] Yazhou Yao, Tao Chen, Guosen Xie, Chuanyi Zhang, Fumin Shen, Qi Wu, Zhen min Tang, and Jian Zhang. Non-salient region object mining for weakly supervised semantic segmentation. In *CVPR*, 2021. 2, 7

[50] 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*, 2020. 2

[51] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 2, 3

[52] Tianfei Zhou, Meijie Zhang, Fang Zhao, and Jianwu Li. Regional semantic contrast and aggregation for weakly supervised semantic segmentation. In *CVPR*, 2022. 7