Training-Free Open-Vocabulary Segmentation with Offline Diffusion-Augmented Prototype Generation

Supplementary Material

In this supplementary material, we delve into additional implementation details pertaining to our prototype generation process, offering information to facilitate reproducibility. A comprehensive list of the used textual prompts is presented to clarify the experimental setup. We systematically explore the impact of varying superpixel hyperparameters on the overall performance of our proposed model. We examine the combined influence of entire caption contexts and word embeddings during prototype generation. Our findings highlight the effectiveness of this approach, particularly for categories consisting of multiple words. We also investigate the impact of employing the unimodal backbone for both local and global matching. Our results demonstrate the advantage of leveraging a multimodal feature extractor like CLIP for global matching. To enhance interpretability, we include visual examples showcasing captions, generated images, and their corresponding attributions and binary masks. Additionally, we include qualitative results across all the considered benchmark datasets. We conduct a thorough examination of both successful cases and instances of failure, supplementing our analysis with "into the wild" examples—segmentation results obtained by prompting our model with diverse free-form textual inputs.

A. Additional Implementations Details

Textual Templates. To encode through the CLIP text encoder both the nouns extracted during prototype generation and the input categories utilized at inference time, we employ the following set of templates $\mathcal{T}$, introduced in [7]:

```
itap of a {}.
a bad photo of the {}.
a origami {}.
a photo of the large {}.
a {} in a video game.
art of the {}.
a photo of the small {}.
```

As discussed in [7], these templates provide a powerful means of contextualizing textual input, making them particularly well-suited for our application in the context of prototype generation and inference.

Prototypes generation. The foundation of our prototype generation lies in the utilization of a dataset of images paired with captions. To ensure the reproducibility of our results, we detail the negative prompts employed during the generation of images with Stable Diffusion in Table 6. These negative prompts play a crucial role in guiding the generation process, aiming to produce prototypes that are realistic and high-quality. The prototypes generation is performed offline and requires around 5.2 sec for each COCO caption. During inference, computing a category embedding and performing prototypes retrieval takes around 10.8 ms and 12.9 ms for the Base and Large versions of FreeDA.

3d	*abstract*	*art*
asymmetric	*bad anatomy*	*bad art*
bad proportions	*blurry*	*canvas frame*
cartoon	*cartoonish*	*cgi*
cloned face	*colorless*	*computer graphic*
cropped	*cut off*	*deformed*
dehydrated	*digital*	*digital art*
disfigured	*doll*	*duplicate*
error	*extra arms*	*extra fingers*
extra legs	*extra limbs*	*fused fingers*
fuzzy	*grainy*	*graphic*
gross proportions	*inaccurate*	*jpeg artifacts*
long neck	*low quality*	*low-resolution*
lowres	*malformed limbs*	*misshaped*
missing arms	*missing legs*	*morbid*
mutant	*mutated*	*mutated hands*
mutation	*mutilated*	*octane*
out of focus	*out of frame*	*oversaturated*
photoshop	*poorly drawn face*	*poorly drawn hands*
render	*retro*	*signature*
text	*too many fingers*	*ugly*
unreal	*unreal engine*	*unrealistic*
username	*video game*	*watermark*
weird colors	*worst quality*	

Table 6. Negative prompts employed in Stable Diffusion during prototypes generation.

B. Additional Experiments and Analyses

Effect of Superpixel Parameters. Felzenszwalb *et al*. [5] introduced an efficient superpixel algorithm that employs a graph-based approach. The algorithm initiates by constructing a graph representation of the image, where each pixel serves as a node, and edges connect neighboring pixels. Edge weights are determined based on the RGB color space differences between adjacent pixels. Consequently, connected components, initially established as individual components for each pixel, are progressively merged. The growth of each component is regulated by the scale of observation parameter k. The algorithm also incorporates two additional parameters: the diameter of the Gaussian filter used for pre-processing to enhance image smoothness and counter artifacts (σ), and the enforced minimum size

Dataset	μ	σ	k
Pascal VOC	100	0.7	20
Pascal Context	100	1.0	20
COCO Stuff	100	1.0	100
Cityscapes	50	0.5	20
ADE20K	100	1.0	20

Table 7. Parameters employed for Felzenszwalb's algorithm on each dataset.

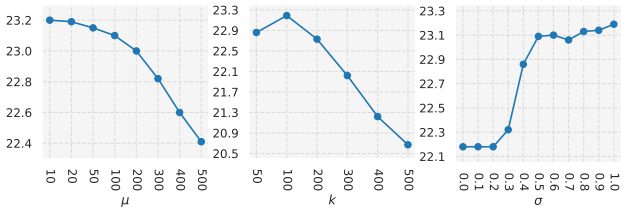

Figure 6. Effect of the variation of superpixel hyperparameters on ADE20K, measured in terms of mIoU.

of superpixels, μ. We employ the implementation of the `skimage`[1] library.

In Table 7, we report the parameter values employed on the examined datasets. Figure 6 further shows the performance variations obtained when altering these parameters on the ADE20K dataset [8, 9]. Notably, minor variations in these parameters have negligible effects on final performance. However, imposing large superpixels through minimum size or scale of observation can significantly degrade the results.

Impact of caption context. In Section 3.1 of the main paper, we outline our methodology for extracting textual key embeddings. Specifically, we employ a linear combination of the word embedding $\hat{t}$ and the caption embedding $\hat{c}$, controlled by a parameter α. In our main results, we set α to 0.9 to effectively incorporate the textual context into the key embedding.

In Table 8, we conduct an ablation study on this choice. The case without caption context corresponds to setting α to 1. It is noteworthy that the inclusion of textual context proves to be particularly beneficial for input categories that consist of more than one word, such as `chest of drawers`. This scenario is prevalent in in-the-wild situations, thus emphasizing the practical utility of our approach in diverse and real-world settings.

Impact of unimodal global matching. In Table 9, we investigate the impact of employing DINOv2 for local and global matching. Since DINOv2 embeddings are not aligned with text, we compute global matching by using the similarity between the `CLS` token of DINOv2 and the representative visual prototypes of the categories. As can be observed, the usage of a text-aligned CLIP backbone improves

[1] https://scikit-image.org/

		mIoU		
	Caption Context	Context	Stuff	ADE
	✗	43.1	27.4	22.2
FreeDA	✓	**43.5**	**28.8**	**23.2**

Table 8. Effect of full caption embeddings on the performance of key embeddings.

Local Backbone	Global Backbone	VOC	Cityscapes	ADE
DINOv2 (ViT-B/14)	DINOv2 (ViT-B/14)	78.4	30.7	17.8
DINOv2 (ViT-L/14)	DINOv2 (ViT-L/14)	74.4	33.5	20.3
DINOv2 (ViT-B/14)	CLIP (ViT-B/16)	85.6	**36.7**	22.4
DINOv2 (ViT-L/14)	CLIP (ViT-L/14)	**87.9**	**36.7**	**23.2**

Table 9. mIoU results with DINOv2 for local/global matching.

performance w.r.t. the unimodal DINOv2 global features.

C. Explainability

A notable advantage of our prototype-based approach lies in its inherent explainability, as the set of referring images used to generate prototypes can be visualized a posteriori. In our approach, in particular, we can visualize the generated images associated with the retrieved prototypes for a given input category, along with the corresponding attribution maps and binary masks.

Figure 9 illustrates the explainability capabilities of our solution, showcasing examples of retrieved prototypes for a specified category, highlighted within the captions in which the corresponding noun was mentioned. We further include the corresponding generated images, attribution maps, and binarized masks, providing a comprehensive view of the explainability achieved by our approach.

D. Additional Qualitative Results

Results on benchmark datasets. Figure 10 shwocases additional qualitative results on Pascal VOC [4], Pascal Context [6], COCO Stuff [1], Cityscapes [3], and ADE20K [8, 9]. These qualitative samples offer a comprehensive view of the performance of our approach, and highlight the versatility and effectiveness of our method across a range of scenes and categories, reinforcing its applicability in various real-world scenarios.

In-the-wild results. Additionally, in Figure 7 we report a collection of in-the-wild examples obtained by prompting our model with diverse free-form textual inputs. Specifically, we extract noun chunks from sample captions of the COCO Captions validation set using the `spaCy`[2] NLP library. After removing stop-words, the noun chunks are utilized as input categories for segmenting the corresponding images. These results extend our analysis beyond curated

[2] https://spacy.io/

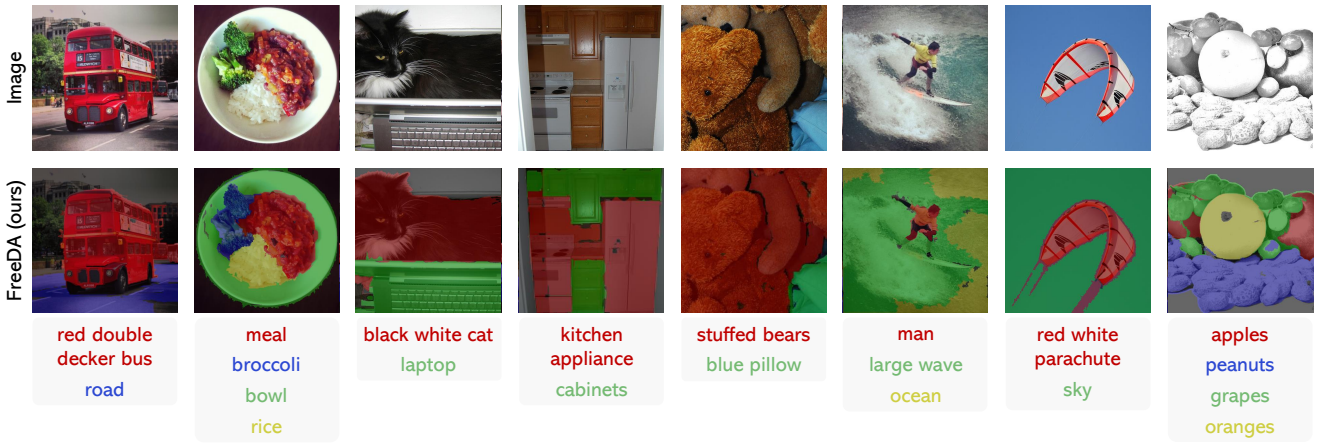

Figure 7. In-the-wild segmentation results obtained by prompting our model with diverse free-form textual inputs.

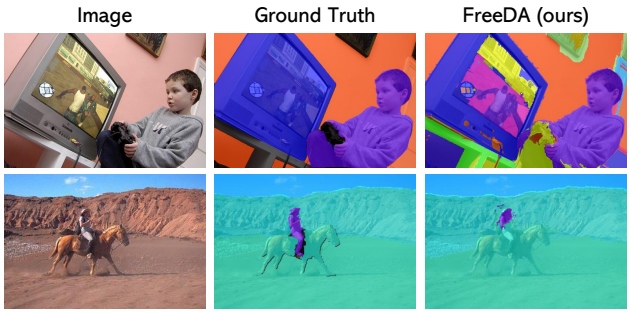

Figure 8. Sample failure cases.

datasets and demonstrate the adaptability and robustness of our approach in handling real-world scenarios with varied and unstructured textual descriptions.

Failure cases. Finally, in Figure 8 we report sample scenarios in which our model encounters challenges and exhibits failure cases. The first row illustrates an image of a TV displaying a video game. Owing to the strong semantic correspondence properties at the token-level of DINOv2, our model tends to segment individual elements shown on the TV screen, thereby impacting the overall segmentation performance for the TV class. The second row of the figure instead presents another failure case featuring an image of a person atop a horse. However, the segmentation is incomplete and only partially captures the person. This limitation can be attributed to the prototypes corresponding to horses ridden by persons, whose noisy binarized masks include their legs. Overall, these failure cases shed light on areas where our model may struggle, emphasizing the need for further refinement and consideration of complex visual contexts.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and Stuff Classes in Context. In *CVPR*, 2018. 2

[2] Junbum Cha, Jonghwan Mun, and Byungseok Roh. Learning To Generate Text-Grounded Mask for Open-World Semantic Segmentation From Only Image-Text Pairs. In *CVPR*, 2023. 5

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *CVPR*, 2016. 2

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, 2012. 2

[5] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59:167–181, 2004. 1

[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 *CVPR*, 2014. 2

[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 *ICML*, 2021. 1

[8] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing Through ADE20K Dataset. In *CVPR*, 2017. 2

[9] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic Understanding of Scenes Through the ADE20K Dataset. *IJCV*, 127(3): 302–321, 2019. 2

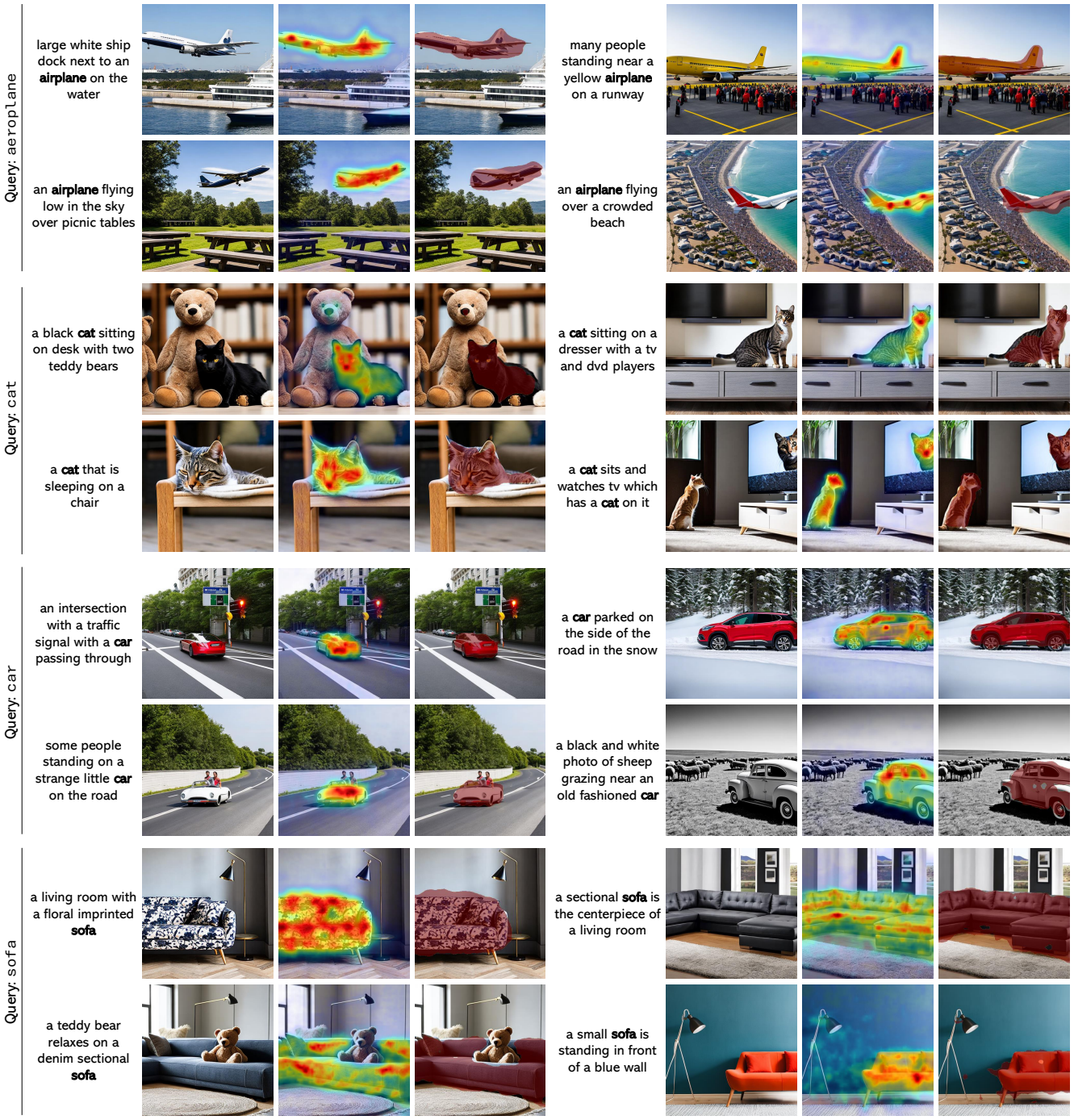

Figure 9. Examples of retrieved prototypes for a specified textual category. From left to right, we show the original COCO caption, the corresponding generated image, the attribution map, and the binarized mask (area highlighted in red).

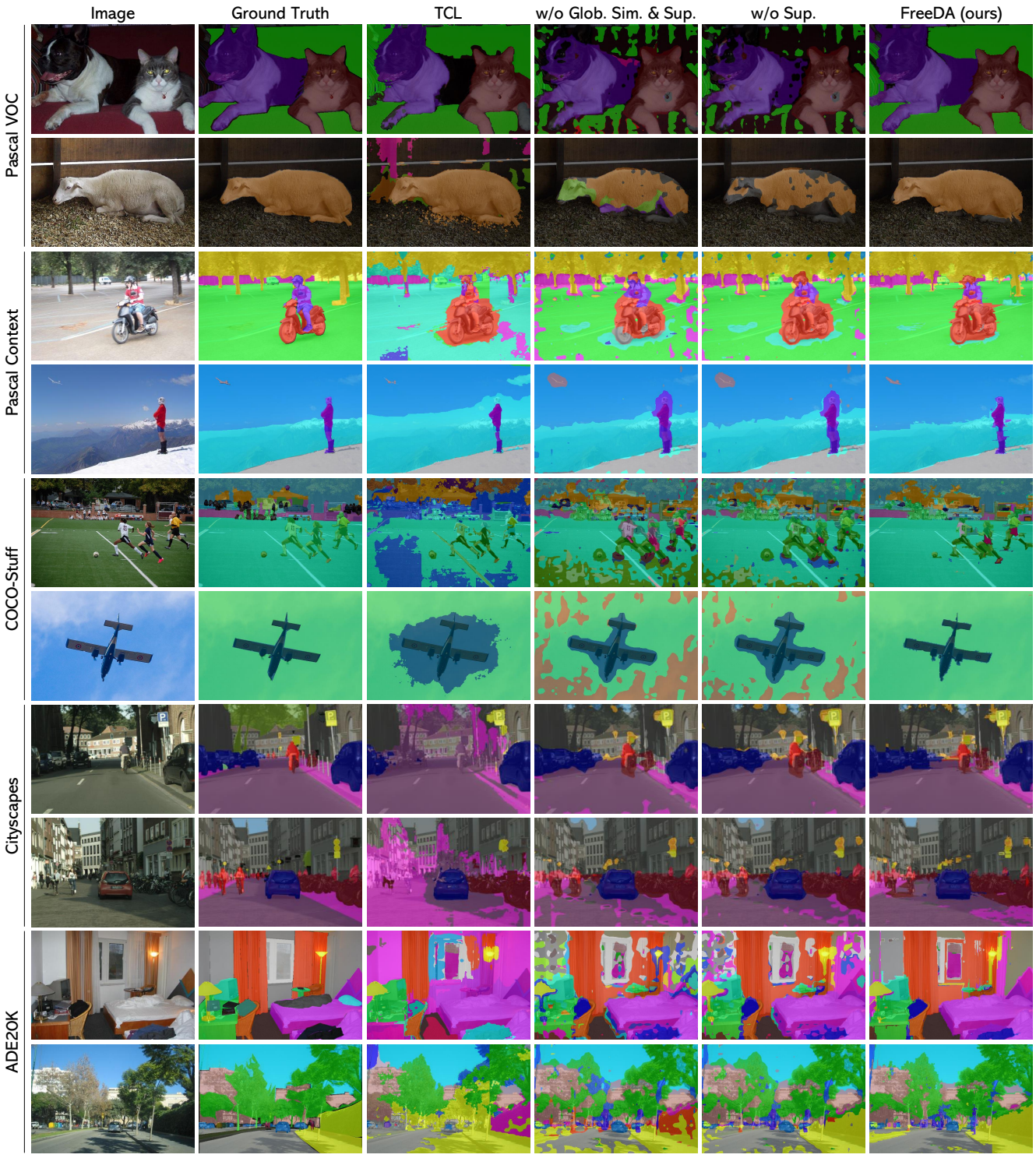

Figure 10. Additional qualitative results of FreeDA in comparison with TCL [2], with and without global similarities and superpixels.