# Supplementary Material
# SceneGraphFusion: Incremental 3D Scene Graph Prediction from RGB-D Sequences

## 8. Network Architecture

We use FC(*in*, *out*) denote a fully-connected layer, and MLP$(\cdot, ..., \cdot)$ as a set of FC layers with a ReLU activation between each FC layer. Our PointNet encoder $f_p$, is a shared-weight MLP$(64, 128, 512)$ followed by a maximum pooling operation to obtain a global feature. The other layers are listed in Tbl. 10.

## 9. Training Details

All the models in our evaluation section (Sec. 6) were trained with the same set up but with different training data for 150 epochs. We use AdamW [3] optimizer with Amsgrad [6] and an adaptive learning rate, inverse proportional to the log of the number of edges. Given a training batch with $n$ edges and $\text{lr}_{base} = 1e^{-3}$, the base learning rate in AdamW is adjusted as follows

$$\text{lr} = \text{lr}_{base} \frac{1}{\ln n}. \tag{11}$$

The training data are the 3D reconstructions created from RGB-D sequences. In order to train our network to handle partial data, subgraphs are randomly extracted during training time. In each iteration, two segments are randomly selected together with their four-hop neighbor segments. We further randomly discard edges with a dropout rate of $50\%$. In addition, we randomly sample points in each segment. The properties described in Sec 3.1 are computed based on sampled points. For the training loss, we follow the approach in [11] with a weight factor of $0.1$ between the object and predicate loss. We use two message passing layers, each with 8 heads.

## 10. Experiment Details

In this section, we detailed the training dataset and the hyper-parameters used in the experiment section (Sec 6.1 Geometric Segments) of our main paper. As mentioned in the main paper, 20 NYUv2 [5] object classes are used. For predicates, we focus on *support* relationships. We further filter out rare relationship. A predicate is discarded if it occurs less than 10 times in the training data or less then 5 times in the test data. This leaves us with 8 predicates, *i.e.* supported by, attached to, standing on, hanging on, connected to, part of, build in, and same part.

The geometrical segmentation method [7] in our framework uses the pyramid level of 2, which scales the input image with a factor of 2, for image segmentation. Further, we filter out segments with less than 512 points.

## 11. 3D Panoptic Segmentation

On Tbl. 12 we report the complete panoptic segmentation evaluation on Tbl. 4. With respect to the panoptic quality (PQ), our method outperforms PanopticFusion in 7 out of 20 classes. The PQ can be broken down into segmentation quality (SQ) and recognition quality (RQ). The SQ evaluates only the matched segments, via an intersection over union (IoU) score over $50\%$. RQ is known as the $F_1$ score. Our method has a similar SQ performance as PanopticFusion while performing worse when compared with the RQ metric. This is likely due to missing scene geometry caused by the incremental segmentation [7] that our approach relies on. By using a metric that is less influenced by missing points, *i.e.* SQ, or ignoring the missing points in the evaluation, our method has equivalent or slightly better performance compare to PanopticFusion [4].

## 12. Robustness against Missing Information

Tbl. 11 shows the complete experiment mentioned in Sec 6.2 of the main paper. We use our network architecture with different attention methods. The update of the node feature $v_i$ in equation 7 can be re-written as follows:

$$\mathbf{v}_i^{\ell+1} = g_e\left(\left[\mathbf{v}_i^{\ell}, \Phi_{j\in\mathcal{N}(i)}\left(\Psi\left(\cdot\right)\right)\right]\right), \tag{12}$$

where $\Phi$ is a permutation in-variance function, *e.g.* sum, mean or max, and $\Psi(\cdot)$ represents an attention function. For *without*, we set $\Psi(\cdot)$ to $\Psi(\mathbf{v}_j^{\ell}) = \mathbf{v}_j^{\ell}$ with $\Phi = \sum$. For *SDPA*, the $\Psi(\cdot)$ is set to $f_{sdpa}(\mathbf{v}_i^{\ell}, \mathbf{v}_j^{\ell})$ with $\Phi = \sum$, where $f_{sdps}$ is the multi-head attention method [8] and for *GAT*,

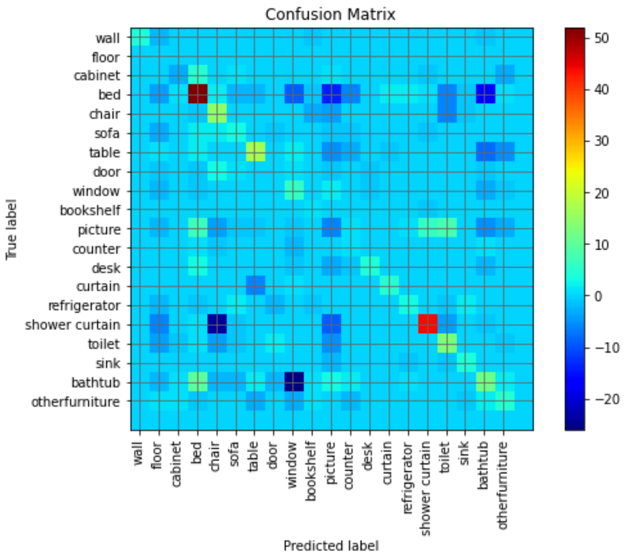

Figure 7. The difference of the confusion matrix without and with (50%) dropping of edges.

we directly use the Pytorch implementation [2] to update nodes respectively.

The proposed attention method consistently outperforms others even when the edges are dropped by 50%. There are some classes that are more robust to missing information, such as a chair, curtain, desk, floor, and wall while some classes are more dependant on others such as bathtub, bed, shower curtain, and window. We visualize this effect by plotting the difference of the confusion matrix with and without dropping of edges, see Fig. 7. It can be seen that beds and pictures are easier to predict when full edges are provided.

## 13. Runtime Analysis

In the following a more detailed runtime analysis is given in terms of the number of segments and data re-usage and graph structure update.

We report the analysis using scene `scene0645_01` which consists of 5230 paired RGB and depth images. The average update of node, edge, GNN features and the class predictions are listed on Tbl. 9. The computation time overtime is reported on Fig. 8. By updating node and edge features with our graph structure, the computation time is significantly reduced. Again, our scene graph prediction method runs in a different thread and will only block the main thread in the data copy and fusion stage.

## 14. Qualitative Result

We demonstrate more qualitative results in the 3D scene graph prediction on both 3RScan [10] and ScanNet [1]. Note that ScanNet does not have ground truth relationships.

|  | # computations | times (ms) |
|---|---|---|
| Node Feature | 2.13 | 2.49 |
| Edge Feature | 20.83 | 1.53 |
| GNN Feature 1 | 20.83 | 14.28 |
| GNN Feature 2 | 57.88 | 44.06 |
| Class Prediction | 57.88 | 9.74 |

Table 9. The average number and time of computation on each feature computation process on the sequence of `scene0645_01`

We therefore use the trained model with 3RScan to do inference on ScanNet scenes. Our method is able to handle the domain gap across these two datasets and predicts reasonable 3D scene graphs on ScanNet scenes.

The results are shown on Fig. 9, Fig. 10 and Fig. 11. Segments are represented by circles and estimated object instances are drawn as rectangles. In our visualization, the class prediction of a segment is correct if no label is shown in the circle and wrong otherwise.

As for relationship prediction, we use green, red and blue to indicate the correct, wrong, and unknown predictions respectively. An unknown prediction is a case where no ground truth data is available. The label on an edge without bracket is the predicted label, with bracket is its ground truth label. To simplify the visualization, we ignore *none*-relationships and merge segments with *same part* relationships in the same box. We also group up predictions of segments with the same label within the same box. The indication of such a grouped prediction is shown by connecting box to box. As for the wrongly predicted segments, their predicted probability remains individual. This indicated with an edge from a circle to a box.

| Function | Layer Definition |
|---|---|
| $g_v, g_a$ | MLP(768, 768, 512) |
| $g_e$ | MLP(1280, 768, 256) |
| $\hat{g}_q, \hat{g}_e$ | FC(512, 512) |
| $\hat{g}_\tau$ | FC(256, 256) |

Table 10. Parameters of the layers in our GNN. FC$(\cdot, \cdot)$ represents fully connected layer, and MLP$(\cdot, \ldots, \cdot)$ represent FC$(\cdot, \cdot)$ layers with ReLU activation between them.

## References

[1] Angela Dai, Angel Xuan Chang, Manolis Savva, Maciej Halber, Tom Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *CVPR*, 2017. 2, 6

[2] Matthias Fey and Jan E. Lenssen. Fast Graph Representation Learning with PyTorch Geometric. In *ICLRW*, 2019. 2

[3] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 1

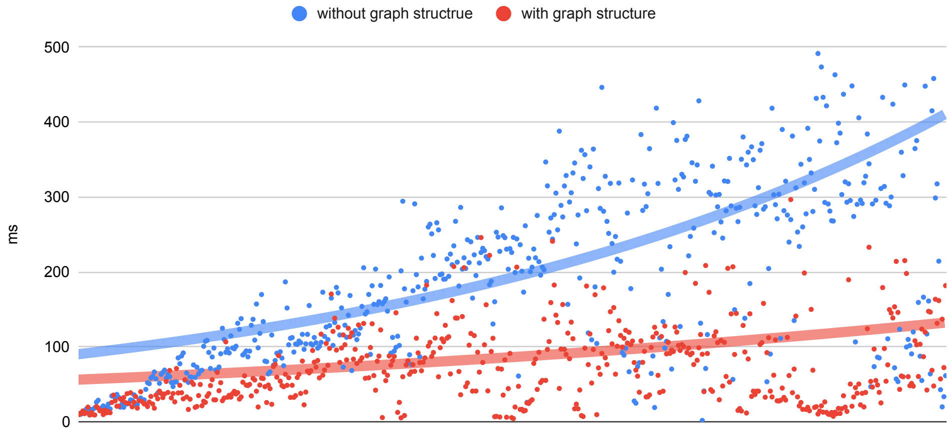

Figure 8. The computation time of the scene graph prediction over time.

| | bath | bed | bkshf | cab. | chair | cntr. | curt. | desk | door | floor | ofurn | pic. | refri. | show. | sink | sofa | table | toil | wall | wind. | avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| without | 50.0 | 3.9 | 0.0 | 27.0 | 51.7 | 16.7 | 62.2 | **20.0** | 16.4 | 96.2 | 15.4 | 8.0 | 4.3 | 11.1 | 52.5 | 45.9 | **54.2** | 41.7 | 67.0 | 26.2 | 33.5 |
| SDPA[8] | 50.0 | 11.1 | **2.3** | 26.0 | 45.7 | 17.7 | 65.2 | 3.9 | 18.7 | 87.4 | 11.2 | 4.8 | 2.5 | 29.4 | 38.1 | 60.8 | 36.8 | 65.0 | 60.1 | 24.0 | 33.0 |
| GAT[9] | 22.0 | 5.7 | 0.0 | 10.9 | 22.8 | 11.0 | 37.6 | 1.8 | 9.4 | 19.8 | 3.1 | 1.3 | 0.0 | 0.0 | 10.4 | 33.0 | 12.0 | 8.7 | 8.7 | 11.9 | 11.5 |
| ours | **83.3** | **24.3** | 0.0 | **43.4** | **69.8** | **30.0** | **68.7** | 4.5 | **29.6** | **98.1** | **26.6** | **10.0** | **34.5** | **66.7** | **65.0** | **74.7** | **54.2** | **86.5** | **75.3** | **41.7** | **49.3** |
| without@p50 | 37.5 | 3.7 | 0.0 | 24.8 | 49.9 | 13.3 | 52.3 | **17.3** | 14.9 | 86.8 | 19.9 | 5.0 | 5.8 | 6.2 | 46.7 | 36.4 | 37.3 | 46.0 | 63.5 | 22.8 | 29.5 |
| SDPA@p50 | 45.5 | 8.5 | 0.0 | 24.2 | 45.0 | 8.8 | **59.9** | 6.5 | 16.1 | 81.2 | 11.0 | 4.0 | 3.4 | 23.1 | 35.5 | 54.2 | 36.3 | 47.7 | 59.6 | 23.1 | 29.7 |
| GAT[9]@p50 | 26.5 | 2.3 | 0.0 | 14.6 | 21.9 | 4.6 | 34.9 | 0.0 | 8.4 | 18.6 | 4.7 | 1.2 | 7.1 | 17.6 | 7.9 | 30.2 | 11.2 | 8.9 | 17.1 | 12.9 | 12.5 |
| ours@p50 | **61.1** | **14.3** | **7.9** | **35.1** | **62.0** | **23.8** | 59.5 | 4.3 | **23.4** | **96.7** | **25.6** | **6.7** | **19.4** | **41.2** | **63.2** | **65.2** | **47.4** | **73.7** | **72.2** | **34.6** | **41.9** |

Table 11. Ablation study: Segment classification of InSeg [7] on 3RScan [10] reporting avg. IoU on segment-level.

| | metric | all | things | stuff | bath | bed | bkshf | cab. | chair | cntr. | curt. | desk | door | floor | ofurn | pic. | refri. | show. | sink | sofa | table | toil | wall | wind. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PanopticFusion [4] | PQ | 33.5 | **30.8** | **58.4** | 31.0 | **35.8** | **16.4** | **23.8** | 46.7 | 10.4 | **16.6** | **16.1** | 18.0 | **76.4** | **27.7** | 26.4 | **39.5** | 36.3 | 36.7 | **42.1** | **34.8** | **76.1** | **40.4** | **19.3** |
| Ours (NN mapping) | PQ | 31.5 | 30.2 | 43.4 | **67.6** | 25.4 | 13.9 | 22.2 | **47.2** | **10.5** | 16.4 | 12.6 | **26.4** | 56.4 | 22.9 | **31.3** | 28.0 | **38.3** | **38.0** | 32.3 | 34.8 | 63.2 | 30.4 | 11.7 |
| Ours (skip missing) | PQ | 36.3 | 51.0 | 34.7 | 68.4 | 28.0 | 16.0 | 26.4 | 58.1 | 15.6 | 24.7 | 17.7 | 28.7 | 64.5 | 26.9 | 35.4 | 30.8 | 40.7 | 41.3 | 38.8 | 45.6 | 66.2 | 37.4 | 15.2 |
| PanopticFusion [4] | SQ | **73.0** | **73.3** | 70.7 | 75.3 | **70.1** | **73.9** | **71.1** | 74.3 | **65.1** | 72.3 | 61.7 | 76.0 | **77.4** | **75.8** | 71.2 | 77.7 | **79.5** | 72.7 | **74.6** | **74.3** | **81.4** | 64.0 | **72.5** |
| Ours (NN mapping) | SQ | 72.9 | 73.0 | **72.6** | **80.6** | 68.2 | 66.9 | **71.1** | **76.5** | 61.7 | **75.1** | **63.8** | **77.4** | 74.8 | 71.6 | **81.5** | **77.8** | 79.1 | **75.4** | 65.3 | 73.3 | 80.2 | **70.4** | 68.2 |
| Ours (skip missing) | SQ | 76.1 | 77.9 | 75.9 | 82.9 | 71.2 | 69.1 | 74.6 | 81.2 | 62.3 | 74.0 | 68.0 | 81.0 | 81.8 | 74.4 | 82.7 | 82.3 | 81.5 | 77.2 | 70.2 | 80.9 | 82.4 | 74.0 | 69.5 |
| PanopticFusion [4] | RQ | **45.3** | **41.3** | **80.9** | 41.2 | **51.1** | **22.2** | **33.5** | **62.8** | 16.0 | **23.0** | **26.0** | 23.6 | **98.7** | **36.5** | 37.1 | **50.8** | 45.7 | **50.5** | **56.3** | 46.9 | **93.5** | **63.1** | **26.7** |
| Ours (NN mapping) | RQ | 42.2 | 40.3 | 59.3 | **83.9** | 37.2 | 20.7 | 31.3 | 61.7 | **17.1** | 21.8 | 19.7 | **34.1** | 75.4 | 31.9 | **38.5** | 36.0 | **48.5** | 50.3 | 49.5 | **47.5** | 78.9 | 43.2 | 17.1 |
| Ours (skip missing) | RQ | 46.8 | 64.7 | 44.8 | 82.5 | 39.4 | 23.2 | 35.4 | 71.6 | 25.0 | 33.3 | 26.0 | 35.4 | 78.8 | 36.1 | 42.8 | 37.4 | 50.0 | 53.5 | 55.3 | 56.4 | 80.4 | 50.6 | 21.9 |

Table 12. The full 3D panoptic segmentation results on the ScanNet v2 open test set.

[4] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things. In *IROS*, 2019. 1, 3

[5] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*, 2012. 1

[6] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the Convergence of Adam and Beyond. In *ICLR*, 2018. 1

[7] Keisuke Tateno, Federico Tombari, and Nassir Navab. Real-Time and Scalable Incremental Segmentation on Dense SLAM. In *IROS*, 2015. 1, 3

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NeurIPS*, 2017. 1, 3

[9] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018. 3

[10] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. RIO: 3D Object Instance Re-Localization in Changing Indoor Environments. In *ICCV*, 2019. 2, 3, 4, 5

[11] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D Semantic Scene Graphs from 3D Indoor Reconstructions. In *CVPR*, 2020. 1

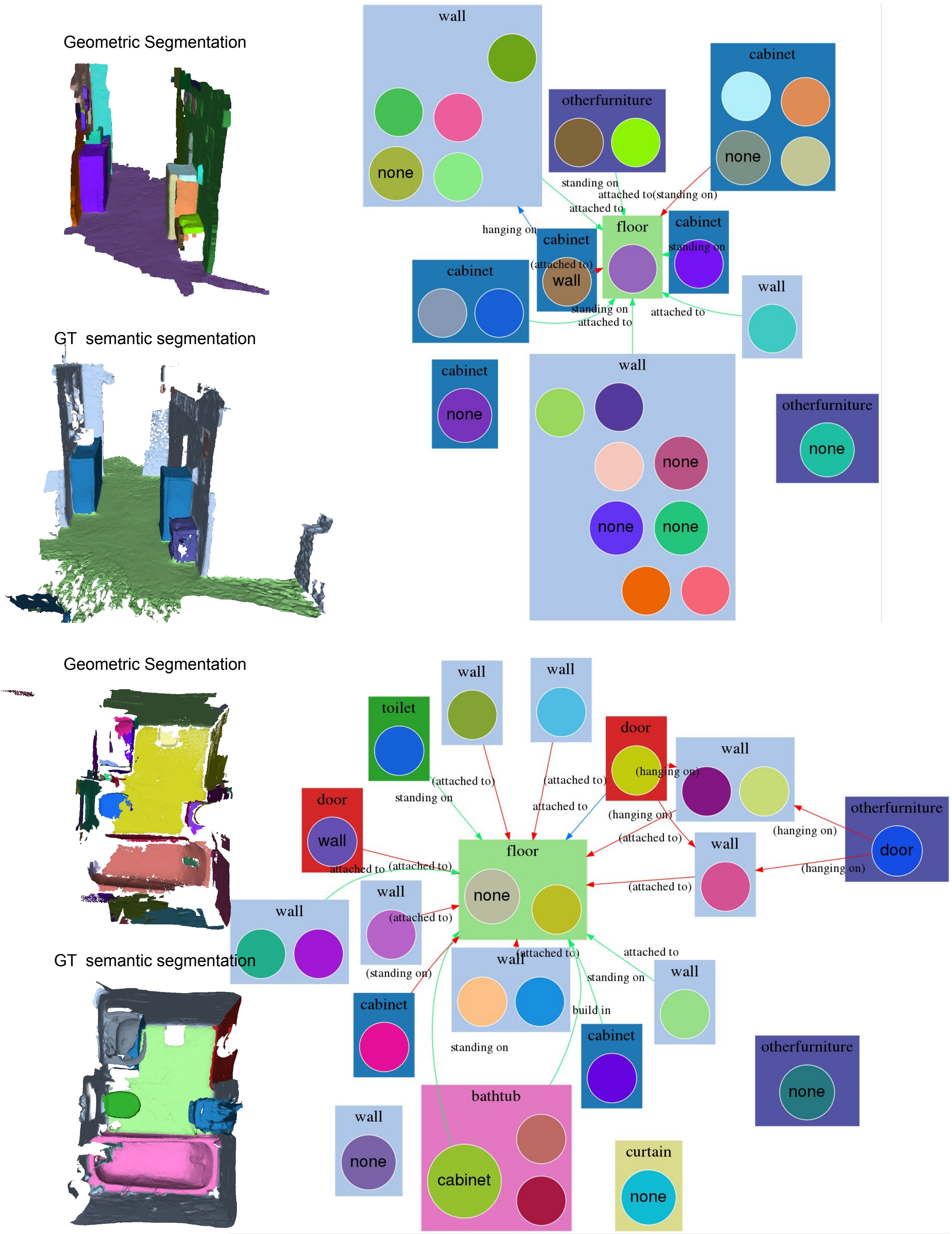

Figure 9. Qualitative results of our method on example scenes from 3RScan [10].

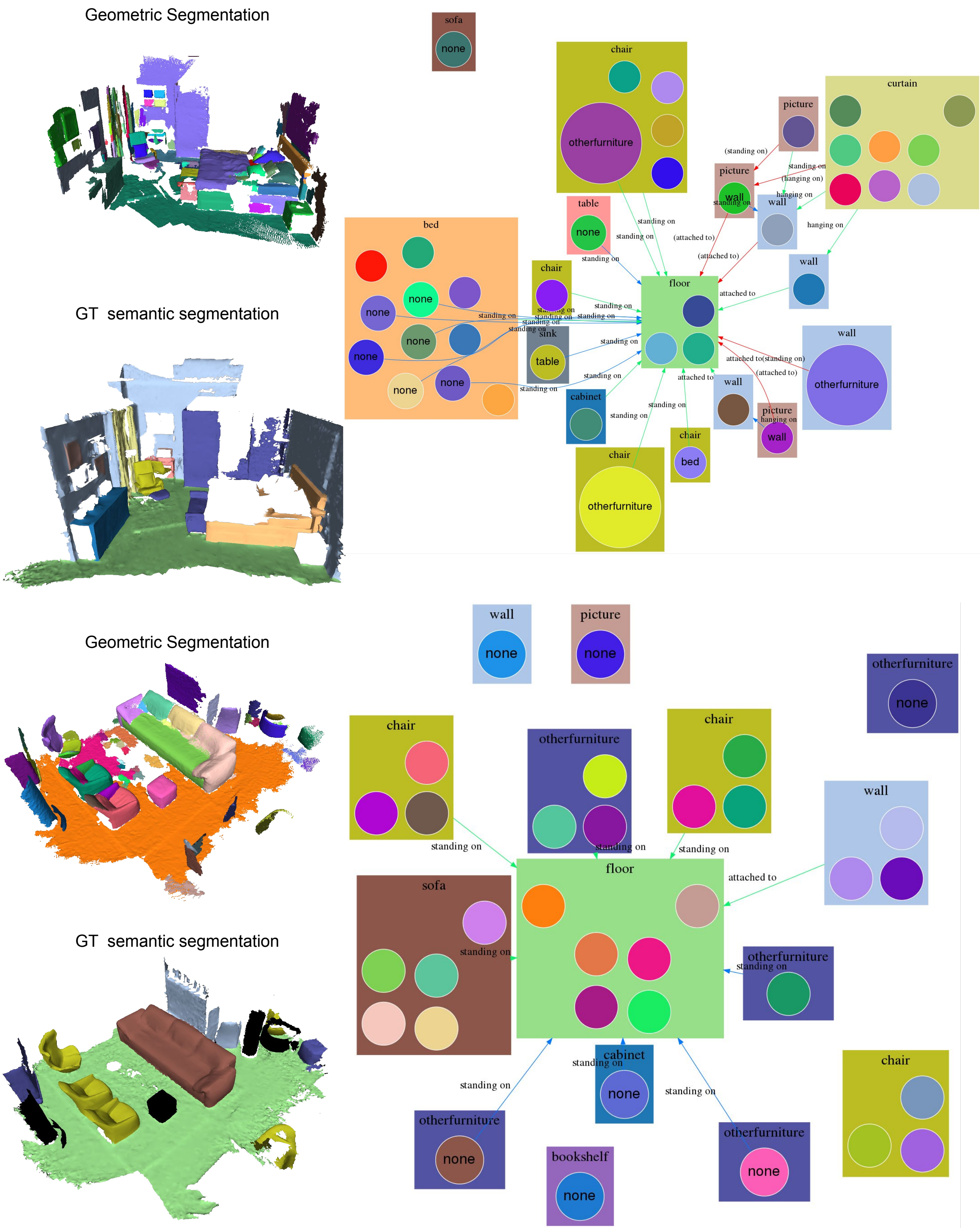

Figure 10. Qualitative results of our method on relative large scenes from 3RScan [10].

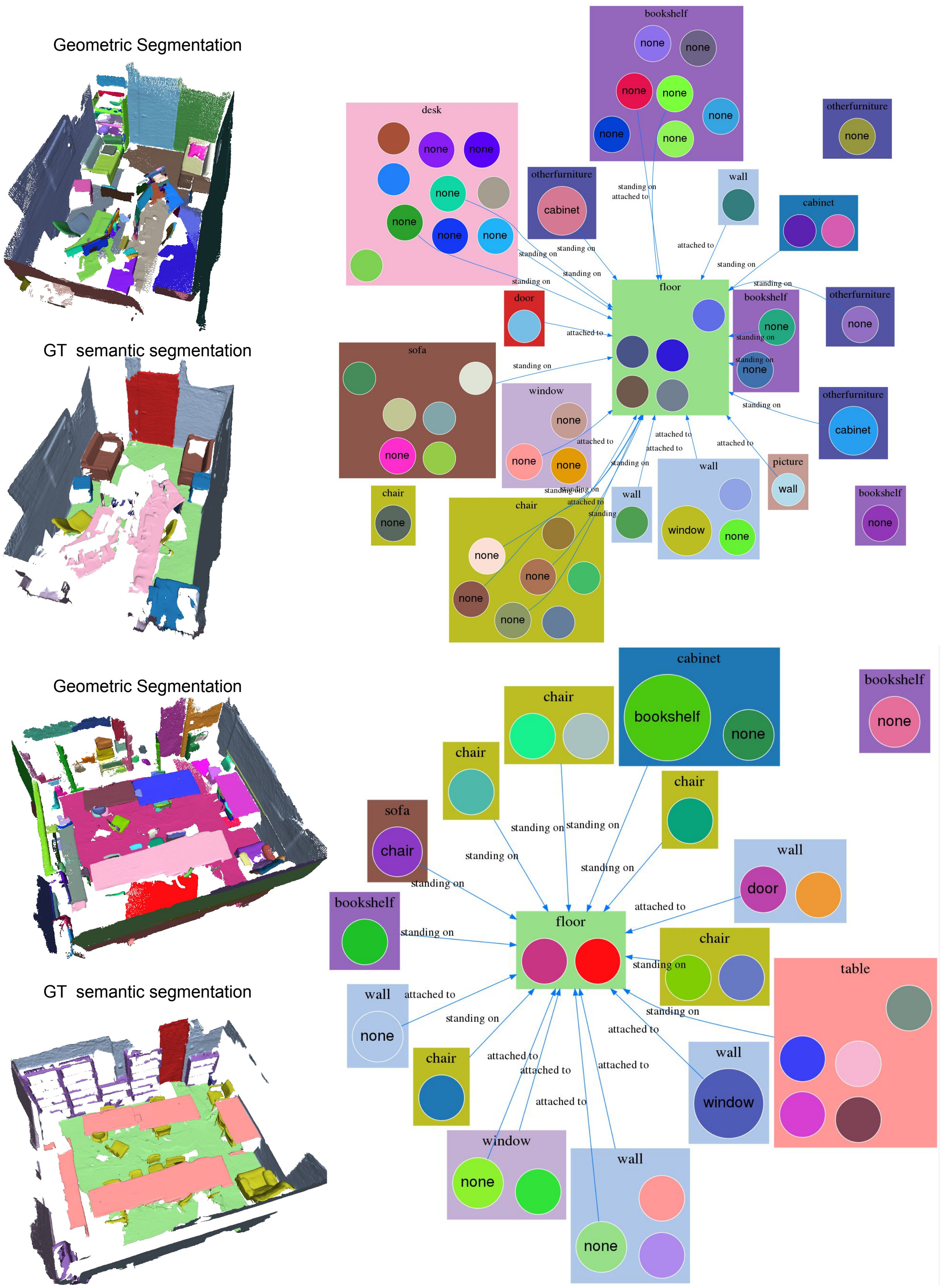

Figure 11. Qualitative results of our method on the scenes from ScanNet [1].