From SAM to CAMs: Exploring Segment Anything Model for Weakly Supervised Semantic Segmentation — Supplementary Material

1. Why use point prompts?

As discussed in Section 3 in the main paper, the CPM employs point prompts to transform CAMs into a format suitable for SAM. In this section, we provide a more comprehensive explanation of selecting a point-based approach over various prompt options[1].

Initially, we tried to convert each class's CAM into binary masks through thresholding, utilizing the resulting mask as a prompt. However, as reported in the official repository's issue[2], we also observed poor performance when SAM was employed with mask prompts. Consequently, we decided not to use mask prompts in the CPM.

Additionally, employing bounding box prompts posed challenges. Not only did it exhibit sensitivity to the threshold for determining box sizes covering the CAM, but the ambiguity arising from the unknown number of objects in the image also presented difficulties. Fig. 1 depicts the challenge from the images potentially containing multiple objects of the same class. In instances where the CAM of a particular class displayed multiple local maxima (peaks), two scenarios could unfold: either various parts of a single object were activated, revealing multiple peaks (top row), or multiple distinct objects were present (bottom row).

For the top row scenario, utilizing a single large bounding box covering all peaks (depicted as a red box) is appropriate. However, this approach could lead SAM to predict the mask of the wrong object for the bottom-row scenario. Conversely, opting for multiple boxes (depicted as blue boxes) allowed for accurate predictions in scenarios with multiple objects. However, this failed to cover the entire region of the object in the top row, resulting in the prediction of sub-part masks only.

Therefore, instead of bounding box prompts, we chose to leverage point prompts (represented as green stars), as this approach effectively addresses both scenarios.

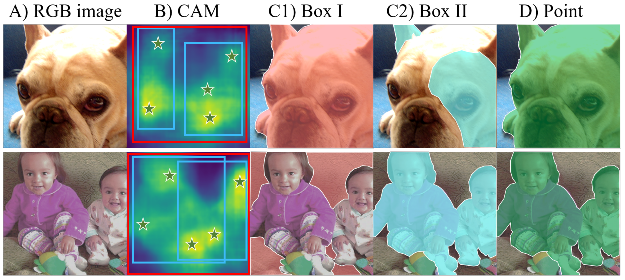

Figure 1. Visualization of why we use point prompts within CPM. **B** showcases three types of prompts that can be converted from CAM: single box covering all the peaks (red), multiple boxes (blue), and points (green). **C1** shows the SAM mask predicted from a single box, while the SAM masks of **C2** are obtained using multiple boxes. Finally, **D** are the masks of using the point prompts, which is our setting in CPM.

2. Detail about LMF

The peaks are defined as local maxima in a square region with a size of 2d+1 pixels [3], where d is set to 10 in our implementation. We drop the peaks at the boundary regions for stable SAM inference. Furthermore, to reject the peaks with insufficiently low CAM scores, which usually are the noise located on the objects of the other class or background, we threshold the peaks with the value of 0.5, which was referred to as τ in the main paper. Here, τ controls the trade-off between precision and recall of the sampled peaks (*i.e.*, point prompts).

Interestingly, the proposed CPM shows robustness against τ, where adjusting its value in the range of 0.4-0.7 almost does not affect the mIoU performance of resulting CAMs (in ±1%). We hypothetically conclude that this robustness mainly comes from neural networks' capability to learn stably even from noisy supervision. This further suggests the superiority of our approach, which directly transfers the knowledge of SAM to the classifier in the training phase, instead of using it in the inference phase like the conventional approaches.

[1] In response to the reviewer's suggestion, we attempted to relocate this section to the main body of the paper. Unfortunately, we could not do so due to page limitations and configuration.

[2] https://github.com/facebookresearch/segment-anything/issues/169

[3] Please refer to *peak_local_max* function of the scikit-image library.

3. Implementation Details

Training Classifier In both the PASCAL and COCO datasets, a batch size of 8 is employed. The specifics related to PASCAL can be found in Section 4.2 of the main paper. In the case of COCO, we set the learning rate to 0.005. The classifier is trained for 400k iters. Similar to the gradual initiation strategy applied in PASCAL, the CPM loss is omitted for the initial 30k iterations for COCO.

Training Semantic Segmentation Model For training the semantic segmentation model with the pseudo-labels, we use the batch size of 8. Learning rate and weight decay are set to 5×10^{-4} and 1×10^{-5}, respectively. The semantic segmentation model is trained for 30 epochs.