Supplementary Material for "Decoupling Zero-Shot Semantic Segmentation"

Jian Ding[1,2], Nan Xue[1], Gui-Song Xia[1]*, Dengxin Dai[2]
[1]CAPTAIN, Wuhan University, China [2]MPI for Informatics, Germany
{jian.ding, xuenan, guisong.xia}@whu.edu.cn, ddai@mpi-inf.mpg.de

A. More Experiments

A.1. Results with the ZS3 setting

Apart from the generalized ZS3 (GZS3), we also report the results achieved with the ZS3 setting, in the evaluation of which the models only predict unseen labels $c \in U$ (see more details in the Sec. 3.1 of our main paper.), and the pixels belong to the seen classes are ignored. The results on COCO-Stuff [1] are reported in Tab. 1. Since most of the existing studies do not report the results with the ZS3 setting, we compared with the SPNet [8] and re-implemented it with FPN and CLIP [7] text embeddings for building a baseline, *i.e.*, SPNet-FPN. The results on ADE20k-Full [9] are reported by Tab. 2.

Table 1. Results on COCO-Stuff with the ZS3 setting. The result of SPNet is directly taken from its original paper [8], and SPNet-FPN is our re-implementation of the SPNet [8] with FPN and CLIP [7] text embeddings, which can be considered as a baseline.

methods	backbone	class embed.	mIoU unseen
SPNet [8]	R-101	ft+w2v	35.2
SPNet-FPN	R-50	clip text	41.3
ZegFormer-seg	R-50	clip text	48.8
ZegFormer	R-50	clip text	**61.5**

A.2. Speed and Accuracy Analyses

Analyses of the Computational Complexity. Given C as the number of channels in a feature map, K as the number of classes, $H \times W$ as the size of feature maps that are used for pixel-wise classification, and N being the number of segments in an image, the complexity of the classification head in the *pixel-level zero-shot classification* is $O(H \times W \times C \times K)$, while the complexity of the classification head of our *decoupling formulation* is $O(N \times C \times K)$. N is usually much smaller than $H \times W$. For instance, in our COCO-Stuff experiments, N is 100, but $H \times W$ is larger than 160×160. Therefore, when K is large, *pixel-level zero-shot classification* will be much slower than the proposed *decoupling formulation of ZS3*.

Table 2. Results on ADE20k-Full achieved with the ZS3 setting. All the models use R-50 as a backbone and CLIP [7] as text embeddings.

methods	mIoU unseen
SPNet-FPN	7.4
ZegFormer-seg	9.4
ZegFormer	**18.7**

Table 3. Results on COCO-Stuff (171 classes) achieved with the GZS3 setting. The FPS is tested on images with the short side of 640 with a single GeForce RTX 3090.

methods	seen	unseen	harmonic	FPS
SPNet-FPN	32.3	11	16.4	17.0
ZegFormer-seg	**37.4**	21.4	27.2	**25.5**
ZegFormer	35.9	**33.1**	**34.4**	6.0

Speed and Accuracy Experiments. We compare the speeds of ZegFormer, ZegFormer-*seg*, and the SPNet-FPN. All these three models use R-50 with FPN as a backbone. ZegFormer-*seg* is an implementation for the *decoupling formulation of ZS3*, while SPNet-FPN is our implementation for *pixel-level zero-shot classification*. ZegFormer is our full model, with a branch to generate image embeddings (see Sec. 3.2 of our main paper for details.) As shown in Tab. 3 and Tab. 4, the ZegFormer-*seg* performs better than SPNet-FPN in both speed and accuracy on COCO-Stuff and ADE20k-Full. The ZegFormer improves the ZegFormer-*seg* by **12 points** in term of mIoU of unseen classes on COCO-Stuff, and still remains an acceptable FPS. We can also see that the speed of SPNet-FPN is slow on ADE20k-Full. This verifies that the speed of *pixel-level zero-shot classification* is largely influenced by K (*number of classes*), as we have discussed before.

*Corresponding author

Table 4. Results on ADE20k-Full (847 classes) achieved with the GZS3 setting. The FPS is tested on images with the short side of 512 with a single GeForce RTX 3090.

methods	seen	unseen	harmonic	FPS
SPNet-FPN	9.2	0.9	1.6	7.9
ZegFormer-seg	18.9	1.3	2.4	**31.3**
ZegFormer	**19.7**	**5.6**	**8.7**	6.3

Table 5. Comparisons with different backbones. In the supervised evaluation, only the pixels of seen classes are evaluated, while the pixels of unseen classes are ignored. The generalized zero-shot evaluation is GZS3, which has been introduced in Sec. 3.1. We can see that SPNet-FPN with *R-101* is comparable with ZegFormer-seg *R-50* in the supervised semantic segmentation, but much lower than ZegFormer-seg with R-50 in the GZS3 evaluation. *S*: seen, *U*: unseen, and *H*: harmonic.

		Supervised	GZS3			
method	backbone	*S*	*S*	*U*	*H*	FPS
SPNet-FPN	R-50	38.5	32.3	11.0	16.4	17.0
SPNet-FPN	R-101	**40.7**	34.6	11.6	17.3	16.2
ZegFormer-seg	R-50	40.3	**37.4**	**21.4**	**27.2**	**25.5**

A.3. Comparisons with Different Backbones.

Since the ZegFormer-seg with R-50 is better than SPNet-FPN with *R-50* in the supervised semantic segmentation, we also report the results of SPNet-FPN with *R-101*. From Tab. 5, we can see that the SPNet-FPN with R-101 is comparable with ZegFormer-seg in the supervised evaluation, but much lower than ZegFormer-seg with the GZS3.

B. More Visualization Results

We visualize the results of ZegFormer-Seg with R-50 and SPNet-FPN with R-50. The two models are *trained* on COCO-Stuff with **156 classes**, and required to segment **847 classes**. The visualization results are shown in Fig. 1, and Fig. 2.

C. More Implementation Details

Our code will be released for the reproduction.

C.1. HyperParameters

Following [3], we use a FPN [6] structure as the pixel decoder of ZegFormer and SPNet-FPN. The output stride of the pixel decoder is 4. Following [2, 3], we use 6 Transformer decoder layers and apply the same loss after each layer. The *mask projection* layer in ZegFormer consists of 2 hidden layers of 256 channels. During training, we crop images from the original images. The sizes of cropped images are 640×640 in COCO-Stuff, and 512×512 in the ADE20k-Full [9] and PASCAL VOC [4]. During testing, we keep the aspect ratio and resize the short size of an image to 640 in COCO-Stuff, and 512 in the ADE20k-Full and PASCAL VOC.

Table 6. Influence of prompt ensemble.

method	prompt ensemble	seen	unseen	harmonic
ZegFormer	✗	35.4	32.7	34.0
ZegFormer	✓	**35.9**	**33.1**	**34.4**

C.2. Prompt Templates

Following the previous works [5, 7], for each category, we used multiple prompt templates to generate the text embeddings then ensemble these text embeddings by averaging. The following is the prompt templates that we used in ZegFormer:

```
'a photo of a {}.',
'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 {}',
'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 photo of 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',
```

To see the influence of prompt templates ensemble, we set a baseline by using only one prompt template, (*i.e.*, "A photo of the {} in the scene.") The comparisons are shown in Tab. 6. We can see that the prompt ensemble will slightly improve the performance.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018. 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 2

[3] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 2021. 2

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 2

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

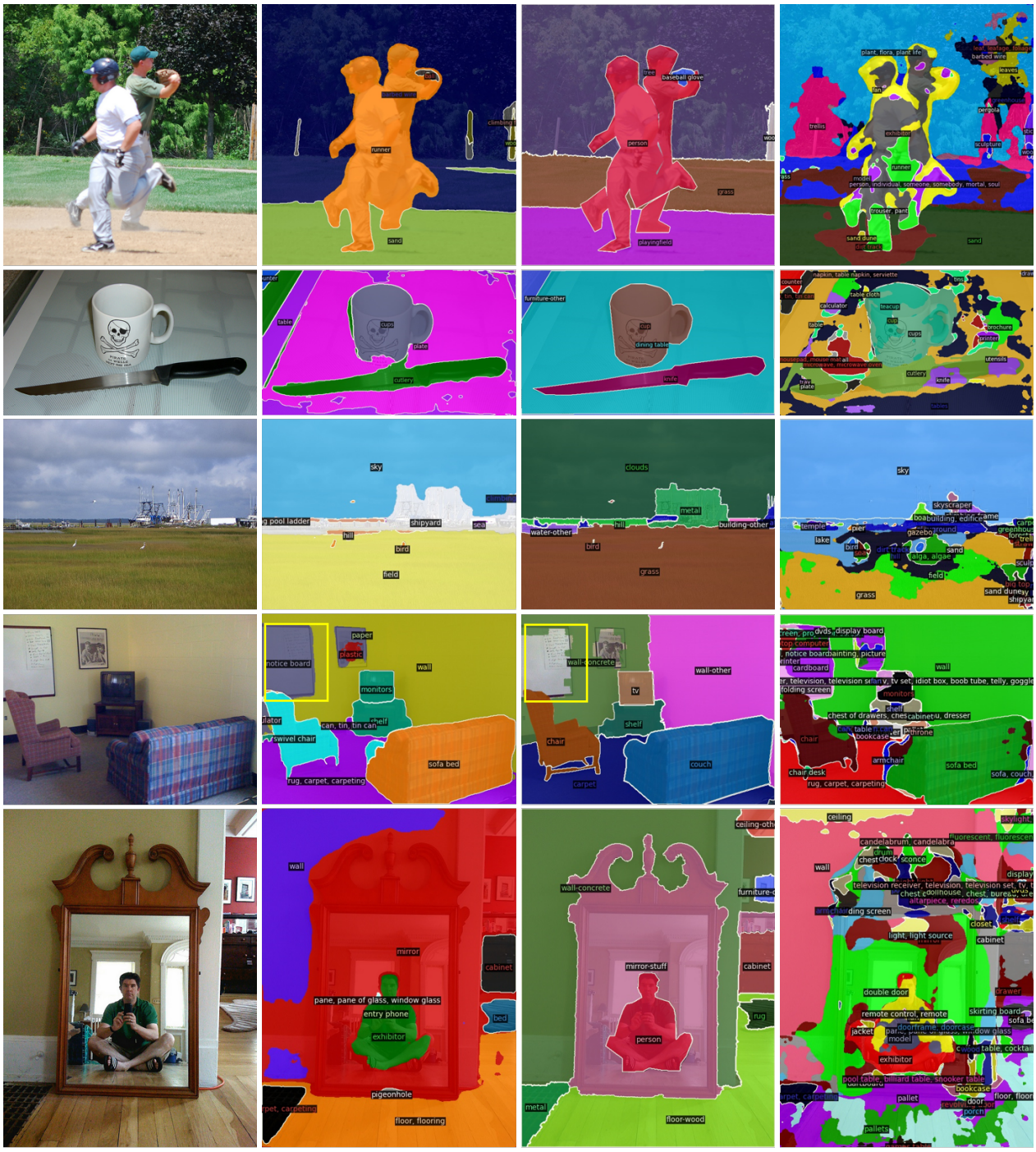

Figure 1. Results on COCO-Stuff. ZegFormer-seg is our proposed model as an implementation of *decoupling formulation of ZS3*, while the SPNet-FPN is a *pixel-level zero-shot classification* baseline. Both the two models are trained with only 156 classes on COCO-Stuff, and required to segment with 847 class names. We can see that the *pixel-level zero-shot classification* is totally failed when there is a large number of unseen classes. In the yellow box are the unannotated category in COCO-Stuff but segmented by our model.

[6] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 2

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual

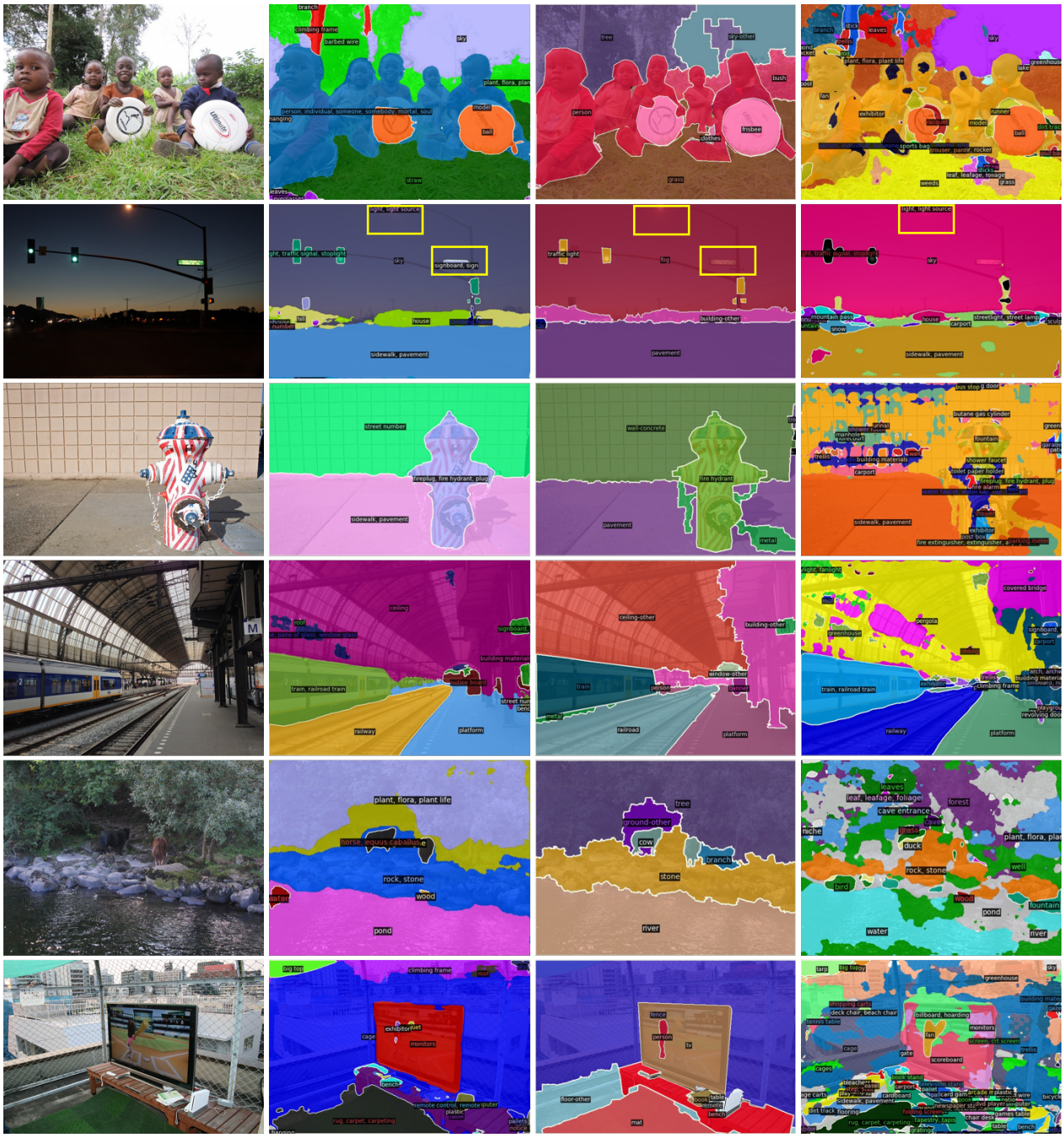

Figure 2. Results on COCO-Stuff. ZegFormer-seg is our proposed model as an implementation of *decoupling formulation of ZS3*, while the SPNet-FPN is a *pixel-level zero-shot classification* baseline. Both models are trained with only 156 classes on COCO-Stuff, and required to segment with 847 class names. In the yellow box are the unannotated category in COCO-Stuff but segmented by our model.

models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 1, 2

[8] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *CVPR*, pages 8256–8265, 2019. 1

[9] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 1, 2