Appendix for *k*-means Mask Transformer

Qihang Yu[1], Huiyu Wang[1], Siyuan Qiao[2], Maxwell Collins[2], Yukun Zhu[2],
Hartwig Adam[2], Alan Yuille[1], and Liang-Chieh Chen[2]

[1] Johns Hopkins University
[2] Google Research

In the appendix, we provide ablation studies, along with both COCO [11] and Cityscapes [6] *test* set results. We also include more visualizations and some failure cases.

A More Experimental Results

A.1 Ablation Studies

We conduct ablation studies on COCO *val* set. To ensure the conclusion is general to different backbones, we experiment with both ResNet-50 [9] and MaX-S [18] (*i.e.*, ResNet-50 with axial-attention blocks [19] in the 3rd and 4th stages). Models are trained with 100k iterations for experiment efficiency.

Different ways for pixel-cluster interaction. The proposed k-means cross-attention adopts a different operation (*i.e.*, cluster-wise argmax) from the original cross-attention (*i.e.*, spatial-wise softmax) [17]. The modification, even though simple, significantly improves the performance at a negligible cost of extra parameters and FLOPs (incurred by the extra prediction heads for deep supervision). In Tab. 1, we provide a comparison with different cross-attention modules serving for the pixel-cluster interaction. In this ablation study, we keep everything the same (*e.g.*, the network architecture and training recipes) except the 'cross-attention modules'. As shown in the table, k-means cross-attention significantly surpasses the original cross-attention by 5.2% PQ with ResNet-50 as backbone. Even when employing a stronger backbone MaX-S, we still observe a significant gain of 4.1% PQ. In both cases, the proposed k-means cross-attention maintains a similar level of parameters and FLOPs.

We have also experimented with another improved cross-attention: dual-path cross-attention, proposed in [18]. The dual-path cross-attention simultaneously updates pixel features and cluster centers, and only shows a marginal improvement (*e.g.*, 0.5% with ResNet-50) over the original cross-attention at the cost of more parameters and FLOPs. Additionally, we attempted to combine both dual-path cross-attention and the proposed k-means cross-attention (called dual-path k-means cross-attention in the table), but did not observe any further significant improvement. Therefore, we did not use it in our final model.

Additionally, we try to add the deep supervision to the cross-attention variant as well, which degrades 1.2% PQ for ResNet-50 backbone and improves 0.1% PQ for MaX-S backbone, indicating that deep supervision, though needed to train the kMaX decoder, is not the reason of the performance improvement.

Table 1: Ablation on different ways for pixel-cluster interaction. The final setting used in kMaX-DeepLab is labeled with gray color

	ResNet-50			MaX-S		
pixel-cluster interaction module	params	FLOPs	PQ	params	FLOPs	PQ
cross-attention [17]	56M	165G	47.5	73M	237G	52.0
dual-path cross-attention [18]	58M	175G	48.0	75M	247G	52.3
k-means cross-attention	57M	168G	52.7	74M	240G	56.1
dual-path k-means cross-attention	59M	176G	53.0	76M	248G	56.2

Table 2: Ablation on the number of kMaX decoders. The three numbers (x, y, z) of each entry in column one correspond to the number of kMaX decoders deployed at output stride 32, 16, and 8, respectively. For simplicity, we only experiment with using the same number of decoders for each resolution. The final setting used in kMaX-DeepLab is labeled with gray color

number of	ResNet-50			MaX-S		
kMaX decoders	params	FLOPs	PQ	params	FLOPs	PQ
(1, 1, 1)	52M	159G	52.5	68M	231G	55.8
(2, 2, 2)	57M	168G	52.7	74M	240G	56.1
(3, 3, 3)	63M	176G	52.8	80M	248G	56.0

Number of kMaX decoders. In Tab. 2, we study the effect of deploying a different number of kMaX decoders at feature maps with output stride 32, 16, and 8. For simplicity, we only experiment with using the same number of decoders for each resolution. We note that a more complex combination is possible, but it is not the main focus of this paper. As shown in the table, using one kMaX decoder at each resolution (denoted as (1, 1, 1) in the table), our kMaX-DeepLab already achieves a good performance of 52.5% PQ and 55.8% PQ with ResNet-50 and MaX-S as backbones, respectively. Adding one more kMaX decoder per resolution (denoted as (2, 2, 2) in the table) further improves the performance to 52.7% PQ and 56.1% PQ with ResNet-50 and MaX-S as backbone, respectively. The performance starts to saturate when using more kMaX decoders. In the end, we employ totally six kMaX decoders, evenly distributed at output stride 32, 16, and 8 (see Fig. **??** for a reference).

Training convergence. As a comparison of training convergence, we train kMaX-DeepLab for 25k, 50k, 100k, 125k, 150k iterations, which gives 48.8%, 51.3%, 52.7%, 53.0%, and 53.0% for ResNet-50 backbone, and 52.4%, 54.6%, 56.1%, 56.1%, 56.2% for MaX-S backbone, respectively. Notably, kMaX-DeepLab not only shows a consistent and significant improvement over its baseline MaX-DeepLab [18], but also shows a trend to converge at 150k, while the MaX-DeepLab requires much more training iterations to converge (*e.g.*, MaX-DeepLab with MaX-S gets 0.8% and 1.1% improvement when trained for 200k, 400k, respectively).

COCO test set. We provide comparison to prior arts on COCO *test* set in Tab. 3. The performance of kMaX-DeepLab on *val* set successfully transfers to *test* set. We analyze the results below w.r.t. different backbones.

1. With ResNet-50 [9], kMaX-DeepLab outperforms MaX-DeepLab [18] with MaX-L by **2.1%** PQ, while requring **7.9**× fewer parameters and **22.0**× fewer computation.
2. Using MaX-S [18] backbone, kMaX-DeepLab outperforms MaskFormer [4] with Swin-L (window size 12) by **3.1%** PQ, while requiring **2.9**× fewer parameters, **3.3**× fewer FLOPs, and runs **3.2**× faster (FPS). Additionally, kMaX-DeepLab surpasses previous state-of-the-art method K-Net [22] by **1.2%** PQ.
3. Using ConvNeXt-L [14] backbone, kMaX-DeepLab sets a new state-of-the-art result with 58.5% PQ, significantly outperforms the best variant of MaskFormer, K-Net, and some recent works CMT-DeepLab, Panoptic SegFormer and Mask2Former by **5.2%**, **3.3%**, **2.8%** PQ, **2.3%**, and **0.2%**, respectively.

Cityscapes test set. The Cityscapes test set results are summarized in Tab. 4, where our kMaX-DeepLab does not use any external datasets [15,11] or test-time augmentation. We observe that kMaX-DeepLab, with single-scale testing, shows a significant improvement of **1.4%** PQ compared to the previous state-of-art Panoptic-DeepLab [2] with SWideRNet-(1, 1, 4.5) [1] as backbone, which adopts multi-scale testing, resulting in over **60**× more computational costs compared to kMaX-DeepLab. Finally, as shown in the table, even compared with other task-specific models, our kMaX-DeepLab also outperforms them in terms of instance segmentation (**1.7%** and **7.9%** AP over Panoptic-DeepLab and PANet [12], respectively) and semantic segmentation (**2.8%** and **1.0%** mIoU better than Panoptic-DeepLab and SegFormer [20], respectively). Our reported PQ, AP, and mIoU are obtained by a single panoptic model without any task-specific fine-tuning. This demonstrates that kMaX-DeepLab is a general method for different segmentation tasks.

B Visualization

To better understand the working mechanism behind kMaX-DeepLab model, we visualize the kMaX-DeepLab clustering process in Fig. 1 and Fig. 2, along with some failure cases in Fig. 3 and Fig. 4. We utilize kMaX-DeepLab with ResNet-50 for all visualizations, including the pixel-cluster assignment (*i.e.*, $\text{argmax}_N(\mathbf{Q}^c \times (\mathbf{K}^p)^{\mathrm{T}})$ in Eq. (7) of main paper) at each kMaX decoder stage and the final panoptic prediction. In the visualization of pixel-cluster assignments, pixels with the same color are assigned to the same cluster and their features will be aggregated to update the corresponding cluster centers.

As shown in Fig. 1 and Fig. 2, kMaX-DeepLab is capable of dealing with small objects and complex scenes, leading to a good panoptic prediction. We further visualize the failure modes of kMaX-DeepLab in Fig. 3 and Fig. 4. kMaX-DeepLab has some limitations, when handling heavily occluded objects and predicting correct semantic classes for challenging masks.

Table 3: COCO *test* set results. Our FLOPs and FPS are evaluated with the input size 1200 × 800 and a Tesla V100-SXM2 GPU. †: ImageNet-22K pretraining. ⋆: Using 256 object queries with drop query regularization. ‡: Using COCO *unlabeled* set

method	backbone	params	FLOPs	FPS	PQ	PQ$^{\text{Th}}$	PQ$^{\text{St}}$
MaX-DeepLab [18]	MaX-S [18]	62M	324G	-	49.0	54.0	41.6
MaX-DeepLab [18]	MaX-L [18]	451M	3692G	-	51.3	57.2	42.4
MaskFormer [4]	Swin-L (W12)† [13]	212M	792G	5.2	53.3	59.1	44.5
K-Net [22]	Swin-L (W7)† [13]	-	-	-	55.2	61.2	46.2
CMT-DeepLab [21]	Axial-R104-RFN† [16]	270M	1114G	3.2	55.7	61.6	46.8
Panoptic SegFormer [10]	Swin-L (W7)† [13]	221M	816G	-	56.2	62.3	47.0
Mask2Former [3]	Swin-L (W12)† [13]	216M	868G	4.0	58.3	**65.1**	48.1
*k*MaX-DeepLab	ResNet-50 [9]	57M	168G	22.8	53.4	59.3	44.5
*k*MaX-DeepLab	MaX-S† [18]	74M	240G	16.9	56.4	62.7	46.9
*k*MaX-DeepLab	ConvNeXt-B† [14]	122M	380G	11.6	57.8	64.3	48.1
*k*MaX-DeepLab	ConvNeXt-L† [14]	232M	744G	6.7	58.0	64.5	48.2
*k*MaX-DeepLab*	ConvNeXt-L† [14]	232M	749G	6.6	58.2	64.7	48.5
*k*MaX-DeepLab‡	ConvNeXt-L† [14]	232M	744G	6.7	**58.5**	64.8	**49.0**

Table 4: Cityscapes *test* set results. †: ImageNet-22K pretraining. TTA: test-time augmentation (which usually incurs at least 10× more computational cost). Our reported PQ, AP, and mIoU are obtained by a single panoptic model (*i.e.*, no task-specific fine-tuning). We mainly consider results without external dataset (*e.g.*, Mapillary Vistas, COCO) for a fair comparison

method	backbone	TTA	PQ	AP	mIoU
Panoptic-DeepLab [2]	Xception-71 [5]	✓	62.3	34.6	79.4
Axial-DeepLab [19]	Axial-ResNet-XL [19]	✓	62.8	34.0	79.9
Panoptic-DeepLab [2]	SWideRNet-(1,1,4.5) [1]	✓	64.8	38.0	80.4
SETR [23]	ViT-L† [7]	✓	-	-	81.1
SegFormer [20]	MiT-B5 [20]	✓	-	-	82.2
Mask R-CNN [8]	ResNet-50 [9]		-	26.2	-
PANet [12]	ResNet-50 [9]		-	31.8	-
*k*MaX-DeepLab	ConvNeXt-L† [14]		66.2	39.7	83.2

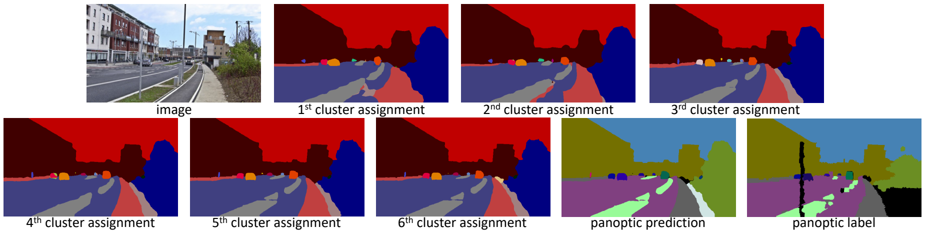

Fig. 1: *k*MaX-DeepLab is capable of capturing extremely small objects, which may be even missing in the ground truth annotation (*e.g.*, the person on the street, in the left side of the image. Best viewed zoom in)

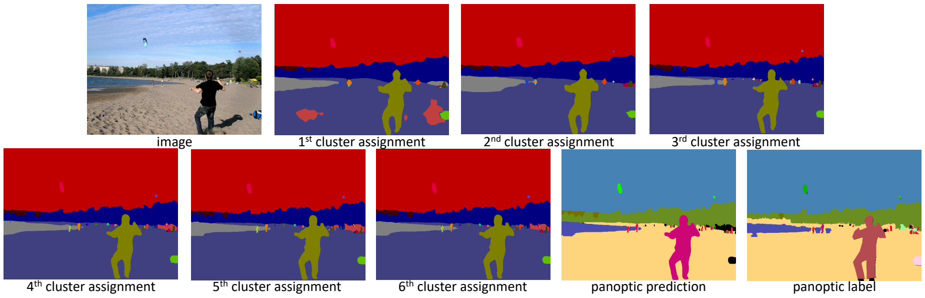

Fig. 2: kMaX-DeepLab is capable of handling images with many small objects in a complex scene

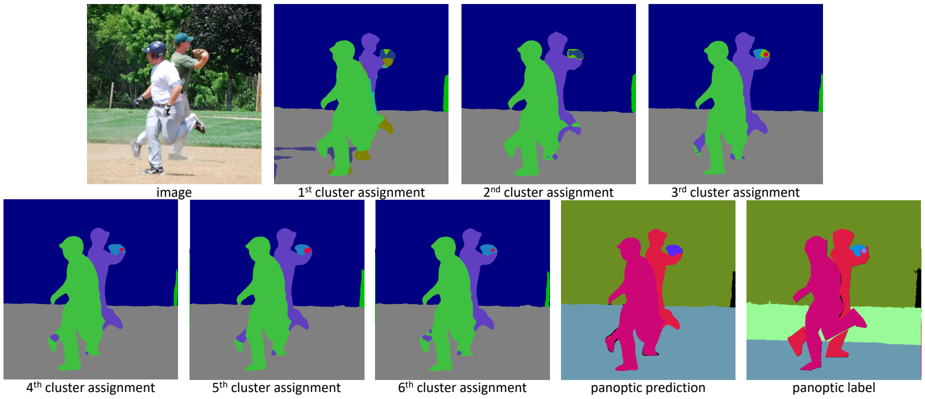

Fig. 3: **[Failure mode]** kMaX-DeepLab struggles to segment both heavily occluded objects and obscure small objects. Specifically, the legs between occluded persons are not well segmented. Additionally, the obscure small baseball is not found at the first two stages. Even though it is recovered in the 3rd clustering stage, it still vanishes in the final prediction. It remains a challenging problem to make the full use of all clustering results to help the final prediction

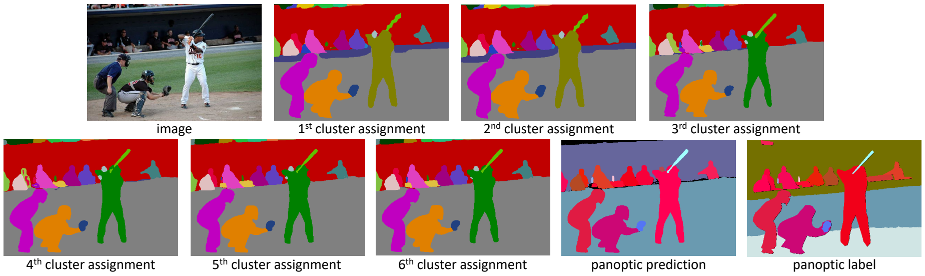

Fig. 4: **[Failure mode]** Although kMaX-DeepLab shows a strong ability to split images into different regions, it may not yield the correct semantic prediction. In this example, kMaX-DeepLab is able to segment out the background regions, but fails to predict the correct semantic labels

References

1. Chen, L.C., Wang, H., Qiao, S.: Scaling wide residual networks for panoptic segmentation. arXiv:2011.11675 (2020) 3, 4
2. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. In: CVPR (2020) 3, 4
3. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. CVPR (2022) 4
4. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: NeurIPS (2021) 3, 4
5. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: CVPR (2017) 4
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016) 1
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 4
8. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017) 4
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 1, 3, 4
10. Li, Z., Wang, W., Xie, E., Yu, Z., Anandkumar, A., Alvarez, J.M., Lu, T., Luo, P.: Panoptic segformer. CVPR (2022) 4
11. 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) 1, 3
12. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: CVPR (2018) 3, 4
13. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021) 4
14. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. CVPR (2022) 3, 4
15. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: ICCV (2017) 3
16. Qiao, S., Chen, L.C., Yuille, A.: Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: CVPR (2021) 4
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) 1, 2
18. Wang, H., Zhu, Y., Adam, H., Yuille, A., Chen, L.C.: Max-deeplab: End-to-end panoptic segmentation with mask transformers. In: CVPR (2021) 1, 2, 3, 4
19. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., Chen, L.C.: Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation. In: ECCV (2020) 1, 4
20. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. In: NeurIPS (2021) 3, 4
21. Yu, Q., Wang, H., Kim, D., Qiao, S., Collins, M., Zhu, Y., Adam, H., Yuille, A., Chen, L.C.: Cmt-deeplab: Clustering mask transformers for panoptic segmentation. In: CVPR (2022) 4

22. Zhang, W., Pang, J., Chen, K., Loy, C.C.: K-net: Towards unified image segmentation. In: NeurIPS (2021) 3, 4
23. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: CVPR (2021) 4