Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries
Supplementary Material

Yuanwen Yue[1] Theodora Kontogianni[2] Konrad Schindler[1,2] Francis Engelmann[2]

[1]Photogrammetry and Remote Sensing, ETH Zurich [2]ETH AI Center, ETH Zurich

In the supplementary material, we first provide additional ablation studies on the attention design and numbers of the two-level queries (Sec. A). Then we describe an alternative 'tight' room layout used by some baselines and report the scores of our model trained with this layout (Sec. B).We also provide further implementation details and more results on semantically-rich floorplans (Sec. C). Finally, we provide details for running competing methods Floor-SP [2] and HEAT [3], as well as a learning-free baseline on the SceneCAD dataset [1] (Sec. D).

A. Additional Ablation Studies

A.1. Self-attention of the two-level queries

In the main paper, the Transformer decoder performs self-attention on all vertex-level queries regardless of the polygon they belong to (Tab. 8, ②). Alternatively, in this experiment, we restrict the vertex-level queries to attend only to vertices within the same polygon. Implementation-wise, we add an attention mask to prevent attention from vertex-level queries in one polygon to vertex-level queries of another polygon. We find that this restricted form of attention leads to an overall reduced performance (Tab. 8, ①). We conclude that the self-attention between vertices across all polygons plays an important role for structured reasoning. In particular, the attention mechanism across multiple polygons seems to help fine-tune the vertex positions of one polygon by attending to the vertex positions of its neighboring polygons.

	Room			Corner			Angle		
Settings	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
① Intra-Poly. Attn.	95.5	94.2	94.8	90.3	81.9	85.9	87.2	79.1	83.0
② Inter-Poly. Attn.	**96.5**	**95.3**	**95.9**	**91.2**	**82.8**	**86.8**	**88.3**	**80.3**	**84.1**

Table 8. Impact of self-attention design on Structured3D val. Using self-attention only between the vertices of a single polygon leads to a drop in the F1 score, showing the usefulness of extending the effect of vertices across all polygons.

Settings			Room		Corner		Angle	
M	N	t (ms)	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
15	30	8.4	95.7	94.5	90.2	82.2	86.5	79.0
20	40	10.8	96.3	95.0	90.8	82.7	87.8	80.0
20	50	12.8	96.2	94.6	90.5	82.1	87.6	79.5
30	40	13.9	96.8	95.5	91.1	82.7	88.1	80.1
30	50	16.8	96.0	94.8	90.6	82.4	87.5	79.7

Table 9. Analysis on number of queries. The reported scores are on Structured3D validation set averaged over three runs.

A.2. Number of queries

We study the effect of different numbers of queries in each level in Tab. 9. The number of queries is based on: 1) the maximum number of rooms and the maximum number of corners in each room in the training dataset. 2) computational efficiency. Although the number of queries empirically has a small impact on reconstruction quality, the results highlight the robustness of our model towards fewer queries and justify the associated gain in runtime (16 ms → 8 ms), especially since we aim to be much faster than prior work. We chose $M = 20$ and $N = 40$ as a good compromise between model performance, inference time, as well as training time.

B. Tight Room Layouts

We represent floorplans as a set of closed polygons, which is consistent with the groundtruth annotations in

	Room			Corner			Angle		
Method	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Floor-SP [2]	89.	88.	88.	81.	73.	76.	80.	72.	75.
MonteFloor [6]	95.6	94.4	95.0	88.5	77.2	82.5	86.3	75.4	80.5
HAWP [8]	77.7	87.6	82.3	65.8	77.0	70.9	59.9	69.7	64.4
LETR [7]	94.5	90.0	92.2	79.7	78.2	78.9	72.5	71.3	71.9
HEAT [3]	96.9	94.0	95.4	81.7	83.2	82.5	77.6	79.0	78.3
RoomFormer (Ours)	97.9	96.7	97.3	89.1	85.3	87.2	83.0	79.5	81.2
RoomFormer* (Ours)	97.2	96.2	96.7	91.6	83.4	87.3	88.3	80.5	84.2

Table 10. Foorplan reconstruction scores on Structured3D test. For a complete and fair comparison, we complement Tab. 1 from the main paper with the result of our model trained on tight room layouts, indicated by *. Cyan and orange mark the two top scores.

Structured3D [9]. The advantage of this representation is that the thickness of inner walls is implicitly provided by the distance between neighboring room polygons (see Fig. 8, *right column*). Alternatively, HEAT [3] predict floorplans as planar graphs, which means that the walls of adjacent rooms are represented by a single shared edge in the graph. In particular, this simpler graph representation can approximate the true floorplan only up to the thickness of the walls. Furthermore, to train HEAT, an additional pre-processing step is required which merges the groundtruth edges of neighboring polygons into a single shared edge. It is important to note that the evaluation still runs on the unmodified groundtruth floorplans. Therefore, during evaluation, HEAT performs a post-processing step to obtain a set of closed room polygons from the estimated planar graph.

To show that the improved performance of our model is independent of the layout representation, we train a model on the same 'tight' room layout representation as HEAT and report the scores in Tab. 10 marked with a star (*). Note again that all models are evaluated on the same non-processed groundtruth annotations of Structured3D [9]. We observe that the room metrics of the model trained on tight room layouts drop a bit compared with our original model, while still outperforming all other methods. The drop in scores is not surprising since the modified training data is only an approximation of the original groundtruth used for evaluation. Furthermore, the room metrics penalize overlap between rooms [6]. Results from tight room layouts are more likely to overlap, which can potentially degrade the room metrics. Interestingly, the angle metrics improve, especially when it comes to angle precision, which already outperforms MonteFloor. In the tight room layout, we encourage corners in adjacent walls to share the same location and have exactly complementary angles. This implicit constraint might help the model to reason about angle relationships, thus benefiting the angle metrics. More qualitative results can be found in Fig. 8.

C. Semantically-Rich Floorplan Models

We proposed three model variants for reconstructing semantically-rich floorplans: (1) SD-TQ: single decoder with two-level queries. (2) TD-TQ: two decoders with two-level queries. (3) TD-SQ: two decoders with single-level queries in the line decoder. Here, we explain additional implementation details along with more results. We also describe our heuristic for drawing the door arcs based on the width of the door segments.

C.1. Implementation details

We describe the implementation details of the three model variants to extend our RoomFormer architecture to predict room types, doors and windows (Fig. 4, main paper).

SD-TQ. Single decoder with two-level queries. We take our original architecture and increase the number of room-level queries M from 20 to 70 while keeping the number of corner-level queries N unchanged. In addition, since there is no need to rasterize lines, we remove the rasterization loss $\mathcal{L}_{\text{ras}}$. We predict all the semantic types (room types, door or window) from the aggregated room-level features of the output embedding of the polygon decoder, as described in Sec. 3.5 of the main paper.

TD-TQ. Two decoders with two-level queries. We add a separate line decoder with the same architecture as the polygon decoder. In the line decoder, we set the number of line-level queries to 50 and the number of corner-level queries to 2. We remove the rasterization loss $\mathcal{L}_{\text{ras}}$ term for the line decoder (still used in the polygon decoder). Room types are predicted from the aggregated room-level features of the output embedding of the polygon decoder, while door/window types are predicted from the aggregated line-level features of the output embedding of the line decoder.

TD-SQ. Two decoders with single-level queries. We add a separate line decoder that takes single-level queries, and predicts the coordinates of the two endpoints of each line directly, similar to [7]. We set the number of single-level queries to 50. Room types are predicted as in TD-TQ. Door/window types are predicted based on the output embedding of the line decoder.

C.2. Plotting doors

To obtain floorplan illustrations that are closer to actual floorplans used by architects, we follow the typical notation and represent doors as arcs. Note that this step is only for visualization purposes and is not part of the annotations in the training datasets. In particular, we cannot predict towards which side a door opens and if it is a double door or a single door. The visualization used in this paper is based on a heuristic which creates a double door when the predicted width of the door exceeds a certain threshold. The exact heuristic is shown in Algorithm 1.

C.3. Additional results

Tab. 11 complements Tab. 4 in the main paper by providing more detailed scores of the three model variants on semantically-rich floorplan reconstruction. By comparing the single-decoder variant (SD-TQ) with the two-decoder variants (TD-TQ and TD-SQ), we find that separate decoders can help improve room type classification. Our polygon queries are designed for geometries with a varying number of vertices. Since a line has a fixed number of 2 vertices, the single-level query variant (TD-SQ) works better than the two-level query variant (TD-TQ). We provide more qualitative results in Fig. 9.

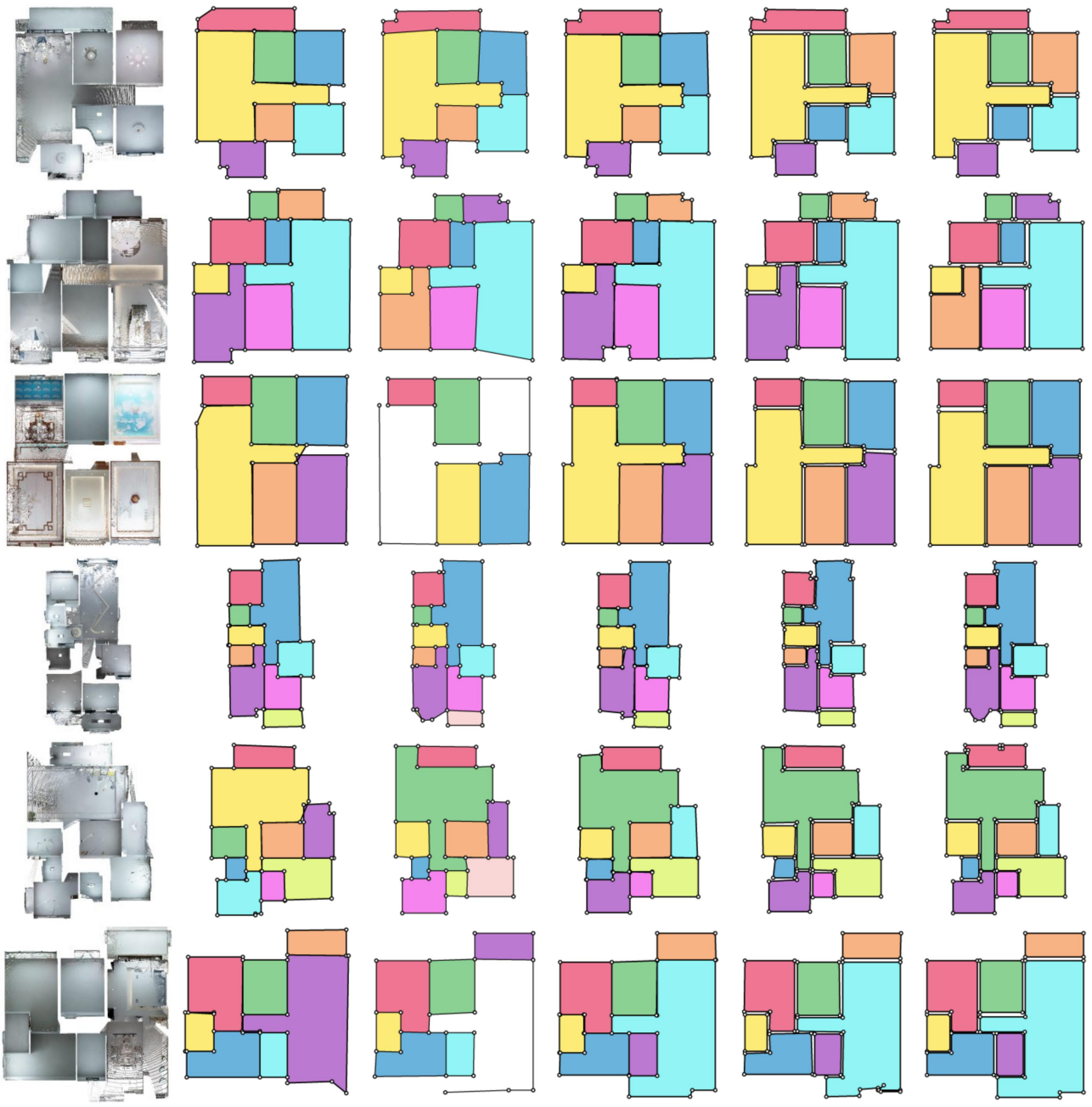

Figure 8. More qualitative evaluations on Structured3D [9]. Colors are assigned based on room locations, *without* semantic meaning. Ours* denotes the result of our model trained on tight room layout. (*Best viewed in color on a screen and zoomed in.*)

	Door/Window			Room*			Room			Corner			Angle		
Method	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
SD-TQ	83.4	79.0	81.1	71.5	70.0	70.7	95.3	93.3	94.3	**86.0**	81.8	83.9	**78.6**	**74.9**	**76.7**
TD-TQ	82.6	**79.1**	80.8	71.9	70.9	71.4	94.0	92.8	93.4	84.2	80.0	82.0	75.6	71.9	73.7
TD-SQ	**85.6**	78.2	**81.7**	**74.8**	**74.0**	**74.4**	**95.4**	**94.4**	**94.9**	85.8	**82.6**	**84.2**	77.3	74.5	75.9

Table 11. Detailed scores of semantically-rich floorplan reconstruction on Structured3D test set [9].

Figure 9. Additional qualitative results on semantically-rich floorplans. (*Best viewed in color on a screen and zoomed in.*)

Algorithm 1 Algorithm for plotting doors as arcs

Input: a set of predicted lines $L = \{l_i\}_{i=1}^{N^l}$, where $l_i = \left(\left(x_1^i, y_1^i\right), \left(x_2^i, y_2^i\right)\right)$
Output: plotting of a set of arcs

calculate the median length m of all the lines L
for l_i in L **do**
  if $\left\|y_2^i - y_1^i\right\| > \left\|x_2^i - x_1^i\right\|$ **then**
    if $y_2^i > y_1^i$ **then** $e_1^i = (x_1^i, y_1^i)$ and $e_2^i = (x_2^i, y_2^i)$
    else $e_1^i = (x_2^i, y_2^i)$ and $e_2^i = (x_1^i, y_1^i)$
    end if
    if len$(l_i) < 1.5 \times m$ **then** draw a quadrant centered at e_1^i with a radius of len(l_i) from e_2^i clockwise
    else draw two opposite quadrants centered at e_1^i and e_2^i with a radius of len$(l_i)/2$ on the right of $\overrightarrow{e_1^i e_2^i}$
    end if
  else
    if $x_2^i > x_1^i$ **then** $e_1^i = (x_2^i, y_2^i)$ and $e_2^i = (x_1^i, y_1^i)$
    else $e_1^i = (x_1^i, y_1^i)$ and $e_2^i = (x_2^i, y_2^i)$
    end if
    if len$(l_i) < 1.5 \times m$ **then** draw a quadrant centered at e_1^i with a radius of len(l_i) from e_2^i counterclockwise
    else draw two opposite quadrants centered at e_1^i and e_2^i with a radius of len$(l_i)/2$ on the left of $\overrightarrow{e_1^i e_2^i}$
    end if
  end if
end for

D. Running Competing Approaches

For comparison on the SceneCAD dataset [1] in the main paper, we select two representative methods from both optimization-based and fully-neural categories Floor-SP [2] and HEAT [3] that offer state-of-the-art results in floorplan reconstruction and have a public codebase. For a more complete comparison, here we also report the performance of a heuristic-guided pipeline that is free of any deep learning components. Here we describe in detail how we adapted those methods for the SceneCAD dataset [1].

Floor-SP: We used the official implementation with a change in the sequential room-wise shortest path module. We project 3D scans into density images of size 256×256 pixels. Unlike Structured3D, the SceneCAD dataset usually contains only one room per scene, which will result in larger occupancy pixel areas for a single room. While this does not cause problems for HEAT and our approach, it can lead to a large search space for the sequential room-wise shortest path module in Floor-SP since each room mask contains a large number of pixels. Using Floor-SP default settings, we observe the solver cannot find a solution for many scenes due to the computational complexity of the larger number of pixels per room. Therefore, we down-sample the density map to 64×64 pixels (a similar size with a single room in the density map of Structured3D) to help reduce the search space. Please note, we train other modules (room mask extraction and corner/edge detection) on the original density map without down-sampling.

		Room	Corner			Angle		
Method	t(s)	IoU	Prec.	Rec.	F1	Prec.	Rec.	F1
Non-learned	0.22	86.0	66.0	80.8	72.7	45.0	55.2	49.6
Floor-SP [2]	26	91.6	89.4	**85.8**	87.6	74.3	71.9	73.1
HEAT [3]	0.12	84.9	87.8	79.1	83.2	73.2	67.8	70.4
RoomFormer (Ours)	**0.01**	**91.7**	**92.5**	85.3	**88.8**	**78.0**	**73.7**	**75.8**

Table 12. Foorplan reconstruction on the SceneCAD val set [1].

HEAT: We used the official implementation with a batch size of 10. We train the model for 400 epochs and found that longer training times would not help to improve the performance further. For the experiments on cross-data generalization, we directly load the released official checkpoints trained on Structured3D and evaluate them on SceneCAD.

Non-learned baseline: We first project the 3D scan along the vertical axis into an occupancy map of size 256×256 pixels. A pixel is occupied if at least one point is projected to this pixel. To mitigate the impact of missing scans, we apply dilation and erosion to fill holes in the occupancy map. Then we employ a learning-free polygon vectorization algorithm [5] to extract closed polygons from the occupancy map, finally followed by a Douglas-Peucker algorithm [4] to remove redundant corners.

We complement Tab. 2 of the main paper with the result of the non-learned baseline (the 1st row in Tab. 12). We report the running time of the polygon vectorization process. Surprisingly, the pipeline achieves a room IoU of 86.0 and a corner recall of 80.8. However, the angle metrics indicate that those polygons usually fail to accurately describe the geometry of the actual room shapes. This suggests that the non-learned baseline can only capture the rough shape of the floorplan compared with methods that utilize deep learning.

References

[1] Armen Avetisyan, Tatiana Khanova, Christopher Choy, Denver Dash, Angela Dai, and Matthias Nießner. SceneCAD: Predicting Object Alignments and Layouts in RGB-D Scans. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 5

[2] Jiacheng Chen, Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floor-SP: Inverse CAD for Floorplans by Sequential Room-Wise Shortest Path. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 5

[3] Jiacheng Chen, Yiming Qian, and Yasutaka Furukawa. HEAT: Holistic Edge Attention Transformer for Structured Recon-

struction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 3, 5

[4] David H Douglas and Thomas K Peucker. Algorithms for the Reduction of the Number of Points Required to Represent A Digitized Line or Its Caricature. *Cartographica: the International Journal for Geographic Information and Geovisualization*, 1973. 5

[5] Muxingzi Li, Florent Lafarge, and Renaud Marlet. Approximating Shapes in Images with Low-complexity Polygons. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[6] Sinisa Stekovic, Mahdi Rad, Friedrich Fraundorfer, and Vincent Lepetit. MonteFloor: Extending MCTS for Reconstructing Accurate Large-Scale Floor Plans. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 3

[7] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. Line Segment Detection Using Transformers Without Edges. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[8] Nan Xue, Tianfu Wu, Song Bai, Fudong Wang, Gui-Song Xia, Liangpei Zhang, and Philip HS Torr. Holistically-attracted Wireframe Parsing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[9] 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 (ECCV)*, 2020. 2, 3