Re:PolyWorld - A Graph Neural Network for Polygonal Scene Parsing

Supplementary Material

In the following pages, we report the hyperparameters of the model and present additional qualitative and quantitative results of Re:PolyWorld for building extraction, floorplan reconstruction, and wireframe parsing. Specifically, we present a comprehensive evaluation of our approach, which includes a comparative analysis with state-of-the-art methods, such as Frame Field Learning (FFL) [3] and PolyWorld [13] for building extraction, Holistic Edge Attention Transformer (HEAT) [5] for floorplan reconstruction, and Holistically Attracted Wireframe Parsing (HAWPv2) [10]. Moreover, we provide detailed quantitative tables, examples of failure cases, as well as an ablation study to evaluate the proposed edge-aware GNN.

Architecture and Hyperparameters

Re:PolyWorld utilizes a Residual Recurrent U-Net model [1] as feature extractor. The feature map $\mathbf{F}$ generated by the backbone, and the vertex descriptors, have $D = 64$ channels. We linearly sample $M = 32$ descriptors between each pair of detected vertices (from vertex i to vertex j). Λ reduces the size of the descriptor by performing a linear projection $\mathbb{R}^{D=64} \rightarrow \mathbb{R}^{D^-=16}$. MLP_{edge} performs the mapping $\mathbb{R}^{MD^-=512} \rightarrow \mathbb{R}^{D_{edge}=128}$ by receiving the concatenation of the reduced descriptors as input (representing the path from vertex i to vertex j). In all our experiments and applications, we used $L = 4$ edge-aware GNN layers with 4 heads each. For the positional refinement, the offset is generated by limiting the maximum permitted value to 5% of the image size by using a tanh activation function. We trained Re:PolyWorld using the same loss hyperparameters described in [13].

Certain variables have been selected based on the specific demands of the application:

- **Building extraction** on the CrowdAI dataset [6] is performed by extracting $N = 256$ vertex candidates from each image and employing a Non-Maximum Suppression (NMS) layer with kernel size of 3, since building vertices can be located in close proximity to each other. To prevent pooling issues in the backbone, we upsampled the images in the data set from its original size of 300×300 pixels to 320×320 pixels.
- **Floorplan** and **architectural reconstruction** on the Structured3D [11] and HEAT [5] data set are performed by detecting $N = 96$ vertices. In this case, a kernel size of 7 is used for the NMS layer.
- **Wireframe parsing** is realized by detecting $N = 320$ vertices and filtering the vertex detection map with the NMS layer with a kernel size of 11. For this application, learning the refinement offset for the vertices is not beneficial (since we aim to detect lines), therefore the model is just trained by minimizing $\mathcal{L}_{det}$ and $\mathcal{L}_{match}$.

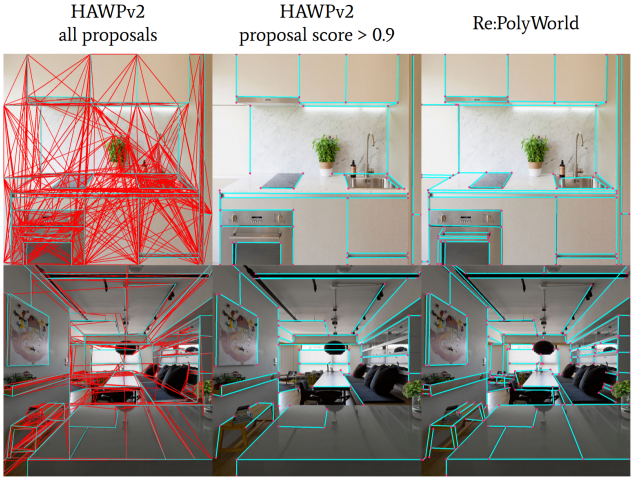

Figure 1. Wireframe parsing results. Columns from left to right: HAWPv2 [10] wireframe proposals (the color gradient encodes the line scores from 0-red to 1-cyan), HAWPv2 filtered wireframes with a threshold of 0.9. Re:PolyWorld wireframes predicted by solving the linear sum assignment.

Additional results

Building extraction: Figure 3 shows a qualitative comparison of Re:PolyWorld, Frame Field Learning (FFL) [3], and PolyWorld [13]. The results of these three distinct polygon extraction approaches are illustrated on various scenes from the CrowdAI [6] test set. It can be observed that Re:PolyWorld, which employs a smaller number of vertices than FFL, produces more regular contours. Furthermore, our approach provides a high degree of generalization on complex and uncommon building shapes, precisely detecting and connecting all the building corners, even in the presence of severe occlusions. Visually, the results of Re:PolyWorld and PolyWorld are comparable, however, quantitative results (Table 1) show that Re:PolyWorld extracts building footprints more accurately, achieving higher semantic and instance segmentation scores, due to the joint analysis of vertices and edges. Additional results of Re:PolyWorld on the CrowdAI data set are shown in Figure 4.

Floorplan reconstruction: Additional evaluation results

on the Structured3D [11] data set are shown in Figure 5. It can be seen that Re:PolyWorld generates more accurate polygons compared to HEAT [5] and also achieves higher recall and precision scores (Table 3). Figure 6 shows additional results produced by Re:PolyWorld for randomly sampled test images of the Structured3D data set.

Wireframe parsing: We provide additional results for wireframe parsing on indoor and outdoor scenes in Figure, 7 and Figure 8, respectively. Unlike HAWPv1 [9] and HAWPv2 [10], Re:PolyWorld does not depend on filtering proposals with low confidence scores. Instead, it addresses an optimal transport problem to generate proposals. This results in a considerably lower number of detected wireframes compared to HAWPv2, as shown in Figure 1. Due to this, Re:PolyWorld does not excel HAWP in terms of Structural Average Precision (sAP), but achieves state-of-the-art Recall performance (Table 4).

Ablation study

We conduct supplementary experiments to evaluate the performance contribution provided by the proposed Edge-Aware Graph Neural Network (EA-GNN). By exploiting the combination of vertices and edges, Re:PolyWorld achieves higher segmentation scores in the building extraction task compared to PolyWorld [13], as shown in Table 1. In this experiment, we use $K = 1$ vertex instances, since PolyWorld is intrinsically unable to produce more than one connection per vertex.

We also evaluate Re:PolyWorld on the Wireframe Parsing dataset [4] by either using or discarding the edge information. This experiment is conducted by training Re:PolyWorld using the original GNN proposed in PolyWorld, thereby eliminating the component of combined vertex and edge information. The results show a higher number of detected lines for our proposed edge-aware Re:PolyWorld approach, along with boosted scores in terms of Structural Average Precision (sAP) and Recall (Table 2). Moreover, the original vertex-only model generates several wrong lines, as shown in Figure 9.

Failure cases

Despite experimental evidence demonstrating that Re:PolyWorld generates a strong set of vertices and robust connections, it is interesting to highlight instances of failure due to the vertex detection network. Figure 2 shows some Re:PolyWorld predictions exhibiting artifacts, resulting from the incomplete identification of vertices by the backbone network. Consequently, the predicted polygons are missing a corner and a significant portion of the object.

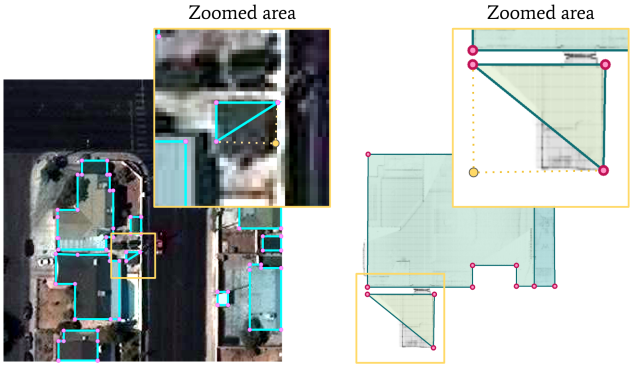

Figure 2. Re:PolyWorld failure cases. Generated polygons lack a corner due to the failure of the backbone to detect a vertex, which is highlighted in yellow in these examples.

Method	edge-aware	offset	AP	AR	IoU	C-IoU	MTA
PolyWorld [13]	no	off	58.7	71.7	89.9	86.9	35.0
		on	63.3	75.4	91.3	88.2	32.9
Re:PolyWorld	yes	off	61.7	74.0	90.6	88.1	35.1
		on	67.2	78.6	92.2	89.7	31.9

Table 1. Ablation study with respect to the proposed edge-aware attention mechanism. Results on the CrowdAI [6] test computed for different configurations of PolyWorld [13] and Re:PolyWorld. For fairness, Re:PolyWorld is trained using $K = 1$ vertex instances, since PolyWorld is intrinsically incapable of generating more than one connection per vertex.

Re:PolyWorld	# lines	sAP^5	sAP^{10}	sAP^{15}	R^5	R^{10}	R^{15}	# GT lines
GNN	66	35.9	40.4	42.7	53.7	57.0	58.6	74.2
EA-GNN	83	44.9	50.2	52.7	60.8	64.6	67.1	

Table 2. Ablation study with respect to the proposed edge-aware attention mechanism. Results obtained on the Wireframe Parsing [4] dataset by Re:PolyWorld in two configurations: by using the vertex-only GNN of PolyWorld [13], and by exploiting the proposed Edge-Aware GNN (EA-GNN).

	Room		Corner		Angle	
Mehtod	Prec	Recall	Prec	Recall	Prec	Recall
HAWP [10]	0.78	0.88	0.66	0.77	0.6	0.7
LETR [8]	0.94	0.9	0.8	0.78	0.72	0.71
HEAT [5]	0.97	0.94	0.82	0.83	0.78	0.79
Floor-SP [2]	0.89	0.88	0.81	0.73	0.8	0.72
MonteFl. [7]	0.96	0.94	0.89	0.77	0.86	0.75
Re:PolyWorld (offset off)	0.99	0.97	0.81	0.86	0.75	0.8
Re:PolyWorld (offset on)	0.99	0.97	0.81	0.86	0.77	0.82

Table 3. Floorplan reconstruction results on the Structured3D dataset [11]. The results of PolyWorld are calculated by discarding the refinement offsets (offset off), and refining the vertex positions (offset on).

References

[1] Md Zahangir Alom, Chris Yakopcic, Mahmudul Hasan, Tarek M Taha, and Vijayan K Asari. Recurrent residual u-net for medical image segmentation. *Journal of Medical Imaging*, 6(1):014006, 2019. 1

Method	# lines	sAP^5	sAP^{10}	sAP^{15}	R^5	R^{10}	R^{15}	# GT lines
HAWPv2@0.0	959	65.7	69.7	71.3	59.5	63.1	64.5	
HAWPv2@0.5	121	61.8	65.2	66.7	59.5	63.1	64.5	
HAWPv2@0.8	72	55.4	58.2	59.4	59.5	63.1	64.5	74.2
HAWPv2@0.9	53	48.9	51.1	52.0	54.2	55.9	56.6	
Re:PolyWorld	83	44.9	50.2	52.7	60.8	64.6	67.1	

Table 4. Results on the Wireframe Parsing dataset [12]. The line proposals of HAWPv2 [10] are filtered with different confidence thresholds (0, 0.5, 0.8, 0.9). "# lines" indicates the average number of detected lines. "# GT lines" is the average number of ground truth lines. The number of lines of Re:PolyWorld is the average number of lines generated by solving the linear sum assignment problem.

[2] Jiacheng Chen, Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2661–2670, 2019. 2

[3] Nicolas Girard, Dmitriy Smirnov, Justin Solomon, and Yuliya Tarabalka. Polygonal building extraction by frame field learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5891–5900, 2021. 1, 4

[4] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 626–635, 2018. 2, 10

[5] Yasutaka Furukawa Jiacheng Chen, Yiming Qian. Heat: Holistic edge attention transformer for structured reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 6

[6] Sharada Prasanna Mohanty, Jakub Czakon, Kamil A Kaczmarek, Andrzej Pyskir, Piotr Tarasiewicz, Saket Kunwar, Janick Rohrbach, Dave Luo, Manjunath Prasad, Sascha Fleer, et al. Deep learning for understanding satellite imagery: An experimental survey. *Frontiers in Artificial Intelligence*, 3, 2020. 1, 2, 4, 5

[7] Sinisa Stekovic, Mahdi Rad, Friedrich Fraundorfer, and Vincent Lepetit. Montefloor: Extending mcts for reconstructing accurate large-scale floor plans. *arXiv preprint arXiv:2103.11161*, 2021. 2

[8] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. Line segment detection using transformers without edges. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4257–4266, 2021. 2

[9] Nan Xue, Tianfu Wu, Song Bai, Fudong Wang, Gui-Song Xia, Liangpei Zhang, and Philip HS Torr. Holistically-attracted wireframe parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2788–2797, 2020. 2

[10] Nan Xue, Tianfu Wu, Song Bai, Fu-Dong Wang, Gui-Song Xia, Liangpei Zhang, and Philip HS Torr. Holistically-attracted wireframe parsing: From supervised to self-supervised learning. *arXiv preprint arXiv:2210.12971*, 2022. 1, 2, 3, 8, 9

[11] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *European Conference on Computer Vision*, pages 519–535. Springer, 2020. 1, 2, 6, 7

[12] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 962–971, 2019. 3, 8, 9

[13] Stefano Zorzi, Shabab Bazrafkan, Stefan Habenschuss, and Friedrich Fraundorfer. Polyworld: Polygonal building extraction with graph neural networks in satellite images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1848–1857, 2022. 1, 2, 4, 10

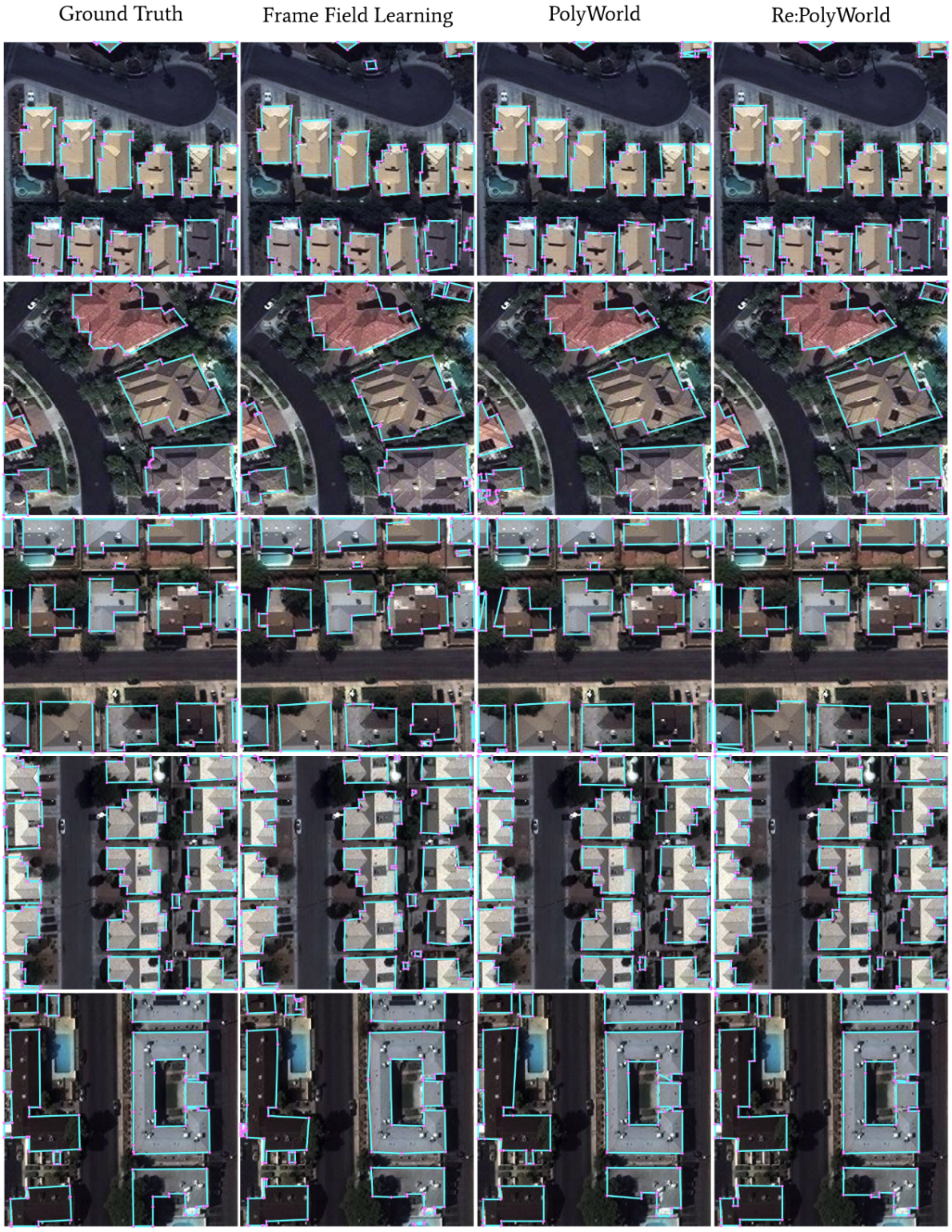

Figure 3. Building extraction results on the CrowdAI [6] test dataset. Columns from left to right: ground truth annotations, Frame Field Learning [3] results, PolyWorld [13] results, Re:PolyWorld results. PolyWorld and Re:PolyWorld annotations are obtained using the positional refinement offset. Please zoom in for better view.

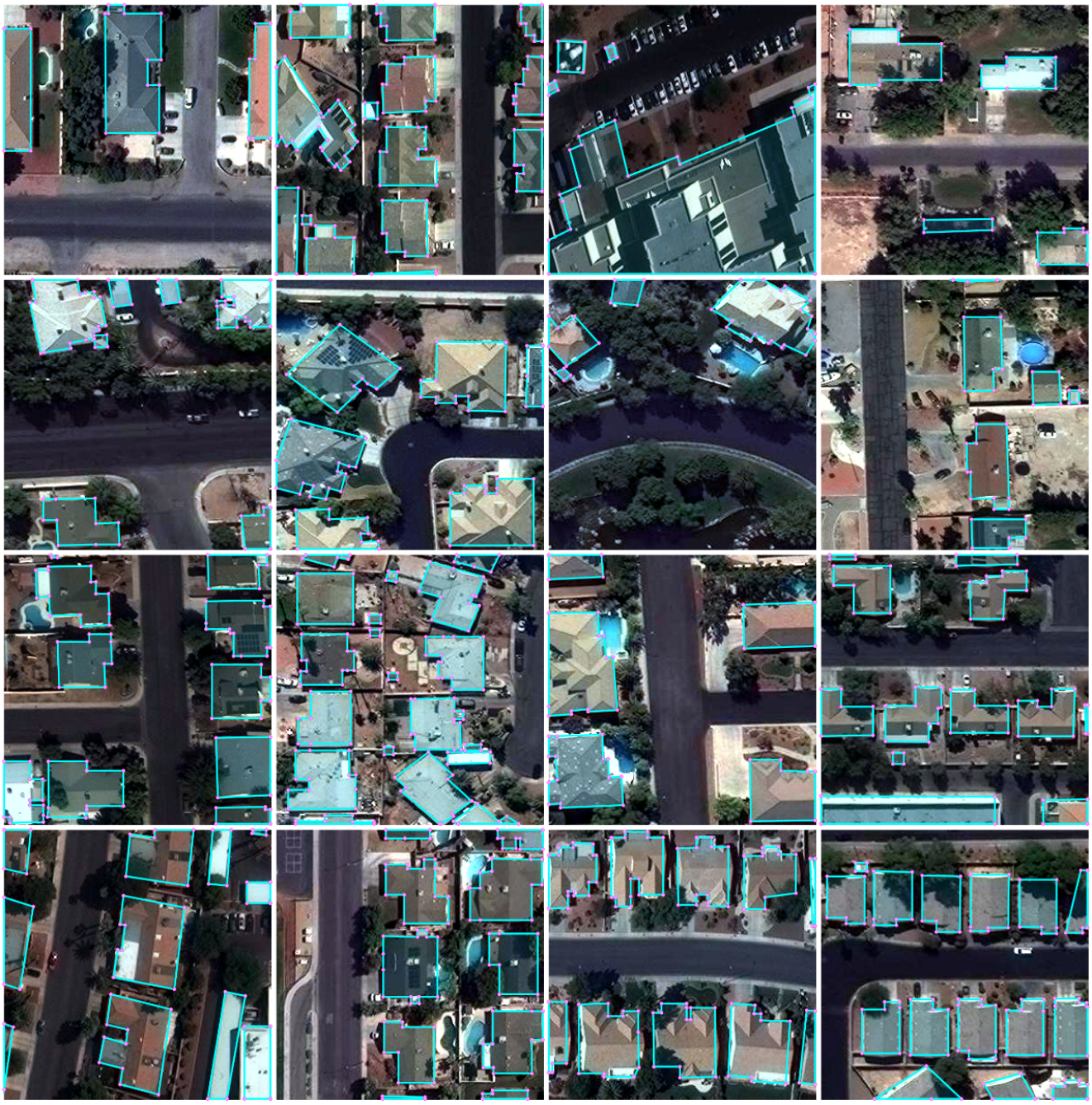

Figure 4. Building extraction results of Re:PolyWorld on **randomly sampled images** from the CrowdAI [6] test dataset. Please zoom in for better view.

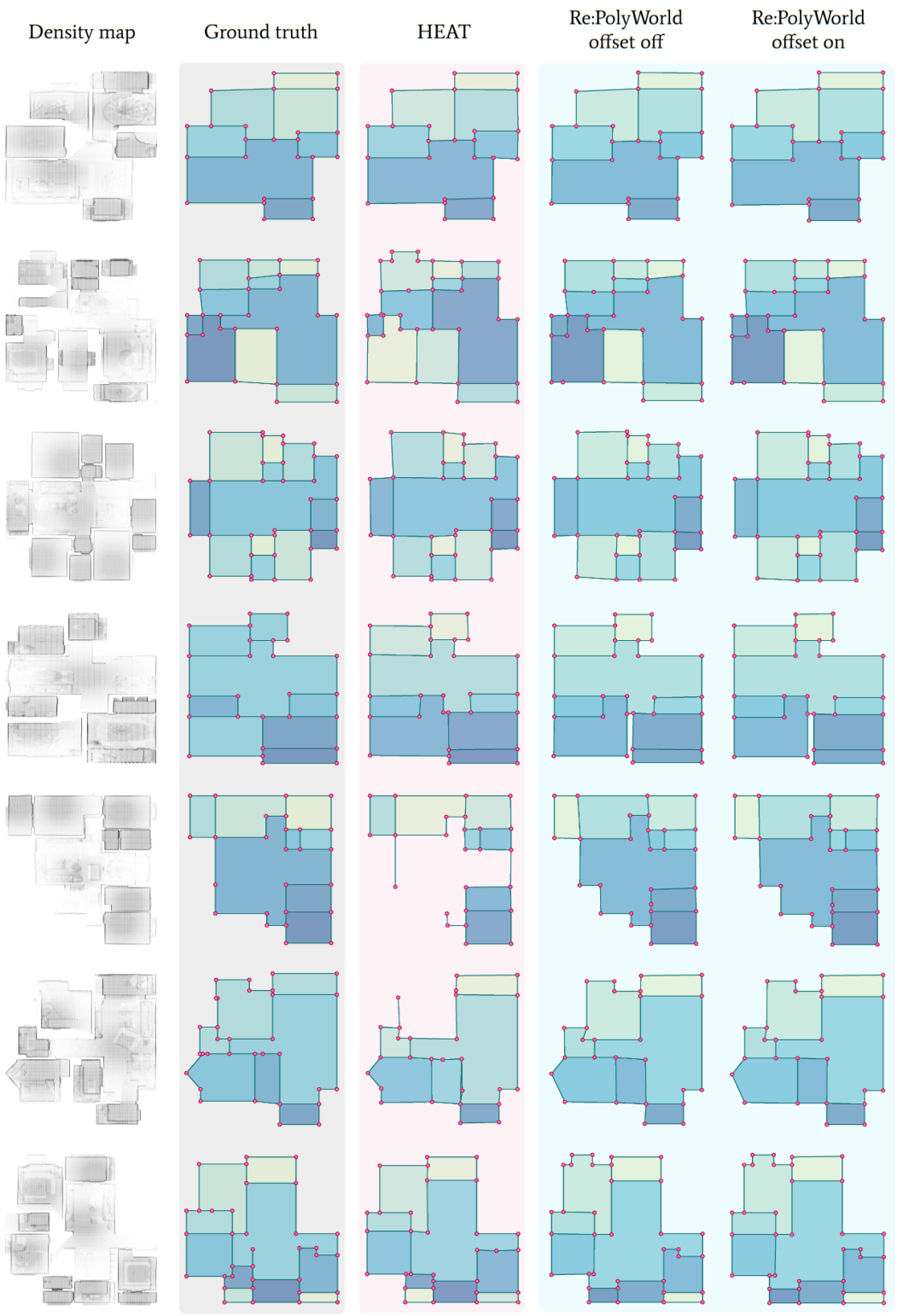

Figure 5. Floorplan reconstruction results on the Structured3D [11] test dataset. Columns from left to right: input density map, ground truth annotations, HEAT [5] results, Re:PolyWorld discarding the positional refinement offset, Re:PolyWorld full method. Please zoom in for better view.

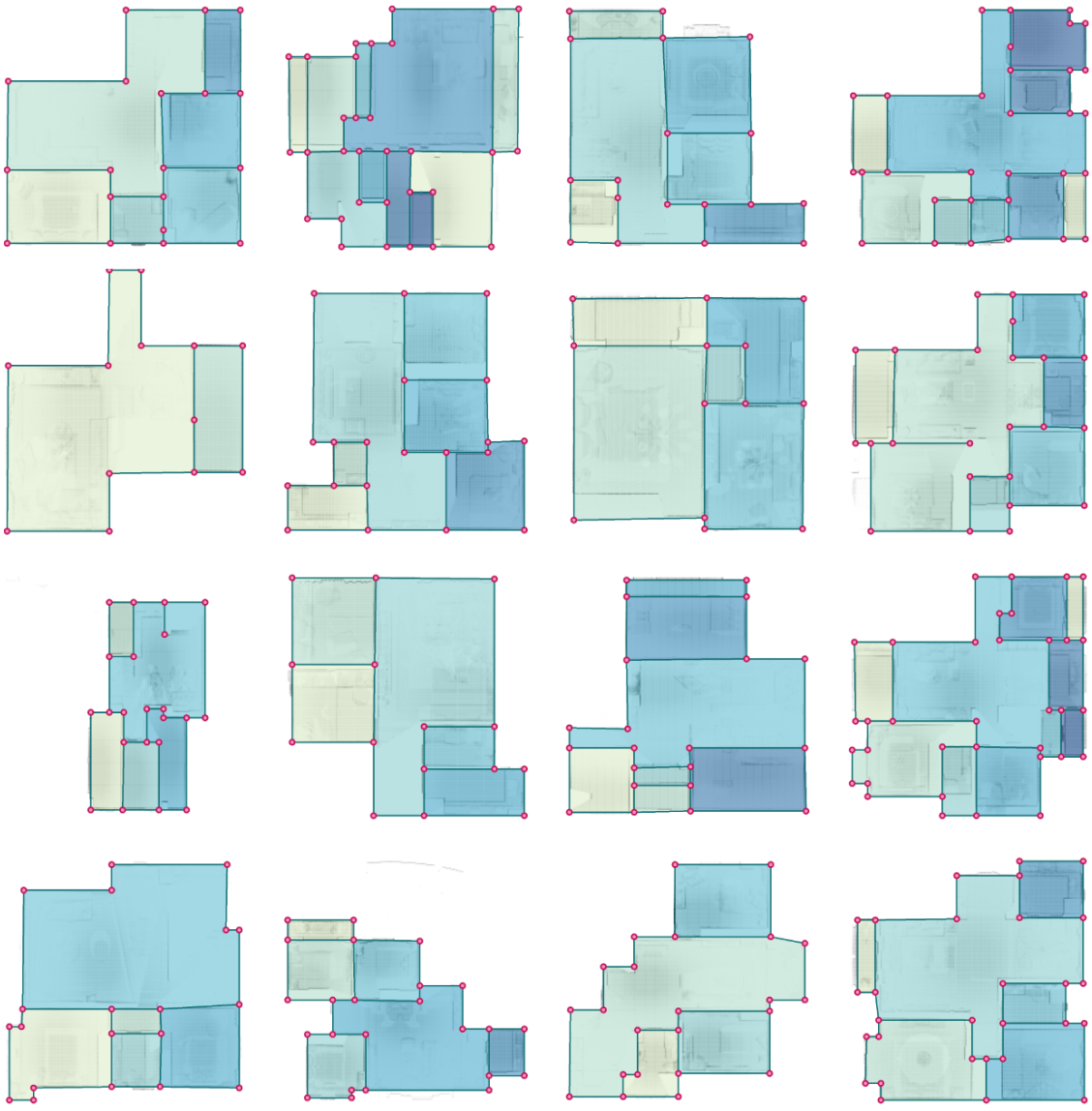

Figure 6. Floorplan reconstruction results of Re:PolyWorld on **randomly sampled images** from the Structured3D [11] test dataset. Polygons generated by Re:PolyWorld overlaid to the input density map. Please zoom in for better view.

Figure 7. Wireframe parsing results on the Wireframe Parsing [12] test dataset. Outdoor images are selected for this figure. Columns from left to right: ground truth wireframes, HAWPv2 [10] predicted wireframes (the visualized lines have $score > 0.9$), Re:PolyWorld predicted wireframes (the visualized lines are obtained by solving the linear sum assignment problem). Please zoom in for better view.

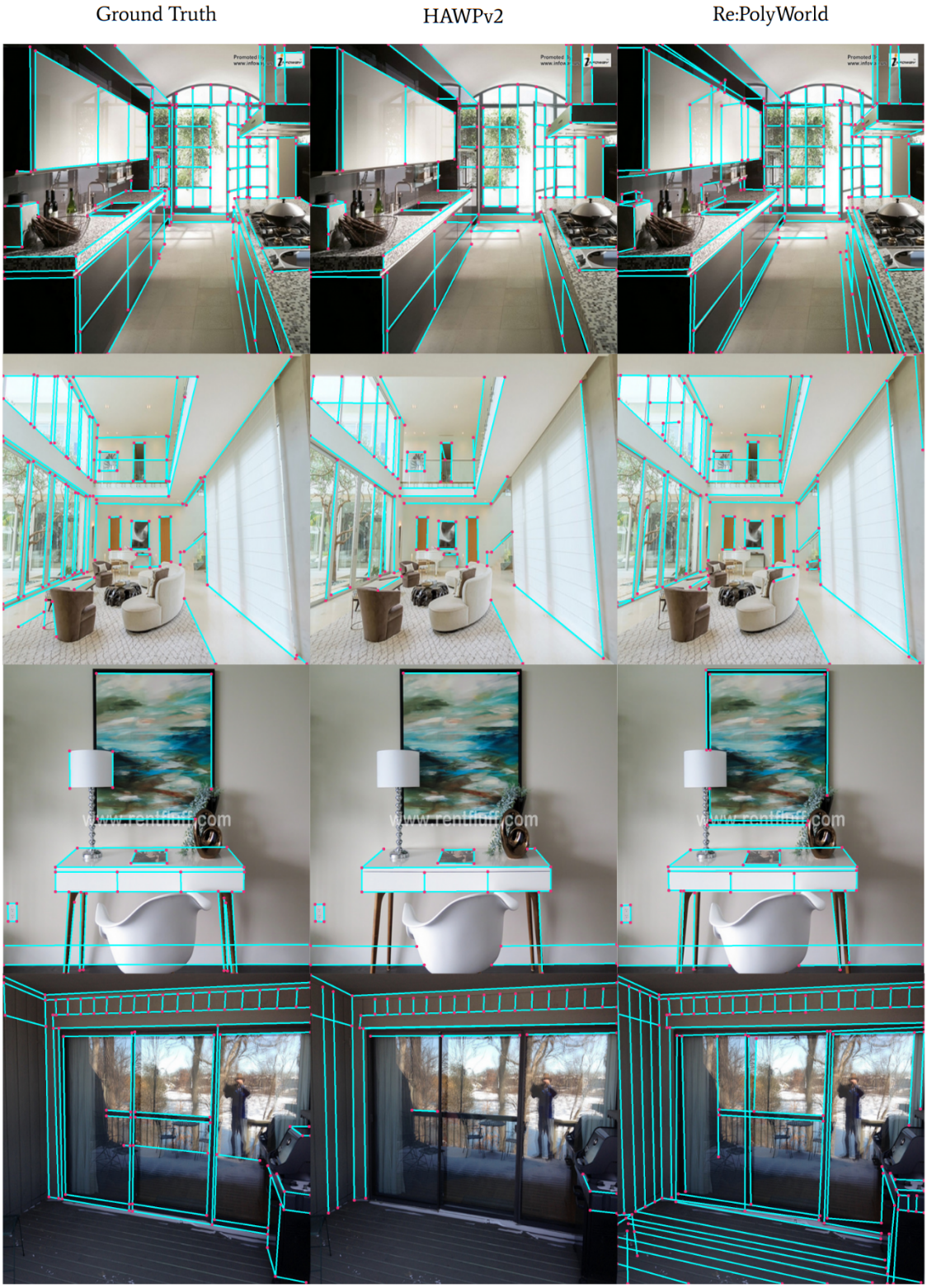

Figure 8. Wireframe parsing results on the Wireframe Parsing [12] test dataset. Indoor images are selected for this figure. Columns from left to right: ground truth wireframes, HAWPv2 [10] predicted wireframes (the visualized lines have $score > 0.9$), Re:PolyWorld predicted wireframes (the visualized lines are obtained by solving the linear sum assignment problem). Please zoom in for better view.

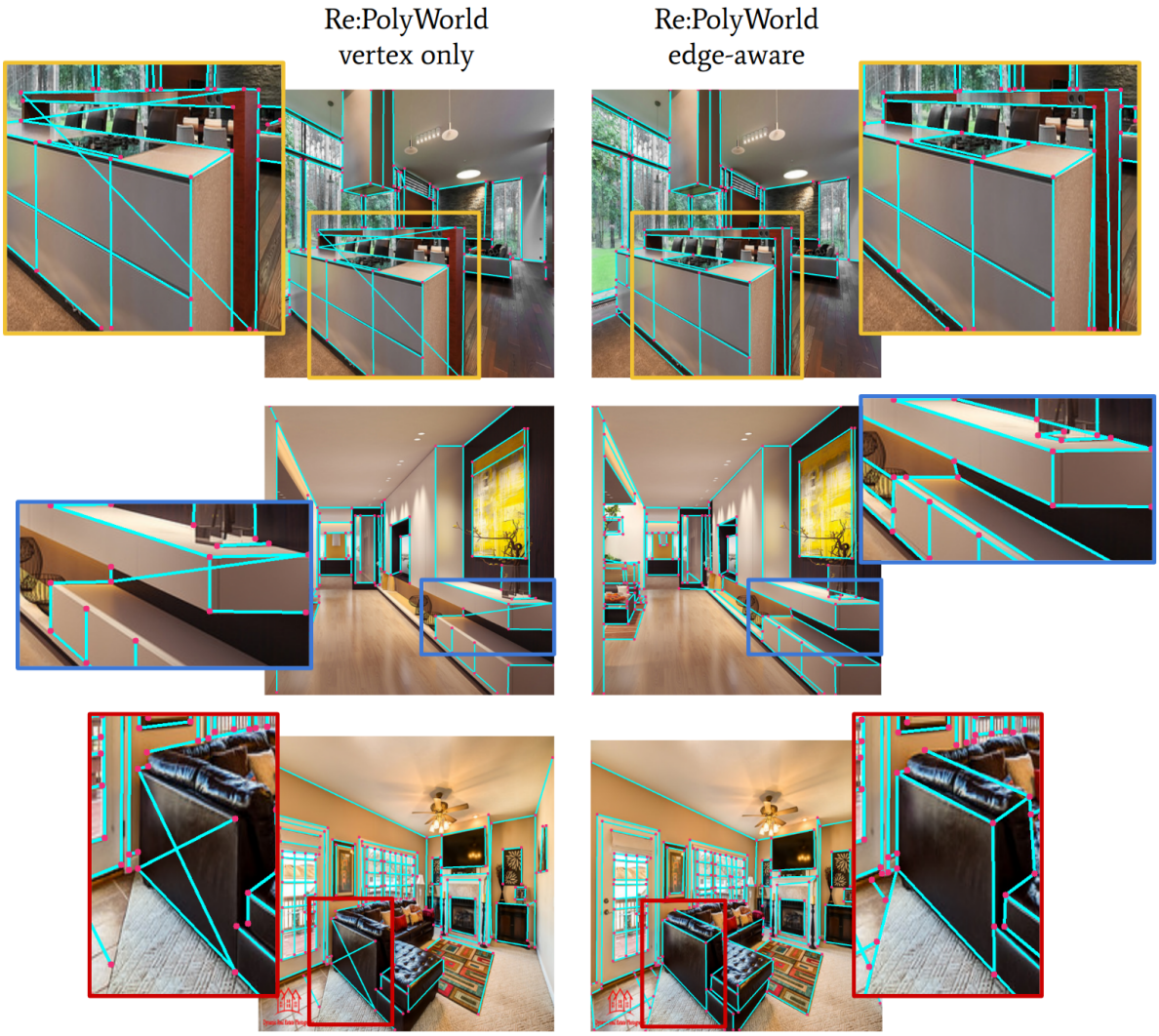

Figure 9. Results obtained in the Wireframe Parsing dataset [4]. On the left column: results obtained by Re:PolyWorld without using edge information (the model runs the GNN proposed in [13]). On the right column: results obtained by Re:PolyWorld using the proposed edge-aware GNN.