Appendix

A. Effect of the number of deep prompt tokens

We figure out that the best number of deep prompt tokens varies for different datasets and the detailed results are shown in Fig. 6. For the PASCAL VOC 2012 dataset (VOC) which contains fewer training samples and categories, 10 tokens are enough to obtain significant performance on both seen and unseen classes. However, for large-scale datasets, more deep prompts, i.e., 100 for COCO-Stuff 164K (COCO) and 35 for PASCAL Context (Context), are beneficial to achieve better segmentation performance. In general, the best number of deep prompt tokens increases with the scale of the dataset and the complexity of the per-pixel classification task increases. Meanwhile, using too many visual prompts may be detrimental to our model instead.

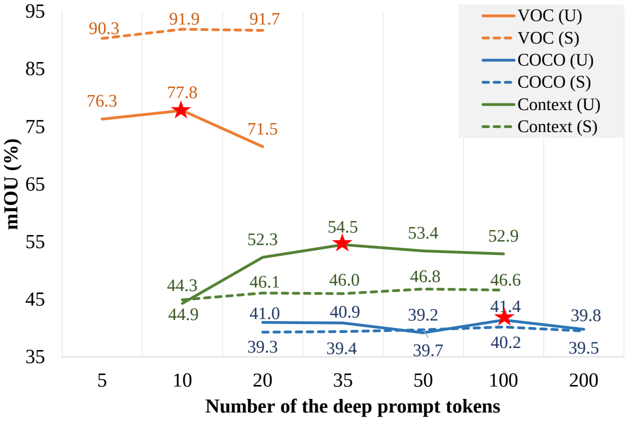


Figure 6. The quantitative results of applying the different numbers of deep prompt tokens. Note that “S” and “U” represent seen and unseen classes separately.

B. Effect of the depth of deep prompt tokens

Except that the number of deep prompt tokens can impact the performance of zero-shot semantic segmentation, we also conduct extensive experiments on PASCAL VOC 2012 (VOC) to explore the effect of inserting the learnable prompts in different layers of the CLIP image encoder. The quantitative results are reported in Tab. 8. For a better explanation, we number the total 12 vision transformer layers in the CLIP image encoder from 1 (“bottom") to 12 (“top") and the layers of inserting prompts are as denoted in the first column of Tab. 8. We figure out that adding prompt tokens on “bottom" layers generally tends to perform better than on “top" layers. Meanwhile, inserting learnable prompt tokens in each ViT layer (layer=1→12) achieves the best performance which is also the default setting in our experiments.

Table 8. Effect of the depth of deep prompt tuning on VOC.

layer	pAcc	mIoU(S)	mIoU(U)	hIoU
1	91.4	87.5	67.8	76.4
1→3	91.7	86.7	70.2	77.6
1→6	92.7	87.8	75.3	81.1
1→9	93.3	88.9	72.4	79.8
1→12	**94.6**	**91.9**	**77.8**	**84.3**
10→12	92.5	88.3	70.9	78.6
7→12	92.5	89.0	68.0	77.1
4→12	93.6	91.5	66.9	77.3

C. Effect of single and multiple text templates

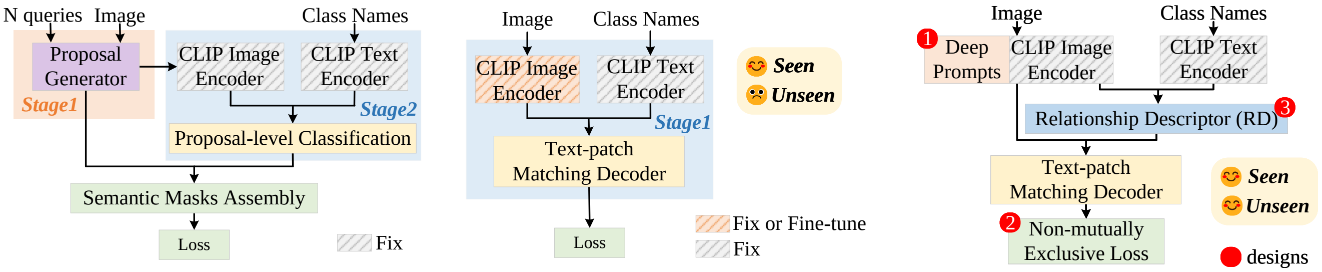
Following the training details of CLIP, we apply a single template “*A photo of a* {}" on PASCAL VOC 2012 (VOC) and multiple templates on large-scale datasets, i.e., COCO-Stuff 164K (COCO) and PASCAL Context (Context), when obtaining the class embeddings from CLIP text encoder. We provide the quantitative results of using single and multiple templates in Tab. 9 where we can see that multiple descriptions achieve reasonable improvements on both two datasets.

Table 9. Comparison of using single and multiple templates on COCO-Stuff 164K and PASCAL Context datasets.

dataset	template	pAcc	mIoU(S)	mIoU(U)	hIoU
COCO	single	61.4	39.5	40.6	40.0
	multiple	**62.0**	**40.2**	**41.4**	**40.8**
Context	single	75.8	45.1	52.1	48.3
	multiple	**76.2**	**46.0**	**54.6**	**49.9**

The details of the 15 augmented templates we used on COCO-Stuff 164K and PASCAL Context datasets are:

‘A photo of a {}.’
‘A photo of a small {}.’
‘A photo of a medium {}.’
‘A photo of a large {}.’
‘This is a photo of a {}.’
‘This is a photo of a small {}.’
‘This is a photo of a medium {}.’
‘This is a photo of a large {}.’
‘A {} in the scene.’
‘A photo of a {} in the scene.’
‘There is a {} in the scene.’
‘There is the {} in the scene.’
‘This is a {} in the scene.’
‘This is the {} in the scene.’
‘This is one {} in the scene.’



(a) Related **Two-Stage** methods. (b) Our **Baseline**: **One-Stage** method. (c) **ZegCLIP:** Baseline with designs.

Figure 7. Brief frameworks of related **two-stage methods**, our **one-stage baseline**, and our proposed **ZegCLIP** model. The happy face in (b)(c) means good performance in seen or unseen classes, while the sad face in (b) represents that the baseline model achieves poor performance in unseen classes.

D. Brief frameworks of related Two-stage and our One-stage method

As we described above, previous zero-shot semantic segmentation methods based on CLIP follow a **two-stage** paradigm as shown in Fig. 7-(a). In stage 1, they need to generate abundant class-agnostic region proposals according to the learnable queries. In stage 2, each cropped proposal region will be encoded via CLIP image encoder to utilize its powerful image-level zero-shot classification capability. The CLIP is still used as a zero-shot image-level classifier.

Instead, we propose a simpler-and-efficient **one-stage** solution as our baseline that directly leverages the feature embedding from CLIP and extends CLIP from image-level classification into pixel-level (or patch-level) as shown in Fig. 7-(b). We introduce a light transformer-based decoder to generate semantic masks by computing the similarities between text-wise and patch-wise embeddings extracted from CLIP. In the baseline, the CLIP text encoder is frozen and the CLIP image encoder can be fixed (Baseline-Fix) or fine-tuned (Baseline-FT).

However, our baseline model still faces the overfitting problem on seen classes. To improve the generalization ability to unseen classes, we propose three important designs on our baseline as shown in 7-(c). After combining three designs, our model ZegCLIP can transfer the zero-shot ability of CLIP from image-level to pixel-level and achieve significant performance on both seen and unseen classes.

In conclusion, in the two-stage methods, N cropped class-agnostic images will be fed into CLIP for image-wise classification which may heavily increase the computational cost. Our proposed one-stage paradigm is simple-but-efficient due to the original image will be encoded only once. The inference speed has been compared in Tab. 3. Our one-stage method ZegCLIP can achieve a speedup of about **5 times faster** than the two-stage method in the inference stage.

E. The details of unseen classes names

For fair comparison, here we provide the detailed unseen class names of PASCAL VOC 2012 (VOC), COCO-Stuff 164K (COCO), and PASCAL Context (Context) dataset in Tab. 10.

Table 10. The details of unseen class names.

Dataset	**The name of unseen classes**
VOC	*pottedplant, sheep, sofa, train, tvmonitor*
COCO	*cow, giraffe, suitcase, frisbee, skateboard carrot, scissors, cardboard, clouds, grass playingfield, river, road, tree, wall concrete*
Context	*cow, motorbike, sofa, cat, boat, fence bird, tv monitor, keyboard, aeroplane*

F. More visualization details

To further demonstrate the effectiveness of our designs on the one-stage baseline (Baseline-FT version), we provide more visualizations including the predicted segmentation results and the semantic masks of different class queries via decoder in Fig. 8. Note that the class names in red are the novel categories.

We can see, after applying our proposed designs, the segmentation performance of (b) ZegCLIP has improved on both seen and unseen classes compared with (a) Baseline-FT. Meanwhile, similar unseen classes can be more clearly classified by our ZegCLIP model as shown in the heat maps. For example, in the “COCO-000000079188” testing image, although Baseline-FT can classify “grass” and “tree” (both are unseen classes) correctly in the semantic masks, our ZegCLIP can distinguish these novel classes discriminatively.

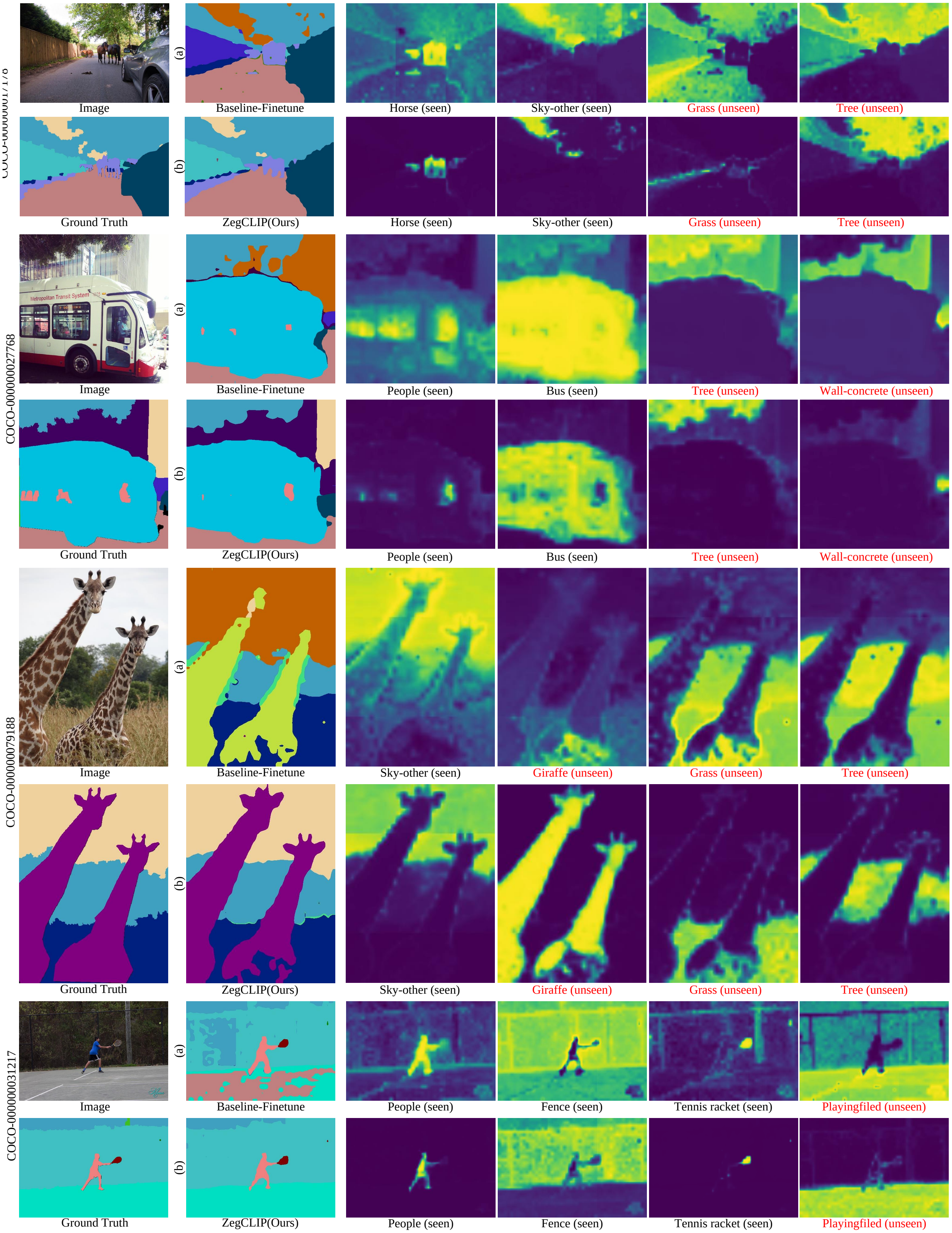


Figure 8. Visualization of semantic masks of different text query embeddings.