Method	Backbone	Params	Pascal Context mIoU (%)	ADE20K mIoU (%)	Cityscapes mIoU (%)
Swin (Upernet decoder) [20]	Swin-L	234M	62.0	53.5	81.3
TSG (Ours)	Swin-L	250M	**64.9** (+2.9)	**55.1** (+1.6)	**83.8** (+2.5)
Mask2Former [5]	Swin-L	216M	65.2	57.7	**84.3**
TSG (Ours) + Mask2Former [5]	Swin-L	232M	**66.1** (+0.9)	**58.0** (+0.3)	**84.3** (+0.0)
Other state-of-the-art methods					
SETR [47]	ViT-L	311M	55.8	50.3	82.2
Segmenter [36]	ViT-L	333M	59.0	52.3	80.7
MaskFormer [6]	Swin-L	212M	-	55.6	-
FSFormer [16]	Swin-L	-	58.9	54.3	84.5
SenFormer [2]	Swin-L	314M	64.5	54.2	84.0
PFT [26] + Mask2Former [5]	Swin-L	232M	-	57.4	-
PFT [26] + Mask2Former [5] + MSDA [49]	Swin-L	232M	-	57.8	-

Table A. Multi-scale testing results on Pascal Context, ADE20k and Cityscapes validation.

Method	aeroplane	bag	bed	bedclothes	bench	bicycle	bird	boat	book	bottle	building	bus	cabinet	car	cat	ceiling	chair	cloth	computer	cow
Swin	81.36	8.56	3.95	38.76	0.00	70.24	80.44	70.05	36.38	72.93	56.76	87.03	31.05	81.52	84.64	54.14	43.35	14.93	26.19	77.20
Swin+BD	81.28	8.06	8.72	36.20	4.62	73.45	84.14	71.42	34.35	75.21	58.11	86.43	31.48	83.72	87.18	50.72	41.66	10.87	27.03	82.21
Ours	83.06	16.07	8.49	39.36	10.71	72.45	88.02	73.49	37.99	77.56	60.42	90.11	32.20	85.08	89.43	51.73	43.02	16.02	29.92	83.94
Improvements (Ours v.s. Swin+BD)	1.78	8.01	-0.23	3.16	6.09	-1.00	3.88	2.07	3.64	2.35	2.31	3.68	0.72	1.36	2.25	1.01	1.36	5.15	2.89	1.73
Method	cup	curtain	dog	door	fence	floor	flower	food	grass	ground	horse	keyboard	light	motorbike	mountain	mouse	person	plate	platform	pottedplant
Swin	22.16	42.41	78.43	13.16	34.83	62.68	30.31	23.56	77.72	51.14	80.30	66.54	36.40	77.55	47.20	0.00	80.20	11.85	40.31	59.75
Swin+BD	21.22	43.66	84.00	12.10	33.04	65.74	26.40	26.65	77.63	52.37	83.06	64.73	37.91	80.09	46.67	12.70	82.22	11.83	42.10	66.30
Ours	29.11	45.92	85.11	15.45	34.74	68.34	44.80	27.58	79.38	54.00	85.31	72.11	42.82	80.79	46.77	39.30	83.89	14.80	43.80	65.62
Improvements (Ours v.s. Swin+BD)	7.89	2.26	1.11	3.35	1.70	2.60	18.40	0.93	1.75	1.63	2.25	7.38	4.91	0.70	0.10	26.60	1.67	2.97	1.70	-0.68
Method	road	rock	sheep	shelves	sidewalk	sign	sky	snow	sofa	table	track	train	tree	truck	tvmonitor	wall	water	window	wood	mIoU
Swin	49.98	39.37	77.26	22.41	20.71	40.26	92.95	60.83	44.17	50.54	55.69	82.91	75.60	23.16	73.55	61.84	83.60	34.04	19.35	50.24
Swin+BD	51.56	32.78	77.54	21.39	21.32	34.53	92.43	67.91	49.34	55.37	55.89	81.83	75.82	28.53	76.64	63.50	83.91	33.36	17.36	51.33
Ours	52.05	39.12	82.58	23.62	22.07	35.56	92.96	72.39	50.15	55.74	58.77	84.79	76.74	31.22	78.36	65.22	85.29	36.81	20.50	54.45
Improvements (Ours v.s. Swin+BD)	0.49	6.34	5.04	2.23	0.75	1.03	0.53	4.48	0.81	0.37	2.88	2.96	0.92	2.69	1.72	1.72	1.38	3.45	3.14	3.12

Table B. Per-class IoU results with Swin-T on Pascal Context validation. 'BD' means our baseline decoder.

A. Multi-scale Testing Results

Table A provides multi-scale testing results on the Pascal Context dataset. Our proposed TSG achieves constantly improvements, compared with our baselines Swin Transformer [20] and Mask2Former [5]. Moreover, our method also outperforms existing approaches in most cases. These results demonstrate the effectiveness of our TSG.

B. Per-class Results

We report per-class IoUs on Pascal Context in Table B. Our method outperform 'Swin+BD' for most of the 59 classes, with minor drops in IoU for 3 classes. In particular, for smaller objects, e.g., 'mouse' and 'flower', our method gets improvements of 26.6% and 18.4%.

C. TSG for Other Segmentation Tasks

Since our TSG is compatible with object-query-based decoders, it can be used in some instance and panoptic segmentation methods. Table C shows the comparisons of our method with baseline Mask2Former [5]. Our TSG performs favorably in instance and panoptic segmentation tasks.

Instance Segmentation	Backbone	AP (%)
Mask2Former [5]	Swin-T	43.0
TSG + Mask2Former Decoder	Swin-T	**44.2** (+1.2)
Panoptic Segmentation	Backbone	PQ (%)
Mask2Former [5]	Swin-T	51.2
TSG + Mask2Former Decoder	Swin-T	**52.7** (+1.5)

Table C. Instance and panoptic segmentation results on COCO 2017 val, with all methods trained on COCO2017 training set.

D. Computational Overhead

Table 1 and Table A show the numbers of parameters in previous methods and our TSG. Meanwhile, on the Pascal Context dataset, the FLOPs of Swin-L and ours are 410G and 498G, respectively. Compared with our baselines Swin Transformer [20] and Mask2Former [5], our method increases computational overheads in a certain degree, but improves the segmentation accuracy. Moreover, previous multi-scale methods such as PFT [26] and SenFormer [2] use similar or much more parameters than ours, while our

TSG achieves higher mIoU.

E. Failure Cases

In Fig. A, we depict some failure cases of our method. Although our TSG is able to reduce over-segmentations, under-segmentations and mis-recognitions caused by sub-optimal scales, TSG still confuses amongst some objects with similar appearances, e.g., ‘wood’, wooden ‘wall’ and ‘building’. These confusions can be alleviated with better backbones having discriminative features.

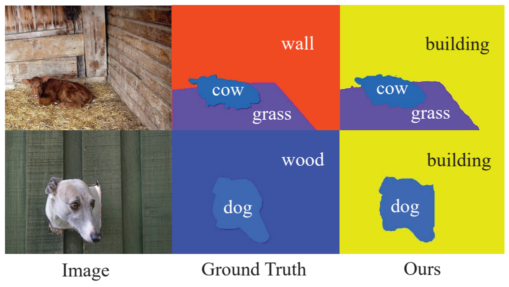

Figure A. Failure cases from our method with Swin-L.