Open-Vocabulary Semantic Segmentation with Image Embedding Balancing

Supplementary Material

1. Additional Experiments

In this section, we conducted some additional experiments. Note that all models use the CLIP [7] ViT-B/16 model and are evaluated on the ADE20K-150 [10] validation set.

1.1. Embedding Balancing Strategy and Weight

In the main paper, we adaptively balance the three image embeddings $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{D} \in \mathbb{R}^{N\times C}$ by computing their geometric mean. This subsection focuses on studying the effects of different averaging strategies (arithmetic mean and geometric mean) and weights (α, β and γ) on the mIoU.

When we take the geometric mean strategy, since there are negative values in the image embeddings $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{D}$, we cannot directly take their geometric mean. Actually, our geometric mean implementation follows ODISE [8], getting different geometric mean of prediction logits for different classes. We first calculate the prediction logits of all the embeddings with all classes:

$$\mathbf{P}_A = \mathbf{A} \times \mathbf{TE}_{test}^T, \tag{1}$$

$$\mathbf{P}_B = \mathbf{B} \times \mathbf{TE}_{test}^T, \tag{2}$$

$$\mathbf{P}_D = \mathbf{D} \times \mathbf{TE}_{test}^T, \tag{3}$$

where $\mathbf{P}_A$, $\mathbf{P}_B$ and $\mathbf{P}_D \in \mathbb{R}^{N\times K}$. Then, we combine the prediction logits of those three kinds of embeddings with geometric mean for training and new classes:

$$\mathbf{P}_{train} = \mathbf{P}_A^{\alpha} \odot \mathbf{P}_B^{\beta} \odot \mathbf{P}_D^{1-\alpha-\beta}, \tag{4}$$

$$\mathbf{P}_{new} = \mathbf{P}_A^{\gamma} \odot \mathbf{P}_B^{\beta} \odot \mathbf{P}_D^{1-\gamma-\beta}, \tag{5}$$

where $\mathbf{P}_{train}$ and $\mathbf{P}_{new} \in \mathbb{R}^{N\times K}$. After that, we remove the columns (the second dimension) corresponding to the new categories in $\mathbf{P}_{train}$, as well as the columns corresponding to the training classes in $\mathbf{P}_{new}$, getting the same $\mathbf{P}_{train} \in \mathbb{R}^{N\times f}$ and $\mathbf{P}_{new} \in \mathbb{R}^{N\times(K-f)}$ in the main paper.

In Tab. 1, we conduct experiments with different image embedding balancing strategies and weights during inference. At first, we set $\alpha = 0.4$, $\beta = 0.6$, $\gamma = 0$. Then we adjust the the three weights and choose a best set of weights for our model. We adjust the weights in the order of α, γ, β. The experiments show that geometric mean is slightly better for the image embedding balancing, so we use geometric mean in our default setting.

adjusted weight	α	β	γ	mIoU$_g$	mIoU$_a$
α	0.4	0.6	0.0	29.3	29.1
	0.3	0.6	0.0	29.7	29.4
	0.2	0.6	0.0	**29.8**	29.6
	0.1	0.6	0.0	29.2	28.8
γ	0.2	0.6	0.0	**29.8**	29.6
	0.2	0.6	0.1	29.6	29.1
	0.2	0.6	0.2	29.2	28.8
β	0.2	0.6	0.0	29.8	29.6
	0.2	0.5	0.0	29.2	28.7
	0.2	0.4	0.0	28.6	28.2
	0.2	0.7	0.0	**30.0**	29.7
	0.2	0.8	0.0	29.6	29.5

Table 1. Ablation study on the image embedding balancing strategy and weight on ADE20K-150 [10] validation set. mIoU$_g$ denotes the mIoU results using geometric mean for embedding balancing. mIoU$_a$ is the mIoU results with arithmetic mean.

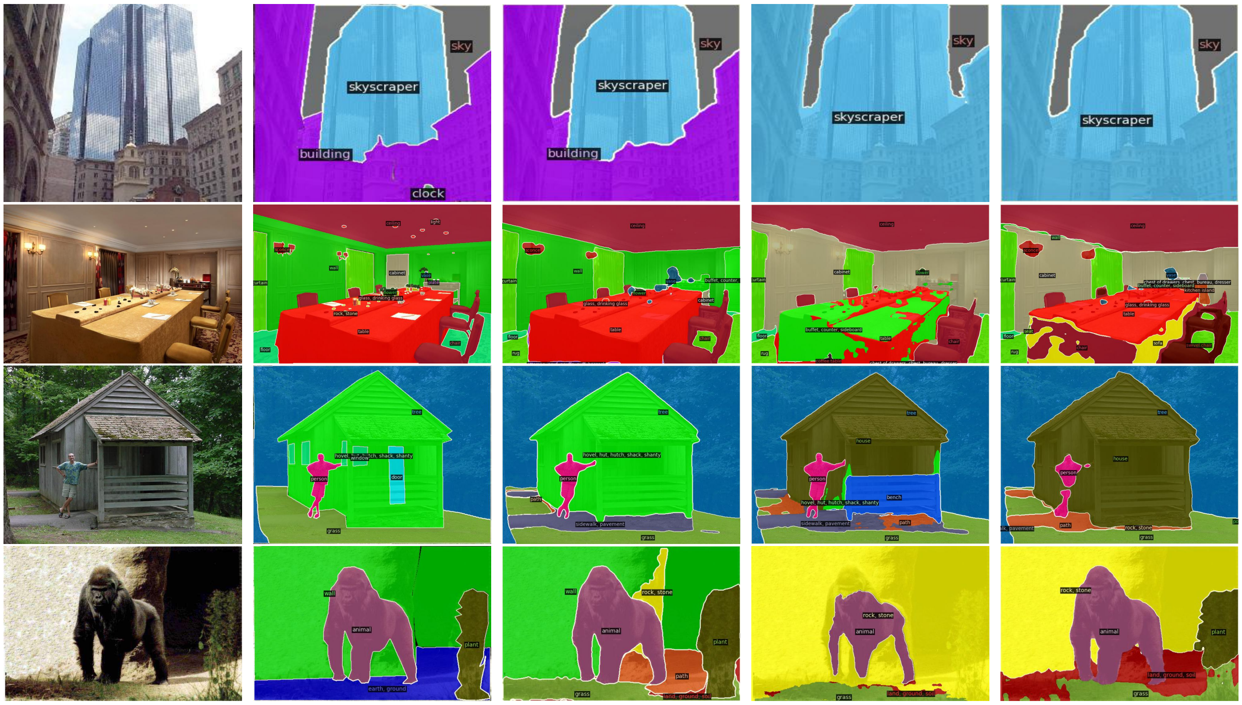
prompt strategy	mIoU
single template	29.1
prompt engineering	**30.0**
prompt tuning	25.3

Table 2. Ablation study on text prompt strategies. *single prompt* denotes using one prompt template: ”a photo of a {}.”. *prompt engineering* means using the 14 prompt templates in ViLD [3]. *prompt tuning* indicates using trainable prompt embeddings introduced in CoOp [11]. We use the CLIP ViT-B/16 based model and evaluate our model on the ADE20K-150 validation set.

2. More Implement Details

2.1. Text Prompt

Following previous works [8, 9], we use the ViLD [3] prompt templates to extract text embeddings. There are 14 prompt templates in ViLD, such as ”a photo of a {}.” and ”There is a {} in the scene”. We use all these 14 templates to extract text embeddings and average the text embeddings of different templates for each class to get the final text embeddings for training and inference. We show the results of different prompt strategies in Tab. 2. When applying prompt tuning, since the trainable prompt embeddings easily overfit to the training classes, the mIoU drops significantly. Prompt engineering with ViLD prompts helps our model the most.



(a) Input (b) GT (c) **EBSeg(ours)** (d) OVSeg [5] (e) SAN [9]

Figure 1. Qualitative results on the ADE20K-150 [10] validation set. We compare our approach with two other methods OVSeg [9] and SAN [5]. Thanks to our AdaB Decoder and SSC Loss, our model shows a stronger generalization ability for new classes that do not exist in the training dataset COCO-Stuff [1], such as *hovel* in the third row and *animal* in the last row. Moreover, with the help of our AdaB Decoder, our model is able to better recognize training classes that exist in the training set, such as *building* in the first row and *table*, *wall* in the second row.

CLIP model	method	params (M)	GFLOPs
ViT-B/16	baseline	23.5	339.9
	EBSeg	**26.6**	**312.3**
ViT-L/14	baseline	24.9	1132.4
	EBSeg	**28.0**	**622.1**

Table 3. Model size of EBSeg. *baseline* denotes that we do not use the additional image backbone and do not downsample the input for CLIP image encoder. The GFLOPs are measured with input images of 640^2 resolution.

2.2. The Semantic Segmentation Loss

During training, following Mask2former [2], we apply binary cross-entropy loss L_{bce} and dice loss L_{dice} to supervise the mask ($\mathbf{M}$) generation process, and apply cross-entropy loss L_{cls} to supervise the mask classification process. Note that we apply L_{cls} to both mask classification results of fully supervised image embeddings $\mathbf{A}$ and mask attention image embeddings $\mathbf{B}$. So our semantic segmentation loss L_{sem_seg} is:

$$L_{sem\_seg} = \lambda_1 L_{bce} + \lambda_2 L_{dice} + \lambda_3 L_{cls}. \tag{6}$$

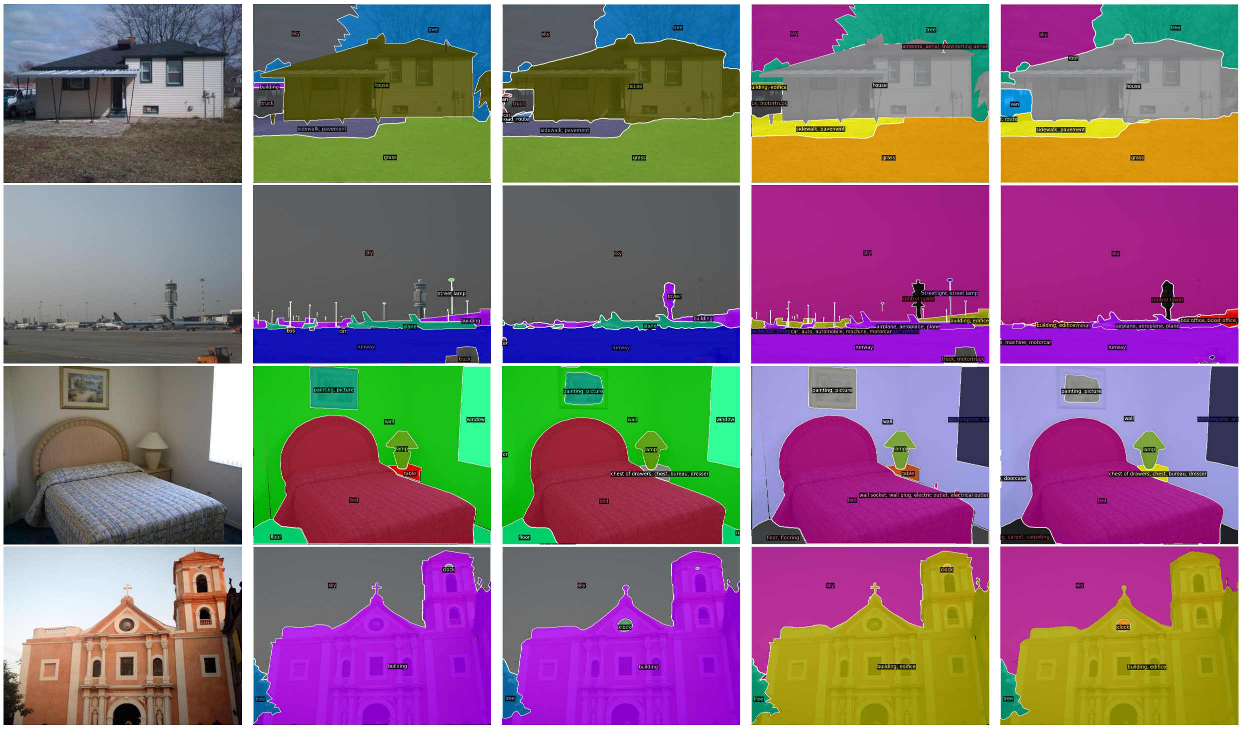
Following the default setting in Mask2former [2], we set $\lambda_1 = 5$, $\lambda_2 = 5$ and $\lambda_3 = 2$.

2.3. Model Size

We list the number of parameters and GFLOPs of our models in Tab. 3. During training, we freeze all parameters of CLIP and SAM [4] image encoders except for their positional embeddings. Since we downsample the input image for CLIP image encoder and use a SAM-B image encoder for all our models (both CLIP ViT-B and ViT-L based models), our models have fewer GFLOPs than a baseline model that do not use an additional image backbone. Please note that during training and inference, we only use the CLIP text encoder once to extract text embeddings in the first iteration. So when measuring GFLOPs, we do not include the CLIP text encoder.

3. More Qualitative Results

In this section, we provide more qualitative results of our model to demonstrate the effectiveness of our proposed approach. We first compare our model with other methods [5, 9] in Fig. 1 on the ADE20K-150 validation set. Then we present more qualitative results of our model on ADE20K-



(a) Input (b) GT (150) (c) prediction (150) (d) GT (847) (e) prediction (847)

Figure 2. Qualitative results on the ADE20K-150/847 [10] validation set. ADE20K-847 has a broader vocabulary than ADE20K-150.

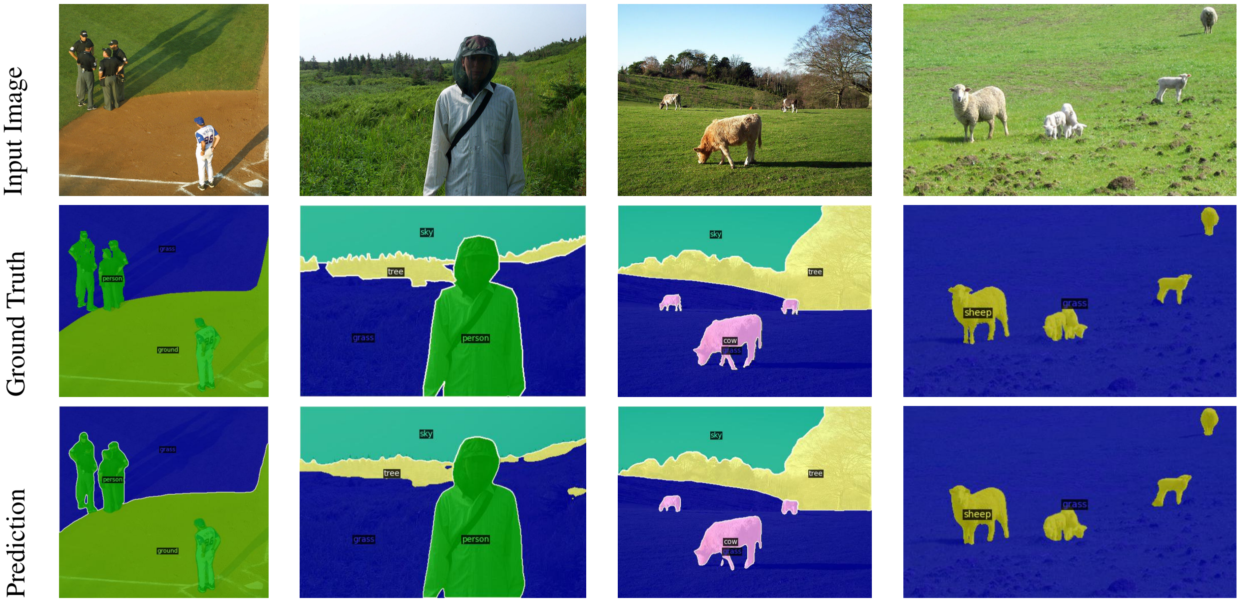


Figure 3. Qualitative results on the PC-59 [6] validation set.

150/847 in Fig. 2, PC-59 [6] in Fig. 3 and PC-459 [6] in Fig. 4. All the visualization results are output by CLIP ViT-B/16 based models.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recog-*

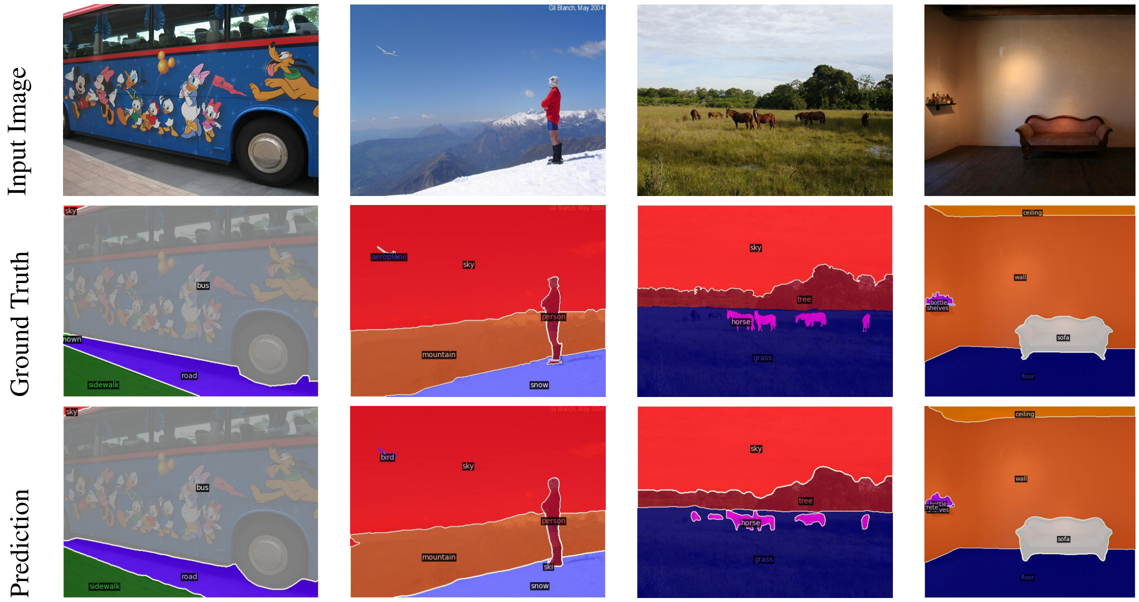


Figure 4. Qualitative results on the PC-459 [6] validation set.

nition, pages 1209–1218, 2018. 2

[2] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 2

[3] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Zero-shot detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2(3):4, 2021. 1

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2

[5] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070, 2023. 2

[6] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 891–898, 2014. 3, 4

[7] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[8] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023. 1

[9] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2945–2954, 2023. 1, 2

[10] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 1, 2, 3

[11] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 1