Supplementary Material for "RankSeg: Adaptive Pixel Classification with Image Category Ranking for Segmentation"

Haodi He[1†], Yuhui Yuan[3†], Xiangyu Yue[2], and Han Hu[3]

[1] University of Science and Technology of China
[2] UC Berkeley
[3] Microsoft Research Asia
✉ {yuhui.yuan,hanhu}@microsoft.com

A. Datasets

ADE20K/ADE20K-Full. The ADE20K dataset [16] consists of 150 classes and diverse scenes with $1,038$ image-level labels, which is divided into 20K/2K/3K images for training, validation, and testing. Semantic segmentation treats all 150 classes equally, while panoptic segmentation considers the 100 thing categories and the 50 stuff categories separately. The ADE20K-Full dataset [16] contains $3,688$ semantic classes, among which we select 847 classes following [6].
PASCAL-Context. The PASCAL-Context dataset [13] is a challenging scene parsing dataset that consists of 59 semantic classes and 1 background class, which is divided into $4,998/5,105$ images for training and testing.
COCO-Stuff. The COCO-Stuff dataset [2] is a scene parsing dataset that contains 171 semantic classes divided into 9K/1K images for training and testing.
COCO+LVIS. The COCO+LVIS dataset [7,9] is bootstrapped from stuff annotations of COCO [10] and instance annotations of LVIS [7] for COCO 2017 images. There are $1,284$ semantic classes in total and the dataset is divided into 100K/20K images for training and testing.
VSPW. The VSPW [12] is a large-scale video semantic segmentation dataset consisting of $3,536$ videos with $251,633$ frames from 124 semantic classes, which is divided into $2,806/343/387$ videos with $198,244/24,502/28,887$ frames for training, validation, and testing. We only report the results on `val` set as we can not access the `test` set.
YouTubeVIS. YouTube-VIS 2019 [14] is a large-scale video instance segmentation dataset consisting of $2,883$ high-resolution videos labeled with 40 semantic classes, which is divided into $2,238/302/343$ videos for training, validation, and testing. We report the results on `val` set as we can not access the `test` set.

B. Comparison with EncNet and ESSNet

Comparison with EncNet. Table 1 compares our method to EncNet [15] based on Segmenter w/ ViT-B/16 and reports the results on the second and last rows.

† Equal contribution.

Table 1: Comparison with EncNet and ESSNet based on Segmenter w/ ViT-B.

Method	ADE20K			COCO-Stuff			COCO+LVIS		
	#params.	FLOPs	mIoU (%)	#params.	FLOPs	mIoU (%)	#params.	FLOPs	mIoU (%)
Baseline	102:50M	78:84G	48:80	102:51M	79:25G	41:85	103:37M	102:53G	19:41
EncNet	109:15M	84:89G	49:06	109:18M	85:29G	42:81	110:90M	108:58G	19:32
ESSNet	**101:42**M	**78:05**G	48:91	**101:43**M	**78:42**G	42:13	**102:29**M	100:25G	19:11
Ours	109:74M	78:71G	**49:68**	109:70M	78:55G	**44:98**	111:45M	**99:59**G	**21:26**

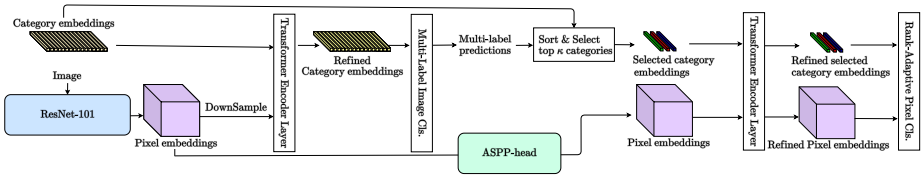
We follow the reproduced EncNet settings in `mmsegmentation` and tune the number of visual code-words as 64 as it achieves the best result in our experiments. According to the comparison results, we can see that our method significantly outperforms EncNet by +0.62%/+2.17%/+1.94% on ADE20K/COCO-Stuff/COCO+LVIS, which further verifies that exploiting rank-adaptive selected-label pixel classification is the key to our method.

Comparison with ESSNet. We compare our method to ESSNet [9] on ADE20K/COCO-Stuff/COCO+LVIS on the last two rows of Table 1. Different from the original setting [9] of ESSNet, we set the number of the nearest neighbors associated with each pixel as the same value of κ in our method to ensure fairness. We set the dimension of the representations in the semantic space as 64. According to the results on COCO+LVIS, we can see that (i) our baseline achieves 19.41%, which performs much better than the original reported best result (6.26%) in [9] as we train all these Segmenter models with batch-size 8 for 320K iterations. (ii) ESSNet achieves 19.11%, which performs comparably to our baseline and this matches the observation in the original paper that ESSNet is expected to perform better only when training the baseline method with much smaller batch sizes. In summary, our method outperforms ESSNet by +0.77%/+2.85%/+2.15% across ADE20K/COCO-Stuff/COCO+LVIS, which further shows the advantage of exploiting multi-label image classification over simply applying k-nearest neighbor search for each pixel embedding.

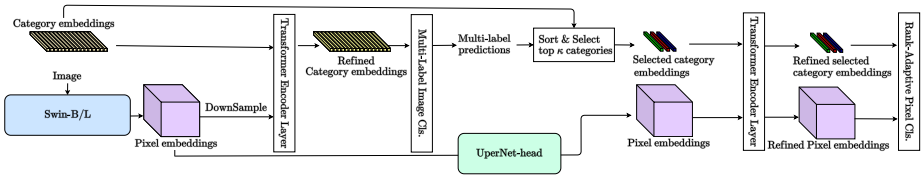
C. DeepLabv3/Swin/BEiT/MaskFormer/Mask2Former + RankSeg

We illustrate the details of combining our proposed joint multi-task scheme with DeepLabv3/Swin/BEiT/MaskFormer/Mask2Former in Figure 1 (a)/(b)/(c)/(d)/(e) respectively.

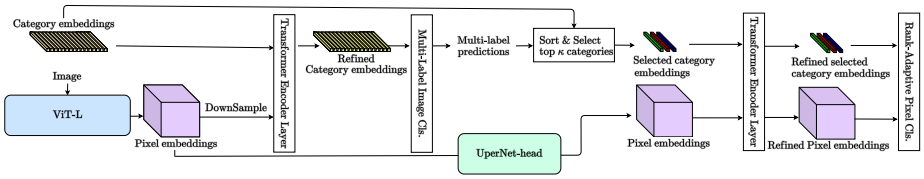
The main difference between DeepLabv3/Swin/BEiT and the Figure 6 (within the main paper) is at DeepLabv3/Swin/BEiT uses a decoder architecture to refine the pixel embeddings for more accurate semantic segmentation prediction. Besides, we empirically find that the original category embeddings perform better than the refined category embeddings used for the multi-label prediction. For MaskFormer and Mask2Former, we apply the multi-label classification scores to select the κ most confident categories and only apply the region classification over these selected categories, in other words, we perform rank-adaptive selected-label region classification instead of rank-adaptive selected-label pixel



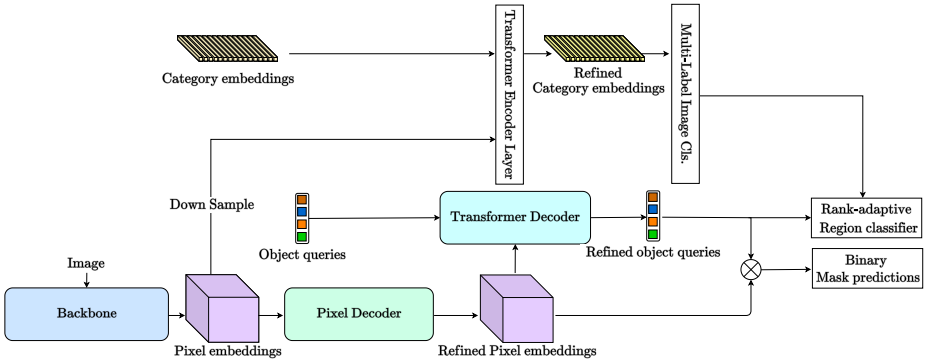
(a) DeepLabv3 + RankSeg



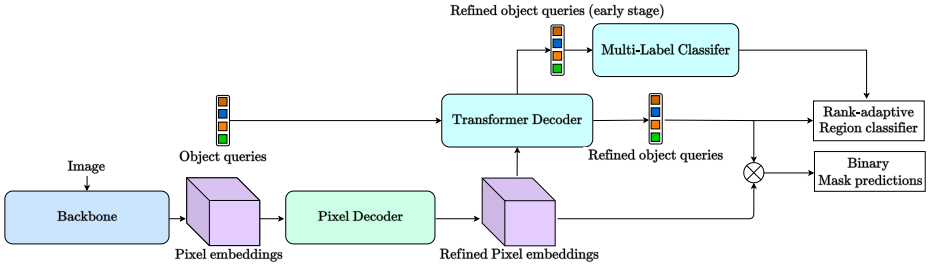
(b) Swin + RankSeg



(c) BEIT + RankSeg



(d) MaskFormer + RankSeg



(e) Mask2Former/SeMask/ViT-Adapter + RankSeg

Fig. 1: The overall framework of combining our method with DeepLabv3 [3], Swin [11], BEIT [1], MaskFormer [6], Mask2Former [5], SeMask [8], and ViT-Adapter [4].

Table 2: Hyper-parameter settings of Mask2Former + RankSeg.

Method	Image semantic seg.	Image panoptic seg.	Video semantic seg.	Video instance seg.
	ADE20K	ADE20K	VSPW	YouTubeVIS 2019
.	150	150	124	40
ml-cls. loss weight	10	10	10	10
seg. loss weight	1	1	1	1

Table 3: Influence of the number of the nearest neighbor within ESSNet. The class embedding dimension is fixed as 64 by default.

# of nearest neighbors	16	32	64	100
mIoU (%)	11:07	16:19	18:45	**19:11**

Table 4: Influence of class embedding dimension within ESSNet. The number of the nearest neighbor is set as 100 by default.

Dimension	16	32	64	128
FLOPs	100:18G	100:21G	100:25G	100:37G
mIoU (%)	19:05	19:19	19:11	**19:20**

Table 5: Dynamic with different confidence thresholds.

Threshold	0:1	0:05	0:02	0:01
mIoU (%)	44:03	**44:64**	44:54	44:56
△	+2:18	+2:79	+2:69	**+2:71**

Table 6: Combination with MaskFormer, SeMask, and ViT-Adapter.

Method	Image semantic seg.		Image panoptic seg.
	ADE20K mIoU (%)		ADE20K PQ (%)
Backbone	Swin-B	Swin-L	ResNet-50
MaskFormer [6]	53:9	55:6	34:7
+ RankSeg	55:1	55:8	36:5
SeMask [8]	–	58:2	–
+ RankSeg	–	58:5	–
ViT-Adapter [4]	–	60:5	–
+ RankSeg	–	60:7	–

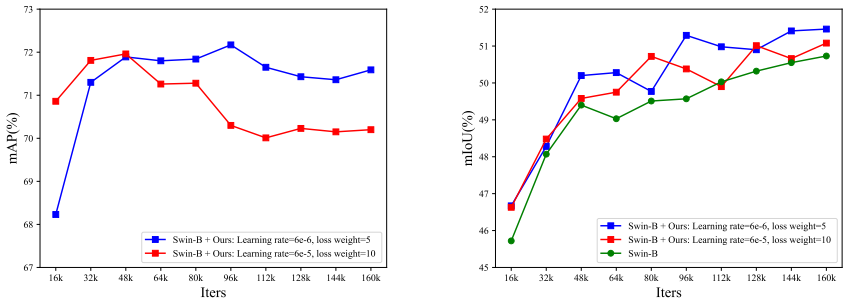
classification for MaskFormer and Mask2Former. We also improve the design of Mask2Former + RankSeg by replacing the down-sampled pixel embeddings with the refined object query embeddings output from the transformer decoder and observe slightly better performance while improving efficiency.

D. Hyper-parameter settings on Mask2Former.

Table 2 summarizes the hyper-parameter settings of experiments based on MaskFormer and Mask2Former. Considering our RankSeg is not sensitive to the choice of κ/multi-label image classification loss weight/segmentation loss weight, we simply set κ=K/multi-label image classification loss weight as 10.0/segmentation loss weight as 1.0 for all experiments and better results could be achieved by tuning these parameters. We also adopt the same set of hyper-parameter settings for the following experiments based on MaskFormer, SeMask, and ViT-Adapter.

E. Ablation study of ESSNet on COCO+LVIS.

We investigate the influence of the number of nearest neighbors and the class embedding dimension in Table 3 and Table 4 based on Segmenter w/ ViT-B.



(a) (b)

Fig. 2: Illustrating the curve of mAPs and mIoUs based on "Swin", "Swin + RankSeg", and "Swin + RankSeg" w/ smaller learning rate and loss weight on the multi-label classification head.

According to Table 3, we can see that ESSNet [9] is very sensitive to the choice of the number of nearest neighbors. We choose 100 nearest neighbors as it achieves the best performance.[4] Table 4 fixes the number of nearest neighbors as 100 and compares the results with different class embedding dimensions. We can see that setting the dimension as 32, 64, or 128 achieves comparable performance.

F. Dynamic κ

We compare the results with dynamic κ scheme in Table 5 via selecting the most confident categories, of which the confidence scores are larger than a fixed threshold value. Accordingly, we can see that using dynamic κ with different thresholds consistently outperforms the baseline but fails to achieve significant gains over the original method (44.98%) with fixed $\kappa = 50$ for all images.

G. Segmentation results based on MaskFormer and SeMask.

Table 6 summarizes the results based on combining RankSeg with MaskFormer [6] and SeMask [8]. According to the results, we can see that our RankSeg improves MaskFormer by 1.2%/1.8% on ADE20K image semantic/panoptic segmentation tasks based on Swin-B/ResNet-50 respectively. SeMask and ViT-Adapter also achieve very strong results, e.g., 58.5% and 60.7%, on ADE20K with our RankSeg.

H. Multi-label classification over-fitting issue.

Figure 2 shows the curve of multi-label classification performance (mAP) and semantic segmentation performance (mIoU) on ADE20K `val` set. These evalu-

[4] Our method sets the number of selected categories as 100 on COCO+LVIS by default.

ation results are based on the joint multi-task method "Swin-B + RankSeg". According to Figure 2 (a), we can see that the mAP of "Swin-B + RankSeg: Learning rate=6e-5, loss weight=10"[5] begins overfitting at 48K training iterations and the multi-label classification performance mAP drops from 71.96% to 70.20% at the end of training, i.e., 160K training iterations.

To overcome the over-fitting issue of multi-label classification, we attempt the following strategies: (i) larger weight decay on the multi-label classification head, (ii) smaller learning rate on the multi-label classification head, and (iii) smaller loss weight on the multi-label classification head. We empirically find that the combination of the last two strategies achieves the best result. As shown in Figure 2, we can see that using a smaller learning rate and smaller loss weight together, i.e., "Swin-B + RankSeg: Learning rate=6e-6, loss weight=5", alleviates the overfitting problem and consistently improves the segmentation performance.

I. Qualitative results

We illustrate the qualitative improvement results in Figure 3. In summary, our method successfully removes the false-positive category predictions of the baseline method.

References

1. Bao, H., Dong, L., Wei, F.: Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021)
2. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: CVPR (2018)
3. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587 (2017)
4. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. arXiv preprint arXiv:2205.08534 (2022)
5. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. arXiv preprint arXiv:2112.01527 (2021)
6. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. arXiv preprint arXiv:2107.06278 (2021)
7. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: CVPR. pp. 5356–5364 (2019)
8. Jain, J., Singh, A., Orlov, N., Huang, Z., Li, J., Walton, S., Shi, H.: Semask: Semantically masked transformers for semantic segmentation. arXiv preprint arXiv:2112.12782 (2021)
9. Jain, S., Paudel, D.P., Danelljan, M., Van Gool, L.: Scaling semantic segmentation beyond 1k classes on a single gpu. In: ICCV. pp. 7426–7436 (2021)
10. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)

[5] The original "Swin-B" [11] sets the learning rate as 6e-5 by default.

11. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
12. Miao, J., Wei, Y., Wu, Y., Liang, C., Li, G., Yang, Y.: Vspw: A large-scale dataset for video scene parsing in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4133–4143 (2021)
13. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: CVPR (2014)
14. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV. pp. 5188–5197 (2019)
15. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: CVPR. pp. 7151–7160 (2018)
16. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR (2017)