# Appendix of Open-Vocabulary Semantic Segmentation with Mask-adapted CLIP

Feng Liang* [1], Bichen Wu[2], Xiaoliang Dai[2], Kunpeng Li[2], Yinan Zhao[2], Hang Zhang† [3],
Peizhao Zhang[2], Peter Vajda[2], Diana Marculescu[1]
[1]The University of Texas at Austin, [2]Meta Reality Labs, [3]Cruise
{jeffliang,dianam}@utexas.edu, {wbc,stzpz,vajdap}@meta.com
https://jeff-liangf.github.io/projects/ovseg

Crop w/ mask Crop w/o mask

Figure 1. Crop without mask will introduce background pixels, making the prediction more difficult.

## A. Crop with or without mask

In the paper, we use the default crop with mask (see left of Figure 1). We also try the direct crop without mask (see right of Figure 1). Following the bottleneck analysis in the introduction, we feed the unmasked crops a pre-trained CLIP for classification. This experiment gives a 13.8% mIoU, which is -6.3% worse than using the masked crops. We hypothesize that the crop with mask introduces many background pixels, making the prediction more difficult. For the example in the right of Figure 1, the "orange" will also be an appropriate category for the unmasked crop.

We note that ZegFormer [2] has also done an ablation study about different strategies to obtain the final crop. We reach a similar conclusion.

## B. Text templates

We use the text templates from ViLD [3]. For each category, we used multiple templates to generate the text embeddings then ensemble these embeddings by a simple average. Text templates are shown as below:

```
'a photo of a {}.',
'This is a photo of a {}',
'There is a {} in the scene',
'There is the {} in the scene',
'a photo of a {} in the scene',
'a photo of a small {}.',
'a photo of a medium {}.',
'a photo of a large {}.',
'This is a photo of a small {}.',
'This is a photo of a medium {}.',
'This is a photo of a large {}.',
'There is a small {} in the scene.',
'There is a medium {} in the scene.',
'There is a large {} in the scene.',
```

*Work done during an internship at Meta Reality Labs.
†Work done while at Meta Reality Labs.

Table 1. The effects of class prediction ensemble. The baseline and our OVSeg model are Swin-Base + Vit-L. We report the mIoU on A-150.

| | MaskFormer only | CLIP only | Ensemble |
|---|---|---|---|
| baseline | **19.6** | 14.3 | 21.8 |
| OVSeg (Ours) | **19.6** | **25.1** | **29.6** |

## C. Class prediction ensemble weight

We set $\lambda = 0.7$ for A-150 and A-847, $\lambda = 0.6$ for PAS-20, PC-59 and PC-459. We further detail the effects of ensemble on A-150 in Table. 1. MaskFormer only or CLIP only denotes the use of the class prediction of MaskFormer or CLIP only. Compared with the baseline, we adapt the CLIP to masked images, leading to a much better CLIP only performance. We also notice ensemble is essential for good performance.

## D. Training hyperparams of R101c model

Our small model is MaskFormer R101c with CLIP ViT-B/16. For MaskFormer training, the backbone weights are initialized from an ImageNet-1K pre-trained model. We use AdamW optimizer with the poly learning rate schedule. The initial learning rate and weight decay are set to $2 \cdot 10^{-4}$ and $10^{-4}$, respectively. We also use a learning rate multiplier

Table 2. Ablation on combining mask prompt tuning (MPT) and fine-tuning (FT). FT ->MPT indicates first FT and then MPT, and vice versa. FT + MPT sim. means optimizing prompts and CLIP simultaneously.

| combination | A-847 | A-150 |
|---|---|---|
| FT ->MPT (default) | **9.0** | **29.6** |
| MPT ->FT | 8.5 (-0.5) | 28.1 (-1.5) |
| FT + MPT sim. | 8.8 (-0.2) | 29.0 (-0.6) |

Table 3. Ablation on prompt depth. We test with and without fully fine-tuned (FT) model.

| prompt depth | A-150 | |
|---|---|---|
| | w/o FT | w/ FT |
| 1 | 25.7 | 29.3 |
| 3 (default) | 26.5 | **29.6** |
| 6 | **26.8** | 29.4 |
| 12 | **26.8** | 29.3 |

0.1 on the backbone. We use a crop size of $512 \times 512$, a batch size of 32 and train the model for 120K iterations. For data augmentations and other hyper-parameters, we follow the setting of [1]. For adapting CLIP ViT-B/16 model, we basically follow the hyperparameters of finetuning ViT-L/16 except we use a larger batch size 1024.

## E. More ablation studies on mask prompt tuning

We explore two other ways to combine mask prompt tuning (MPT) and fine-tuning (FT) as in Table 2. Our default setting (FT ->MPT) is first doing FT and then applying MPT to the already fine-tuned model. We don't change the weights of fine-tuned CLIP. The other option is first doing MPT and then doing FT with fixed mask prompts (MPT ->FT). This combination produces poor results (-1.5% drop on A-150). We conjecture mask prompts learned with original CLIP provide a bad prior when we fune-tune the entire CLIP model. We also explore learning mask prompts and fine-tune CLIP weight *simultaneously* (FT + MPT sim.). This doesn't bring favorable results either.

We further ablate the effects of prompt depth in Table 3. The depth can be selected from $\{1, 3, 6, 12\}$. We use two different scenarios: without fine-tuning (w/o FT) for mask prompt tuning only, with fine-tuning (w/ FT) for applying mask prompt tuning over a already fine-tuned model. For w/o FT case, one layer prompt can bring significant improvement, *e.g.*, from baseline's 21.8% to 25.7%. Deeper prompts result in better performance, because more parameters are introduced with more prompts. Interestingly, deeper prompts (going from 3 to 12) don't bring further improvement for w/ FT case. We choose prompt depth as 3 for default setting.

Table 4. Comparison between different prompt tuning methods.

| Method | baseline | MPT (ours) | VPT |
|---|---|---|---|
| mIoU on A-150 | 21.8 | **26.5** | 25.5 |

Table 5. The source of mask-category training pairs.

| Training pairs | Stuff | Cap. | Stuff + Cap. |
|---|---|---|---|
| mIoU on A-150 | 23.0 | **28.8** | 26.7 |

## F. Compare masked prompt tuning (MPT) to Deep Visual Prompt Tuning (VPT) [4]

We compared our MPT to VPT [4]. With the Swin-Base + ViT-L/14 baseline, we added 50 learnable tokens to the image input tokens. VPT used "deep prompts" with depth 6, resulting in 25.5% mIoU on A-150, which is 1.0% worse than MPT (case (a) in Table 3). This could be due to the use of masked prompts in MPT, which prevent zero masked tokens and mitigate domain distribution shifts in the CLIP model. Additionally, MPT requires no additional computation, while VPT requires 40% more computation to process the extra tokens. We plan to include this ablation study in our final draft.

## G. Combine training pairs from COCO-stuff and COCO-Caption pseudo segments.

We combined GT COCO-stuff annotations (case (1) in Tab.2) with caption pseudo-labeled annotations (case (3) in Tab.2), resulting in 1.4M pairs with 12K nouns in Table 5. However it underperformed compared to using only pseudo-labeled annotations (26.7% mIoU *vs.* 28.8% mIoU on A-150). We believe the class distribution was dominated by the GT COCO-stuff annotations and resulted in overfitting. Future work could explore a more balanced data selection (*e.g.* 10% GT + 90% pseudo-labeled annotations) to potentially improve performance.

## H. Class-wise IoU over seen and unseen categories.

We detail the class IoU on all 150 categories in ADE20K-150 (model trained on COCO) in Figure 2, and we annotated seen *vs.* unseen classes and their IoUs. Seen categories mean there are *similar* categories in COCO-stuff training set. Unseen categories denote the novel categories

in ADE20K. The average IoU of seen and unseen categories are 37.6% and 21.9%, respectively, showing that our model performs better on seen categories. This is also observed in other open vocabulary segmentation work, such as [2].

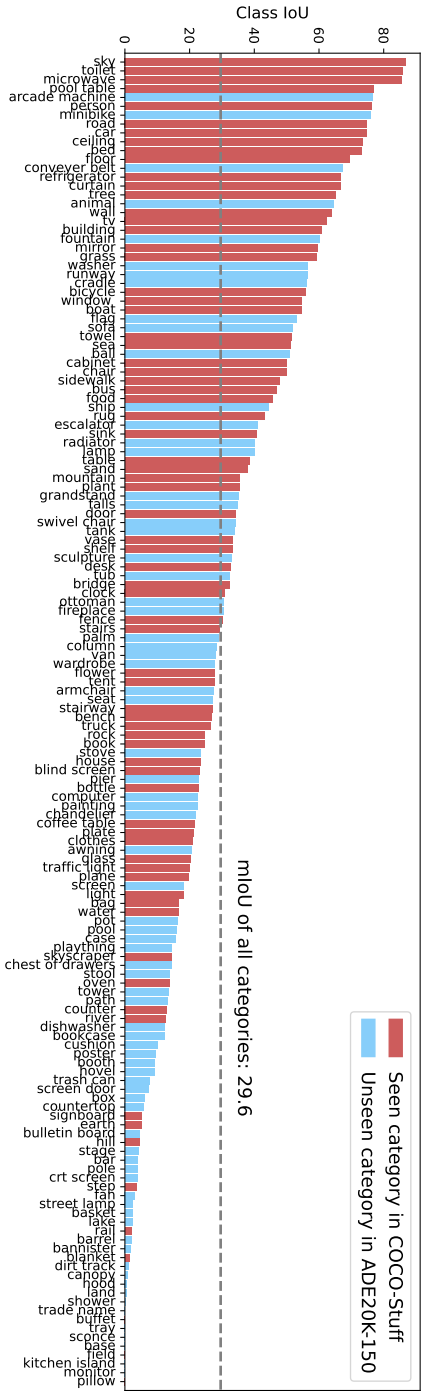


Figure 2. Class IoU on all 150 categories in ADE20K (model trained on COCO). It is expected the model performs better on seen categories in training set.

# I. Inference speed discussions

We followed the two-stage framework of SimBaseline [5] with a focus on accuracy improvement. Our study also evaluated the inference time of MaskFormer and CLIP region classification. For our OVSeg model (Swin-Base + ViT-L), the inference time of MaskFormer and CLIP is roughly 0.2s and 0.6s, respectively, per image on an NVIDIA A5000 GPU. We acknowledge that processing hundreds of regions with CLIP is time-intensive and understand that improving the efficiency of two-stage frameworks is a crucial area of research. It is out of the scope of this work and we plan to address this challenge in future work.

# References


[1] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 2

[2] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11592, 2022. 1, 3

[3] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 1

[4] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 2

[5] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. *arXiv preprint arXiv:2112.14757*, 2021. 3