CLIPer: Hierarchically Improving Spatial Representation of CLIP for Open-Vocabulary Semantic Segmentation

Supplementary Material

6. More implementation details

For the attention maps from CLIP, we first sum all the attention maps from early transformer blocks without averaging across heads, and then normalize the attention maps along the spatial dimension and the probability dimension to sum to 1. We zero out the small value in attention maps before averaging or multiplication fusion from CLIP or Stable Diffusion. After generating the final attention maps for each category, we compute their respective classification scores using max pooling. We then select only the attention maps with high classification scores and apply argmax operation to derive the final results.

7. Additional experiments

Different Stable Diffusion or DINO models. Table 9 shows the performance of using different Stable Diffusion (SD) models and DINO models for fine-grained compensation (FGC), which adopt different training datasets and have different parameters. SD V2.1 consistently delivers superior performance across all three datasets: VOC, Context, and Object. This demonstrates the effectiveness of the advancements introduced in SD V2.1, which containing more detailed spatial representation. Compared to different DINO models, we observe that more pre-training data or larger model can not guarantee better result.

Attention	#Params	Pre-training	VOC	Context	Object
(a) DINO-B/8	85.8M	ImageNet-1M	**68.2**	**35.8**	**41.9**
(b) DINO-B/16	85.8M	ImageNet-1M	66.1	36.7	42.5
(c) DINOv2-B/14	86.6M	LVD-142M	64.6	35.2	42.3
(d) DINOv2-L/14	304.4M	LVD-142M	64.8	35.1	42.4
(e) DINOv2-G/14	1136.5M	LVD-142M	65.0	35.1	42.3
(f) SDv1.4	893.7M	LAION-2B	68.8	37.1	41.3
(g) SDv1.5	893.7M	LAION-2B	69.1	37.2	41.5
(h) SDv2.1	900.1M	LAION-5B	**69.8**	**38.0**	**43.3**

Table 9. Comparison (mIoU) with using proxy attention in DINOs for our FGC. (a) and (g) correspond to Table 8 of paper.

Different time-steps. We further investigate the impact of varying the time-step in Stable Diffusion. Table 10 summarizes the performance on three datasets: VOC [12], Context [25], and Object [3]. Time-step values ranging from 41 to 49 to identify the optimal setting in total of 50 time-steps.

Different input size. We also explore the impact of varying input sizes, specifically the short side of the input image, on model performance. Table 11 summarizes the results across three datasets: VOC, Context, and Object. Input sizes of 224, 336, and 448 are tested to evaluate the performance.

Time-step	VOC	Context	Object
41	69.8	38.0	43.0
43	70.0	38.0	43.1
45	69.8	**38.0**	**43.3**
47	70.3	38.0	43.2
49	**70.4**	38.0	43.2

Table 10. **Effect of time-step in Stable Diffusion**. There is only a slight variation when time-step changes.

Short size	VOC	Context	Object	Avg.
224	68.3	37.7	43.5	49.8
336	69.8	38.0	43.3	**50.4**
448	69.9	37.8	42.4	50.0

Table 11. **Comparison for different input sizes**. The size of 336 achieves best averaged performance on three datasets.

An input size of 336 achieves the best result on the Context dataset (38.0%) and performs competitively on VOC (69.8%) and Object (43.3%), which provides the most balanced performance.

Impact of sliding-window inference. We present more results of our method with or without sliding-window inference strategy in Table 12. We perform experiments using two backbones, including ViT-B/16 and ViT-L/14.

8. Additional visualizations

To further demonstrate the effectiveness of our proposed method, we present additional qualitative results in Fig. 7. These visualizations provide comprehensive segmentation results across diverse datasets and scenarios, highlighting the strengths of our CLIPer model in diverse real-world scene.

Model	Encoder	VOC	Context	Object	VOC20	Context59	Stuff	Cityscapes	ADE	Avg.
CLIPer*	ViT-B/16	60.1	34.8	36.0	84.0	38.5	25.3	36.0	19.8	41.8
CLIPer*‡		62.1	35.7	37.5	85.0	39.6	27.6	37.0	20.6	43.1
CLIPer		65.9	37.6	39.0	85.2	41.7	27.5	38.3	21.4	44.4
CLIPer‡		66.5	38.3	40.0	86.0	42.4	28.6	38.7	22.0	45.3
CLIPer*	ViT-L/14	61.2	34.3	39.6	88.2	39.8	25.8	37.9	22.8	43.6
CLIPer*‡		64.0	35.7	41.4	89.0	40.9	27.1	39.4	22.9	45.1
CLIPer		69.8	38.0	43.3	90.0	43.6	28.7	41.6	24.4	47.3
CLIPer‡		72.2	39.5	44.7	89.8	44.6	30.4	42.5	25.0	48.6

Table 12. **Impact of sliding-window inference under different settings**. * denotes our CLIPer without using FGC module, and ‡ denotes using sliding-window inference.

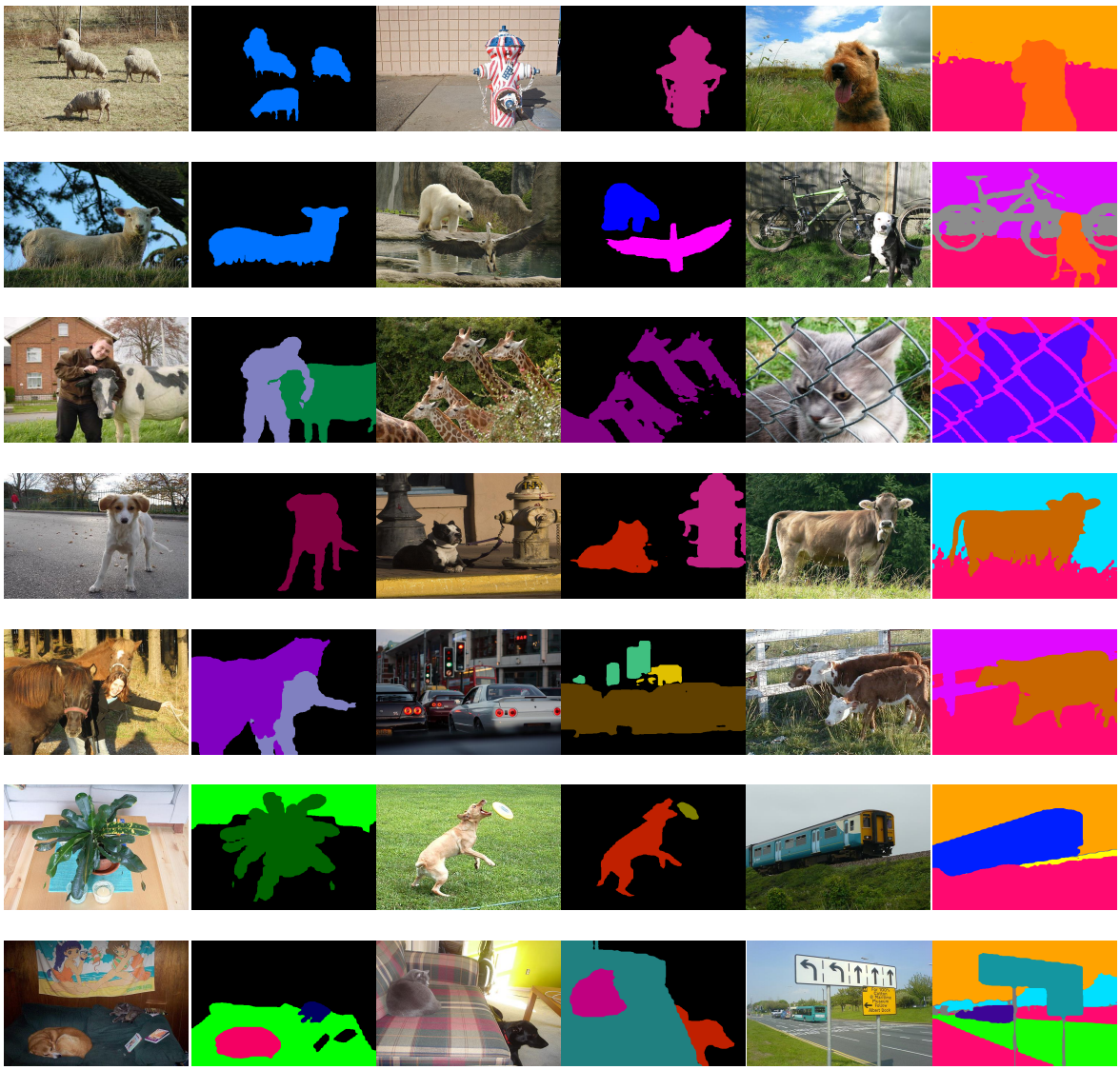

Figure 7. **Additional qualitative results of our proposed CLIPer**. Our CLIPer performs accurate segmentation on these examples.