Supplementary: AutoFocusFormer: Image Segmentation off the Grid

Chen Ziwen[1*], Kaushik Patnaik[2], Shuangfei Zhai[2], Alvin Wan[2]
Zhile Ren[2], Alex Schwing[2], Alex Colburn[2], Li Fuxin[1,2]
[1]Oregon State University, [2]Apple Inc.
{chenziw, lif}@oregonstate.edu
{kaushik_patnaik,szhai,alvinwan,zhile_ren,aschwing,alexcolburn,fli26}@apple.com

Anchors	Space-filling curve type	Silhouette Coefficient ↑			
		Stage 1	Stage 2	Stage 3	Stage 4
✓	horizontal scanline	**0.24**	**0.24**	0.22	**0.24**
✗	horizontal scanline	-0.01	-0.20	-0.16	0.03
✓	Peano	0.2	0.23	**0.29**	0.21
✗	Peano	0.15	0.15	0.14	0.17
✓	Hilbert	0.22	0.23	0.22	0.19
✗	Hilbert	0.14	0.15	0.16	0.18

Table 1. Ablation studies on the anchors and the type of space-filling curves used in the balanced clustering algorithm. For the cases without anchors, the space-filling curve is directly applied on the tokens. The metric scores are averaged over a random batch of 256 images from ImageNet.

A. Additional Experimental Results

A.1. Additional Ablation Studies

A.1.1 Ablation on Balanced Clustering Algorithm

We show more results and comparisons regarding our balanced clustering algorithm in Table 1. We study the benefits of space-filling anchors and different types of space-filling curves. We measure the quality of the resulting clusters using the silhouette coefficient [12] metric. The silhouette coefficient ranges from -1 to 1, measuring how clearly distinguishable the clusters are. A larger value indicates better clusters. Specifically, the silhouette score for the i-th token is calculated as

$$\frac{b_i - a_i}{\max(a_i, b_i)}, \tag{1}$$

where a_i is the mean distance between the position of the i-th token and all other tokens in the same cluster, and b_i is the mean distance between the position of the i-th token and all tokens in the next nearest cluster. The final silhouette coefficient is the average score of all the tokens. The numbers in Table 1 are averaged over a random batch of 256 images from ImageNet.

Our default setting is to use space-filling anchors, and apply a simple horizontal scanline as the space-filling curve on the anchors. A horizontal scanline sweeps the rows from left to right in odd rows and from right to left in even rows. We experimented with two other more complicated curves here: the Peano [11] and the Hilbert [6] curves. Both are recursive curves establishing a surjective mapping from a unit interval to a unit square. We also studied direct application of space-filling curves to tokens without us of the anchors. Results show that the anchors are necessary for obtaining more separated clusters. Also surprisingly, the simple horizontal scanline attains better cluster quality than the more complicated space-filling curves when the anchors are used. The visualization of the clustering results are shown in Fig. 1.

A.2. Additional Segmentation Experiments

A.2.1 Additional Results on COCO and ADE20K

COCO instance segmentation. For instance segmentation on COCO (Table 2), we present very significant AP improvement for the Mini size, showing the capability of our model of being more efficient with limited resources. For Tiny and Small, we obtained par results with Swin with 10% decrease in FLOPs. For the 1/5 downsampling-rate models, we see they have significant computational benefits with little performance drop with respect to their 1/4 counterparts. We observe AP improvements for small objects (AP^S) and regressions for large objects (AP^L). We suspect that the standard decoder heads were not aggregating information well when the sampling rate is very uneven for large objects, and we aim to improve the decoder in future work.

Semantic segmentation with HCFormer. HCFormer [13] performs prediction on the feature map at the coarsest level. Each token on the finer level will learn 9 similarity values to the tokens in a 3×3 window in the coraser level, and the model uses the similarity values to interpolate the pre-

*Work done while Chen Ziwen was an intern at Apple Inc.

Backbone	Segmentation Head	Search Space	Epochs	AP	AP^S	AP^M	AP^L	# Params	FLOPs
EdgeViT-XS [10]	Mask R-CNN [5]	-	12	38.3	-	-	-	26.5M	-
PVT v2-B1 [14]	Mask R-CNN [5]	-	12	38.8	-	-	-	33.7M	-
LightViT-T [7]	Mask R-CNN [5]	-	36	38.4	-	-	-	28M	187G
Swin-Mini‡	Mask2Former* [1]	100 queries	50	33.1	13.8	35.2	53.7	25.8M	149G
AFF-Mini	Mask2Former* [1]	100 queries	50	**42.3**	21.2	**45.6**	63.7	25.8M	148G
AFF-Mini-1/5	Mask2Former* [1]	100 queries	50	**42.3**	**21.8**	**45.7**	**64.0**	25.8M	**120G (-19% vs. Swin)**
PVT v2-B3 [14]	Mask R-CNN [5]	-	12	42.5	-	-	-	64.9M	-
LightViT-S [7]	Mask R-CNN [5]	-	36	39.9	-	-	-	38M	204G
SpineNet-96 [2]	Mask R-CNN [5]	1000 proposals	350	41.5	-	-	-	55.2M	315G
Swin-Tiny	Mask2Former [1]	100 queries	50	45.0	24.5	48.3	**67.4**	47M	232G
AFF-Tiny	Mask2Former* [1]	100 queries	50	**45.3**	**24.8**	**49.2**	66.9	46M	204G (-12% vs. Swin)
AFF-Tiny-1/5	Mask2Former* [1]	100 queries	50	44.5	24.5	47.8	66.3	46M	**152G (-34% vs. Swin)**
LightViT-B [7]	Mask R-CNN [5]	-	36	41.2	-	-	-	54M	240G
PVT v2-B5 [14]	Mask R-CNN [5]	-	12	42.5	-	-	-	101.6M	-
SpineNet-190 [2]	Mask R-CNN [5]	1000 proposals	500	46.1	-	-	-	176.2M	2077G
Swin-Small	Mask2Former [1]	100 queries	50	**46.3**	25.3	**50.3**	**68.4**	69M	313G
AFF-Small	Mask2Former* [1]	100 queries	50	**46.4**	**27.0**	49.8	67.6	61.4M	281G (-10% vs. Swin)
AFF-Small-1/5	Mask2Former* [1]	100 queries	50	45.7	26.1	49.2	67.5	61.4M	**206G (-34% vs. Swin)**

Table 2. Instance segmentation on COCO instance val2017. "1/5" means the backbone uses 1/5 downsampling rate instead of the traditional 1/4 downsampling rate. * The segmentation head is modified to accept point cloud input. ‡ This Swin backbone is trained using the same architecture configuration and training settings as our model. The random seed is fixed at 0.

Backbone	Segmentation Head	Crop Size	mIoU	FLOPs
Swin-Small	HCFormer [13]	512	48.8	56G
PVT v2-B5 [14]	Semantic FPN [8]	512	48.7	91.9G
AFF-Small	HCFormer* [13]	512	**49.2**	**51.1G**

Table 3. Semantic segmentation on ADE20K val with HCFormer head. * The segmentation head is modified to accept point cloud input.

Class	Swin-Tiny	AFF-Tiny	Swin-Small	AFF-Small
person	36.3	38.6	36.8	39.2
rider	29.0	30.4	28.7	33.3
car	59.3	60.9	59.9	61.4
truck	41.4	43.1	42.0	41.1
bus	60.4	65.1	65.2	66.9
train	43.7	52.3	51.7	55.4
motorcycle	24.5	26.2	25.2	28.9
bicycle	23.2	24.9	24.7	25.6
average	39.7	42.7	41.8	44.0

Table 4. Class-wise Instance Segmentation AP on CityScapes (backbone Swin vs. AFF) with Mask2Former segmentation head.

diction all the way up to the highest resolution. We replace the square window by 9 nearest neighbors in the coarser level, while the calculation of similarity values stays the same. In Table 3, we show semantic segmentation results on the ADE20K [15] dataset with the HCFormer [13] head for the AFF-Small model. We achieve a +0.4% increase in the mIoU metric with -8% FLOP count.

A.2.2 Class-wise Segmentation Results

To facilitate understanding how AFF improves over the baselines, in addition to score breakdown according to object sizes, we further provide class-wise segmentation score breakdown in Table 4 and Table 5. However, through theses results, we don't see apparent correlation between score improvement and classes. We guess that the improvement from AFF is more correlated with object sizes than specific categories.

B. Segmentation Training Setting Details

We largely follow the settings of Mask2Former [1] in training including weight decay, augmentations and training steps. More specifically, we use the AdamW [9] optimizer with the step learning rate scheduler. We use a weight decay of 0.05. We apply a learning rate multiplier 0.1 to the backbone. We set $\alpha = 4$ for ADE20K and COCO, and $\alpha = 8$ for Cityscapes. We use a learnable shepard power initialized at 6 for ADE20K and Cityscapes, and a fixed power 4 for COCO.

For ADE20K, we train for 80K steps with a batch size of 32 and a base learning rate 0.0002. The FLOP count is calculated on a random 512×512 image, as we crop all images to this size during training.

For COCO, we train for 50 epochs with a batch size of 64 and a base learning rate 0.0002. We apply the large-scale jittering (LSJ) augmentation [3,4] with a random scale sampled from range 0.1 to 2.0 followed by a fixed size crop to 1024×1024 during training. During inference we use the standard Mask R-CNN [5] inference setting where we resize an image with shorter side to 800 and longer side up to 1333. The FLOP count is averaged over 100 validation images for the COCO FLOP count. We scale the learning rate down by 0.1 at 0.9 and 0.95 fractions of the total training steps.

For Cityscapes, we train for 45K steps with a batch size of 32 and a base learning rate 0.0002. During training, we use a crop size of 512×1024. During inference, we use the entire image (1024×2048). We use 100 queries for all models.

For all training tasks, we do not use test-time augmentation

Class	Swin-Tiny	AFF-Tiny	Class	Swin-Tiny	AFF-Tiny	Class	Swin-Tiny	AFF-Tiny
person	50.541	50.210	bicycle	23.690	23.861	car	45.443	46.281
motorcycle	40.718	40.840	airplane	60.491	59.508	bus	70.112	71.876
train	71.873	71.853	truck	42.820	44.763	boat	29.597	30.855
traffic light	30.971	30.314	fire hydrant	70.358	68.929	stop sign	68.312	68.553
parking meter	50.348	50.078	bench	24.617	25.122	bird	33.864	34.639
cat	76.870	77.594	dog	68.567	70.120	horse	47.912	47.360
sheep	53.354	54.451	cow	56.130	55.105	elephant	66.213	65.602
bear	77.146	81.873	zebra	65.191	66.634	giraffe	61.708	61.146
backpack	23.438	23.936	umbrella	54.531	53.804	handbag	22.801	24.080
tie	37.348	36.612	suitcase	50.478	50.670	frisbee	68.770	69.198
skis	7.103	7.668	snowboard	31.047	31.945	sports ball	50.537	50.470
kite	38.308	38.900	baseball bat	38.899	38.439	baseball glove	45.918	48.275
skateboard	37.358	41.363	surfboard	40.484	41.238	tennis racket	61.187	61.463
bottle	42.015	42.877	wine glass	37.538	38.286	cup	47.234	49.267
fork	23.725	24.974	knife	18.923	19.837	spoon	19.997	22.391
bowl	45.014	45.610	banana	27.437	25.364	apple	24.022	25.128
sandwich	47.375	47.423	orange	37.039	37.065	broccoli	24.394	25.494
carrot	24.957	24.435	hot dog	45.251	41.675	pizza	58.760	57.535
donut	55.920	57.046	cake	49.705	48.989	chair	26.461	27.313
couch	48.393	47.571	potted plant	27.185	26.932	bed	45.991	44.751
dining table	22.249	23.094	toilet	68.757	68.878	tv	67.036	66.700
laptop	69.644	70.101	mouse	66.043	61.897	remote	39.732	40.769
keyboard	55.528	56.306	cell phone	41.274	42.047	microwave	65.497	64.513
oven	39.090	38.031	toaster	40.422	39.761	sink	41.816	42.876
refrigerator	66.878	66.622	book	15.363	16.021	clock	56.141	57.378
vase	41.719	42.520	scissors	35.620	37.310	teddy bear	55.229	52.801
hair drier	9.155	14.401	toothbrush	28.801	26.657			

Table 5. Class-wise Instance Segmentation AP on COCO (backbone Swin vs. AFF) with Mask2Former segmentation head.

or multi-scale testing. For all segmentation results, we report the best validation result in one run with seed fixed at 0. Validation results are reported every 2500 steps.

C. Qualitative Comparisons

In Fig. 2, we provide a qualitative comparison of AFF-Small and Swin-Small with Mask2Former segmentation head on the Cityscapes panoptic segmentation data, along with the remaining token locations in stage 2, 3 and 4. Our model is able to retain tokens on very small objects even in the last stage, which provides the foundation to capture crowded, small objects, such as the people sitting in the cafe in the first example in Fig. 2.

In Fig. 3, we provide a qualitative comparison between AFF-Tiny and Swin-Tiny with Mask2Former segmentation head on the ADE20K semantic segmentation data. Our model better captures small objects (e.g., the pole in the first row, the chickens in the third row, and the rug in the fourth row) with fewer false positives in small objects (compared to the Swin baseline in the second row).

References

[1] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1299, 2022. 2

[2] Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V Le, and Xiaodan Song. Spinenet: Learning scale-permuted backbone for recognition and localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11592–11601, 2020. 2

[3] Xianzhi Du, Barret Zoph, Wei-Chih Hung, and Tsung-Yi Lin. Simple training strategies and model scaling for object detection. *arXiv preprint arXiv:2107.00057*, 2021. 2

[4] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2918–2928, 2021. 2

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 2

[6] David Hilbert. Über die stetige abbildung einer linie auf ein flächenstück. In *Dritter Band: Analysis· Grundlagen der Mathematik· Physik Verschiedenes*, pages 1–2. Springer, 1935. 1

[7] Tao Huang, Lang Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Lightvit: Towards light-weight convolution-free vision transformers. *arXiv preprint arXiv:2207.05557*, 2022. 2

[8] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6399–6408, 2019. 2

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

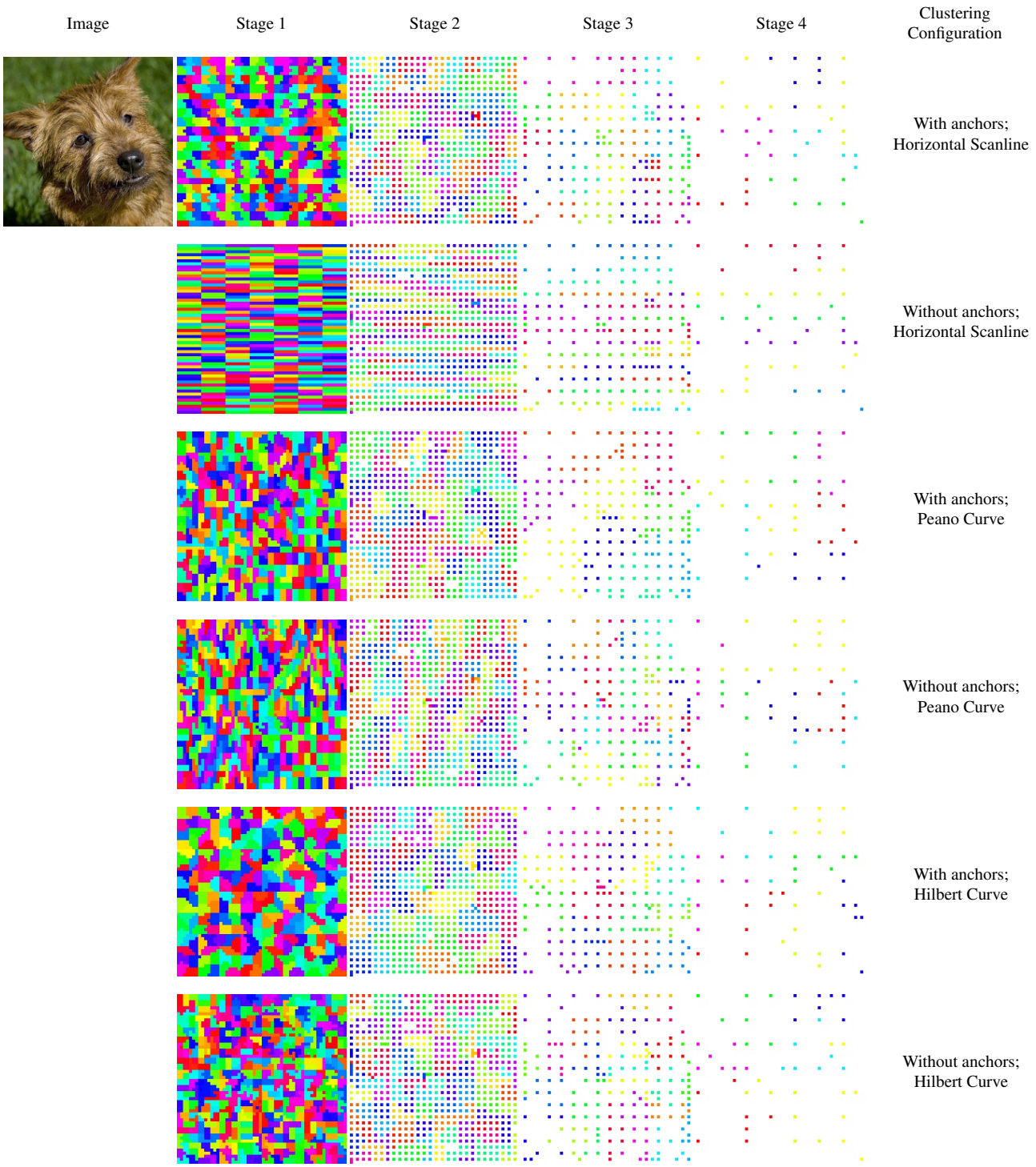


Figure 1. Visualization of the balanced clustering results with different configurations of anchors and space-filling curves. For the cases without anchors, the space-filling curve is applied directly on the tokens. From the results, we observe the use of anchors to be critical for obtaining more rounded and separated clusters. Although Peano and Hilbert are recursive curves, the uneven density of the tokens due to adaptive sampling still breaks the local euclidean metric if we directly apply these curves on the tokens.

[10] Junting Pan, Adrian Bulat, Fuwen Tan, Xiatian Zhu, Lukasz Dudziak, Hongsheng Li, Georgios Tzimiropoulos, and Brais

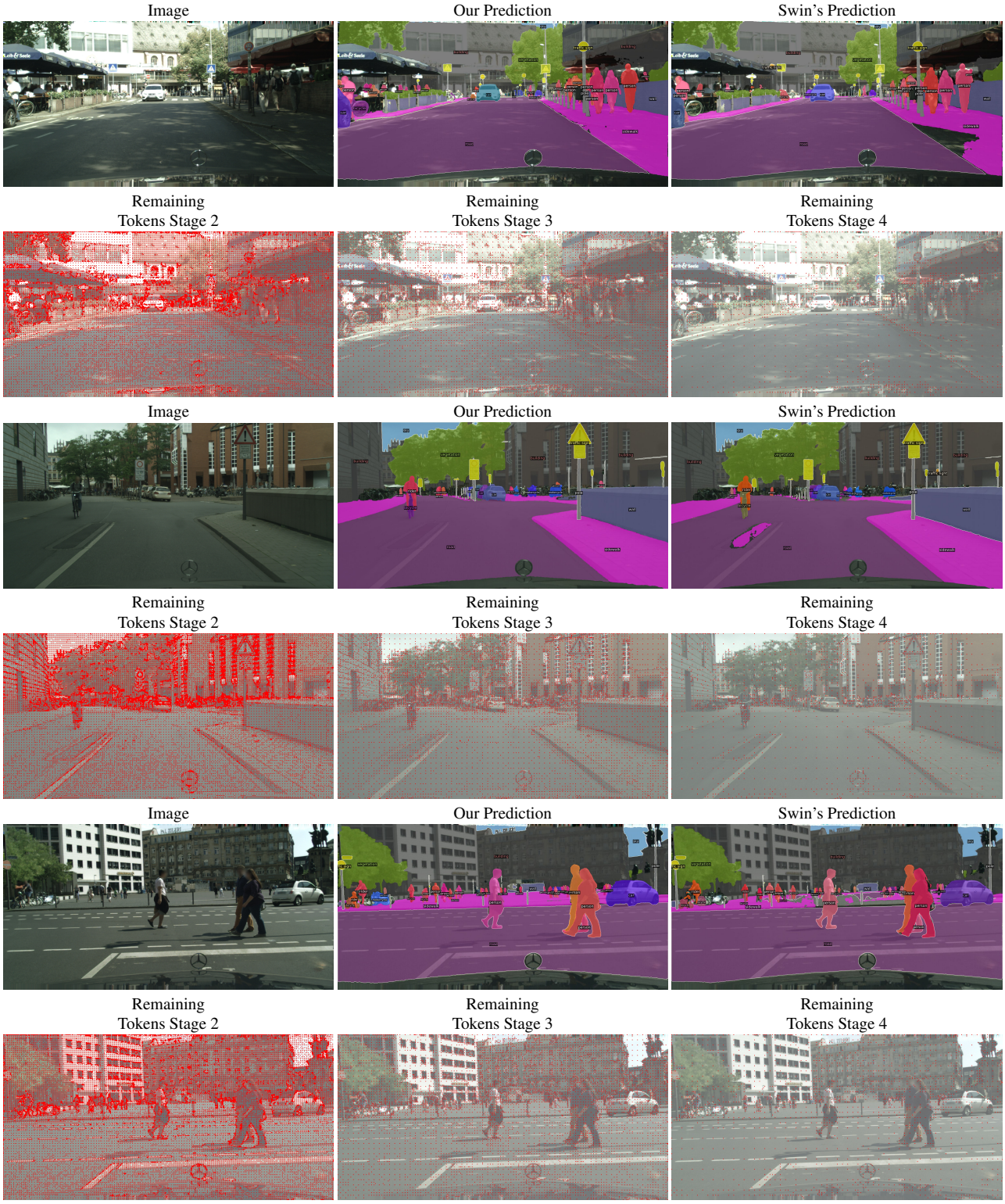


Figure 2. Additional qualitative comparison between AFF-Small and Swin-Small with Mask2Former segmentation head on Cityscapes panoptic segmentation. The red pixels in the even rows indicate the locations of the remaining tokens in stage 2, 3 and 4.

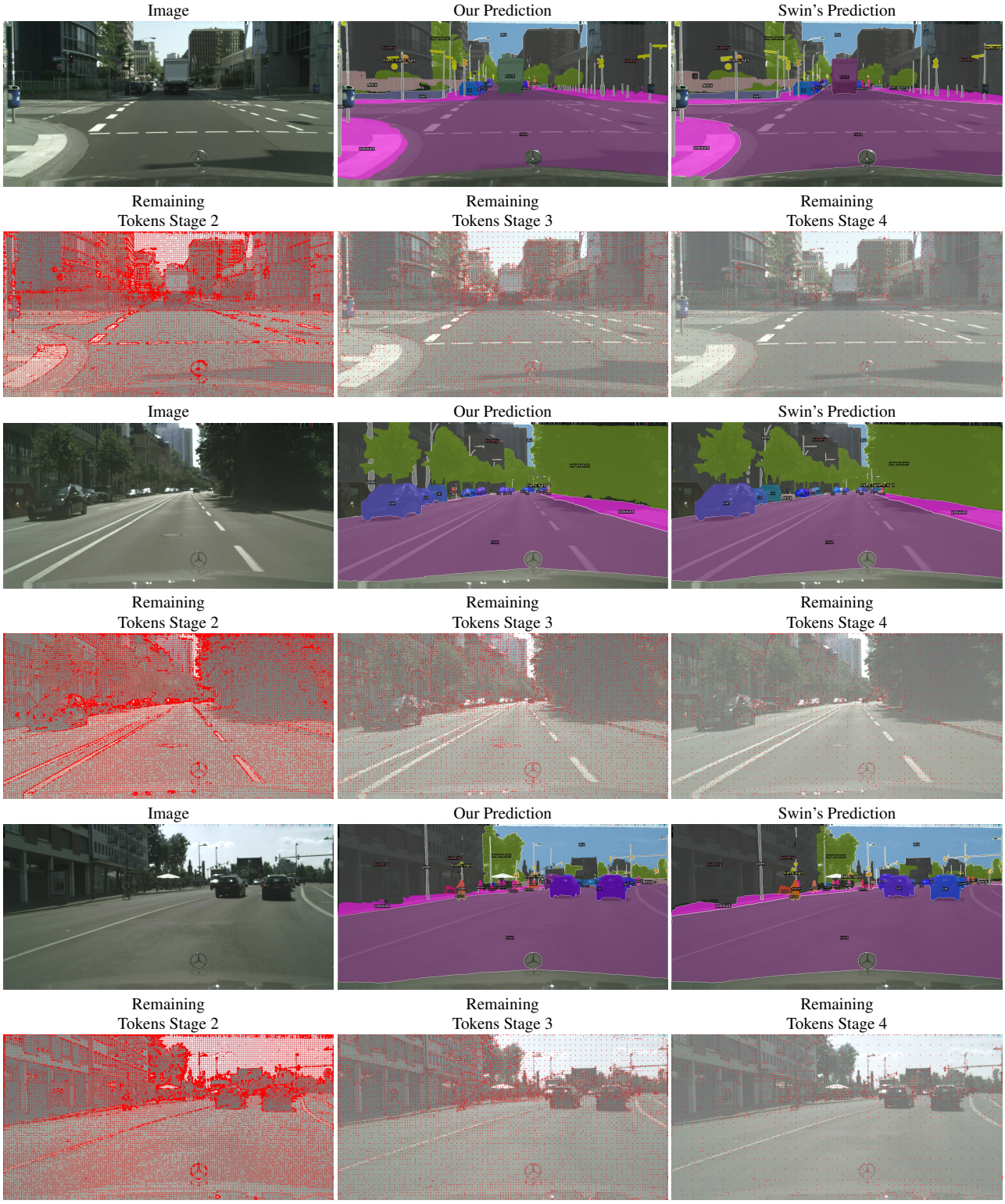
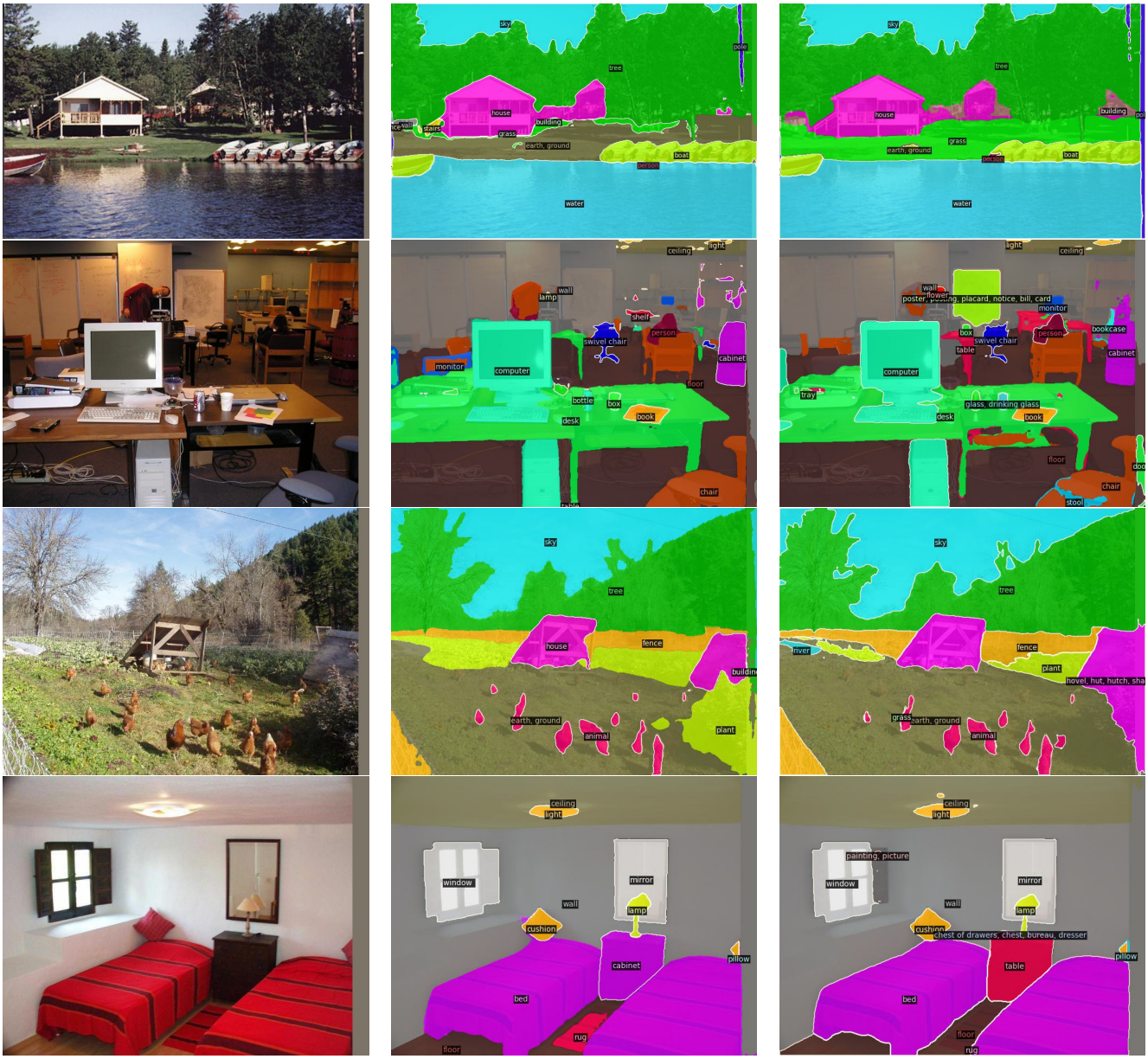


Figure 2. (Continued) Additional qualitative comparison between AFF-Small and Swin-Small with Mask2Former segmentation head on Cityscapes panoptic segmentation. The red pixels in the even rows indicate the locations of the remaining tokens in stage 2, 3 and 4.



(a) Original images. (b) AFF-Tiny predictions. (c) Swin-Tiny prediction.

Figure 3. Qualitative comparison between AFF-Tiny and Swin-Tiny with Mask2Former segmentation head on ADE20K semantic segmentation. First column: original image. Second column: AFF prediction. Third column: Swin prediction.

Martinez. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 294–311. Springer, 2022. 2

[11] Giuseppe Peano. Sur une courbe, qui remplit toute une aire plane. In *Arbeiten zur Analysis und zur mathematischen Logik*, pages 71–75. Springer, 1990. 1

[12] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 1

[13] Teppei Suzuki. Clustering as attention: Unified image segmentation with hierarchical clustering. *arXiv preprint arXiv:2205.09949*, 2022. 1, 2

[14] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 2

[15] 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 (CVPR), pages 633–641, 2017. 2