Supplementary Material for 'Towards Part-Based Understanding of RGB-D Scans'

In this supplemental material, we detail our network architecture in Section 1; in Section 2, we provide details of our baselines designs; in Section 4, we provide specifications of parts that we used in our experiments; in Section 5, we additionally provide more quantitative results, visualize examples of part priors combinations for each main category and examples of our predictions compared to ground-truth.

1. Network Architecture Details

We detail our network architecture specification in Tables 3-4. Table 3 describes the layers for encoding the detected objects to a feature code. The feature code is then input to a decoder which predicts the semantic part structure, as detailed in Table 5; here, the output of the last layer, `lin3`, represents a tuple of children latent codes, which predict part prior weights, as specified in Section 3.4 of the main paper. The final part refinement is then described in Table 4. Our volumetric object encoder and part refinement are fully convolutional, while the semantic part structure prediction operates on the latent feature representations of shapes and parts with MLP structure.

	mAP@25 (↑)						
Method	chair	table	cab.	bkshlf	bed	bin	**avg**
MLCVNet + StructureNet	45.7	25.7	19.8	50.0	36.4	53.0	38.4
RevealNet	70.3	40.6	90.5	87.2	22.7	20.6	55.3
Ours	78.4	47.2	90.5	77.8	22.7	72.4	64.8

Table 1: Evaluation of instance completion on Scan2CAD [1]. We evaluate object completion as a union of predicted part decompositions, in comparison with state-of-the-art instance completion [4] and the union of StructureNet [5] parts as instances.

2. Additional Baseline Training Details

In all our experiments in comparison with state of the art, we leveraged a combination of various approaches. For the task of Semantic Part Completion, we performed scan completion with SG-NN [3] and object detection with MLCVNet [8]. Our UNet baseline is developed as a baseline without any semantic part structure or geometric part prior inference; it consists of only a 3D voxel encoder (four convolutional blocks consisting of 3D convolutions (with 16, 32, 64, 128 output channels) using Group Normalization and ReLU activation) and 3D voxel decoder (five convolutional blocks consisting of 3D transposed convolutions (with 128, 64, 32, 16, 1 output channel(s), equipped with "add" skip connections) and a 3D convolution, using Group Normalization and ReLU activation) with 45 output feature channels, corresponding to binary masks for each possible part type, and trained with a binary cross entropy loss. The UNet bottleneck has a spatial resolution of $4 \times 4 \times 4$. Without the explicit part structure representations, this UNet baseline tends to predict noisy part masks, or part types from incorrect classes which remain functionally different.

Note that for experiments with StructureNet [5], we used the same experimental setup as described in their original paper, training different models for each class category. Since StructureNet operates in the canonical space of the objects, we provided our predicted object orientations from our approach to guide the StructureNet predictions.

3. Comparison to Sung et al. 2015

We compare with the approach of Sung et al. [7] on their benchmark for shape completion of chairs and tables. [7] follows a leave-one-out approach by training on all but one left-out shape; our approach is trained on PartNet objects

	Chamfer Distance (↓)								IoU (↑)							
Method	chair	table	cab.	bkshlf	bed	bin	class avg	inst avg	chair	table	cab.	bkshlf	bed	bin	class avg	inst avg
StructureNet [5]	0.019	0.089	0.048	0.032	0.069	0.105	0.061	0.049	18.5	1.0	10.1	16.8	6.8	12.1	10.9	12.8
Ours	0.029	0.089	0.055	0.037	0.058	0.081	0.058	0.048	27.6	8.0	17.3	20.9	19.8	28.7	20.4	22.6

Table 2: Evaluation on semantic part completion on Scan2CAD [1] with ground truth 3D object detection (oriented 3D bounding boxes) as input.

Encoder	Input Layer	Type	Input Size	Output Size	Kernel Size	Stride	Padding
conv0	scan occ. grid	Conv3D	(1, 32, 32, 32)	(16, 16, 16, 16)	(5, 5, 5)	(2, 2, 2)	(2, 2, 2)
gnorm0	conv0	GroupNorm	(16, 16, 16, 16)	(16, 16, 16, 16)	-	-	-
relu0	gnorm0	ReLU	(16, 16, 16, 16)	(16, 16, 16, 16)	-	-	-
pool1	relu0	MaxPooling	(16, 16, 16, 16)	(16, 8, 8, 8)	(2, 2, 2)	(2, 2, 2)	(0, 0, 0)
conv1	pool1	Conv3D	(16, 8, 8, 8)	(32, 8, 8, 8)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
gnorm1	conv1	GroupNorm	(32, 8, 8, 8)	(32, 8, 8, 8)	-	-	-
relu1	gnorm1	ReLU	(32, 8, 8, 8)	(32, 8, 8, 8)	-	-	-
pool2	relu1	MaxPooling	(32, 8, 8, 8)	(32, 4, 4, 4)	(2, 2, 2)	(2, 2, 2)	(0, 0, 0)
conv2	pool2	Conv3D	(32, 4, 4, 4)	(64, 2, 2, 2)	(5, 5, 5)	(2, 2, 2)	(2, 2, 2)
gnorm2	conv2	GroupNorm	(64, 2, 2, 2)	(64, 2, 2, 2)	-	-	-
relu2	gnorm2	ReLU	(64, 2, 2, 2)	(64, 2, 2, 2)	-	-	-
pool3	relu2	MaxPooling	(64, 2, 2, 2)	(64, 1, 1, 1)	(2, 2, 2)	(2, 2, 2)	(0, 0, 0)
conv3	pool3	Conv3D	(64, 1, 1, 1)	(128, 1, 1, 1)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
gnorm3	conv3	GroupNorm	(128, 1, 1, 1)	(128, 1, 1, 1)	-	-	-
relu3	gnorm3	ReLU	(128, 1, 1, 1)	(128, 1, 1, 1)	-	-	-
flat0	node feature	Flatten	(128, 1, 1, 1)	(128)	-	-	-

Table 3: Layer specification for detected object encoder.

Child decoder	Input Layer	Type	Input Size	Output Size
lin0	node feature	Linear	128	1280
relu0	lin0	ReLU	1280	1280
reshape0	relu0	Reshape	1280	(10, 128)
node_exist	reshape0	Linear	(10, 128)	(10, 1)
concat0	(reshape0, reshape0)	Concat.	(10, 128), (10, 128)	(10, 10, 256)
lin1	concat0	Linear	(10, 10, 256)	(10, 10, 128)
relu1	lin1	ReLU	(10, 10, 128)	(10, 10, 128)
edge_exist	relu1	Linear	(10, 10, 128)	(10, 10, 1)
mp	(relu1, edge_exist, reshape0)	Mes. Passing	(10, 10, 128), (10, 10, 1), (10, 128)	(10, 384)
lin2	mp	Linear	(10, 384)	(10, 128)
relu2	lin2	ReLU	(10, 128)	(10, 128)
node_sem	relu2	Linear	(10, 128)	(10, #classes)
lin3	relu2	Linear	(10, 128)	(10, 128)
relu3	lin3	ReLU	(10, 128)	(10, 128)

Table 4: Layer specification for decoding an object into its semantic part structure.

Prior refiner	Input Layer	Type	Input Size	Output Size	Kernel Size	Stride	Padding
concat0	(prior, scan occ. grid)	Concat.	(1, 32, 32, 32), (1, 32, 32, 32)	(2, 32, 32, 32)	-	-	-
conv0	concat0	Conv3D	(2, 32, 32, 32)	(8, 32, 32, 32)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
bnorm0	conv0	BatchNorm	(8, 32, 32, 32)	(8, 32, 32, 32)	-	-	-
relu0	bnorm0	ReLU	(8, 32, 32, 32)	(8, 32, 32, 32)	-	-	-
conv1	relu0	Conv3D	(8, 32, 32, 32)	(16, 32, 32, 32)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
bnorm1	conv1	BatchNorm	(16, 32, 32, 32)	(16, 32, 32, 32)	-	-	-
relu1	bnorm1	ReLU	(16, 32, 32, 32)	(16, 32, 32, 32)	-	-	-
conv2	relu1	Conv3D	(16, 32, 32, 32)	(8, 32, 32, 32)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
bnorm2	conv2	BatchNorm	(8, 32, 32, 32)	(8, 32, 32, 32)	-	-	-
relu2	bnorm2	ReLU	(8, 32, 32, 32)	(8, 32, 32, 32)	-	-	-
conv3	relu2	Conv3D	(8, 32, 32, 32)	(1, 32, 32, 32)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
add3	(prior, conv3)	Add	(1, 32, 32, 32), (1, 32, 32, 32)	(1, 32, 32, 32)	-	-	-
sigmoid3	add3	Sigmoid	(1, 32, 32, 32)	(1, 32, 32, 32)	-	-	-

Table 5: Layer specification for final part mask refinement.

that do not intersect with any of the evaluation instances. Our approach outperforms [7], with Chamfer Distance of 0.77 and 0.76 in comparison with 0.86 and 0.85 of [7] on chairs and tables, respectively. We show additional qualitative comparisons in Figure 1.

4. Part Types

In Figure 2, we visualize all part types which we trained on. Note that the classes 'cabinet' and 'bookshelf' share the same set of parts, so we use the same part types and priors.

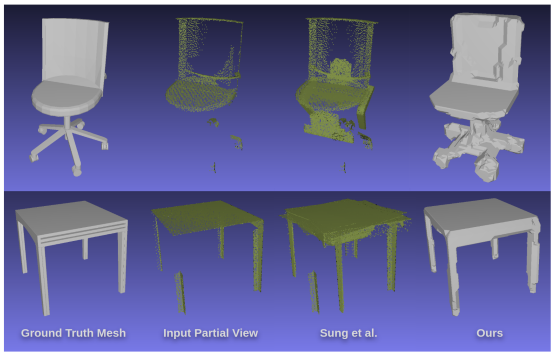

Figure 1: Qualitative comparison with Sung et al. [7] on their benchmark for shape completion. The larger missing regions (chair legs, table leg) are challenging, and [7] struggles to fit the correct structures, whereas our strong priors on semantic part structure and geometric part priors provide a coherent shape prediction.

5. Additional Results

Additional Quantitative Results In Table 1 we additionally evaluate object instance completion using an mAP@25 metric, in comparison to state-of-the-art RevealNet [4] and a combination of MLCVNet [8] with StructureNet [5]. Additionally, in Table 2, we evaluate our approach with ground truth 3D detection, i.e., ground truth oriented 3D bounding boxes for each object in the scene. Under ground truth detection, our structural part priors enable more robust part decomposition than StructureNet [5].

Additional Part Prior Visualizations We show additional examples of computed part priors for each object class category in Figure 3. All priors are visualized with three level-sets.

Additional Qualitative Semantic Part Completion Results Figure 4 shows additional examples of our predictions compared with ground-truth. Our method predicts meaningful part completion across a variety of object categories.

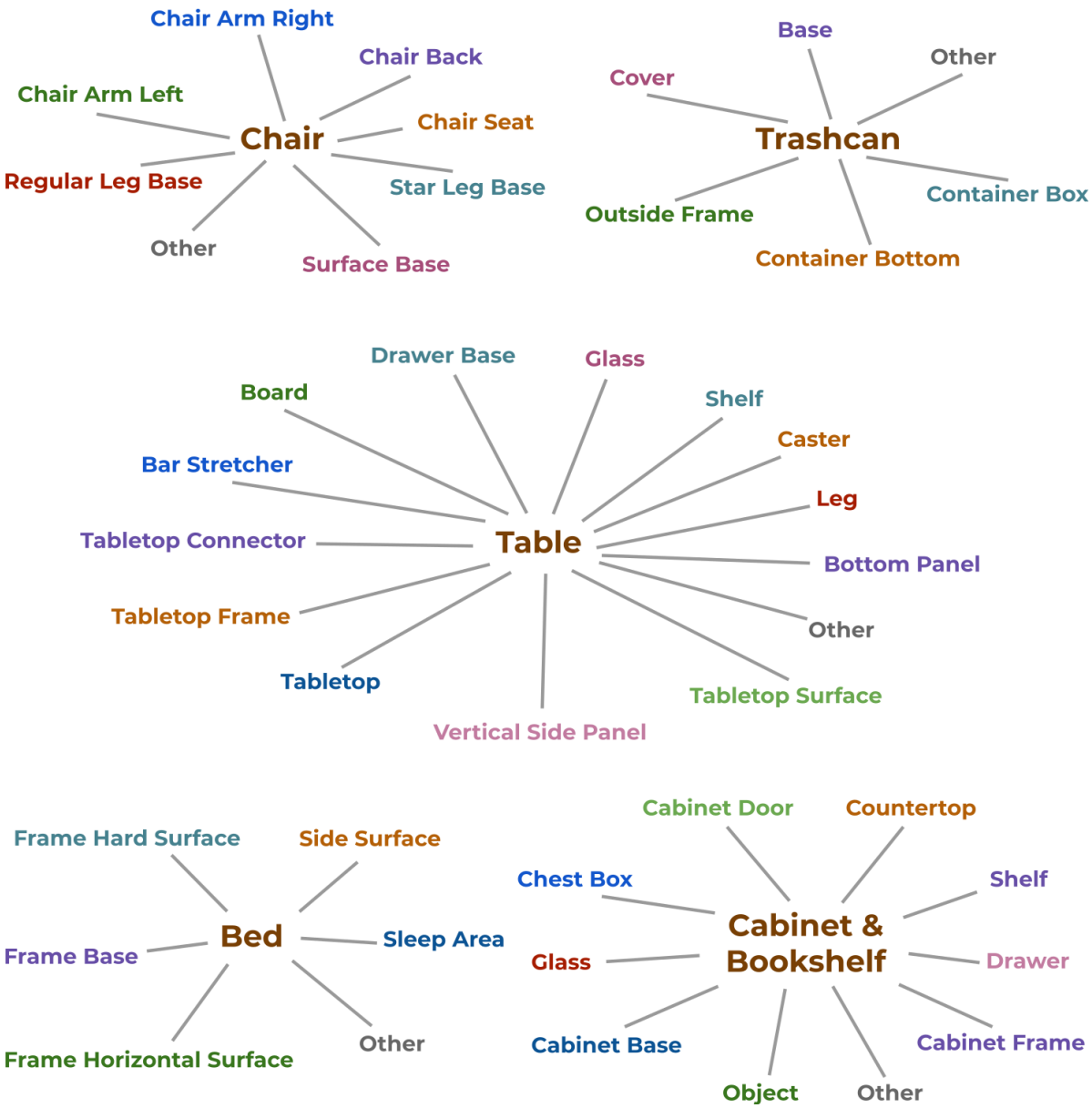

Figure 2: Part specification for the parts used in our approach. Note that 'cabinet' and 'bookshelf' classes have the same set of parts.

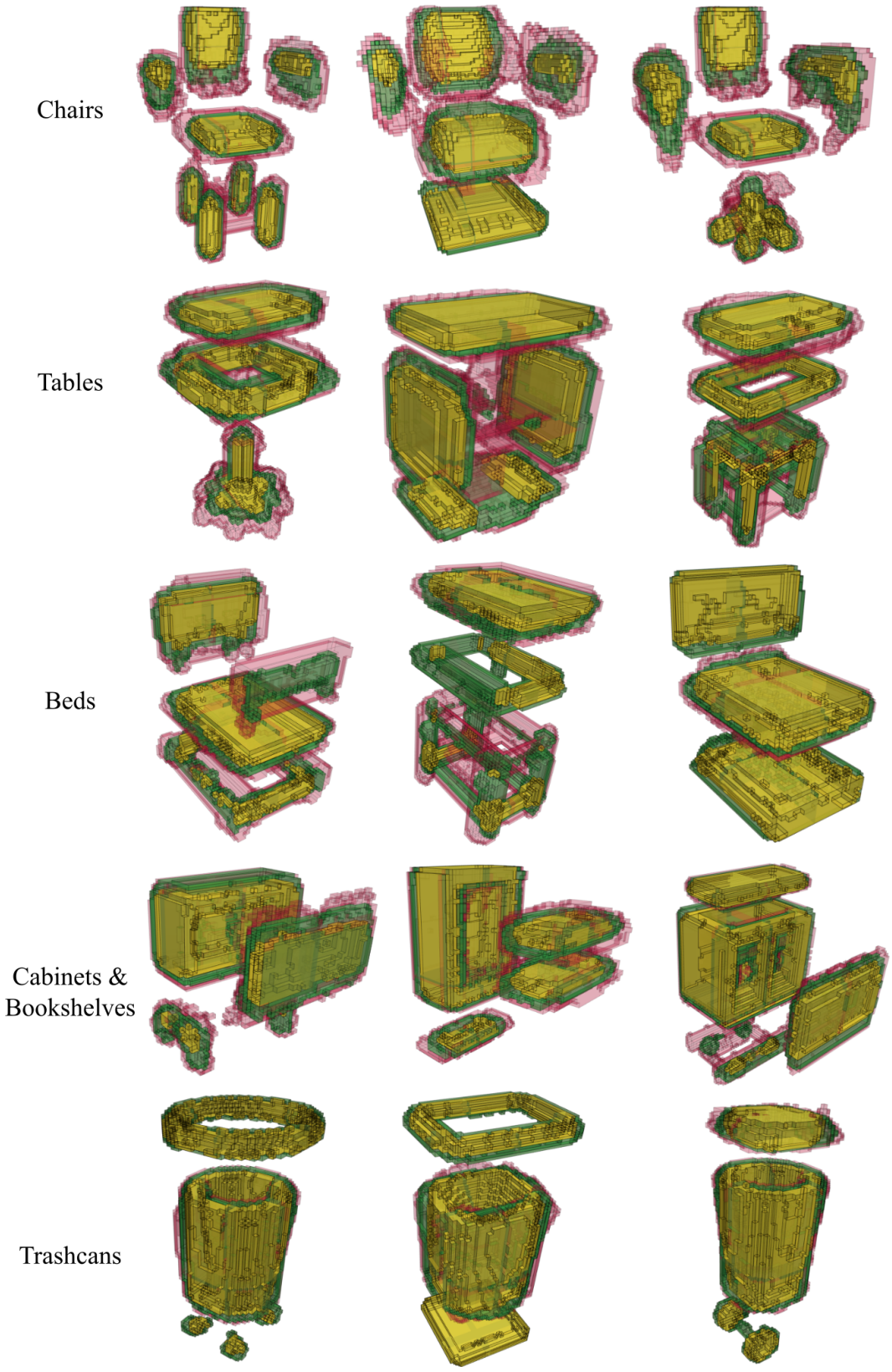

Figure 3: Visualization of various part priors.

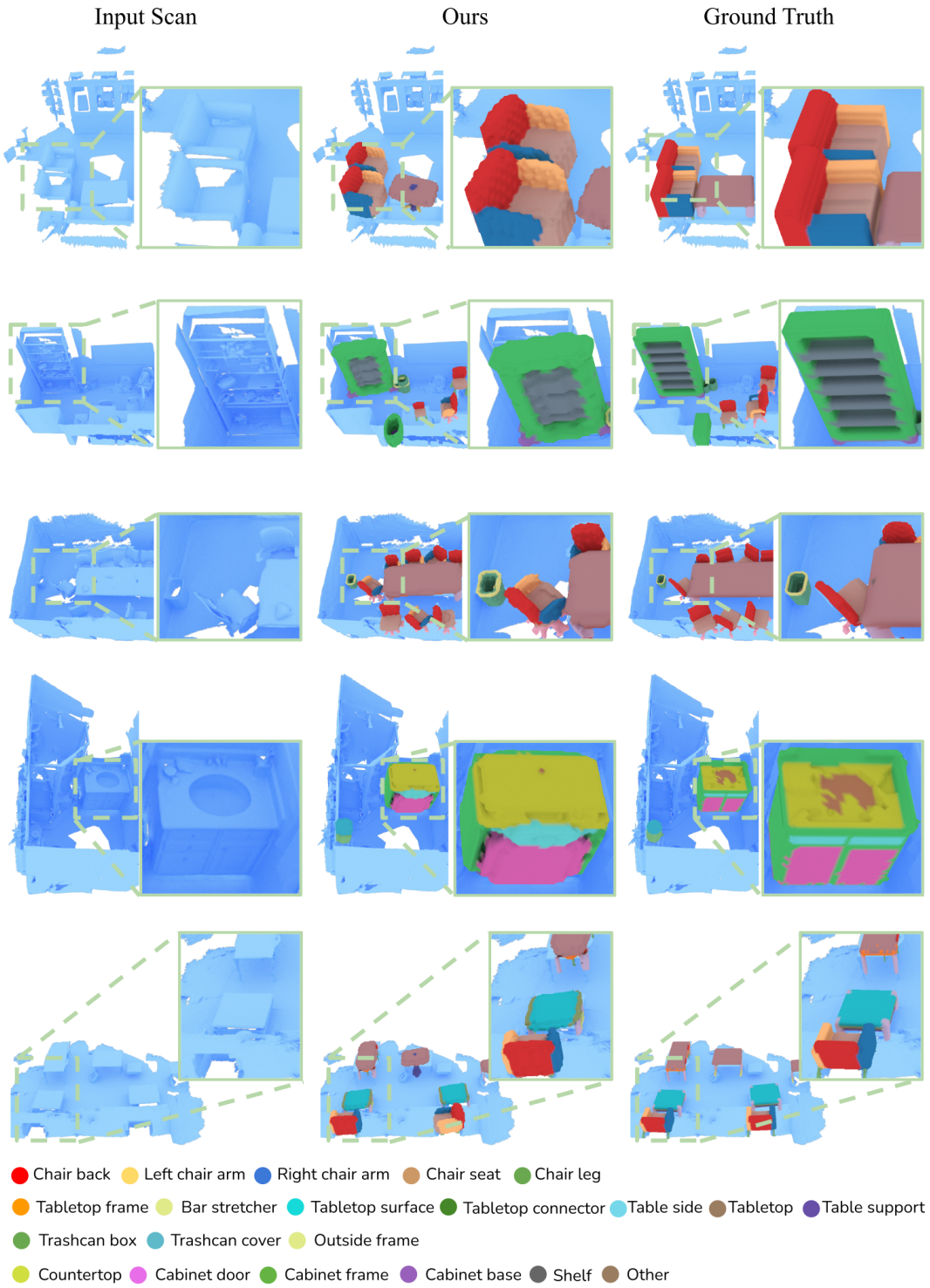

Figure 4: Additional qualitative results for our method on ScanNet [2] scenes and ground truth from Scan2CAD [1] and PartNet [6].

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Nießner. Scan2cad: Learning CAD model alignment in RGB-D scans. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2614–2623, 2019. 1, 6

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 6

[3] Angela Dai, Christian Diller, and Matthias Nießner. Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2020. 1

[4] Ji Hou, Angela Dai, and Matthias Nießner. Revealnet: Seeing behind objects in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2098–2107, 2020. 1, 3

[5] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*, 2019. 1, 3

[6] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019. 6

[7] Minhyuk Sung, Vladimir G Kim, Roland Angst, and Leonidas Guibas. Data-driven structural priors for shape completion. *ACM Transactions on Graphics (TOG)*, 34(6):1–11, 2015. 1, 2, 3

[8] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level context votenet for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10447–10456, 2020. 1, 3