Supplementary Material: Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++

David Acuna[1,3,*] Huan Ling[1,2*] Amlan Kar[1,2,*] Sanja Fidler[1,2]
[1]University of Toronto [2]Vector Institute [3]NVIDIA[†]
{davidj, linghuan, amlan, fidler}@cs.toronto.edu

1. Appendix

In the appendix, we provide additional qualitative and quantitative results for our Polygon-RNN++ model. In particular, we show an additional ablation study that emphasizes the importance of the evaluator network. We also illustrate the performance of our model across the course of training during the RL training phase. We provide further details on automatic full-image object instance segmentation. Finally, we provide several qualitative examples for both the automatic and interactive modes.

Training with RL. In Fig. 1, we plot the performance in terms of mean IoU during RL training on Cityscapes [1]. Additionally, we show the average length of polygons. Note that nothing in our model prevents predicting polygons that include self-intersection. We also investigate this issue in this plot. In comparison with the MLE model, we can see that we obtain an increase in terms of mean IoU. Note also that by directly optimizing IoU, the average length of the predicted polygons and the number of polygons with at least one self-intersection decrease.

Evaluator Network. In Table 1 we compare different decoding strategies, and showcase the importance of the evaluator network. We separate two cases: handling multiple first vertex candidates, as well as multiple sequences (polygons) that follow from each first vertex candidate. We see that while beam search for both the first vertex and the full sequence outperforms greedy decoding, the improvement is minor (0.5%, second row). Following K first vertices using greedy decoding for the sequence and using the evaluator network to choose between the K polygons results in 2.2% over beam search (third row). Using beam search for the sequence (and evaluator network to choose between the K polygons in the end) further increases performance (0.3%, fourth row). In the last row, we use beam search until the last predicted polygon vertex, and use the evaluator network to also choose between the K last vertex candidates (for each first vertex candidate). This gets us another 0.2% (last row).

Output Resolution and Sensitivity to T of GGNN. Note that our GGNN formulation is efficient and can handle large output sizes. We experimented with two output resolutions, 112×112, and 224×224. The result for the 224×224 was only 0.02% better than that of 112×112, but required longer training times. Thus, our choice in the paper is 112×112. In Table 2 we report results for different T (number of propagation steps), showing stable results across several options.

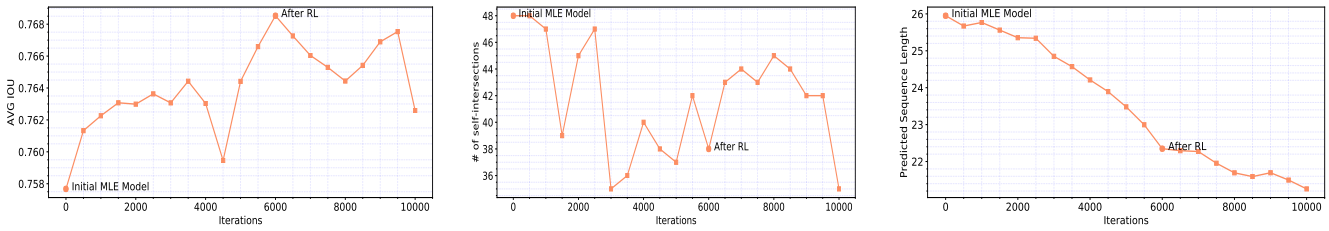
Automatic Mode in Cityscapes. In Figure 3 and 4 we provide a qualitative comparison between our model in automatic mode and the ground-truth polygons. The first column illustrates the predicted full image while the second shows the GT polygons. We remind the reader that here, our model exploits ground-truth bounding boxes.

Full-image Instance-Level Segmentation on Cityscapes. We evaluate our model on the task of instance segmentation. In our scenario, this can also be seen as an automatic full image annotation task. Since PolygonRNN++ requires bounding boxes, we use FasterRCNN [6] for object detection on the whole image. In particular, we train the best FasterRCNN model of [4] (pre-trained on MS-COCO) on the Cityscapes dataset (fine annotations only). The predicted boxes are then fed to our model to produce polygonal instance segmentations. Evaluating Polygon-RNN++ with FasterRCNN on the **Cityscapes test set** achieves 22.8% AP and 42.6% AP_{50}.

A major limitation of our model in this scenario is that it only predicts one polygon per bounding-box. As a result, multi-component object-masks, coming from occluding objects, are heavily penalized by the evaluation procedure.

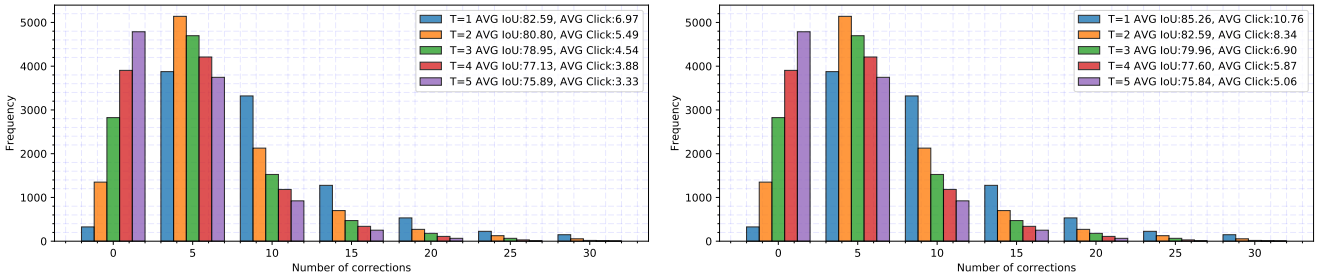
In order to tackle this issue, we further use semantic information. Specifically, we use the predicted semantic segmentation results from [9]. We then perform a logical "and" operation between the predicted class-semantic map and our instance polygonal prediction. Following this scheme, we achieve 25.49% AP and 45.47% AP_{50} on the **test set**.

*authors contributed equally
†work done when D.A. was at UofT



(a) Average IOU Inference Greedy Pass (b) Number of Self-Intersections (c) Polygons AVG Length

Figure 1: Performance during RL training on our *validation* set on the Cityscapes dataset.



(a) $T_2 = 0.7$ (b) $T_2 = 0.8$

(c) $T_2 = 0.9$ (d) $T_2 = 1.0$

Figure 2: Interactive Mode on Cityscapes for different values of T_2

In Figure 5, we show more qualitative results of our full instance segmentation model (i.e. with boxes from Faster-RCNN). Results on the Cityscapes Instance Segmentation Benchmark are reported in Table 3.

Interactive mode in Cityscapes. Figure 2 shows the histogram of the number of corrections (clicks) for different values of T_2 and T. It also shows the average IoU and the average number of clicks for the given thresholds. We can see that most of the predictions can be successfully corrected with 5 clicks. Figure 7 shows a few qualitative examples of our interactive simulation on the Cityscapes dataset. The automatically predicted polygons are shown in the first column while the second column shows the result after a certain number of corrections. The last one depicts the ground-truth polygons. For all instances, we show the required number of clicks and the achieved IoU.

Automatic Mode in Out-of-Domain Imagery. In Figures 8, 9, 10, 11, 12 we analyze the qualitative performance of our model on different datasets that capture both shifts in environment KITTI [2] and domain (general scenes [10], aerial [8], medical [5, 7, 3]). We emphasize that here we use our model trained on Cityscapes **without any fine-tuning** on these datasets. The first column shows prediction in automatic mode while the second column visualizes the GT instances. Note that in some of these datasets we do not have GT polygons as only segmentation masks are provided. In those cases, the GT image is labeled as *Mask* and the number of clicks is not shown.

Automatic Mode and Online Fine-Tuning. Figure 13 illustrates the performance of the proposed Online Fine-tuning algorithm on different datasets. We first show the prediction of the model without any fine-tuning. We then illustrate the automatic predictions of the fine-tuned model. The GT

First Vertex	Sequence	Bicycle	Bus	Person	Train	Truck	Motorcycle	Car	Rider	Mean
Greedy	Greedy	57.57	75.74	68.78	59.40	75.97	58.19	75.88	65.47	67.13
BeamSearch (BS)	BS	58.50	74.43	68.68	61.73	75.90	58.44	75.51	66.73	67.49
Eval. Net	Greedy	61.62	79.31	70.37	62.17	77.45	60.71	77.80	68.30	69.72
Eval. Net	BS	62.04	79.19	**70.87**	62.61	77.81	**62.00**	77.87	68.25	70.08
Eval. Net	BS- Eval. Net	**62.34**	**79.63**	70.80	**62.82**	**77.92**	61.69	**78.01**	**68.46**	**70.21**

Table 1: The role of the evaluation network. Here *Greedy* denotes greedy decoding of the polygon, and *BS* indicates beam-search. We use $K = 5$. Performance (IoU in %) in automatic mode for all Cityscapes classes.

T_{GGNN} (# of propagation steps)	3	5	7
AVG IoU (%)	71.37	71.38	71.46

Table 2: Performance (automatic mode) for different number of propagation steps (T_{GGNN}) in GGNN. Experiment done on Cityscapes. We report the performance averaged across all categories.

Model	AP	AP 50 %
PANet	36.4	63.1
Mask R-CNN	32.0	58.1
SegNet	29.5	55.6
GMIS	27.6	44.6
PolygonRNN++	25.5	45.5
SGN	25.0	44.9

Table 3: Performance on official Cityscapes Instance Labeling benchmark (test). We report best result for each method.

instances are shown in the last column. In all cases, the fine-tuned predictions are generated after the last chunk has been seen (illustrated in Figure 9 of the main paper).

Extended Evaluation on ADE Val Set. In Figure 6, we report the performance of PolygonRNN++ on the ADE validation set without any fine-tuning, and running in automatic mode (with ground-truth boxes). Note that, for the ease of visualization we only illustrate the top and bottom 50 categories, sorted by performance. Only instances with more than 20 categories are shown.

References

[1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[2] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012.

[3] S. Gerhard, J. Funke, J. Martel, A. Cardona, and R. Fetter. Segmented anisotropic ssTEM dataset of neural tissue. *figshare*, 2013.

[4] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, pages 3296–3297, July 2017.

[5] A. H. Kadish, D. Bello, J. P. Finn, R. O. Bonow, A. Schaechter, H. Subacius, C. Albert, J. P. Daubert, C. G. Fonseca, and J. J. Goldberger. Rationale and Design for the Defibrillators to Reduce Risk by Magnetic Resonance Imaging Evaluation (DETERMINE) Trial. *J Cardiovasc Electrophysiol*, 20(9):982–7, 2009.

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[7] A. Suinesiaputra, B. R. Cowan, A. O. Al-Agamy, M. A. Elattar, N. Ayache, A. S. Fahmy, A. M. Khalifa, P. Medrano-Gracia, M.-P. Jolly, A. H. Kadish, D. C. Lee, J. Margeta, S. K. Warfield, and A. A. Young. A collaborative resource to build consensus for automated left ventricular segmentation of cardiac MR images. *Medical Image Analysis*, 18(1):50 – 62, 2014.

[8] X. Sun, C. M. Christoudias, and P. Fua. Free-shape polygonal object localization. In *ECCV*, 2014.

[9] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[10] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

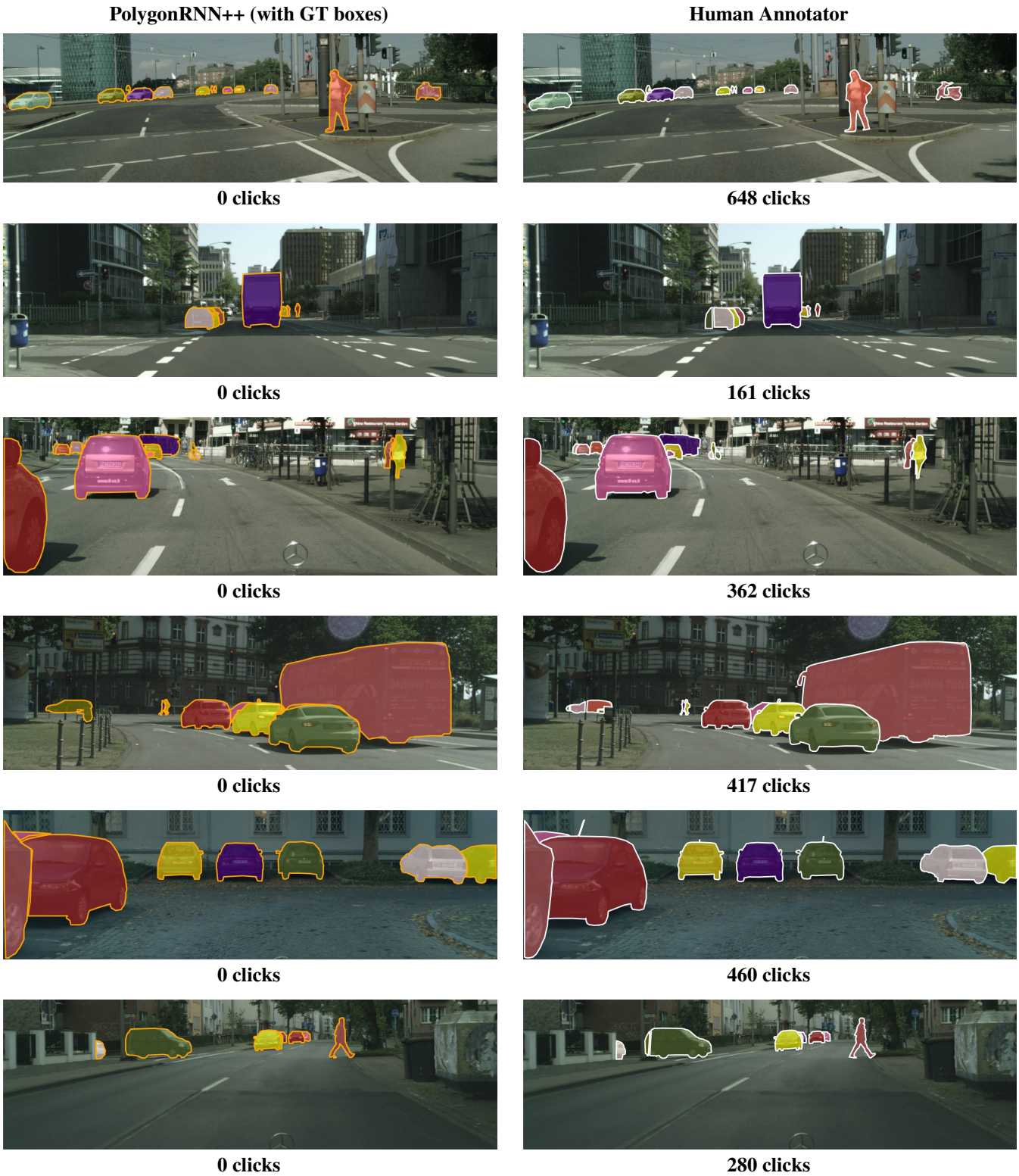


Figure 3: **Automatic mode on Cityscapes dataset**: Qualitative comparison between a human annotator vs PolygonRNN++ in automatic mode on Cityscapes. This model exploits GGNN to output a polygon at a higher resolution. Note that our model relies on bounding boxes.

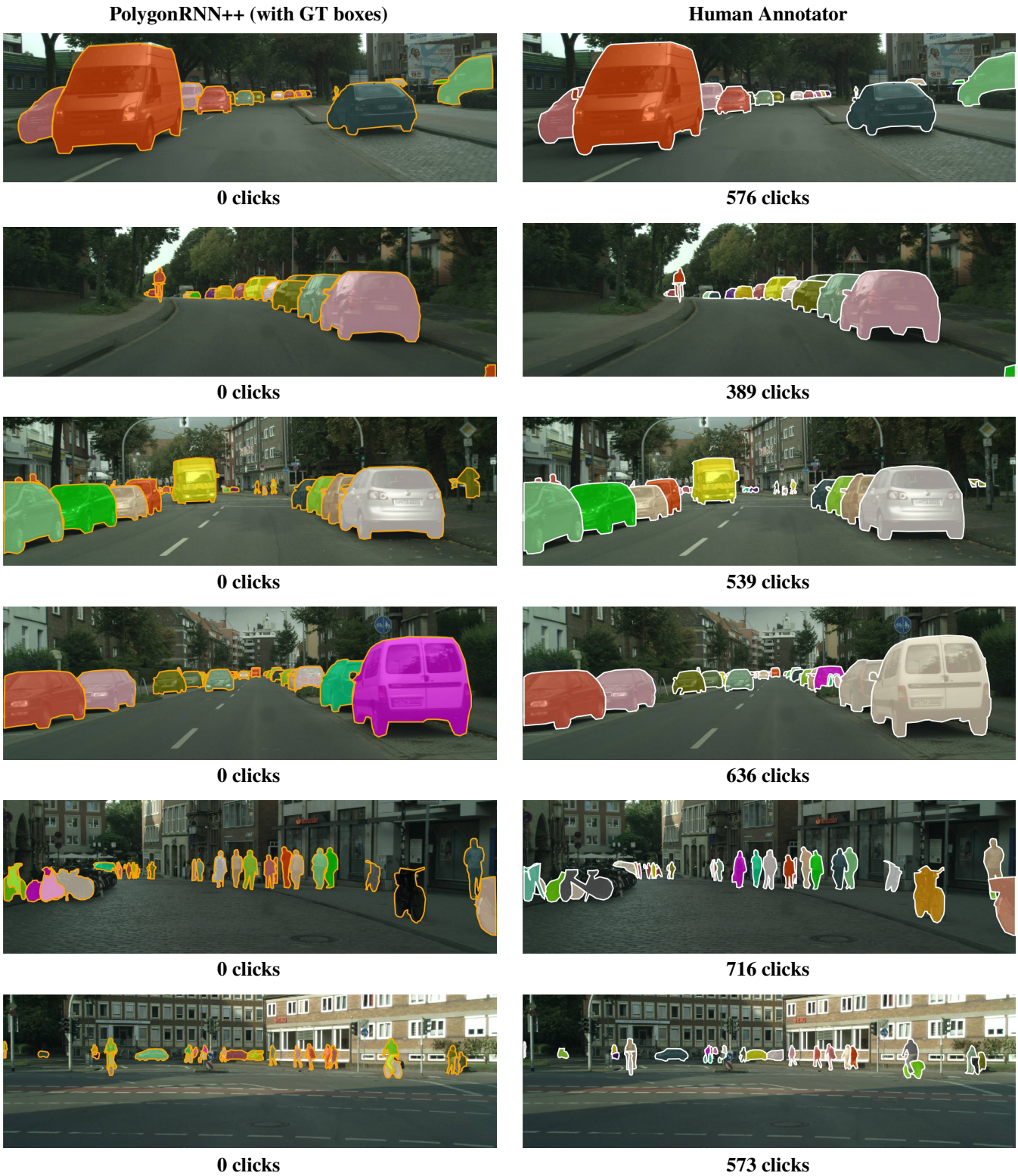
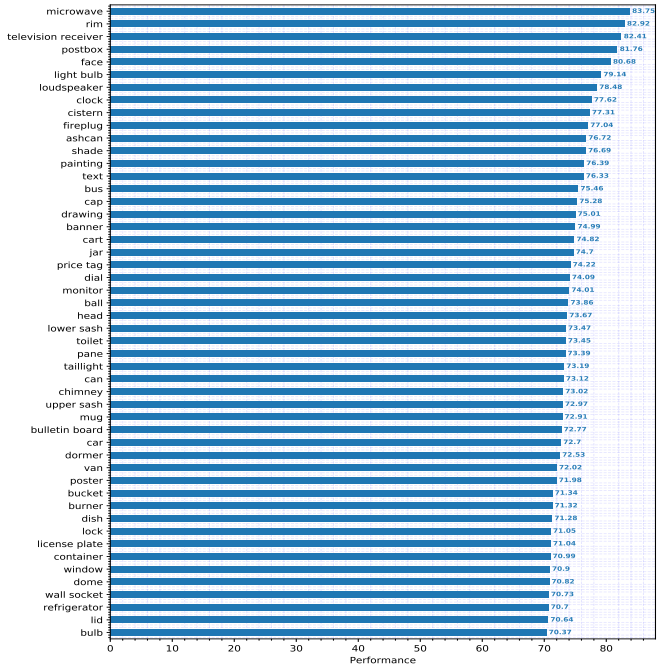


Figure 4: **Automatic mode on Cityscapes dataset**: Qualitative comparison between a human annotator vs PolygonRNN++ in automatic mode on Cityscapes. This model exploits GGNN to output a polygon at a higher resolution. Note that our model relies on ground-truth bounding boxes.

Figure 5: **Full-image instance segmentation**: Qualitative results of full image prediction using Polygon-RNN++ with boxes from Faster-RCNN



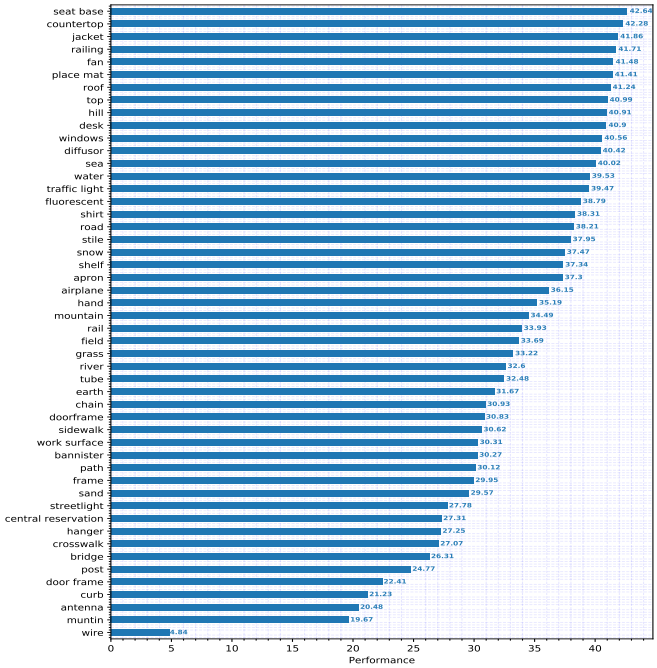


Figure 6: **Automatic mode on ADE val**: Performance of PolygonRNN++ **without fine-tuning** on ADE validation set. Only categories with more than 20 instances are shown. **Left:** Top 50 categories in terms of IoU performance. **Right:** Bottom 50 categories in terms of IoU performance.

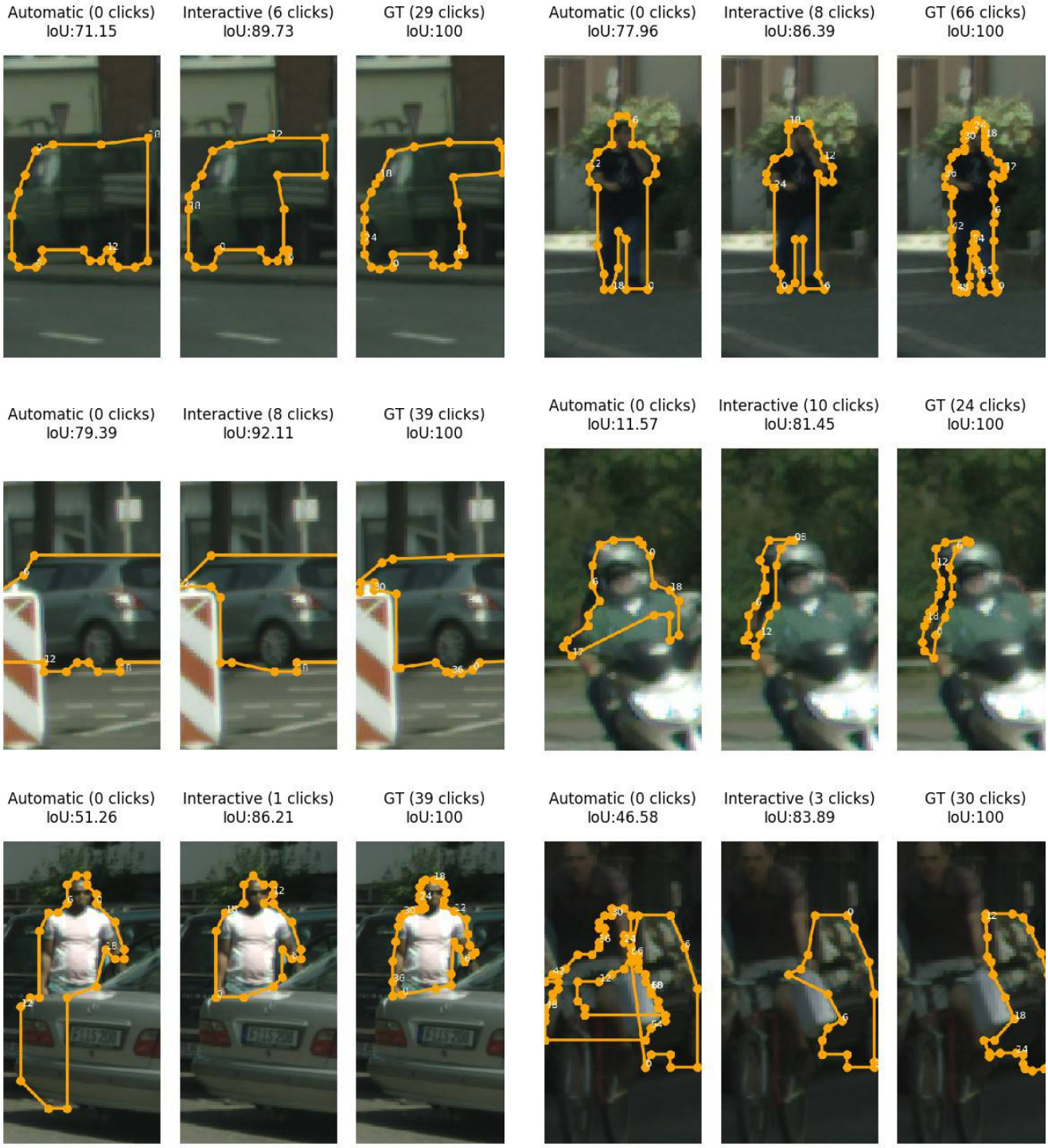


Figure 7: **Interactive mode on the Cityscapes dataset**. Here we show only predicted and corrected vertices at the 28×28 resolution (no GGNN is used here). Notice a major failure case in automatic mode in the bottom right example (where self-intersection occurs), which however gets quickly corrected. Note that the annotator is simulated (we correct a vertex if it deviates from the ground-truth vertex by a threshold T).

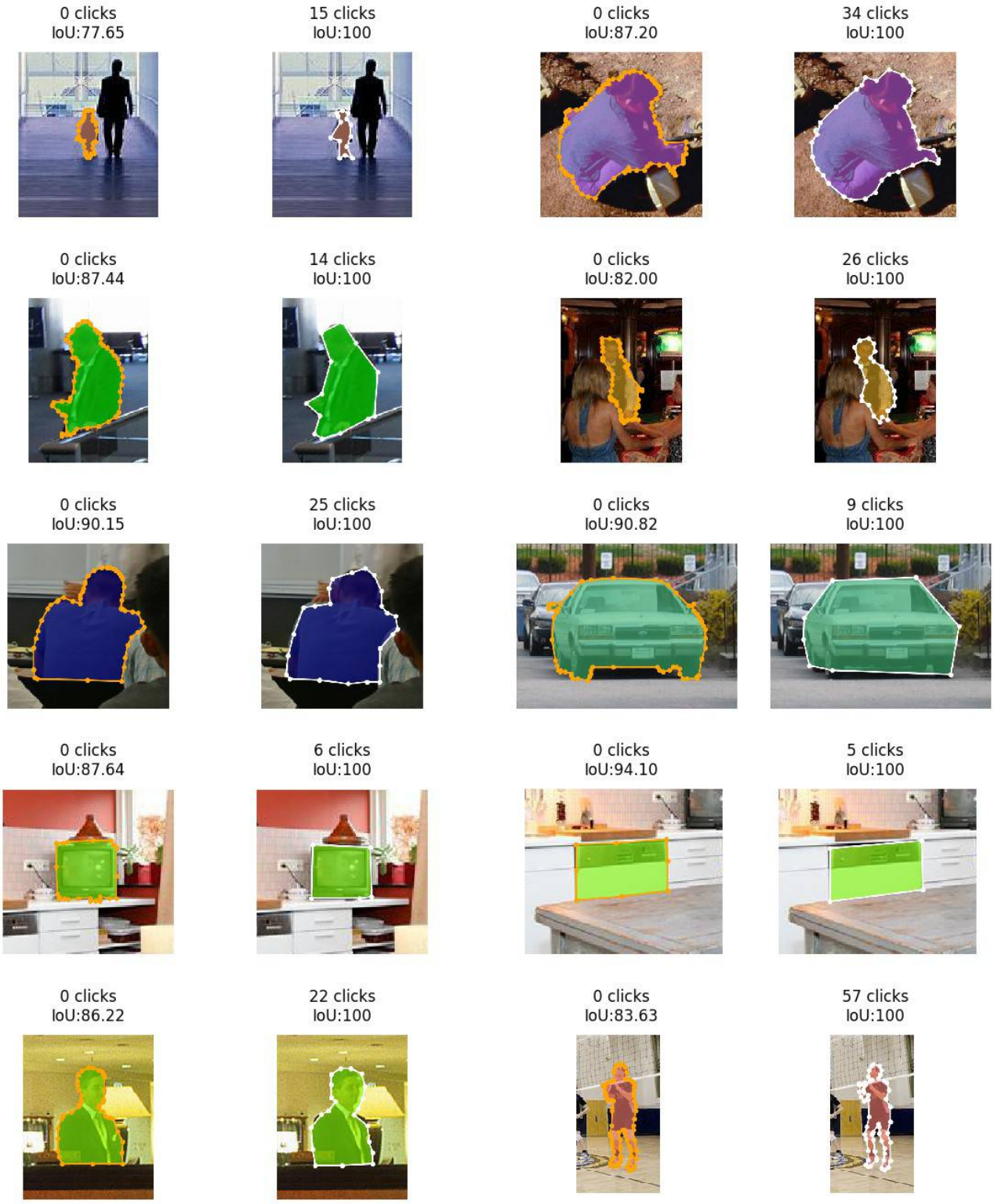


Figure 8: **Cross-domain without fine-tuning: trained on Cityscapes → tested on ADE20k, automatic mode**. Qualitative comparison between a human annotator vs PolygonRNN++ in automatic mode in ADE20K **without fine-tuning**. Note that our model relies on bounding boxes. Notice also that in some cases our predictions achieve a much higher level of detail than human provided annotations.

Figure 9: **Cross-domain without fine-tuning: trained on Cityscapes → tested on Rooftop-Aerial, automatic mode**. Qualitative comparison between a human annotator and PolygonRNN++ in automatic mode in the Rooftop-Aerial dataset **without fine-tuning**. Note that our model relies on ground-truth bounding boxes.

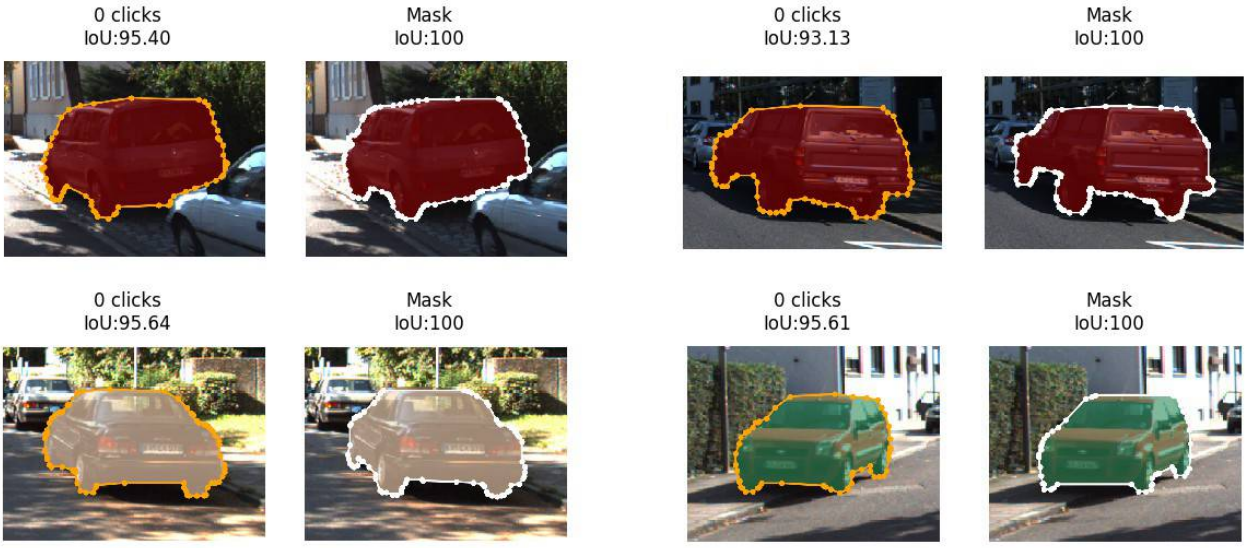


Figure 10: **Cross-domain without fine-tuning: trained on Cityscapes → tested on KITTI, automatic mode**. Qualitative comparison between a human annotator and PolygonRNN++ in automatic mode in KITTI **without fine-tuning**. Note that our model relies on ground-truth bounding boxes.

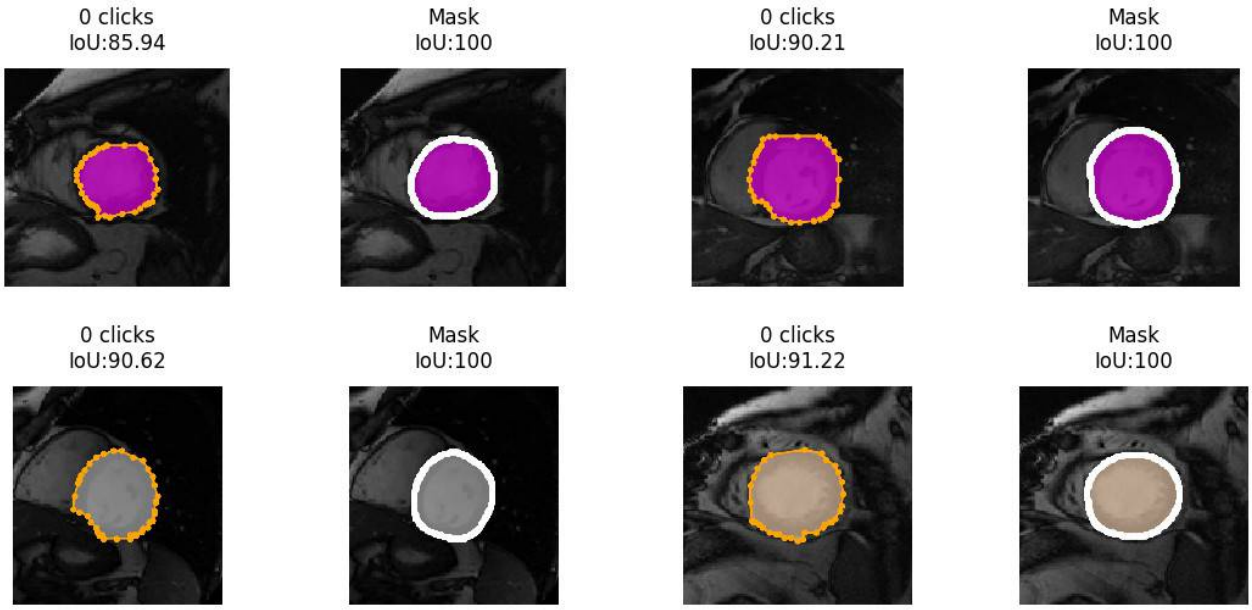


Figure 11: **Cross-domain without fine-tuning: trained on Cityscapes → tested on Cardiac MR, automatic mode**. Qualitative comparison of ground-truth masks vs PolygonRNN++ in automatic mode in the Cardiac MR dataset [5, 7] **without fine-tuning**. Note that our model relies on bounding boxes.

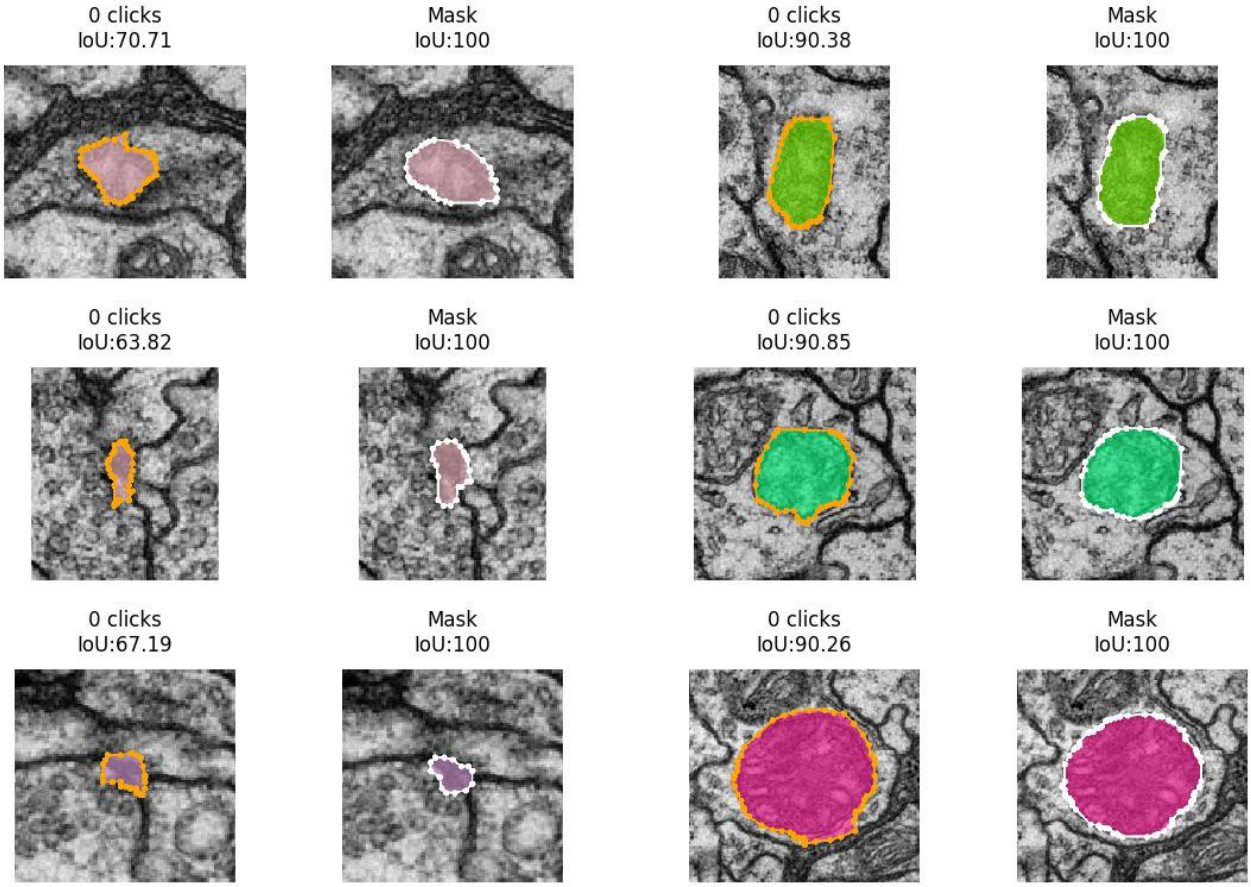


Figure 12: **Cross-domain without fine-tuning: trained on Cityscapes → tested on ssTEM, automatic mode**. Qualitative comparison of ground-truth masks vs PolygonRNN++ in automatic mode in the ssTEM dataset[3] **without fine-tuning**. Note that our model relies on ground-truth bounding boxes.

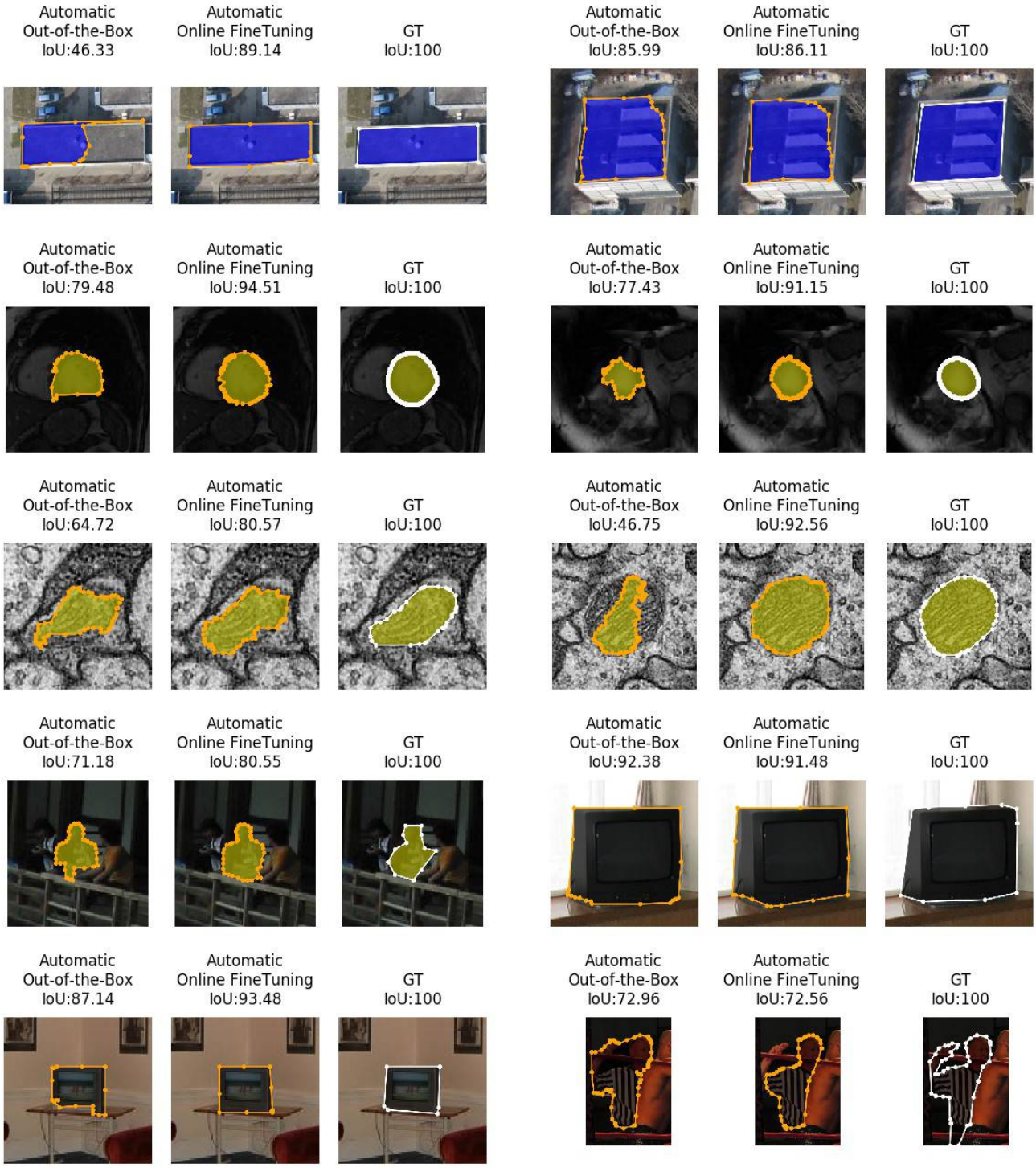


Figure 13: **Cross-domain with fine-tuning, automatic mode**. Qualitative results on different out-of-domain datasets after using the proposed Online Fine-tuning algorithm.

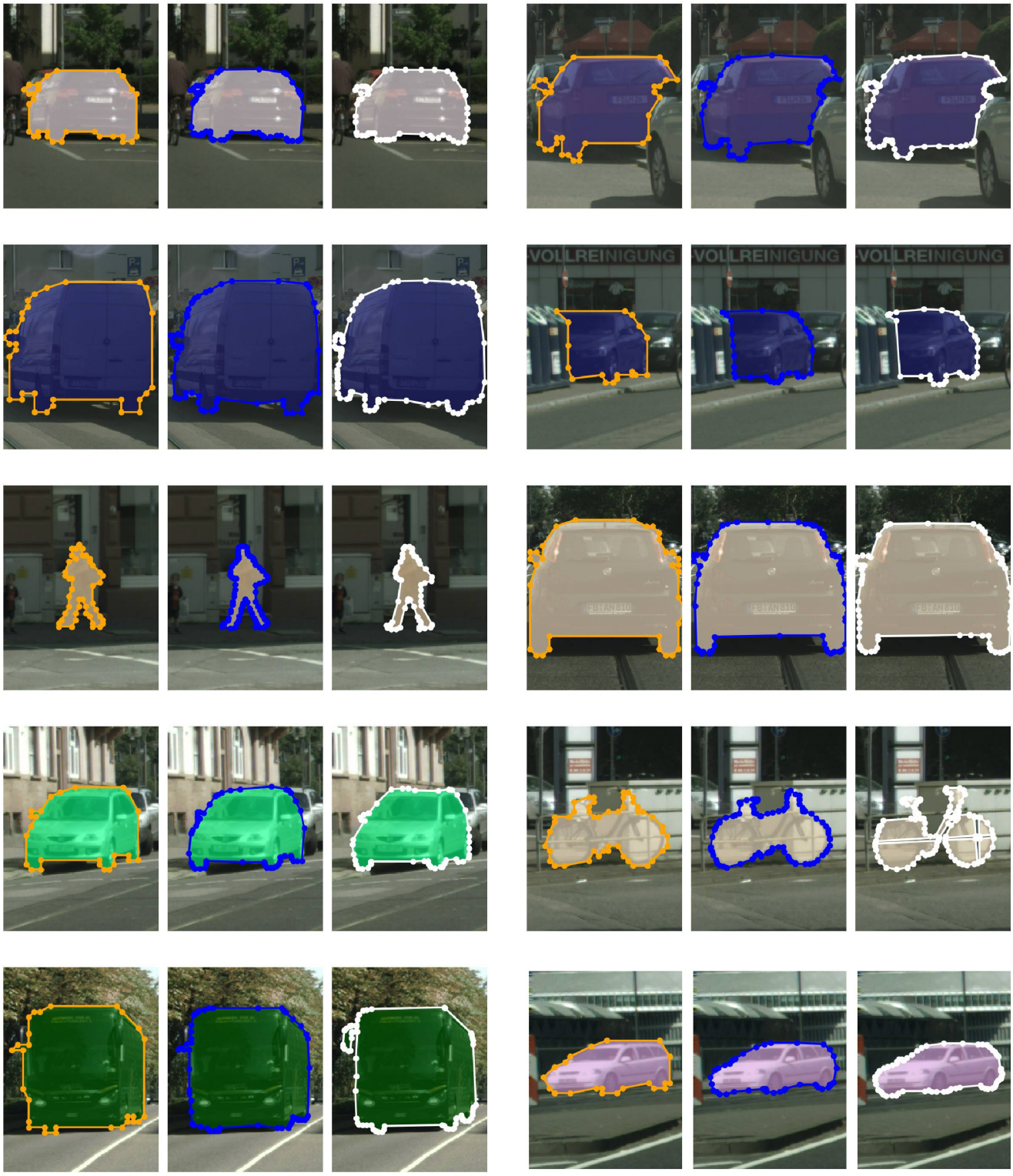


Figure 14: **Upscaling with the GGNN:** Performance of PolygonRNN++ after and before upscaling the output polygon with GGNN **Left:** Output of PolygonRNN++ before upscaling **Center:** Output of PolygonRNN++ after GGNN **Right:** GT