Supplementary Materials for CMT-DeepLab: Clustering Mask Transformers for Panoptic Segmentation

Qihang Yu[1] Huiyu Wang[1] Dahun Kim[2] Siyuan Qiao[3] Maxwell Collins[3] Yukun Zhu[3]
Hartwig Adam[3] Alan Yuille[1] Liang-Chieh Chen[3]
[1]Johns Hopkins University [2]KAIST [3]Google Research

In the supplementary materials, we provide more technical details, along with more ablation and comparison results with other concurrent works. We also include more visualizations and comparisons over the baselines. Additionally, we provide a comprehensive comparison, in terms of training epochs, memory cost, parameters, FLOPs, and FPS, across different methods. We also report results with a ResNet-50 backbone for a fair comparison across different methods, along with additional results on Cityscapes. Finally, we summarize the limitations of our work and potential negative impacts.

1. More Technical Details

Backbones. In Fig. 1, we provide an architectural comparison of MaX-DeepLab-S [9] and CMT-DeepLab built upon Axial-R50/104 [10]. Specifically, we simplify the backbone from MaX-DeepLab-S [9] by removing transformer modules in the backbone (light blue), and stacking more blocks in the decoder module (light orange). The Axial-R104 backbone is obtained by scaling up Axial-R50 (*i.e.*, four times more layers in the stage-4).

Recursive Feature Network. We construct Recursive Feature Network (RFN) in a manner similar to [7]. More specifically, we stack two models together, with a skip-connection from the decoder features at stride 4 in the first network to the encoder features at stride 4 in the second network. Instead of using the complicated fusion module proposed in [7], we simply average the features for fusion. Moreover, the two networks share the same set of cluster centers (*i.e.*, object queries), which are sequentially updated from the first network to the second one. We also add supervision for the first network but use the Hungarian matching results based on the final output.

2. More Results and In-depth Analysis

Effect of frequent pixel update (our second solution). As discussed in the main paper, the clustering results will be also used to update pixel features besides cluster centers to ensure a frequent pixel update. We tried removing the pixel feature updates from clustering transformer, which leads to a degradation of 0.4% PQ.

Comparison with more concurrent works. Also shown in Tab. 1, we compare our CMT-DeepLab with the baseline MaX-DeepLab [9], and *concurrent* works MaskFormer [2] and K-Net [11] on the *test-dev* set. As shown in the table, our best model (using 200K iterations and RFN) attains the performance of 55.7% PQ on the test-dev set, which is 4.4% and 2.4% better than MaX-DeepLab-L [9] and MaskFormer [2]. Our best model is 0.5% PQ better than K-Net [11], which adopts a different framework (*i.e.*, dynamic kernels) than mask-transformer-based approaches. In addition to PQ, we further look into RQ and SQ for performance analysis. We observe that with a similar performance to K-Net [11] in RQ, our best model performs better in SQ. Specifically, our best model yields 83.6% SQ, which is 1.2%, 1.6%, and 1.1% better than K-Net, MaskFormer, and MaX-DeepLab-L, respectively. Interestingly, our lightweight variant, CMT-DeepLab with Axial-R50, achieves 83.0% SQ, which is still better than all the other methods. We attribute our better performance in SQ to the proposed clustering mask transformer layer, which yields denser attention maps to facilitate segmentation tasks.

Accuracy-cost Trade-off Comparison. We provide a comprehensive comparison of training cost (epochs, memory), model size (params, FLOPs, FPS), and performance (PQ) in Tab. 2. The training memory is measured on a TPU-v4, while other statistics are measured with a Tesla V100-SXM2 GPU. We use TensorFlow 2.7, cuda 11.0, input size 1200×800, and batch size 1. For MaskFormer (PyTorch-based), we cite the numbers from their paper. As shown in the table, our CMT-DeepLab-S (Axial-R50) outperforms MaskFormer-SwinB by 1.2% PQ with comparable model size and inference cost. Our CMT-DeepLab-S also outperforms MaskFormer-SwinL while using much fewer model parameters and running faster. All our models outperform MaX-DeepLab. Notably, our best model CMT-DeepLab-L-RFN (Axial-R104-RFN) outperforms MaX-DeepLab-L by 4.2% PQ while using only 60% model parameters and 33.6% FLOPs.

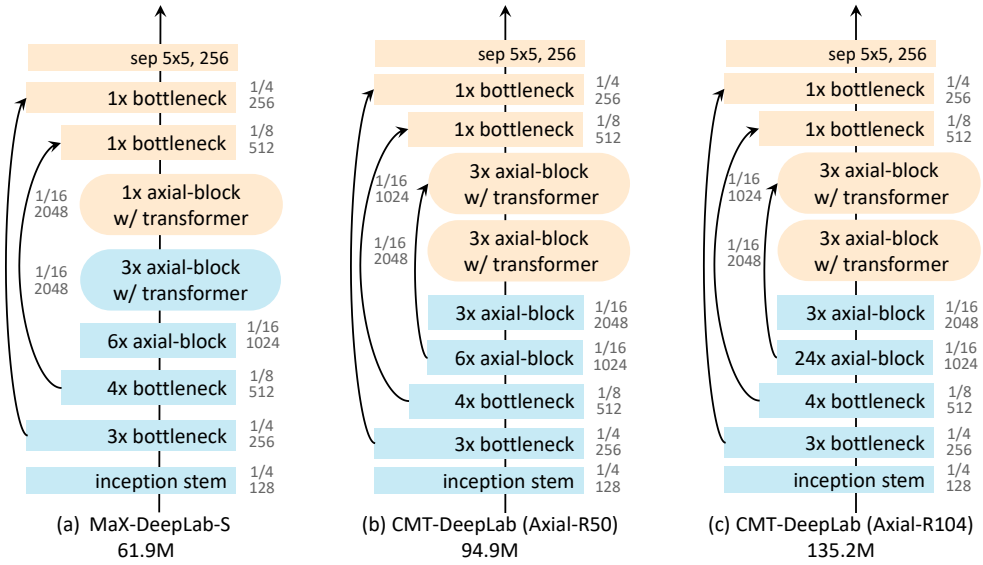

Figure 1. A visual comparison of architecture between MaX-DeepLab-S and CMT-DeepLab. Pretrained backbone part is labeled in blue color.

method	backbone	params	val-set PQ	val-set PQ^{Th}	val-set PQ^{St}	test-dev PQ	test-dev PQ^{Th}	test-dev PQ^{St}	SQ	RQ
MaX-DeepLab-S [9]	MaX-S [9]	61.9M	48.4	53.0	41.5	49.0	54.0	41.6	-	-
MaX-DeepLab-L [9]	MaX-L [9]	451M	51.1	57.0	42.2	51.3	57.2	42.4	82.5	61.3
MaskFormer† [2]	Swin-B‡ [6]	102M	51.8	56.9	44.1	-	-	-	-	-
MaskFormer† [2]	Swin-L‡ [6]	212M	52.7	58.5	44.0	53.3	59.1	44.5	82.0	64.1
K-Net† [11]	R101-FPN [4]	-	49.6	55.1	41.4	-	-	-	-	-
K-Net† [11]	R101-FPN-DCN [3]	-	48.3	54.0	39.7	-	-	-	-	-
K-Net† [11]	Swin-L‡ [6]	-	54.6	60.2	46.0	55.2	61.2	46.2	82.4	**66.1**
CMT-DeepLab	Axial-R50‡ [10]	94.9M	53.0	57.7	45.9	53.4	58.3	46.0	83.0	63.6
CMT-DeepLab	Axial-R104‡	135.2M	54.1	58.8	**47.1**	54.5	59.6	46.9	83.2	64.7
CMT-DeepLab	Axial-R104‡-RFN	270.3M	55.1	60.6	46.8	55.4	61.0	**47.0**	83.5	65.6
CMT-DeepLab (iter 200k)	Axial-R104‡-RFN	270.3M	**55.3**	**61.0**	46.6	**55.7**	**61.6**	46.8	**83.6**	65.9

Table 1. Results comparison on COCO val and test-dev set. ‡: ImageNet-22K pretraining. †: Concurrent works. We update comparison with concurrent works, and also our improved results with longer training iterations.

method	epochs	memory	params	FLOPs	FPS	PQ
MaskFormer-SwinB [2]	300	-	102M	411G	8.4	51.8
MaskFormer-SwinL [2]	300	-	212M	792G	5.2	52.7
MaX-DeepLab-S [9]	216	6.3G	62M	291G	11.9	48.4
MaX-DeepLab-L [9]	216	28.7G	451M	3317G	2.2	51.1
CMT-DeepLab-S	54	10.2G	95M	396G	8.1	53.0
CMT-DeepLab-L	54	11.8G	135M	553G	6.0	54.1
CMT-DeepLab-L-RFN	54	25.8G	270M	1114G	3.2	55.1
CMT-DeepLab-L-RFN	108	25.8G	270M	1114G	3.2	55.3

Table 2. A comprehensive accuracy-cost trade-off comparison.

Backbone Differences. As different backbones are adopted for different methods (*e.g.*, MaX-S/L [9], Swin [6]), it hinders a direct and fair comparison across different methods. To this end, we provide results based on a ResNet-50 backbone across different models on COCO *val* set. As shown in Tab. 3, our CMT-DeepLab significantly outperforms MaX-DeepLab and concurrent works (MaskFormer and K-Net).

	MaskFormer [2]	K-Net [11]	MaX-DeepLab [9]	CMT-DeepLab
PQ	46.5	47.1	46.0	**48.5**

Table 3. Results comparison with ResNet-50 as the backbone.

Results on Cityscapes. We provide additional results on Cityscapes in Tab. 4. For a fair comparison, we adopt the **same** setting, including pretrain weights (IN-1k), training hyper-parameters (e.g., iterations 60k, learning rate 3e-4, crop size 1025×2049), and post-processing scheme (pixel-wise argmax as in MaX-DeepLab). As shown in the table, our CMT-DeepLab-S significantly outperforms MaX-DeepLab-S by 2.9% PQ and 1.6% mIoU.

method	PQ	RQ	SQ	mIoU
MaX-DeepLab-S	61.7	74.5	81.5	79.8
CMT-DeepLab-S	64.6	77.4	82.6	81.4

Table 4. Cityscapes *val* set results.

3. Visual Comparison

Visualization Details. To visually compare the clustering results/attention maps, we firstly follow DETR [1] to average values across multi-heads to obtain a single attention map, which is then transformed into a heatmap in a manner similar to CAM [12] by normalizing the values to the range [0, 255]. Note that we do not apply any smoothing techniques (*e.g.*, square root), which in fact adjust the learned attention values. These differences make the visualization differ from those in the paper of MaX-DeepLab [9]. All visualizations are done with CMT-DeepLab based on Axial-R50, and MaX-DeepLab-S, with input size 641×641.

Clustering results. In Fig. 2, Fig. 3, Fig. 4, and Fig. 5, we provide more clustering visualization results. We observe the same trend as we presented in the main paper that the clustering results, providing denser attention maps, are close-to-random at the beginning and are gradually refined to focus on different objects. Interestingly, we also observe some exceptions (see Fig. 2, Fig. 3, Fig. 4), where the clustering results start with a good semantic-level clustering, indicating that some cluster centers can embed semantic information and thus specialize in some classes.

Attention map comparison with MaX-DeepLab. In Fig. 6, Fig. 7, and Fig. 8, we show more attention map comparison with MaX-DeepLab. As shown in those figures, CMT-DeepLab provides a much denser attention map than MaX-DeepLab.

4. Limitations

Motivated from a clustering perspective, CMT-DeepLab generates denser attention maps and thus leads to a superior performance in the segmentation task. However, the proposed clustering mask transformer, though significantly improves the segmentation quality (SQ), does not bring the same-level performance boost on the recognition ability (RQ). Specifically, we have adopted some simple scaling-up strategies, including increasing model size, input size, or training iterations. Those strategies result in a large performance gain in RQ as a compensation, but with a cost at parameters, computation, or training time. It thus remains an interesting problem to explore in the future that how to improve its recognition ability efficiently and effectively.

5. Potential Negative Impacts

In this paper, we present a new panoptic segmentation framework, inspired by the traditional clustering-based algorithm, generates denser attention maps and further achieves new state-of-the-art performance. The findings described in this paper can potentially help advance the research in developing stronger, faster, and more elegant end-to-end segmentation methods. However, we also note that there is a long-lasting debate on the impacts of AI on human world. As a method improving the fundamental task in computer vision, our work also advances the development of AI, which means there could be both beneficial and harmful influences depending on the users.

License of used assets. COCO dataset [5]: CC-by 4.0. ImageNet [8]: https://image-net.org/download.php.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[2] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 1, 2

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2

[4] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 3

[6] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2

[7] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. *arXiv:2006.02334*, 2020. 1

[8] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115:211–252, 2015. 3

[9] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021. 1, 2, 3, 6

[10] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation. In *ECCV*, 2020. 1, 2

[11] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. In *NeurIPS*, 2021. 1, 2

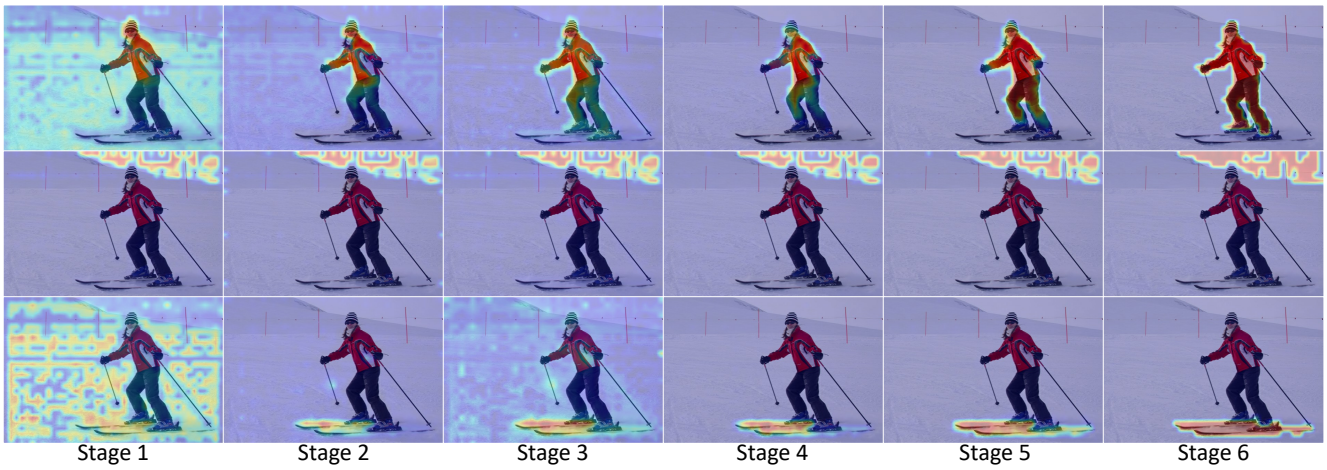

Figure 2. Visualization of clustering results at different stages (*i.e.*, transformer layers). We note that clustering results for person (row 1) and skis (row 3) start from a close-to-random distribution at the beginning and are gradually refined to focus on corresponding target. But we also find some cluster centers, *e.g.*, sky in row 2, are specialized in some semantic classes and start at a good semantic clustering.

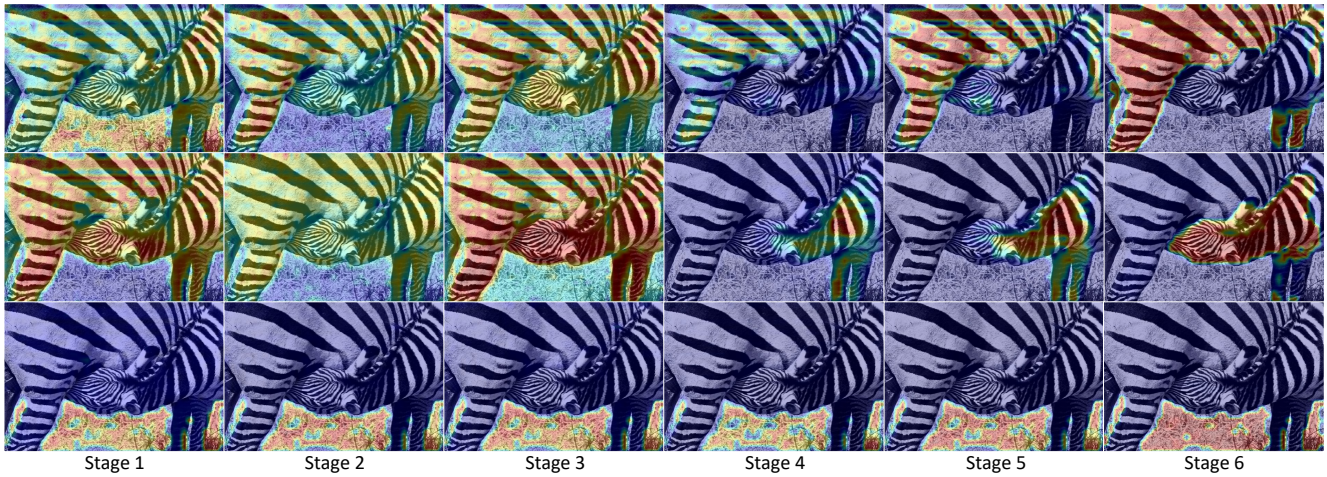

Figure 3. Visualization of clustering results at different stages (*i.e.*, transformer layers). Both row 1 and 2 experience a semantic-to-instance refinement during the clustering process (*e.g.*, in col 3, both clustering results capture all zebras.), which finally falls onto corresponding zebra. The cluster center on row 3 initializes with a good clustering result for grass, which coincides with the observation that some cluster centers intrinsically embed semantic information.

[12] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 3

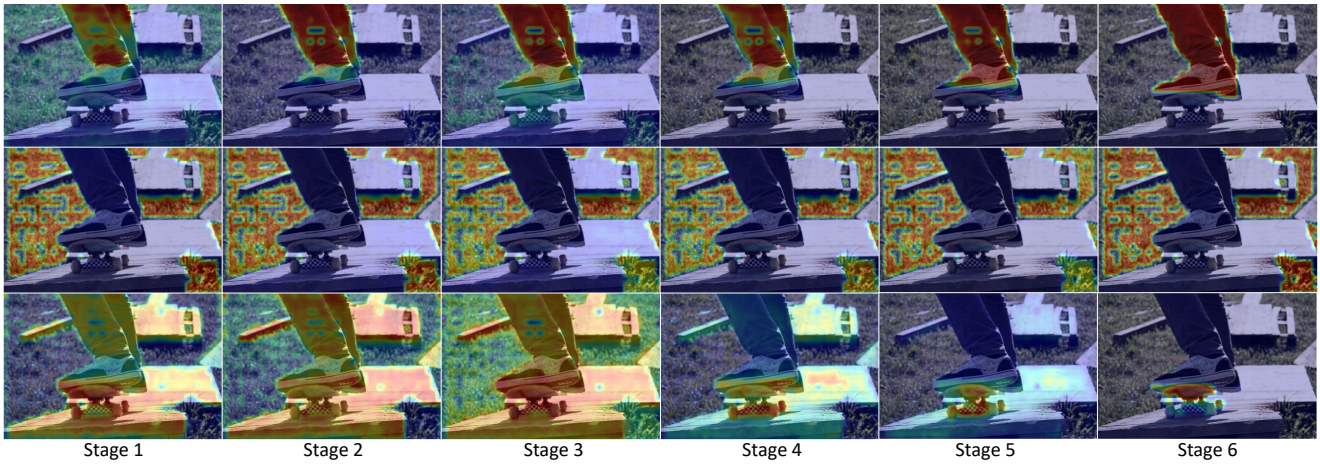

Figure 4. Visualization of clustering results at different stages (*i.e.*, transformer layers). Row 1, 3 gradually falls into the target person and skateboard, while row 2 starts with a good clustering for grass.

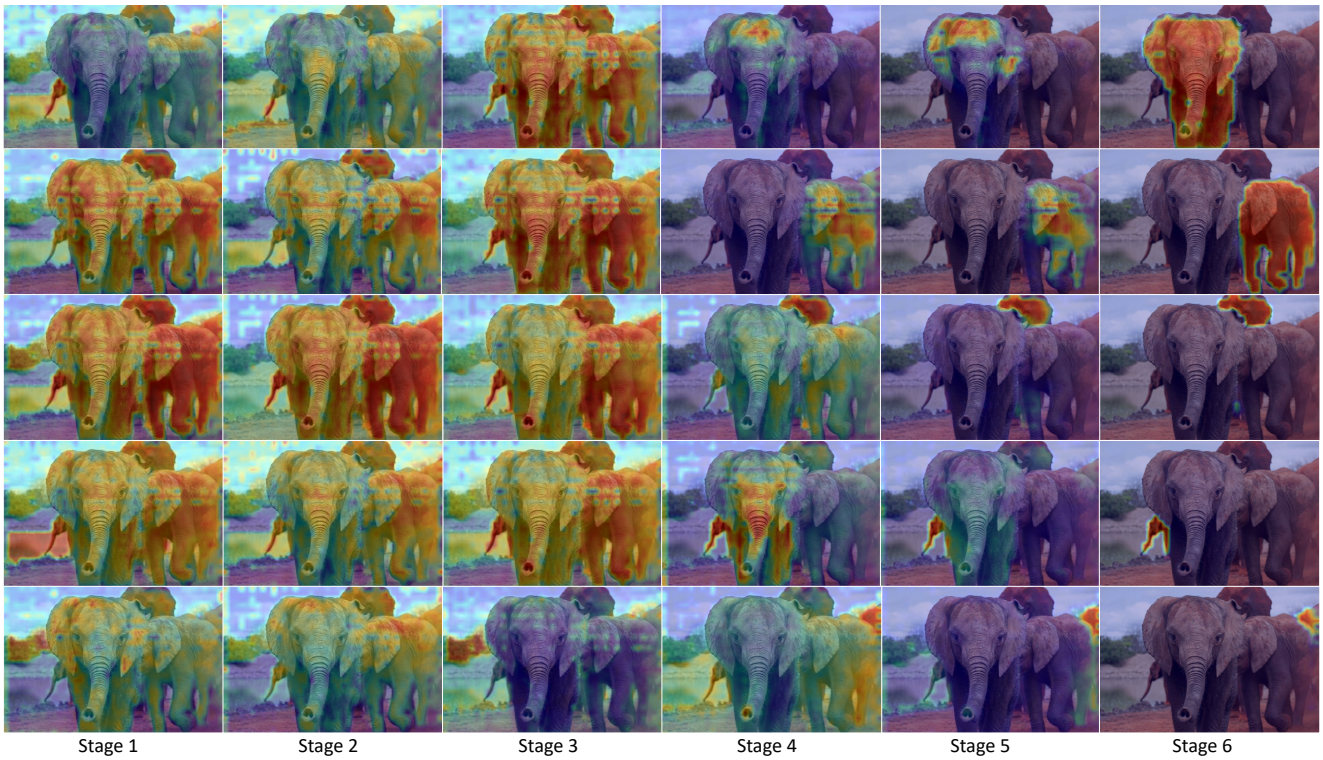

Figure 5. Visualization of clustering results at different stages (*i.e.*, transformer layers). Each row corresponds to an elephant instance prediction. Similarly, most results start from a close-to-random clustering and gradually converge to the target in a semantic-to-instance manner.

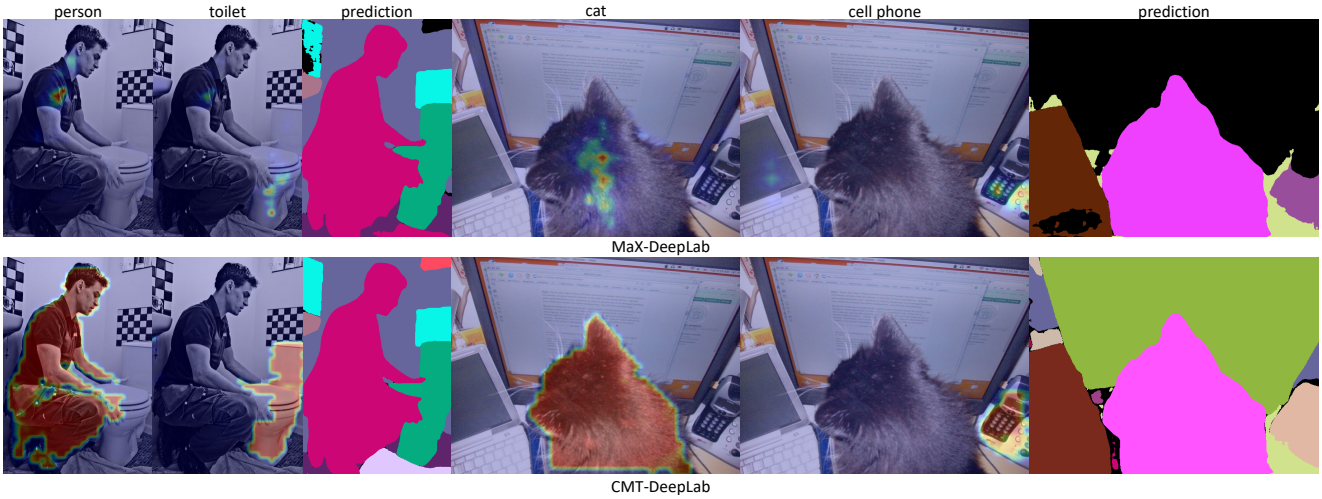

Figure 6. Visual comparison between CMT-DeepLab and MaX-DeepLab [9]. CMT-DeepLab provides a denser attention map to update cluster centers, which leads to superior performance in dense prediction tasks.

Figure 7. Visual comparison between CMT-DeepLab and MaX-DeepLab [9]. CMT-DeepLab provides a denser attention map to update cluster centers, which leads to superior performance in dense prediction tasks.

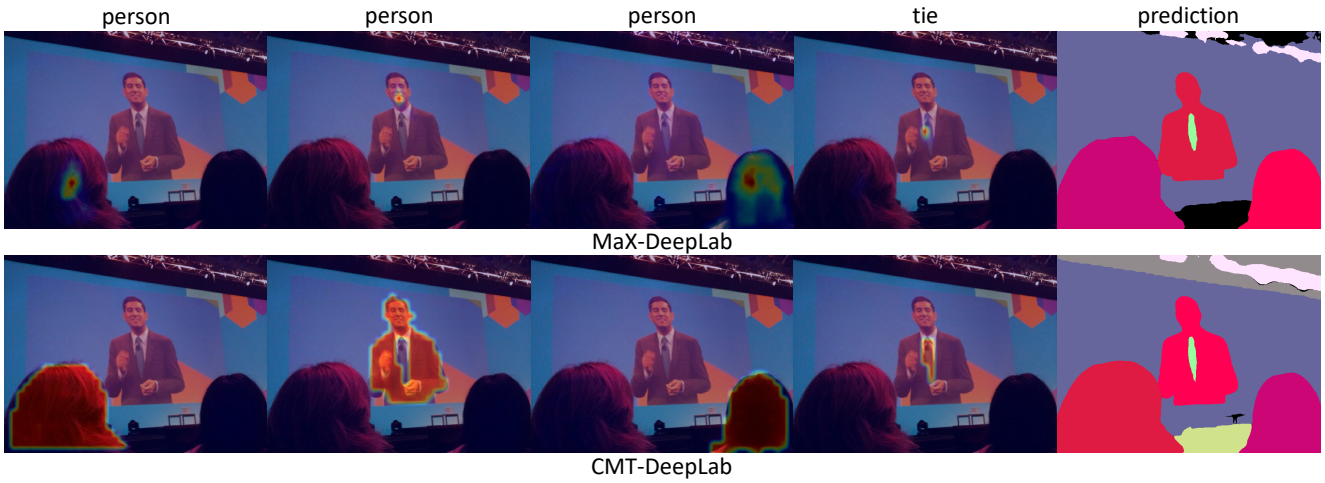

Figure 8. Visual comparison between CMT-DeepLab and MaX-DeepLab [9]. CMT-DeepLab provides a denser attention map to update cluster centers, which leads to superior performance in dense prediction tasks.