Voxel-based Network for Shape Completion by Leveraging Edge Generation - Supplementary Material

Xiaogang Wang Marcelo H Ang Jr Gim Hee Lee
National University of Singapore
xiaogangw@u.nus.edu {mpeangh,gimhee.lee}@nus.edu.sg

1. Ground Truth Edge Generation

We obtain the ground truth edges $\widehat{\mathrm{P}_e}$ with [1]. Specifically, object edges are identified by evaluating the query point p from its k-nearest neighbors. We first find the k-nearest neighbors of each query point from the object and denote c as the center of these neighbors. We then calculate the minimum distance v among all the neighboring points to the query point. A query point p is classified as the edge point if $||c - p|| > \lambda \cdot v$. We set $\lambda = 5$ and $k = 100$ for our created dataset, and set $\lambda = 1.8$ and $k = 150$ for the Completion3D dataset.

2. Evaluation Metrics

Following previous methods [10, 6, 7, 9], we use the CD and Fréchet Point Cloud Distance (FPD) [7, 5] as the evaluation metrics for the synthetic datasets and fidelity and registration errors for the KITTI dataset.

FPD. FPD evaluates the distribution similarity by the 2-Wasserstein distance between the real and fake Gaussian measured in the feature spaces of the point sets, *i.e.*

$$\mathrm{FPD}(\mathrm{X}, \mathrm{Y}) = \|\mathrm{m_X} - \mathrm{m_Y}\|_2^2 + \mathrm{Tr}(\Sigma_\mathrm{X} + \Sigma_\mathrm{Y} - 2(\Sigma_\mathrm{X}\Sigma_\mathrm{Y})^{\frac{1}{2}}). \quad (1)$$

Fidelity. Fidelity measures the average distance from each point in the input to its nearest neighbor in the output. It evaluates how well the input points are preserved in the output.

Registration Errors. Rotation and translation errors are the evaluation metrics for the point cloud registration. More specially, it measures the registration performances between neighboring frames in the same Velodyne sequence. Two types of inputs are evaluated: the partial points from the raw scans, and the generated complete points by different models. The rotation error is computed as $2cos^{-1}(2 < q_1, q_2 >^2 -1)$, where q_1 and q_2 are the ground truth rotation and the rotation computed from ICP, respectively. The translation error is computed as $||t_1 - t_2||_2$, in which t_1 is the ground truth translation and t_2 is the translation generated by ICP, respectively.

3. More Details of Conversion between Points and Grids

1) Conversion from points to grids. We calculate the initial grid features as the coordinate differences between points and their corresponding eight nearby grid vertexes in five different scales. This results in five tensors of sizes $3 \times 2048 \times 8$, $3 \times 1024 \times 8$, $3 \times 512 \times 8$, $3 \times 256 \times 8$ and $3 \times 128 \times 8$, respectively. Corresponding to the different point resolutions $\{2048, 1024, 512, 256, 128\}$, the five voxel resolutions are $\{32^3, 16^3, 8^3, 4^3, 2^3\}$. A set of grid features $\mathrm{P}_f^i, \{i = 0, 1, 2, 3, 4\}$ are obtained from the initial grid features by several convolutional blocks (§3.2). The quantitative comparison between our proposed grid transformation and the Gridding of GRNet is shown in rows 3 and 4 of Table 6 in the main paper, i.e., our transformation achieves 24.5% relative improvement compared to Gridding (3.600 vs 4.768).

2) Conversion from grids to points. GRNet predicts 262,144 (64^3) points from every vertex feature, and then samples 2048 points as the coarse output P_C and use MLPs to generate dense points from P_C. In contrast, we directly predict the dense points by adding the point offsets to the grid centers. The number of points for each grid cell are decided by the binary score p_c and density value δ_c (§3.3.3).

4. Network Architecture Details

We express the 3D convolutions with its number of output channels, the kernel size, the stride and padding values. For example, C3D(O1K3S1P1) indicates a 3D convolutional layer with the number of output channel as 1, kernel size as $3 \times 3 \times 3$, the stride and padding values as 1. DC3D

	PCN [10]	PCN-FC [10]	CDA [2]	TopNet [6]	CRN [7]	GRNet [9]	DPC [11]	MSN [3]	VE-PCN
Para. (M)	6.85	53.2	51.85	9.96	5.14	76.71	6.66	30.32	35.00
Time (ms)	57.5	21.4	614.9	63.1	61.3	124.3	331.3	346.7	450.1

Table 1: Space and time comparisons of different methods.

Methods	CD
full pipeline	**2.669**
without $\mathcal{L}_{\text{CD}}^{\text{S}}$	2.886
without $\mathcal{L}_{\text{CD}}$	3.408
without $\mathcal{L}_d$	2.843
without $\mathcal{L}_{\text{BCE}}^{\text{E}}$	2.842

Table 2: Ablation studies on the different losses. Results are obtained by evaluating mean CD per point (10^{-4}) on our dataset.

represents a 3D deconvolutional layer. We set the dilation value for all our convolutions as 1 except the first 3D convolution in the residual blocks of the edge generator.

The network architectures of the edge generator and shape encoder are shown in Figures 1 and 2, respectively. Figure 3 shows architectures of the refinement cells and the shape decoder. Figure 3 (a) and (b) show the first four refinement cells and the last refinement cell, respectively. Every cell shares similar architectures but with different feature dimensions. C_1, O_1 and O_2 in the first four refinement cells are $\{128, 128, 128\}$, $\{256, 128, 128\}$, $\{192, 64, 64\}$ and $\{129, 32, 32\}$, respectively.

Figure 3 (c) and (d) show the architecture of point generator in the shape completion module and the edge generator. Figure 3 (c) illustrates the prediction architectures for the classification score p_c and density value δ_c of each grid cell. Figure 3 (d) presents the point set generation for each grid. Prior to feeding the grid features into the convolutional layers, grid features are concatenated with 2 dimensional randomly sampled values to increase the point diversity in a local patch.

5. Time and Space Complexity Analysis

The number of parameters and inference time of different methods are shown in Table 1. Some layers in our network are 1D CNNs instead of 3D CNNs (e.g. the architecture of point generator) and thus we consume much smaller parameters than voxel-based methods (CDA and GRNet). We compute the average inference time of 5000 forward steps on a Titan X GPU, and our time complexity is comparable to other methods.

6. More Ablation Studies

More ablation studies on different losses are shown in Table 2. We test the effects of $\mathcal{L}_{\text{CD}}^{\text{S}}$, $\mathcal{L}_{\text{CD}}$, $\mathcal{L}_o$ and $\mathcal{L}_{\text{BCE}}^{\text{E}}$ by setting the corresponding weights to be 0. The results are obtained by testing on our created dataset.

7. More Experimental Results

7.1. Results on the PCN Dataset

We show the results of our method on the PCN dataset in Table 3. All the other results are cited from the state-of-the-art work PMP-Net [8]. We achieve lower average CD errors compared to all prior works and obtain better performances on the majority of object categories.

7.2. More Results on the Completion3D Dataset

More qualitative results on the validation data are shown in Figures 4 and 5.

7.3. More Results on our Dataset

Table 4 shows the FPD evaluations from various methods. More qualitative results on seen categories of our dataset are shown in Figures 6 and 7. More qualitative results on unseen categories are shown in Figure 8.

To further test the robustness of different models, we occlude the partial input with different occlusion ratios p that ranges from 20% to 40%. We directly adopt the models trained on seen categories for testing. The quantitative results are shown in Table 5. More qualitative results are shown in Figures 9 and 10.

7.4. More Results on the KITTI Dataset

Figure 11 shows the completion results on the KITTI dataset. We evaluate the performances by calculating the registration errors following PCN [10].

7.5. More Results on the Edge Generation

We show more experimental results on point cloud edge generation in Figure 12.

8. Comparisons to SK-PCN [4]

SK-PCN proposes a similar thought that adopts skeleton generations to help the shape completion. However, our edges are different from their meso-skeleton. The differences are shown in Figure 13 (The results of SK-PCN are Fig. 9 of their paper). Their meso-skeleton focus on the overall shapes. In contrast, our edges focus on high frequency components (*e.g.* thin structures), which are difficult to generate in existing methods. This can be evidenced from the limitation cases in Fig. 10 of SK-PCN

Methods	Mean Chamfer Distance (CD) per point (10^{-3})								
	Average	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Vessel
FoldingNet	14.31	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99
TopNet	12.15	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12
AtlasNet	10.85	6.37	11.94	10.10	12.06	12.37	12.99	10.33	10.61
PCN	9.64	5.50	10.63	8.70	11.00	11.34	11.68	8.59	9.67
GRNet	8.83	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04
CRN	8.51	**4.79**	9.97	**8.31**	9.49	8.94	10.69	7.81	8.05
PMP-Net	8.66	5.50	11.10	9.62	9.47	**6.89**	10.74	8.77	**7.19**
Ours	**8.32**	4.80	**9.85**	9.26	**8.90**	8.68	**9.83**	**7.30**	7.93

Table 3: Quantitative results on the PCN dataset.

	PCN [10]	PCN-FC [10]	CDA [2]	TopNet [6]	CRN [7]	GRNet [9]	DPC [11]	MSN [3]	VE-PCN
FPD	5.584	6.634	9.142	7.7536	3.054	6.513	8.347	3.904	**1.882**

Table 4: FPD comparisons on different methods. The lower, the better.

Ratios	Mean Chamfer Distance per point (10^{-4})								
	PCN [10]	PCN-FC [10]	CDA [2]	TopNet [6]	CRN [7]	GRNet [9]	DPC [11]	MSN [3]	VE-PCN
20%	5.642	6.230	7.485	6.732	3.554	4.016	2.975	3.213	**2.565**
30%	5.991	6.704	7.771	7.245	3.932	4.646	4.681	4.084	**3.055**
40%	7.066	7.934	8.741	8.622	**5.094**	10.746	8.210	6.546	5.129

Table 5: Quantitative comparison of occluded point clouds under different occlusion rates.

paper. Moreover, SK-PCN generates the complete points by learning displacements from skeletal points with a local adjustment strategy. In contrast, we synthesize the complete points by injecting the edge features into the completion decoder with a voxelization strategy. This voxel structure further enables our point generation to be confined within well-defined spaces of the grid cells, and thus eliminates the generation of spurious points that are commonly seen in other methods.

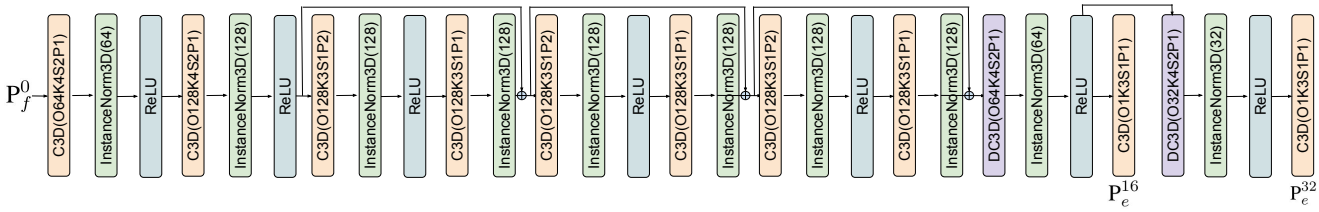

Figure 1: Architecture details of the edge generator.

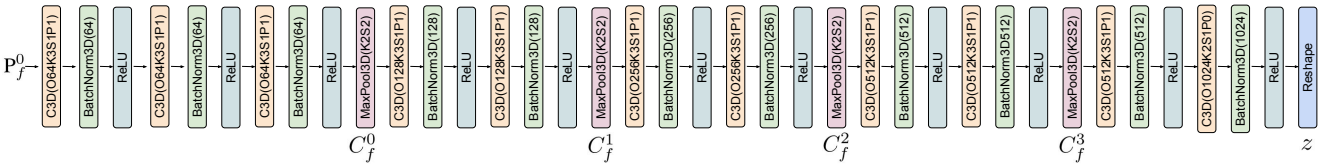

Figure 2: Architecture details of the shape encoder.

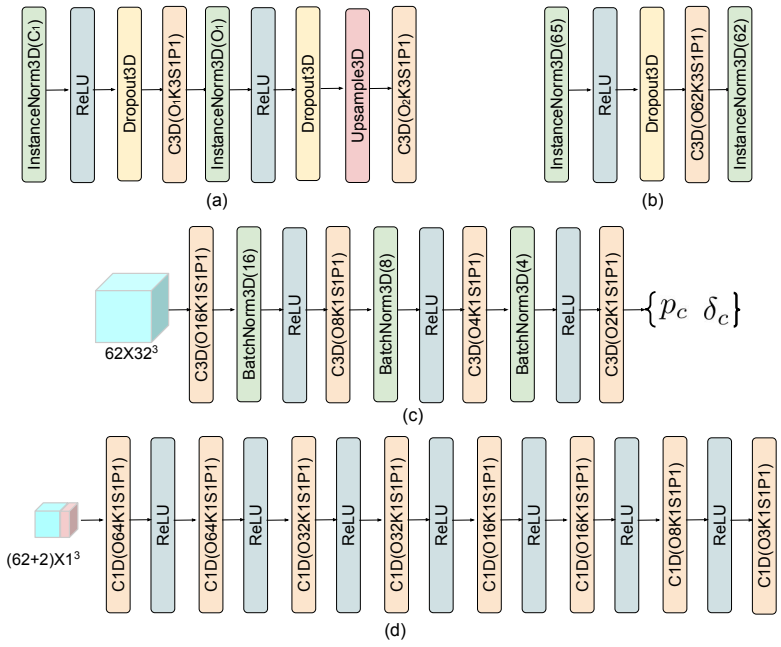

Figure 3: Architecture details of the refinement cells and shape decoder.

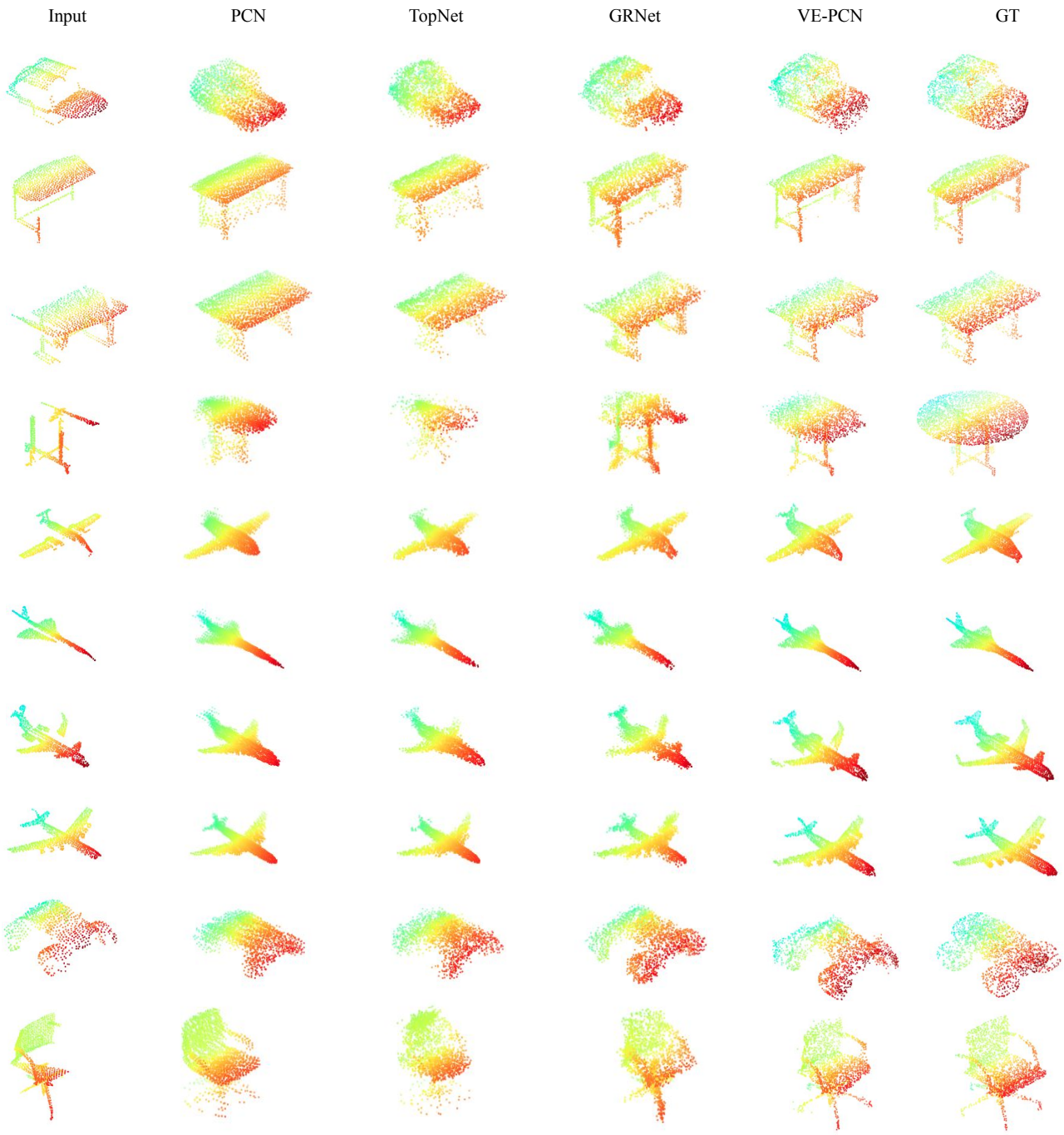

Figure 4: Qualitative comparisons on the Completion3D dataset (1/2).

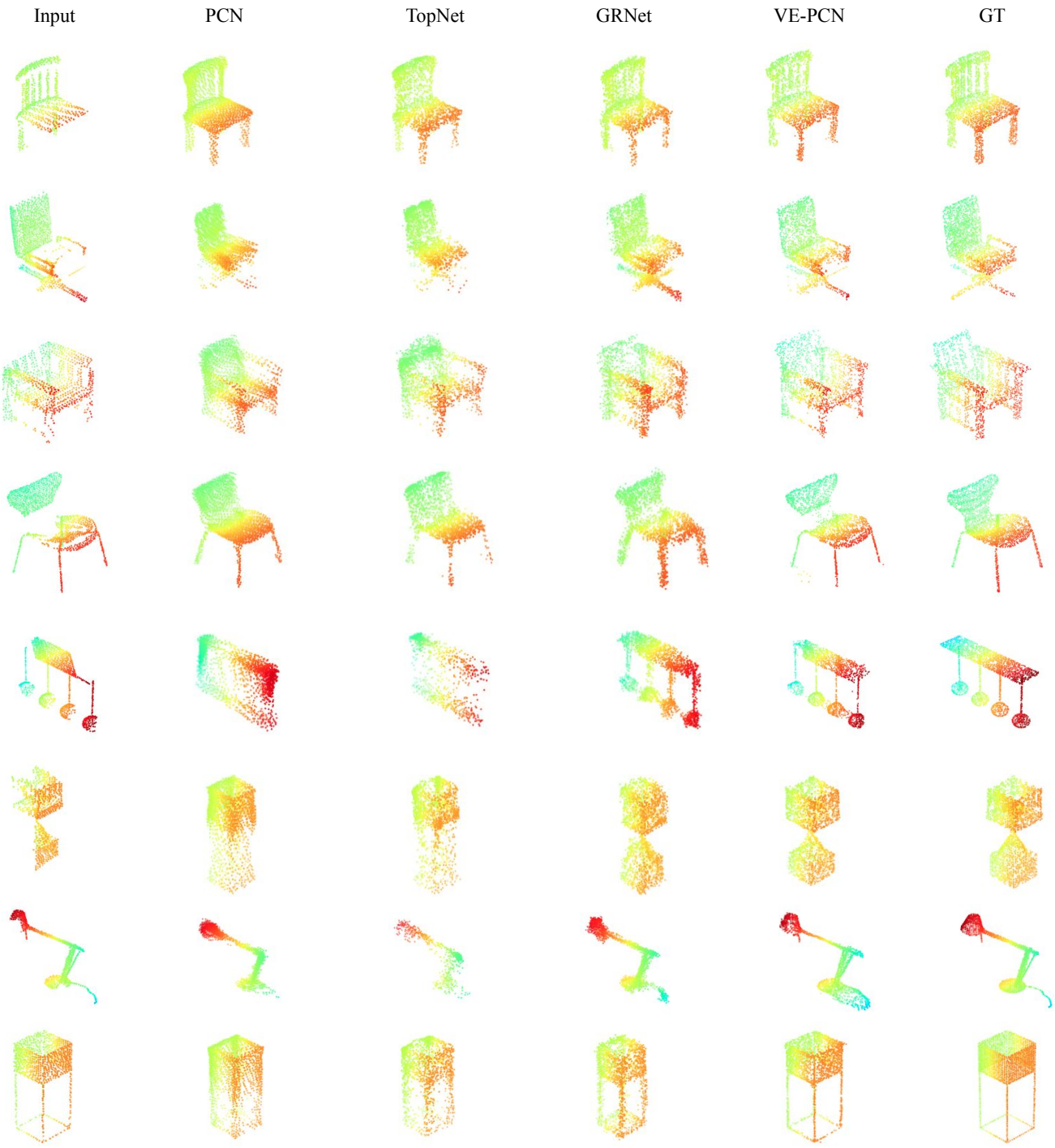

Figure 5: Qualitative comparisons the Completion3D dataset (2/2).

Figure 6: Qualitative comparisons on seen categories of our dataset (1/2).

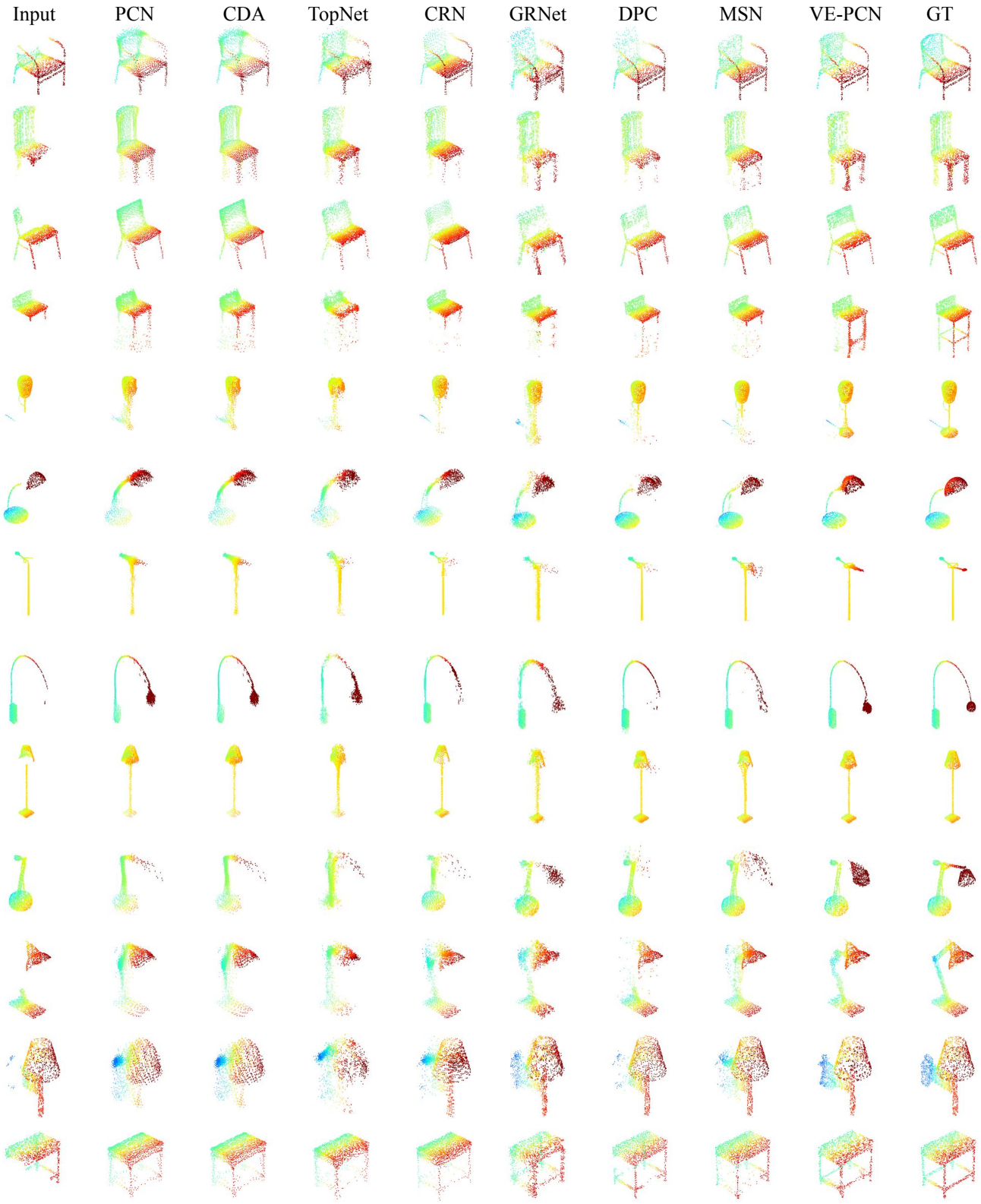

Figure 7: Qualitative comparisons on seen categories of our dataset (2/2).

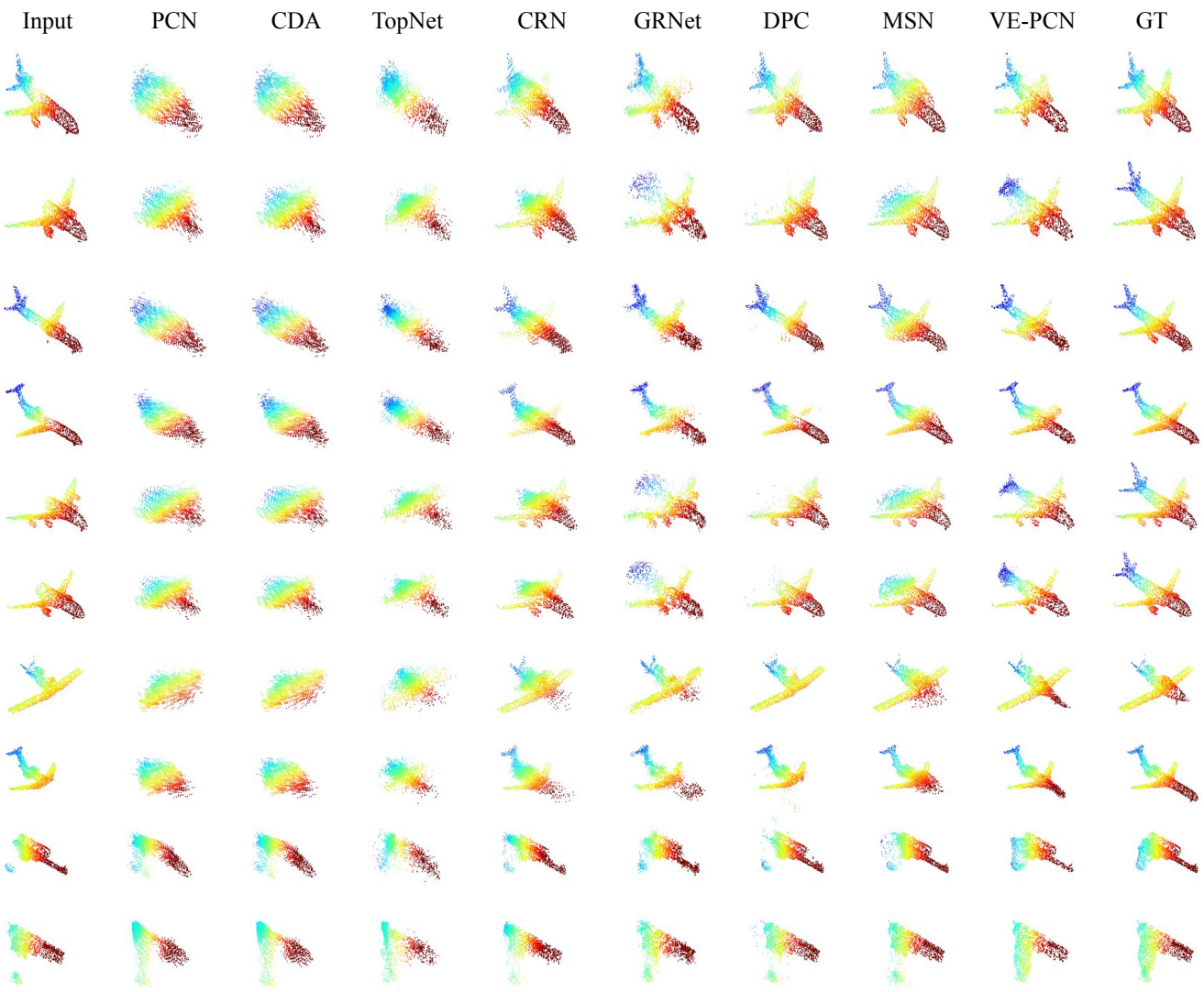

Figure 8: Qualitative comparisons on unseen categories of our dataset.

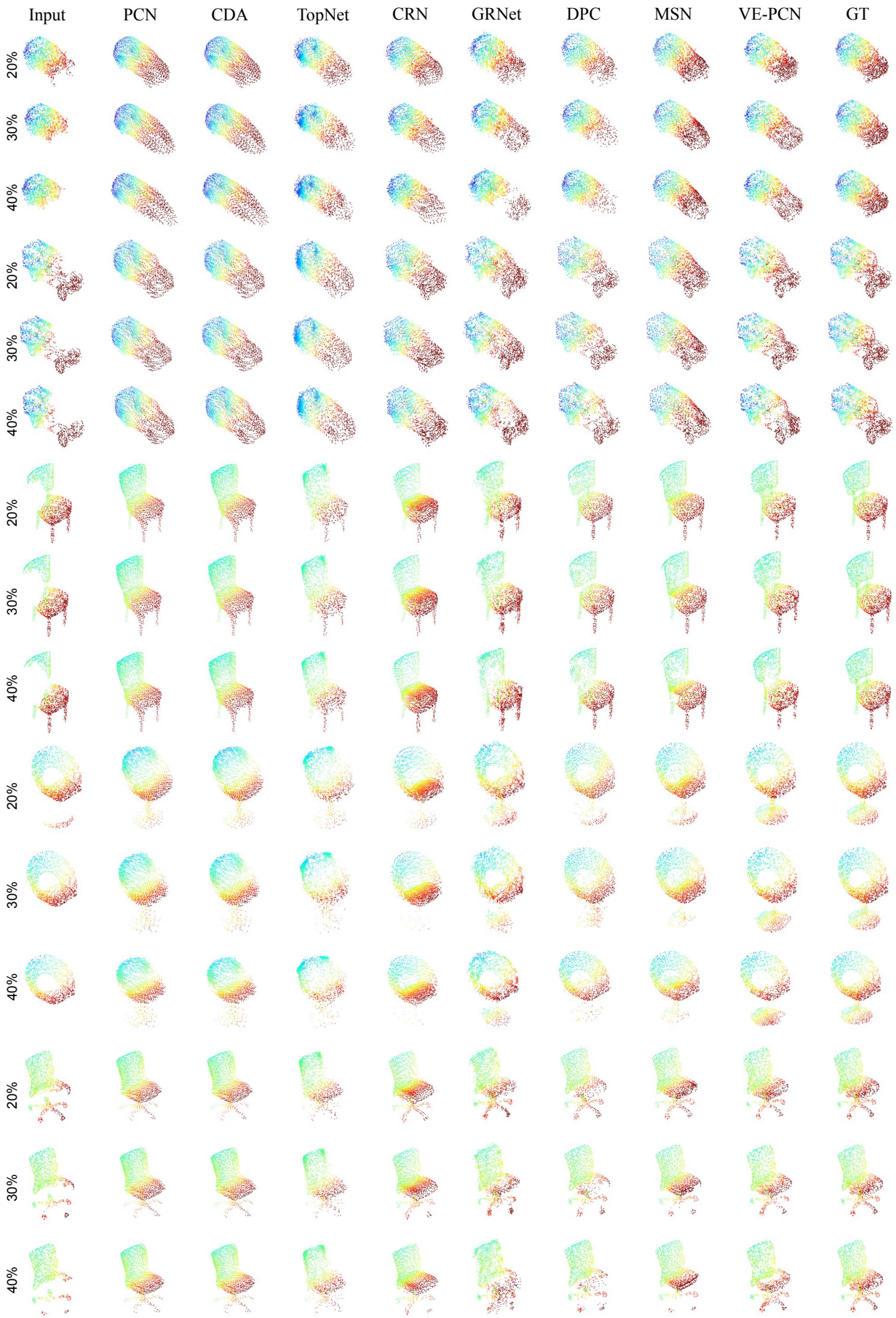

Figure 9: Qualitative comparisons on different occlusion ratios (1/2).

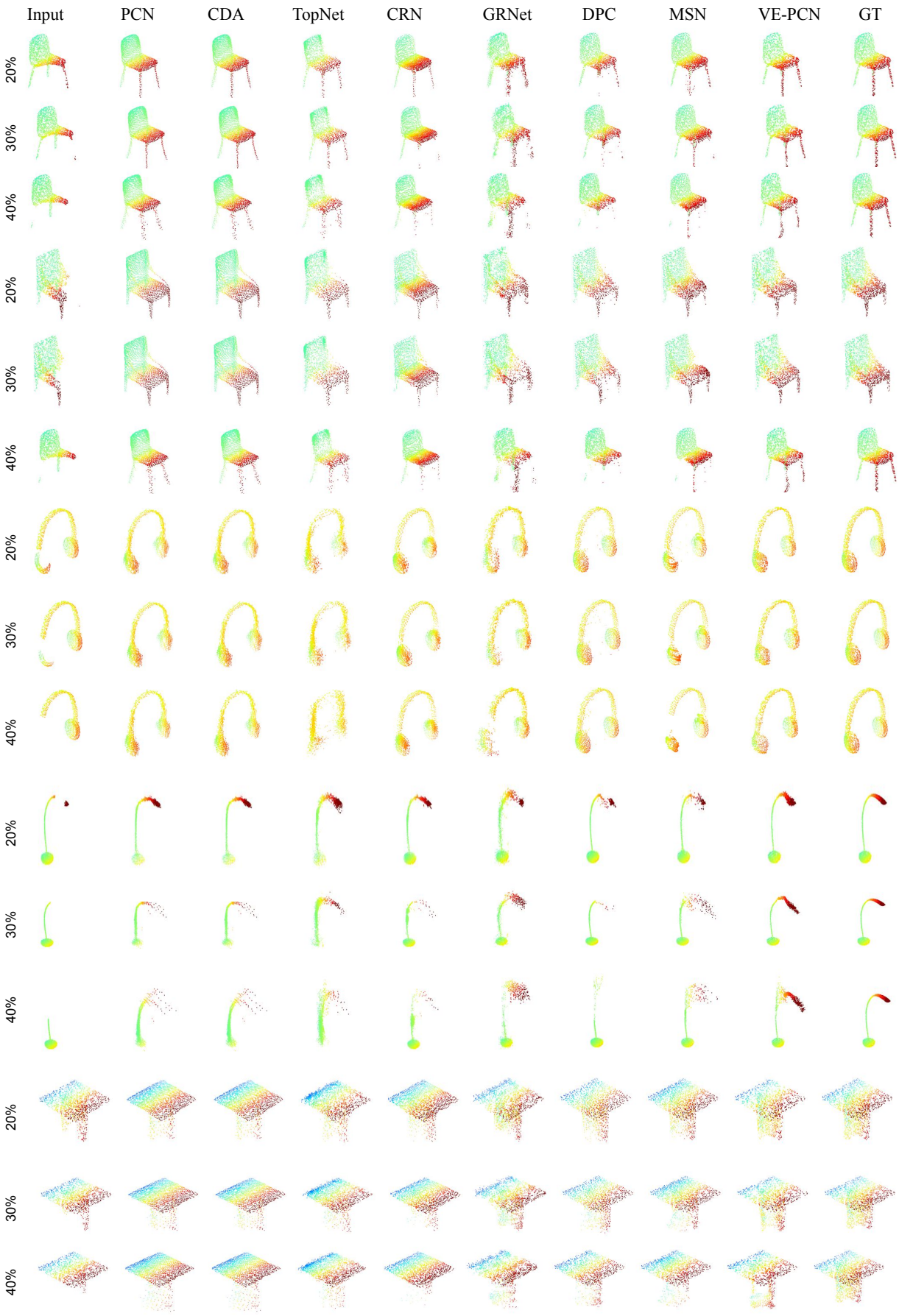

Figure 10: Qualitative comparisons on different occlusion ratios (2/2).

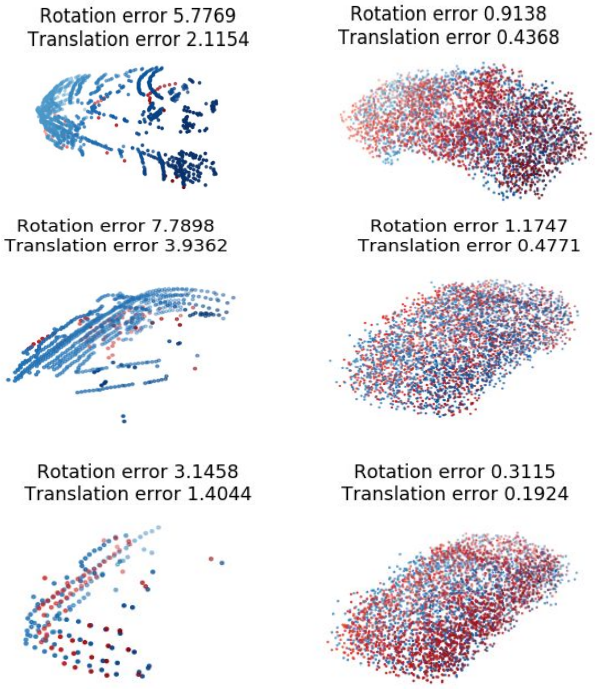

Figure 11: Completion results on KITTI.

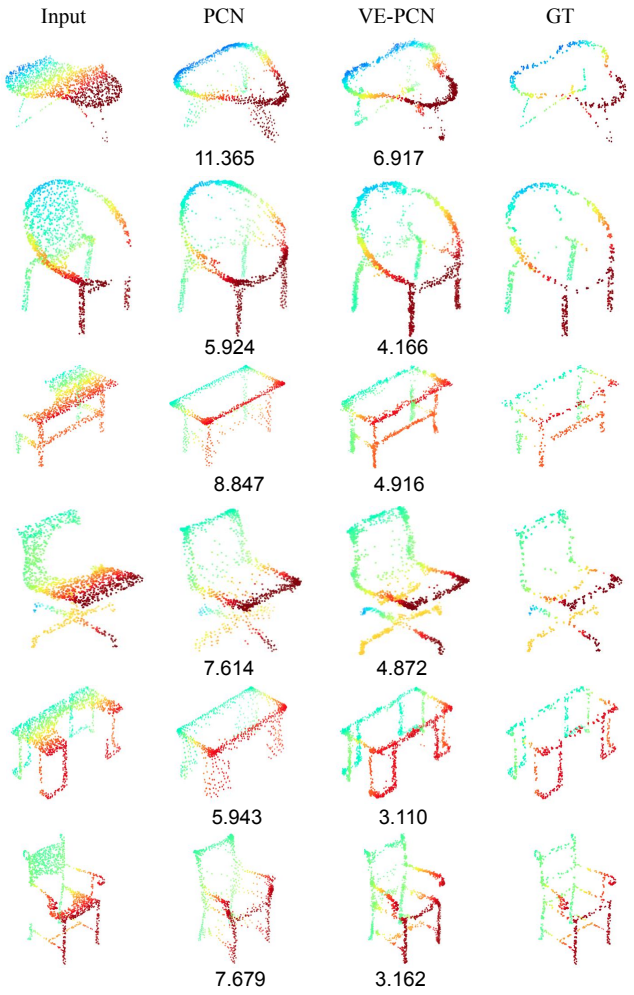

Figure 12: Edge generation results. Bottom values are the CD errors per point (10^{-4}).

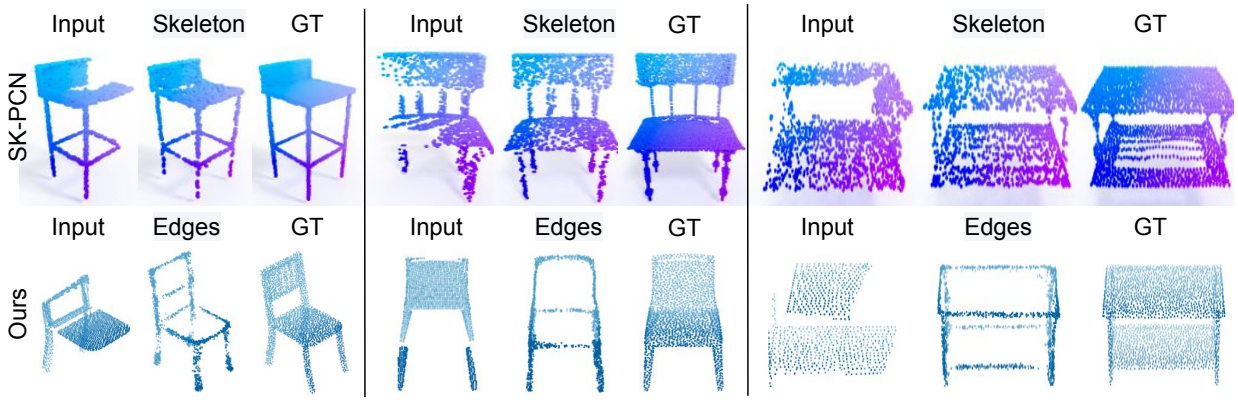

Figure 13: Edge (ours) vs. mes-skeleton (SK-PCN) generation.

References

[1] Syeda Mariam Ahmed, Yan Zhi Tan, Chee Meng Chew, Abdullah Al Mamun, and Fook Seng Wong. Edge and corner detection for unorganized 3d point clouds with application to robotic welding. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7350–7355. IEEE, 2018.

[2] Isaak Lim, Moritz Ibing, and Leif Kobbelt. A convolutional decoder for point clouds using adaptive instance normalization. In *Computer Graphics Forum*, volume 38, pages 99–108. Wiley Online Library, 2019.

[3] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11596–11603, 2020.

[4] Yinyu Nie, Yiqun Lin, Xiaoguang Han, Shihui Guo, Jian Chang, Shuguang Cui, and Jian.J Zhang. Skeleton-bridged point completion: From global inference to local adjustment. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16119–16130. Curran Associates, Inc., 2020.

[5] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[6] Lyne P Tchapmi, Vineet Kosaraju, S. Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] Xiaogang Wang, Marcelo H. Ang Jr. , and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[8] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net: Point cloud completion by learning multi-step point moving paths. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7443–7452, 2021.

[9] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020.

[10] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision*, pages 728–737. IEEE, 2018.

[11] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. Detail preserved point cloud completion via separated feature aggregation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 512–528. Springer, 2020.