# Supplementary material:

# Deep Floor Plan Recognition using a Multi-task Network with Room-boundary-Guided Attention

## Overview

This supplementary document is composed of the following sections.

- In Section A, more visual comparisons between our method and other state-of-the-arts with post-processing are provided.
- In Section B, more visual comparisons between our method and other state-of-the-arts without postprocessing are provided.
- In Section C, limitations of our method are provided.
- In Section D, our cross-and-within-task weighted loss is analyzed.
- In Section E, the effect of our weighted loss in semantic segmentation network is analyzed.
- In Section F, more results produced from our method, including the reconstructed 3D models, are shown.

## A. Additional Results: Visual Comparisons with Postprocessing

Figures 1-4 show more visual comparisons by applying our method and other state-of-the-arts, *i.e.*, Raster-to-vector [3], DeepLabV3+ [1], and PSPNet [4], to various input floor plans. Note that, we apply postprocessing to all these methods, except the Raster-to-vector results, since its pipeline already contained a postprocessing step to connect room regions. The input floor plans in Figures 1-3 are selected from the challenging R3D dataset with irregular shapes, while the input floor plans in Figure 4 are selected from the R2V dataset. Since Raster-to-vector method will produce large missing parts on the R3D dataset, thus we only compare our method with DeepLabV3+ and PSPNet in Figures 1-3. From the results we can see that our method produced the best results compared to others, demonstrating the superiority of our method on floor plan recognition.

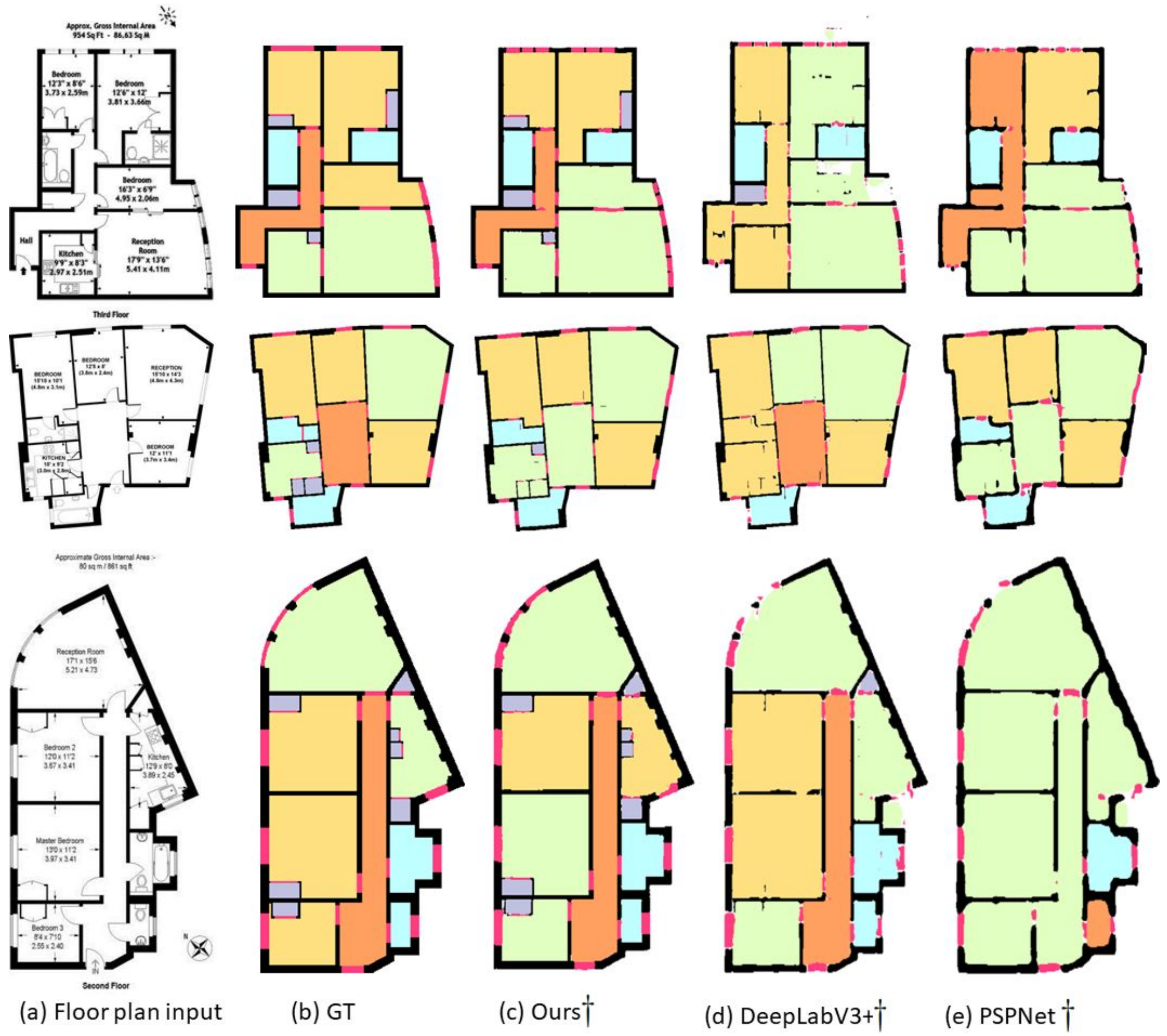

Figure 1. Additional comparison results (1/3) on the R3D dataset by applying (c) our method, (d) DeepLabV3+, and (e) PSPNet to the floor plan inputs shown in (a). GT denotes ground truth. Symbol † indicates the method with our postprocessing.

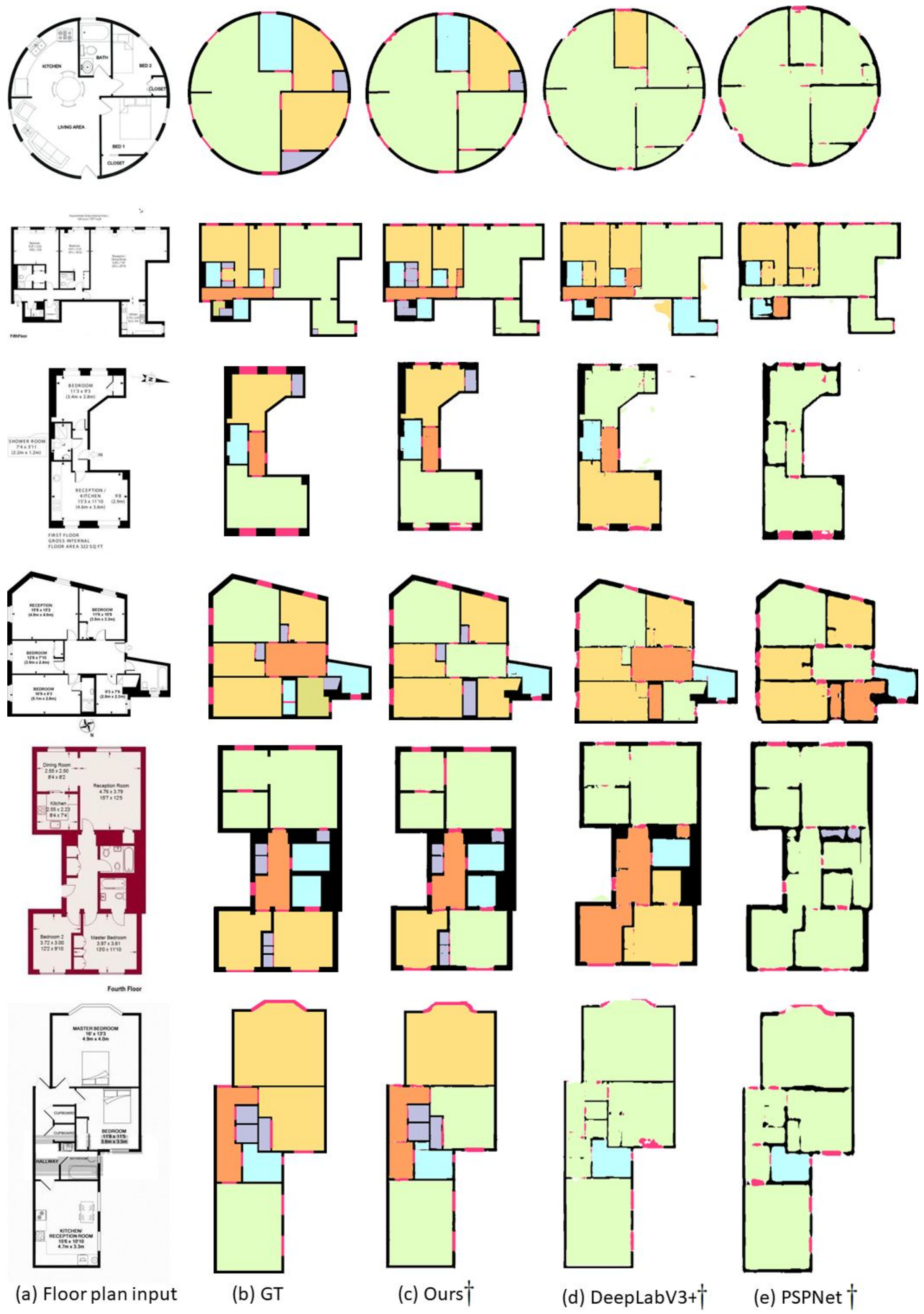

Figure 2. Additional comparison results (2/3) on the R3D dataset by applying (c) our method, (d) DeepLabV3+, and (e) PSPNet to the floor plan inputs shown in (a). GT denotes ground truth. Symbol † indicates the method with our postprocessing.

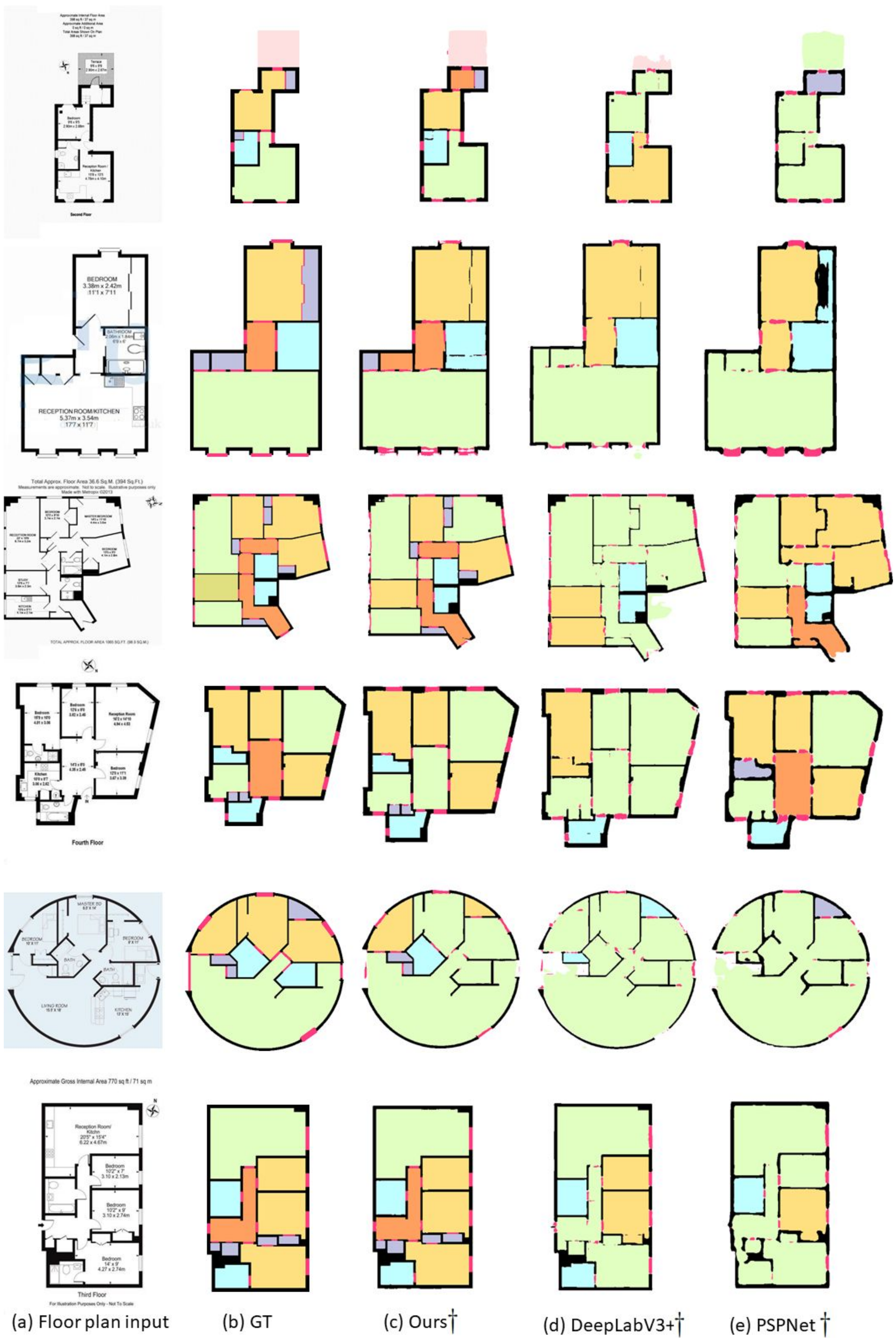

Figure 3. Additional comparison results (3/3) on the R3D dataset by applying (c) our method, (d) DeepLabV3+, and (e) PSPNet to the floor plan inputs shown in (a). GT denotes ground truth. Symbol † indicates the method with our postprocessing.

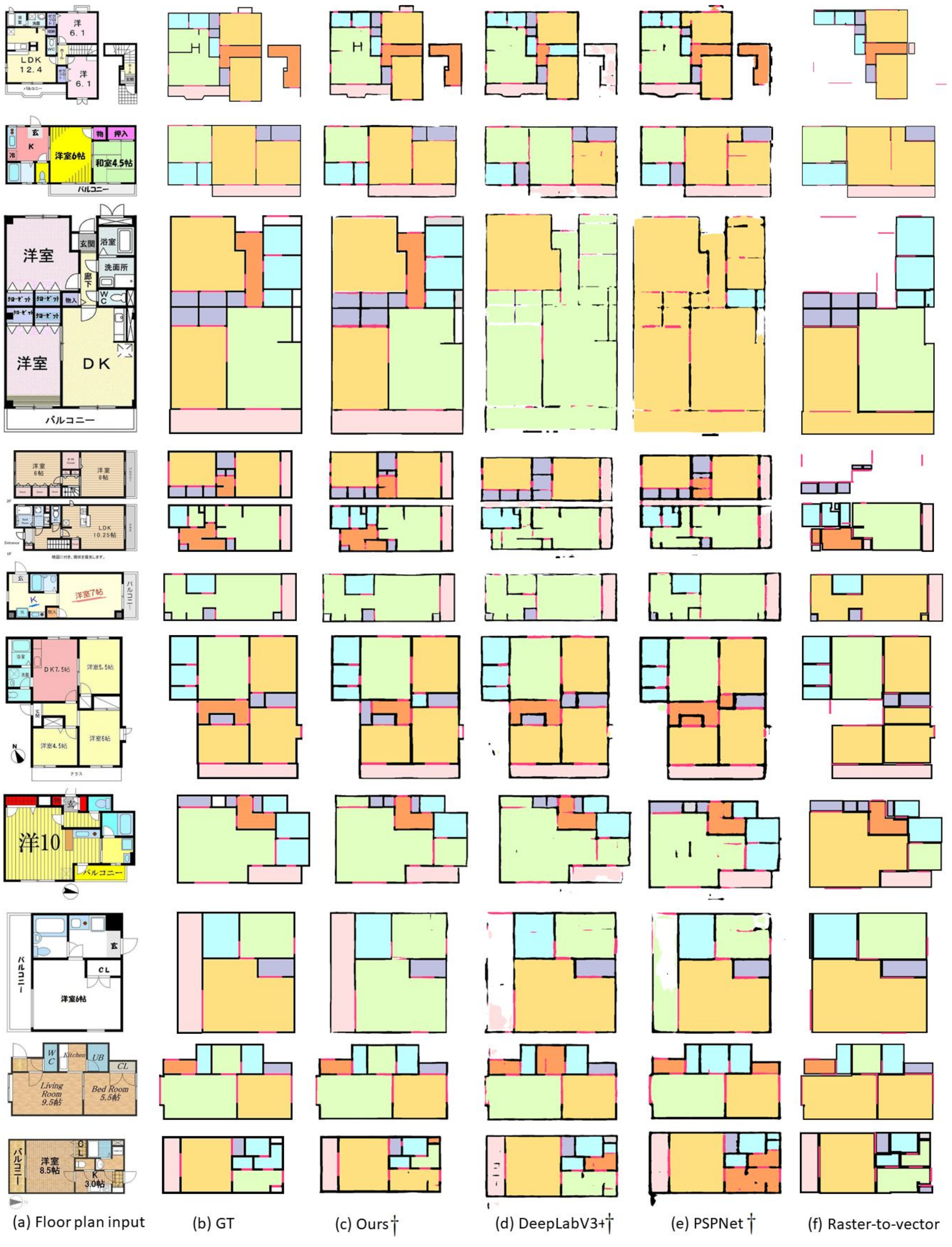

Figure 4. Additional comparison results on the R2V dataset by applying (c) our method, (d) DeepLabV3+, (e) PSPNet, and (f) Raster-to-vector to the floor plan inputs shown in (a). GT denotes ground truth. Symbol † indicates the method with our postprocessing

## B. Additional Results: Visual Comparisons without Postprocessing

In this section, we provide visual comparisons by employing our method and others without postprocessing. Specifically, in Figures 5&6, we compared our method with DeepLabV3+ and PSPNet by training and testing on the R3D dataset and R2V dataset, respectively. From these results, we can see that even without postprocessing, our method can still outperform others.

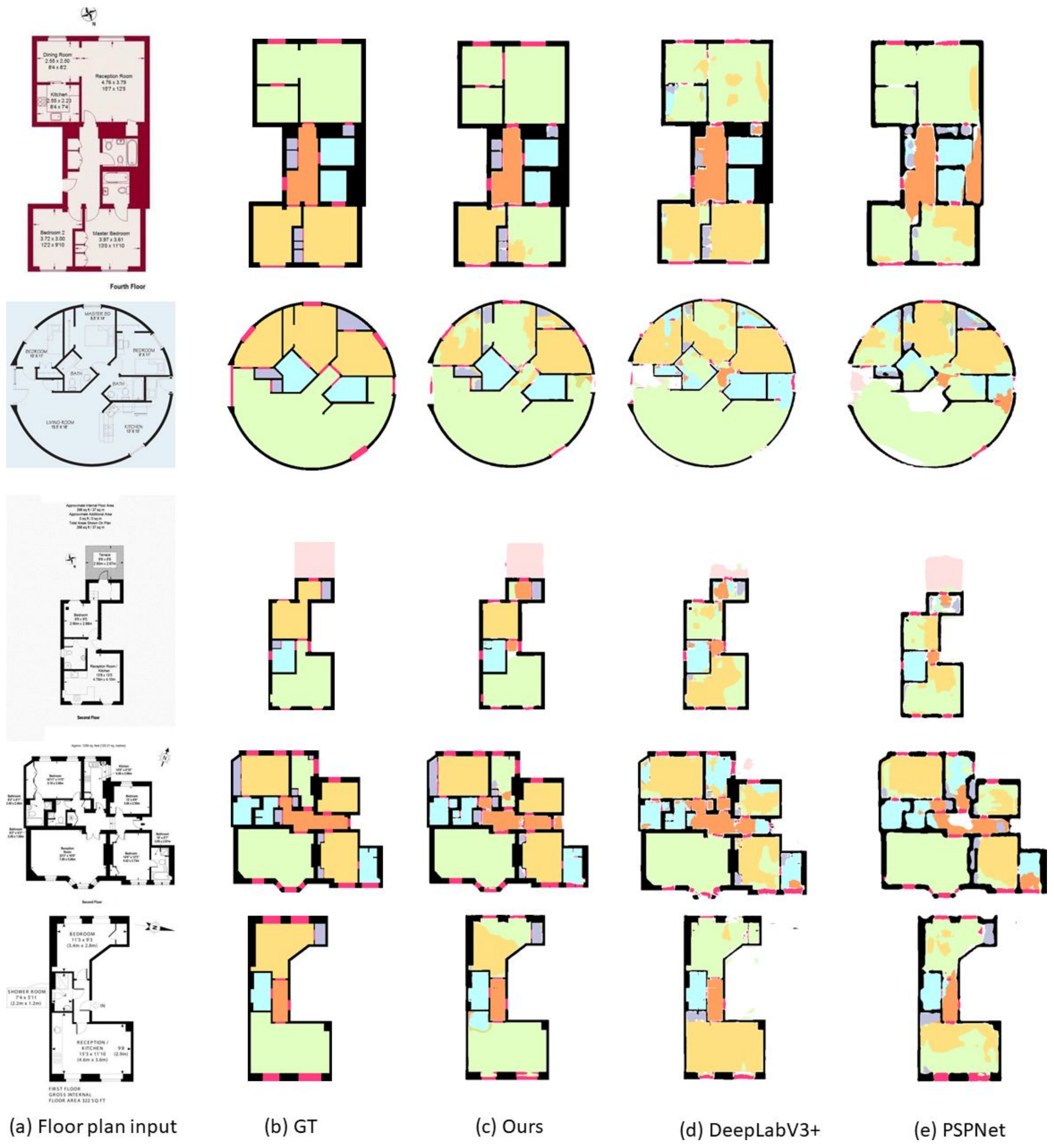

Figure 5. Visual comparisons by applying (c) our method, (d) DeepLabV3+, and (e) PSPNet to the floor plan inputs shown in (a) without postprocessing to floor plans on the R3D dataset. GT denotes ground truth.

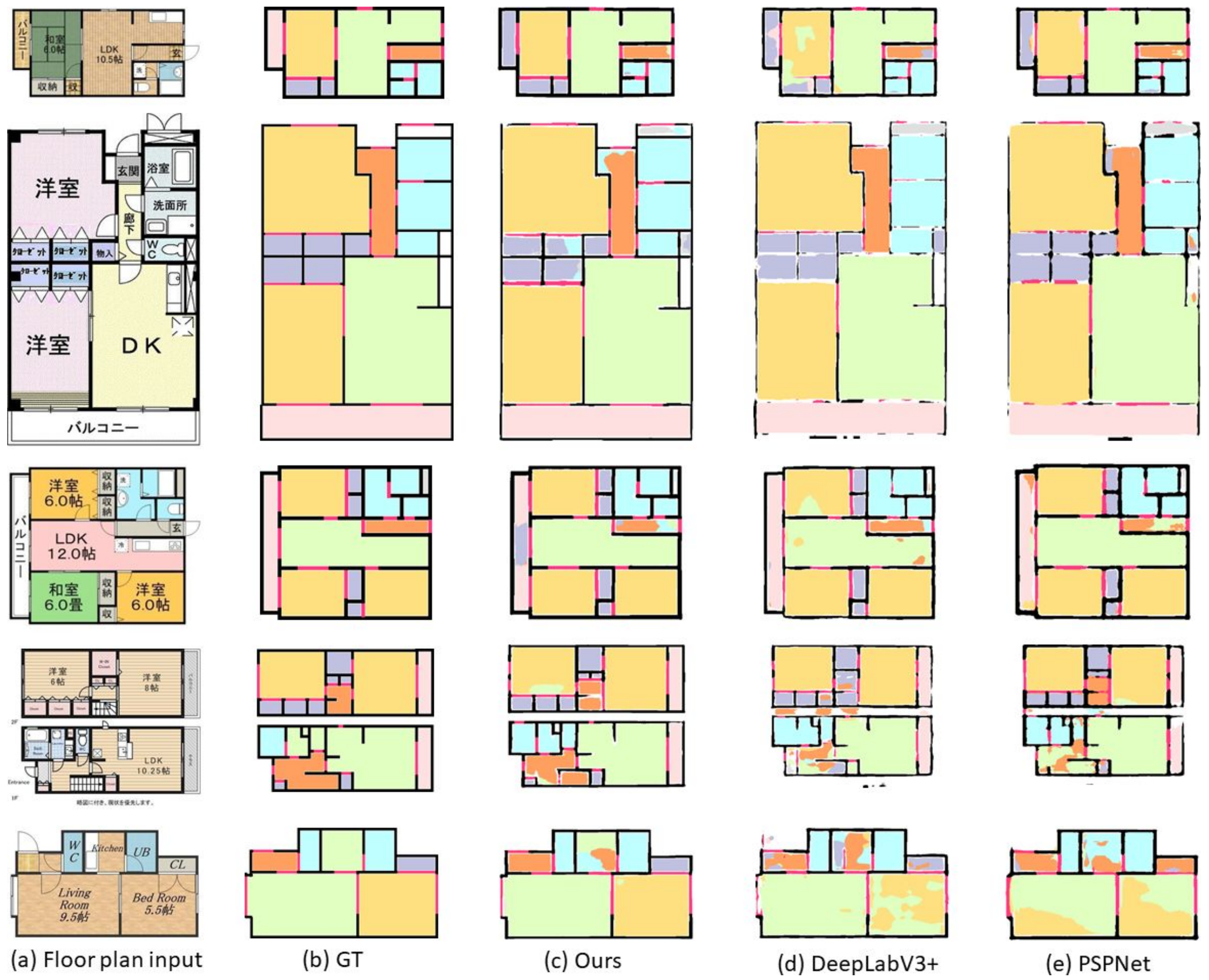

Figure 6. Visual comparisons by applying (c) our method, (d) DeepLabV3+, and (e) PSPNet to the floor plan inputs shown in (a) without postprocessing to floor plans on the R2V dataset. GT denotes ground truth.

## C. Limitations

Figure 7 shows two very challenging cases, where our method fails to detect the room type pixels. In the first case on top, our network fails to recognize some inside and outside regions (see the red box in Figure 7 (top)) due to the specially-shaped corridor in the floor plan. In the second case on bottom, our network misrecognizes the large icon on the top-right corner (see the red arrow in Figure 7 (bottom)) as a wall element, since the size of the arrow appears like part of a wall. This motivates us to further explore the semantics of symbols in floor plans, and perhaps also take the room type features to help improve the room boundary prediction.

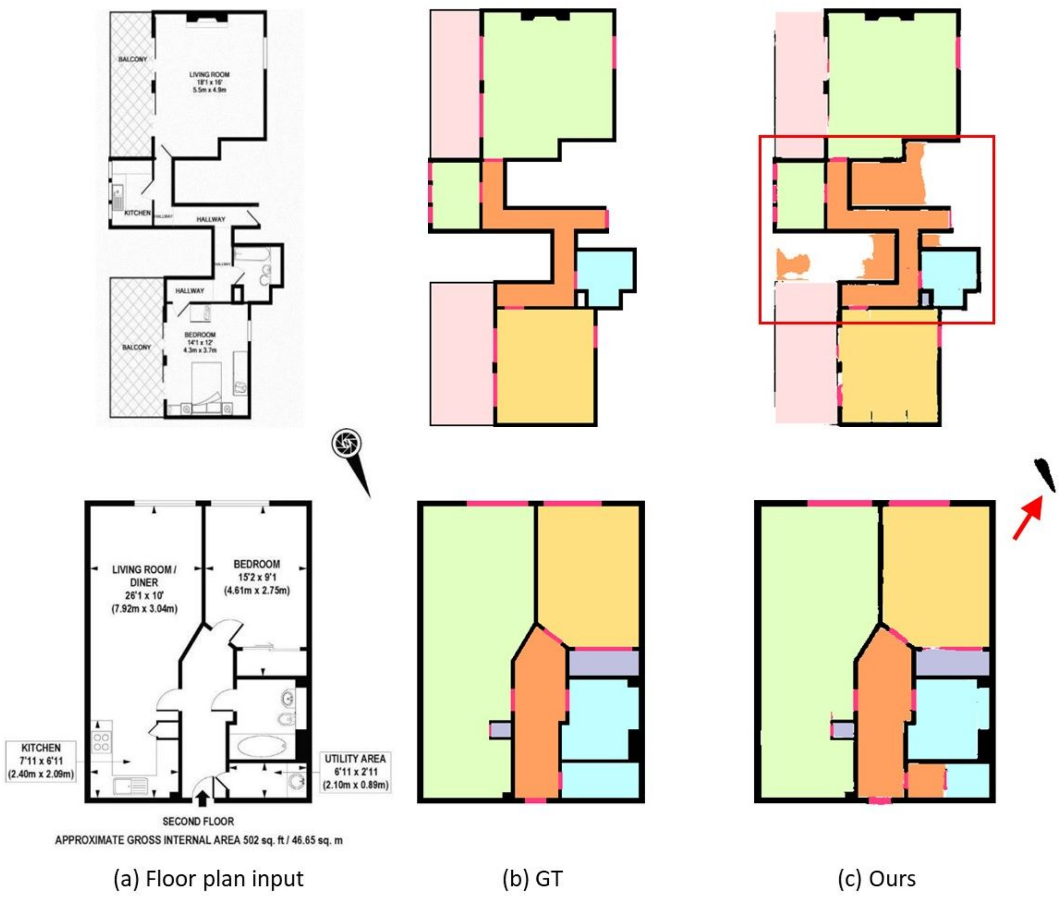

Figure 7. Two challenging cases.

## D. An Analysis on the Cross-and-Within-Task Weighted Loss

The cross-and-within-task weighted loss is evaluated in this section. The mathematical details can be found in Eqs.(4) & (6) of our main manuscript.

- *Baseline #1: Softmax loss.* In the first version, the Softmax loss is directly used as the prediction error for the room boundary and room type prediction, which are denoted as $\mathcal{L}_{rb}$ and $\mathcal{L}_{rt}$, respectively. The overall loss function is the sum of them with equal weights.

$$\mathcal{L} = \mathcal{L}_{rb} + \mathcal{L}_{rt}, \tag{1}$$

  where $\mathcal{L}_* = \sum_i -y_i \log p_i$ with $y_i$ being the index of the $i$-th element and $p_i$ being the prediction label for the $i$-th element.

- *Baseline #2: Cross-task weighted loss.* In the second version, the Softmax loss is still employed as the prediction error inside the room boundary and room type prediction task. However, cross-task weights are used to balance the two tasks.

$$\mathcal{L} = w_{rb}\mathcal{L}_{rb} + w_{rt}\mathcal{L}_{rt}, \tag{2}$$

  where $w_{rb}$ and $w_{rt}$ are weights given by Eq. (7) in our submitted manuscript.

- *Baseline #3: Within-task weighted loss.* Instead of using the basic Softmax loss that treats the floor plan elements equally, the within-task weighted loss is utilized to balance the floor plan elements within each task in the third version. The final loss is also the simple sum of all losses.

$$\mathcal{L} = \mathcal{L}_{rb} + \mathcal{L}_{rt}, \tag{3}$$

  where $\mathcal{L}_* = w_i \sum_i -y_i \log p_i$; see Eq. (4) in our submitted paper for the definition of $w_i$.

Table 1 compares the results of the four variations of our networks having different loss functions based on the R3D dataset [2]. The results demonstrate that our proposed cross-and-within-task weighted loss made our network achieve the best performance.

Table 1. A comparison of networks with different loss functions.

| Metrics | Methods | | | |
|---|---|---|---|---|
| | Baseline #1 | Baseline #2 | Baseline #3 | Our complete version |
| $overall\_accu$ | 0.89 | 0.91 | 0.90 | **0.91** |
| $average\ class\_accu$ | 0.76 | 0.77 | 0.78 | **0.80** |

## E. An Analysis on using Weighted Loss in Semantic Networks

In our main paper, we did not use our weighted loss in DeepLabV3+ & PSPNet, and just used the original softmax cross-entropy loss in these networks. In this section, we tried our weighted loss in these networks, and obtained the results shown below. Here, we could see that our weighted loss can improve the accuracy of minor classes, e.g., balcony. Since the loss needs to balance the weights of the major and minor classes, it may not always improve the overall accuracy.

Table 2. The accuracy comparison by using our weighted loss and original softmax cross-entropy loss in DeepLabV3+ and PSPNet, respectively.

| | DeepLabV3+ | | PSPNet | |
|---|---|---|---|---|
| | our weighted loss | softmax cross-entropy loss | our weighted loss | softmax cross-entropy loss |
| balcony accuracy | **0.15** | 0.08 | **0.47** | 0.41 |
| overall accuracy | **0.85** | 0.85 | **0.85** | 0.84 |

## F. Additional Results: Floor Plan Element Recognition and 3D Model Reconstruction

More sample results produced by our method are shown in Figures 8 - 17. The recognized wall regions are used to recover the 3D model of the apartment.

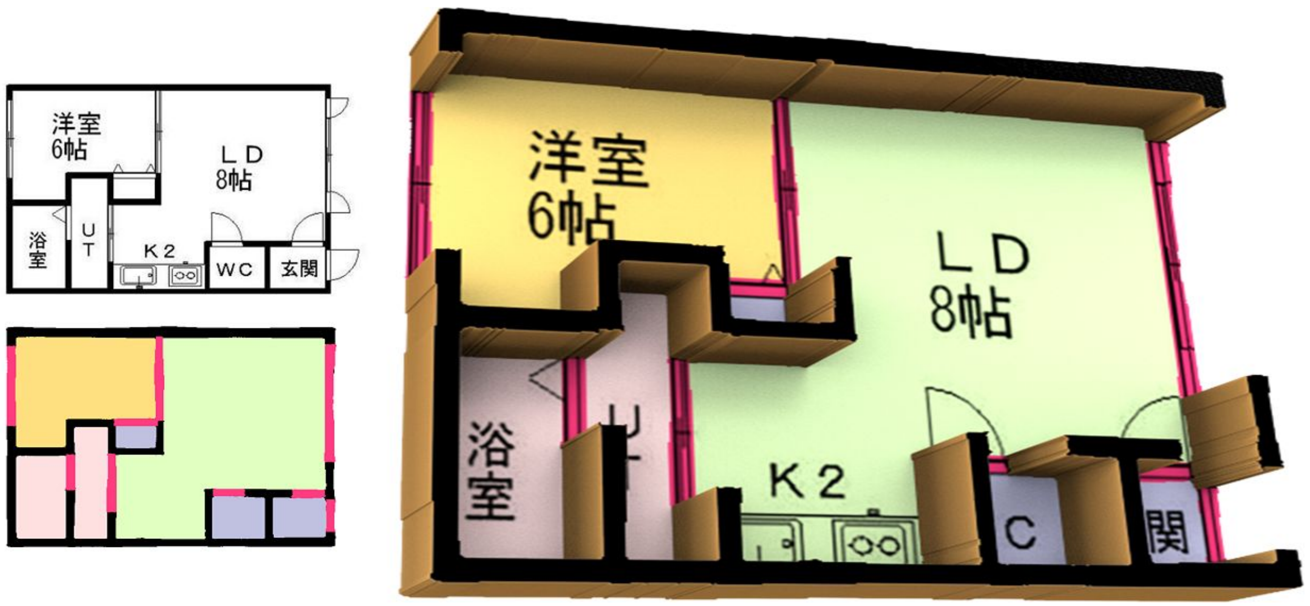

Figure 8. Sample results 1: The floor plan recognition result (lower-left) and the reconstructed 3D model (right).

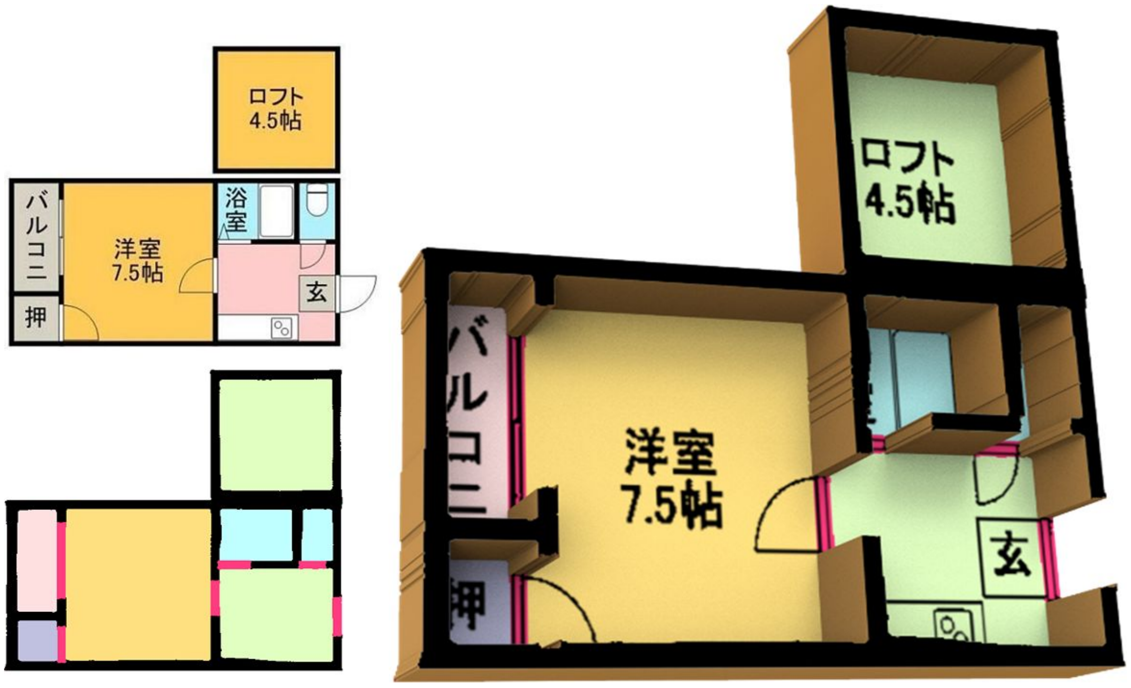

Figure 9. Sample results 2: The floor plan recognition result (lower-left) and the reconstructed 3D model (right).

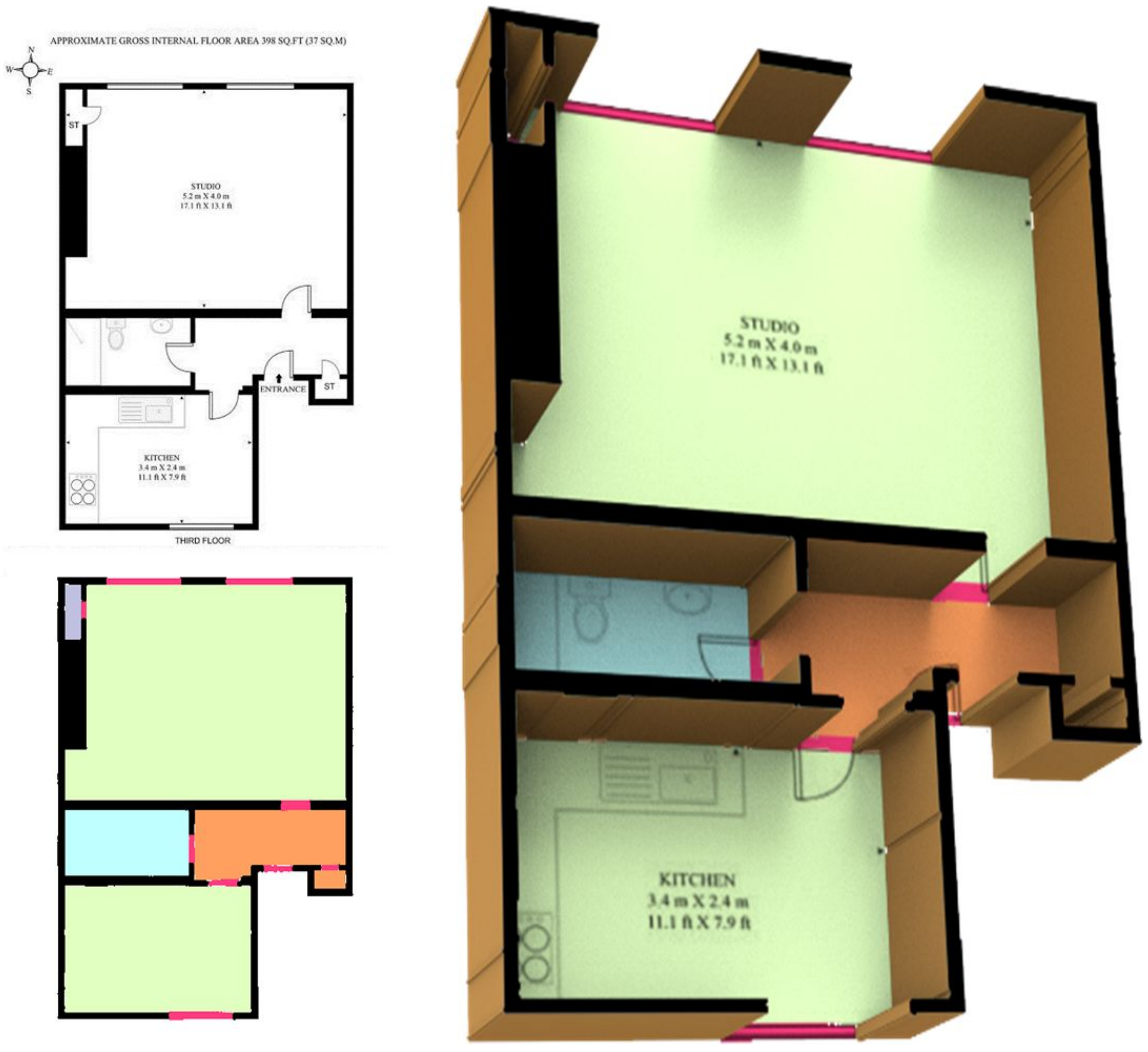

Figure 10. Sample results 3: The floor plan recognition result (lower-left) and the reconstructed 3D model (right).

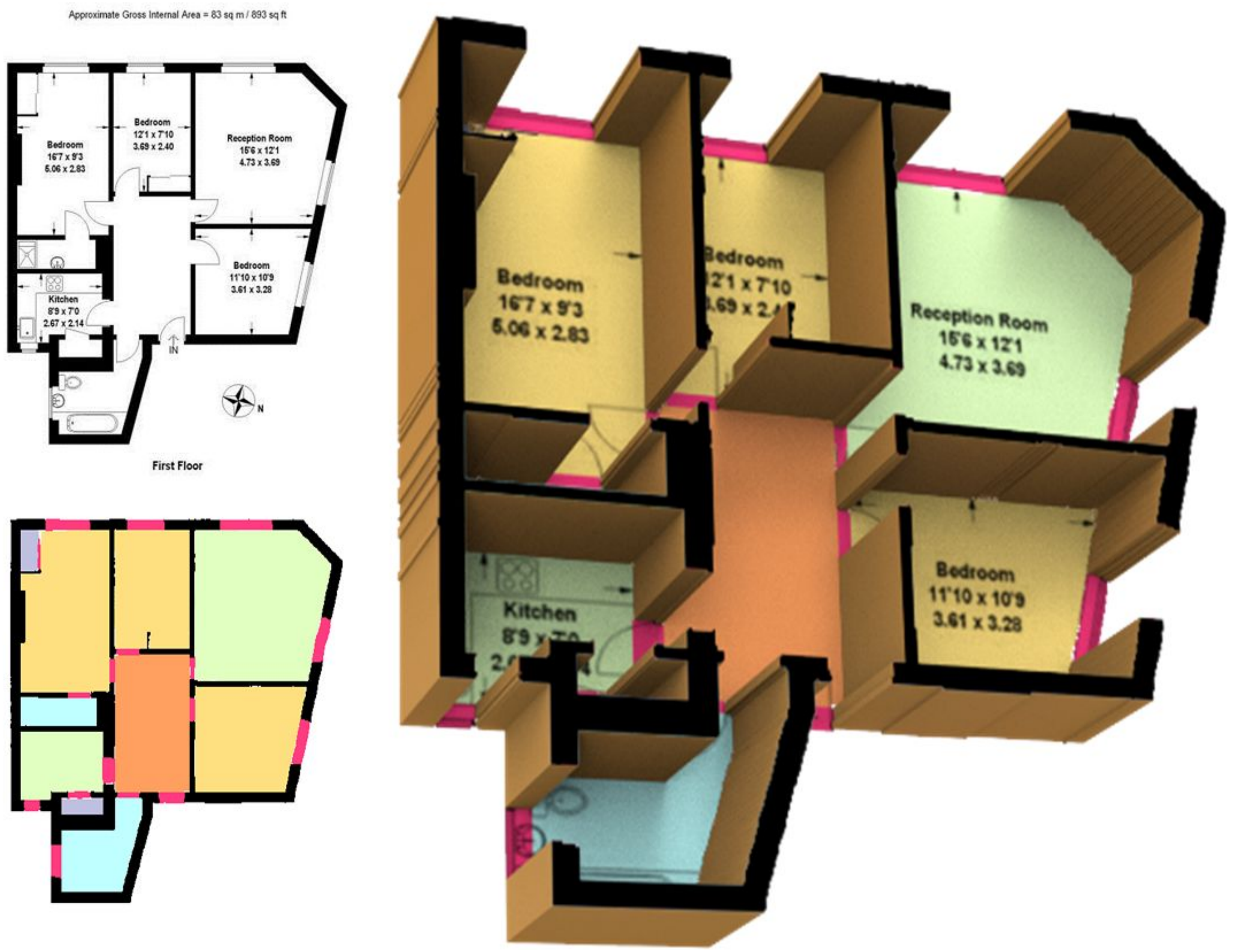
Approximate Gross Internal Area = 83 sq m / 893 sq ft
Bedroom 16'7 x 9'3 5.06 x 2.83
Bedroom 12'1 x 7'10 3.69 x 2.40
Reception Room 15'6 x 12'1 4.73 x 3.69
Bedroom 11'10 x 10'9 3.61 x 3.28
Kitchen 8'9 x 7'0 2.67 x 2.14
IN
N
First Floor

Figure 11. Sample results 4: The floor plan recognition result (lower-left) and the reconstructed 3D model (right).

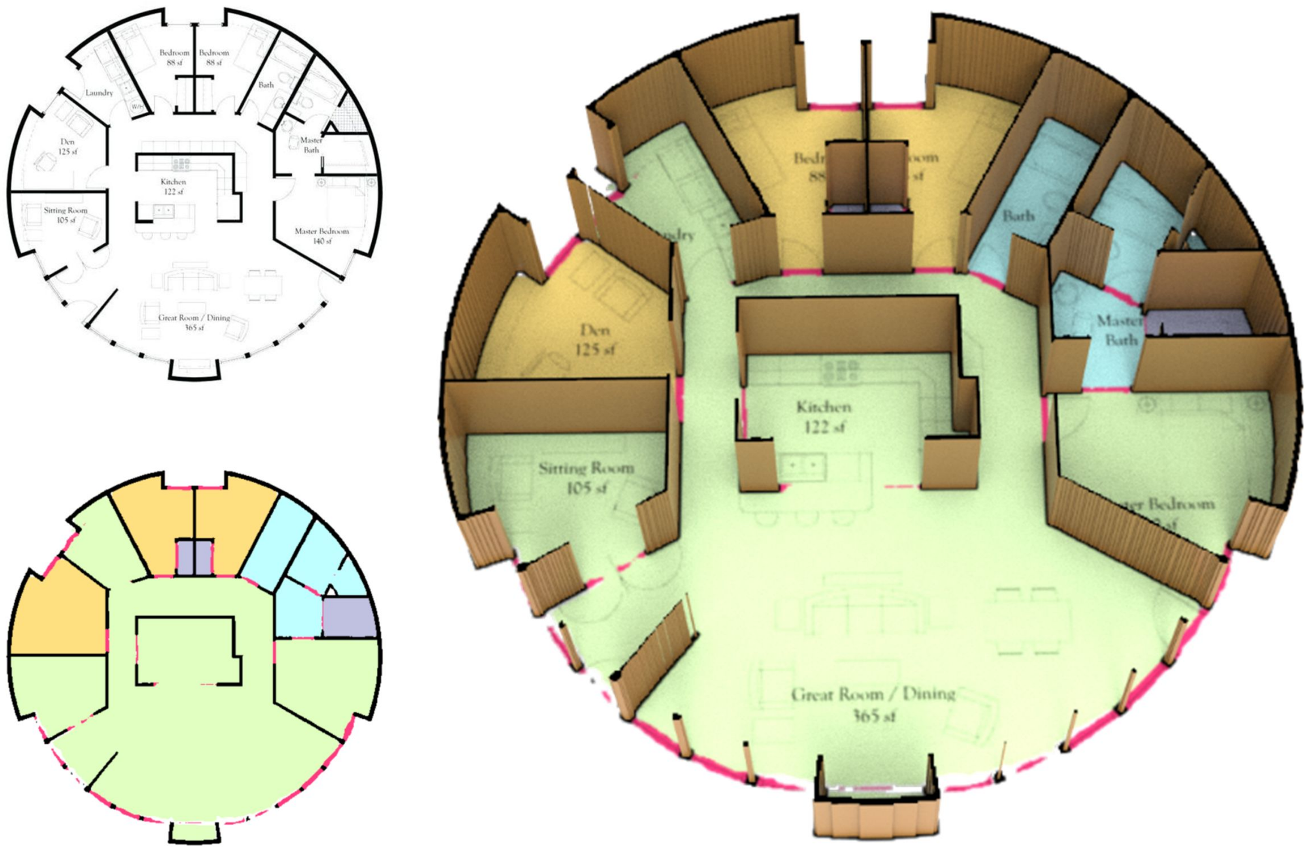

Figure 12. Sample results 5: The floor plan recognition result (lower-left) and the reconstructed 3D model (right).

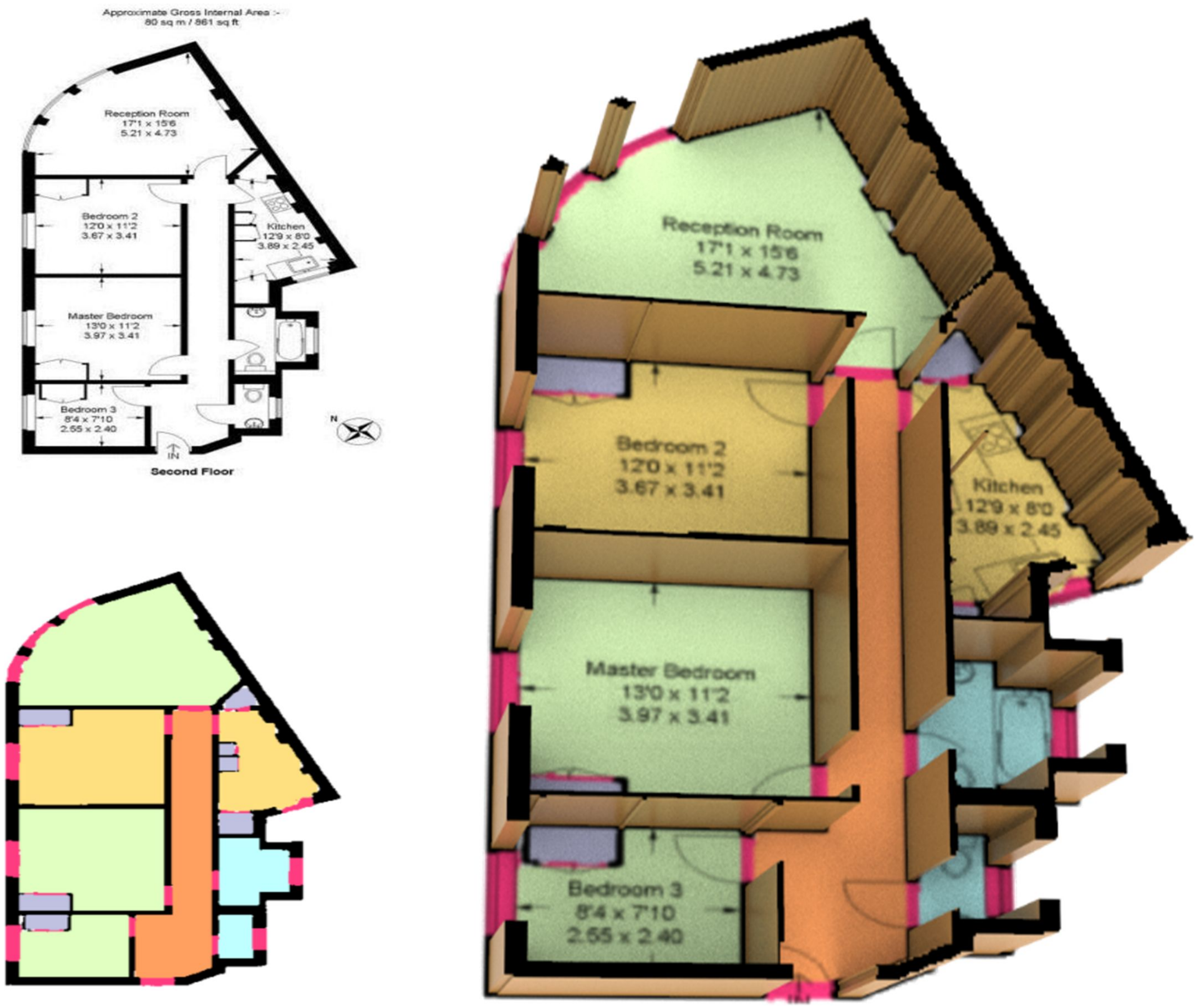

Figure 13. Sample results 6: The floor plan recognition result (lower-left) and the reconstructed 3D model (right).

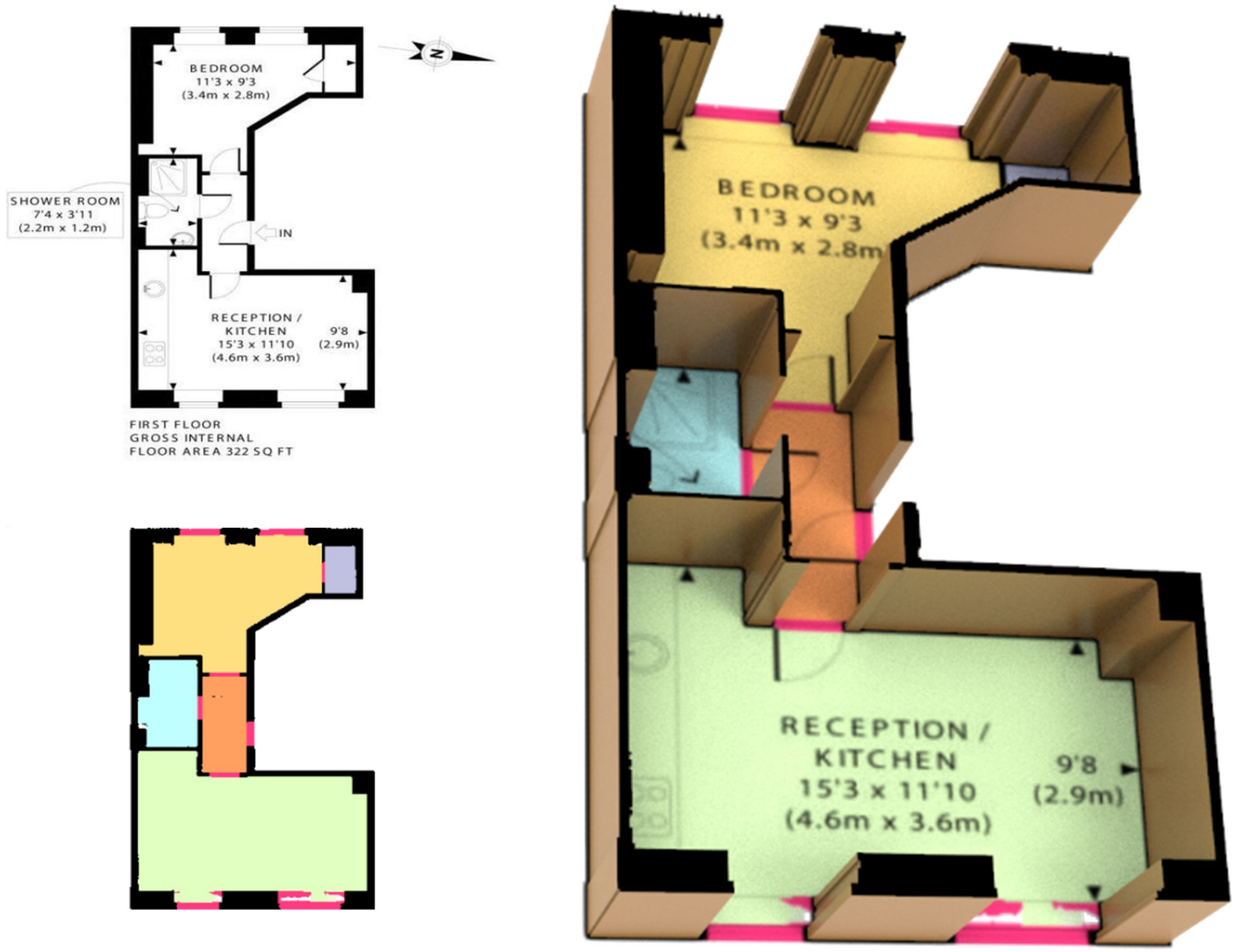

Figure 14. Sample results 7: The floor plan recognition result (lower-left) and the reconstructed 3D model (right).

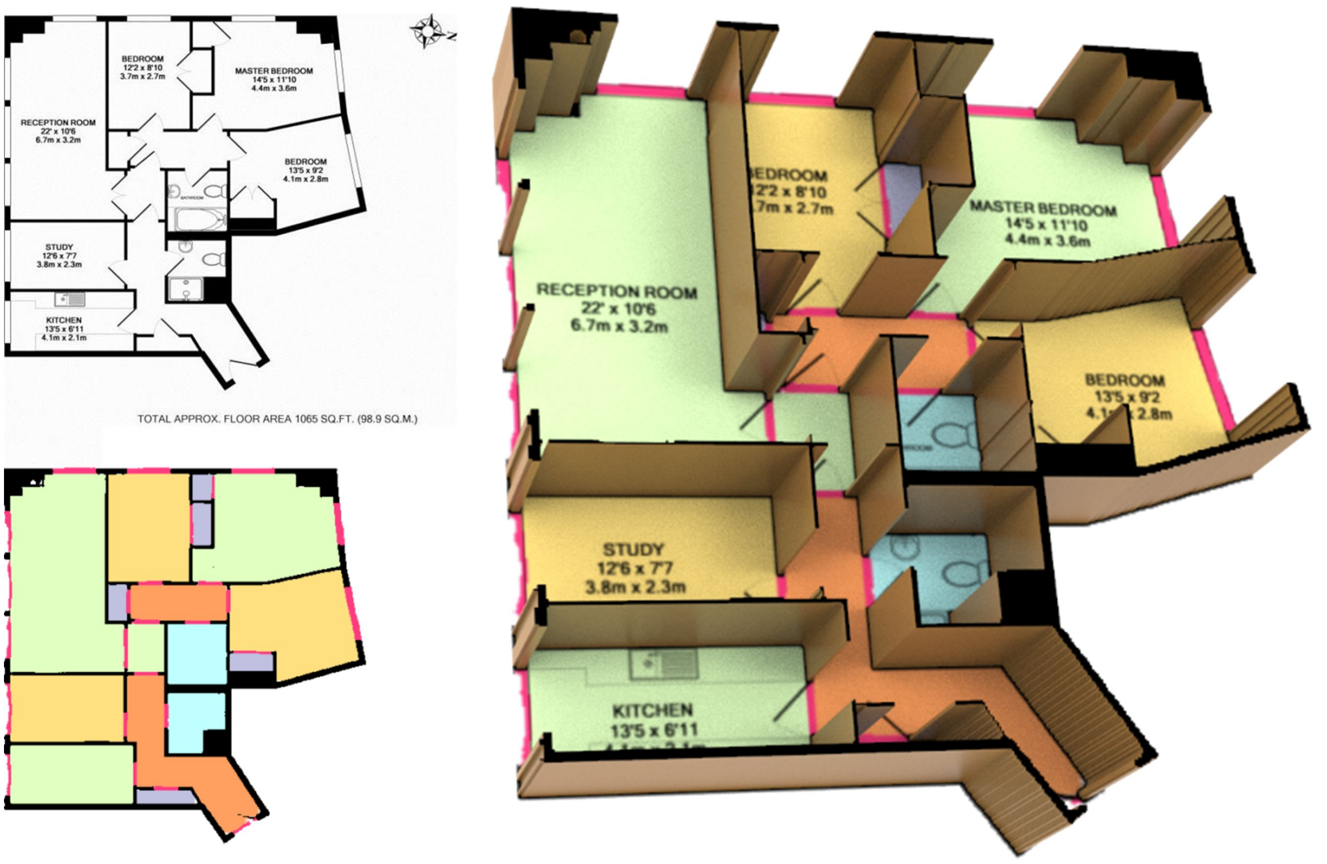

Figure 15. Sample results 8: The floor plan recognition result (lower-left) and the reconstructed 3D model (right).

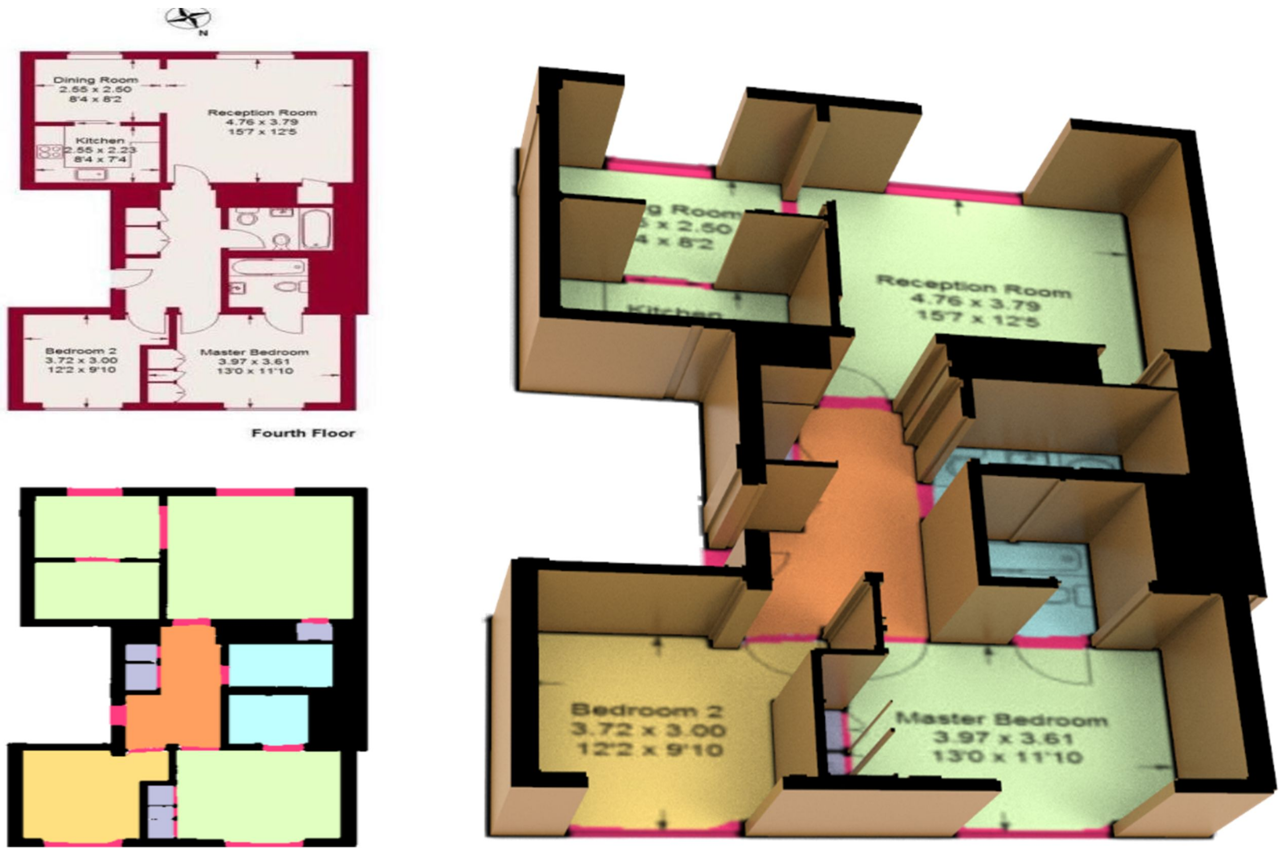

Figure 16. Sample results 9: The floor plan recognition result (lower-left) and the reconstructed 3D model (right).

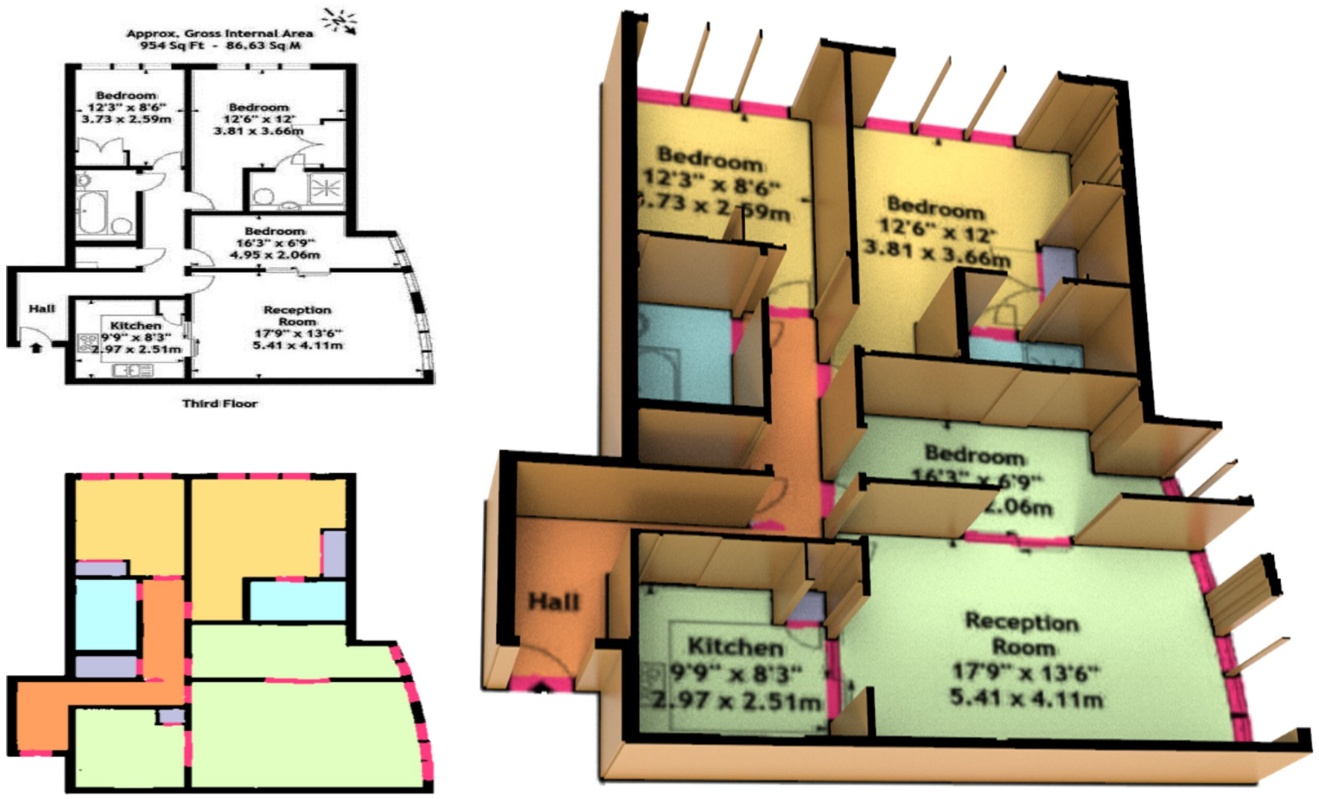

Figure 17. Sample results 10: The floor plan recognition result (lower-left) and the reconstructed 3D model (right).

## References

[1] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conf. on Computer Vision (ECCV)*, 2018. 3

[2] C. Liu, A. Schwing, K. Kundu, R. Urtasun, and S. Fidler. Rent3D: Floor-plan priors for monocular layout estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 10

[3] C. Liu, J. Wu, P. Kohli, and Y. Furukawa. Raster-to-vector: Revisiting floorplan transformation. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2017. 3

[4] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3