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

- Problem / objective
  - Weakly Supervised Semantic Segmentation
- Contribution / Key idea
  - CLIP-based single-stage pipeline for Weakly supervised semantic segmentation (**WeCLIP**)

- **Motivation**

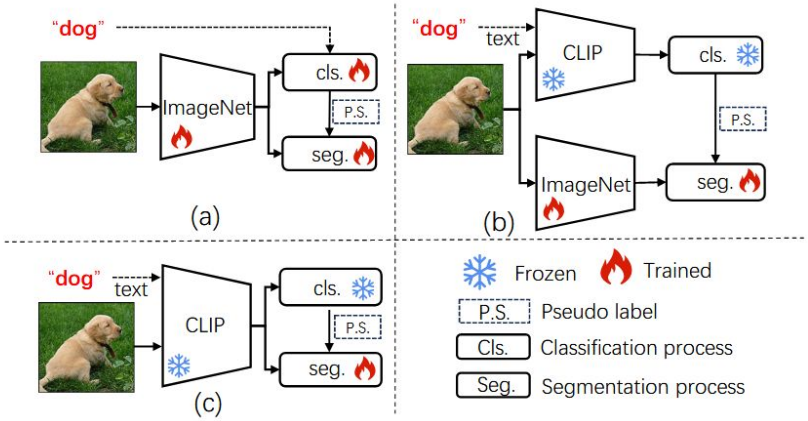

Figure 1. Comparisons between our approach and other single-stage or CLIP-based approaches. **(a) Previous single-stage approach,** which uses a trainable ImageNet [11] pre-trained backbone with trainable classification and segmentation process. **(b) Previous CLIP-based approach,** which is a multi-stage 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.

- **Overview**

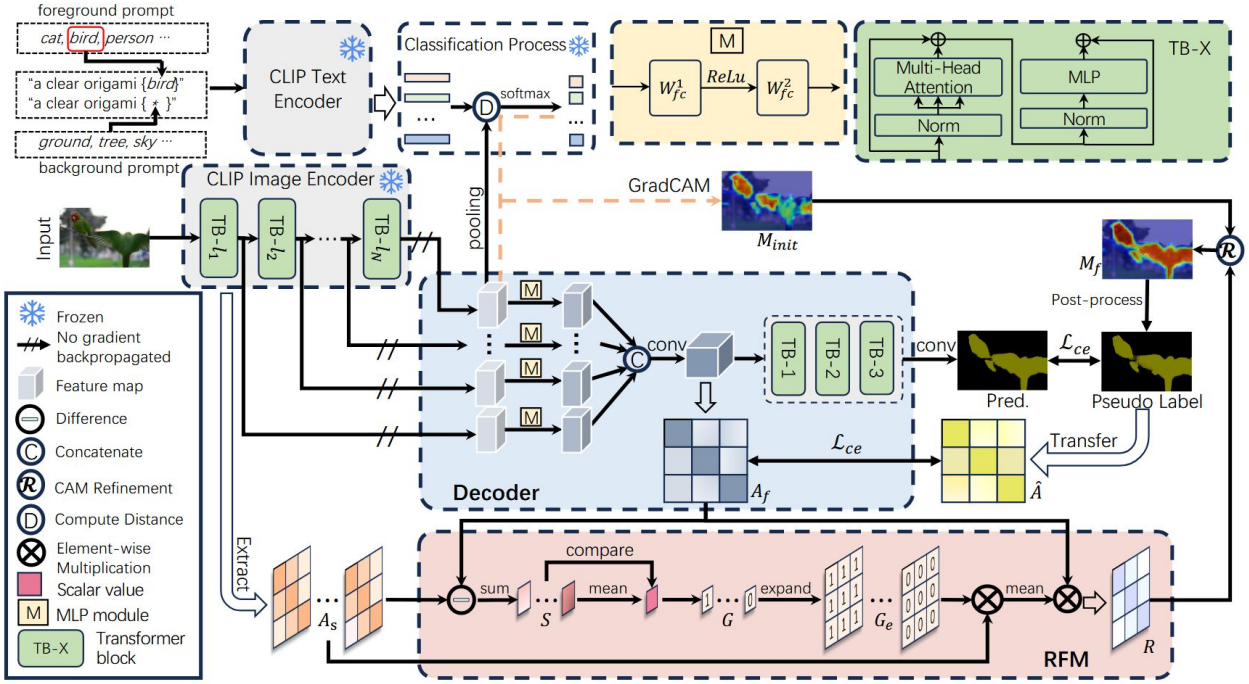

Figure 2. Framework of our WeCLIP. The image is input to the Frozen CLIP image encoder to generate the image features, and class labels are used to build text prompts and then input to the Frozen CLIP text encoder to generate the text features. The classification scores are generated based on the distance 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.

- **Overview**

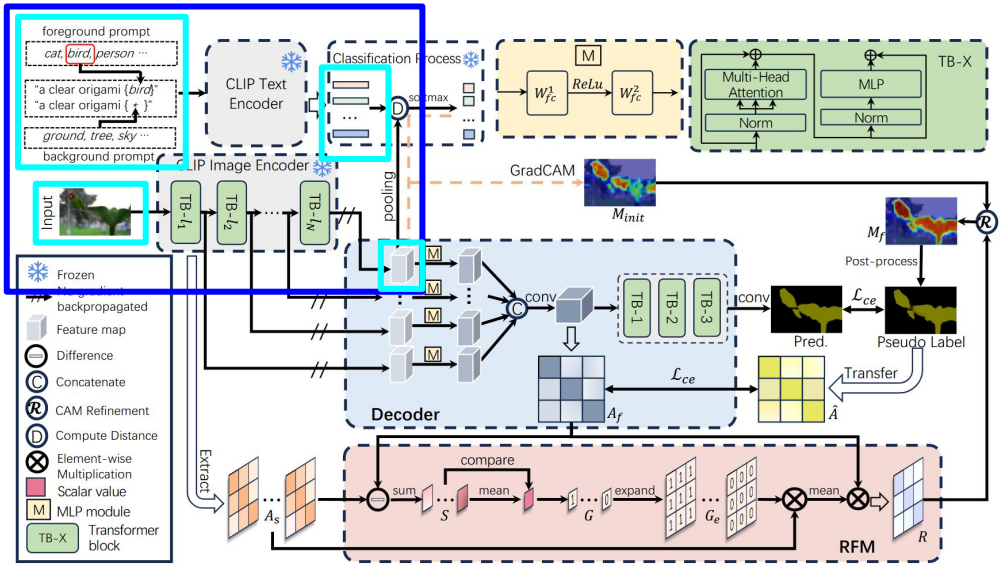

Figure 2. Framework of our WeCLIP. The image is input to the Frozen CLIP image encoder to generate the image features, and class labels are used to build text prompts and then input to the Frozen CLIP text encoder to generate the text features. The classification scores are generated based on the distance 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.

1. Image features, Text features 생성

- Image -> CLIP image encoder -> Image features
- Foreground/Background class labels -> CLIP text encoder -> Text features

- **Overview**

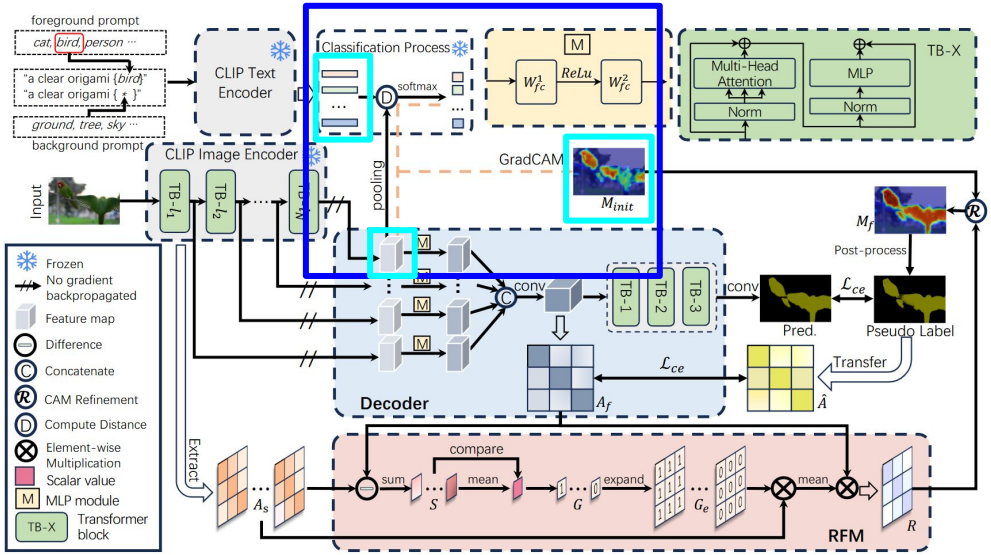

Figure 2. Framework of our WeCLIP. The image is input to the Frozen CLIP image encoder to generate the image features, and class labels are used to build text prompts and then input to the Frozen CLIP text encoder to generate the text features. The classification scores are generated based on the distance 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.

2. Initial CAM 생성

- Pooled image features 와 text features 간 거리 기반으로, classification scores 획득
- Classification scores 기반으로, GradCAM을 통해 initial CAM 생성

- **Overview**

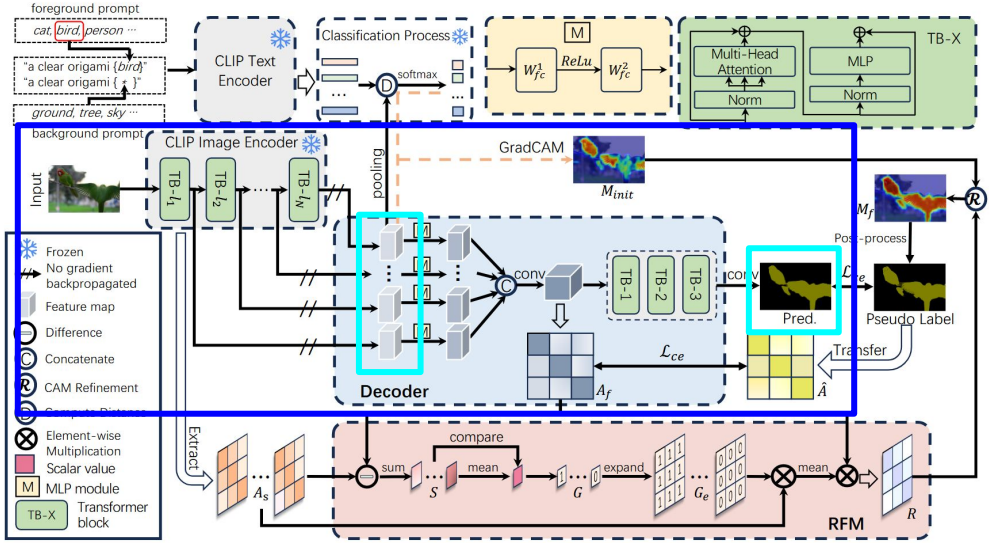

Figure 2. Framework of our WeCLIP. The image is input to the Frozen CLIP image encoder to generate the image features, and class labels are used to build text prompts and then input to the Frozen CLIP text encoder to generate the text features. The classification scores are generated based on the distance 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.

3. Decoder

- Decoder의 인풋: Image features from the last layer of each transformer block in the frozen CLIP image encoder
- Decoder의 아웃풋: Final segmentation predictions
- 동시에, decoder의 intermediate feature maps 통해 affinity map 생성(-> RFM의 인풋)

- **Overview**

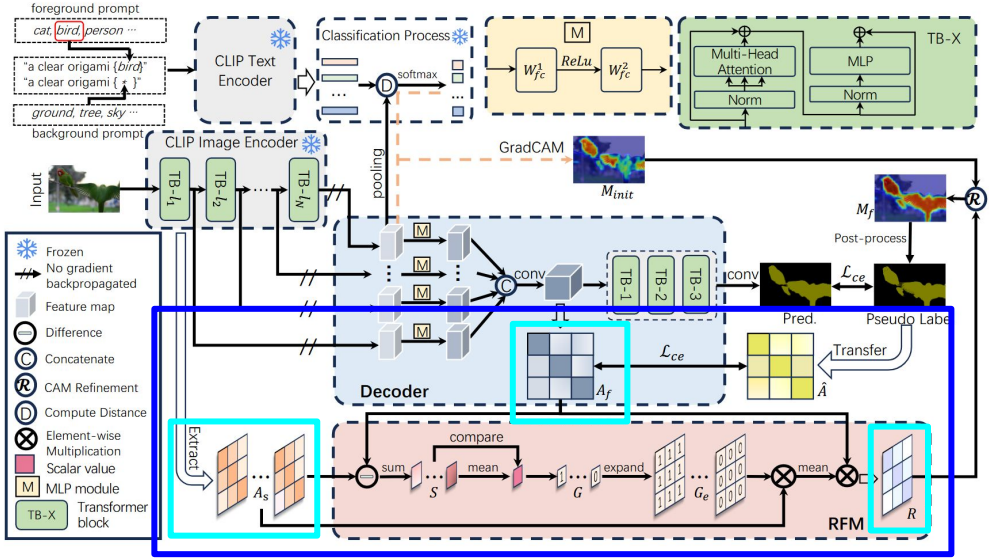

Figure 2. Framework of our WeCLIP. The image is input to the Frozen CLIP image encoder to generate the image features, and class labels are used to build text prompts and then input to the Frozen CLIP text encoder to generate the text features. The classification scores are generated based on the distance 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.

4. RFM

- RFM의 인풋: Affinity map과 Multi-head attention maps from each block of the frozen CLIP image encoder
- RFM의 아웃풋: Refining map

- **Overview**

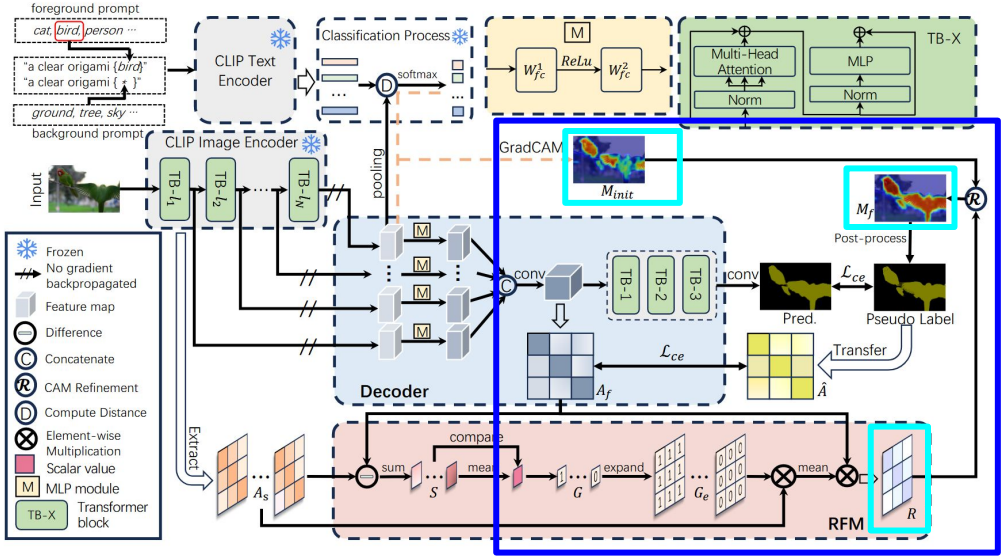

Figure 2. Framework of our WeCLIP. The image is input to the Frozen CLIP image encoder to generate the image features, and class labels are used to build text prompts and then input to the Frozen CLIP text encoder to generate the text features. The classification scores are generated based on the distance 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.

5. Final pseudo-label

- Refing map으로 initial CAM을 refine하여 refined CAM 생성
- Refined CAM에 post-processing을 통해 final converted pseudo label 생성
- Final pseudo-label을 사용하여 학습