ME-PCN: Point Completion Conditioned on Mask Emptiness
(Supplementary Material)

Bingchen Gong[1], Yinyu Nie[2], Yiqun Lin[3], Xiaoguang Han[*3], and Yizhou Yu[*1]
[1]The University of Hong Kong, [2]Technical University of Munich, [3]SSE, CUHK(SZ)

1. Network Architecture and Parameters

We provide the details of parameters and layer information of our network in this section. It discusses the encoder for $\mathcal{R}^*_{ray}$, $\mathcal{R}^d_{ray1}$ and $\mathcal{R}^d_{ray1}$. The coarse decoder is adopted from [3] and will not be covered here.

Emptiness $\mathcal{R}^*_{ray}$ Encoder Our approach takes a partial point cloud $\mathbf{Q}$ and sampled rays $\mathcal{R}^*_{ray}$ as inputs and encodes them into a global feature vector (GFV) with emptiness semantics, which will be used to predict complete point cloud with a coarse-to-fine strategy. The parameters for each layer in our encoder is shown in Table 1. For vanilla MSN+ray, we use the same encoder architecture as in Table 1.

Input	Partial Points Q	Ray Samples $\mathcal{R}^*_{ray}$
3 MLPs	64, 128, 1024	64, 128, 1024
Max-pool	1024	1024
Concatenate	2048	
Output	Global Feature Vector g'	

Table 1: Encoder of ME-PCN

Input	Partial Points $\mathbf{Q}$		Ray Samples $\mathcal{R}^*_{ray}$	
2 MLPs	128, 256		128, 256	
Max-pool	$f_i : 256$	256	256	$g_i : 256$
Concatenate		GFV g': 512		
Tile & Cat	$\widetilde{F} : \lvert Q\rvert \times 768$		$\widetilde{G} : \lvert\mathcal{R}^*_{ray}\rvert \times 768$	
2 MLPs	512, 1024		512, 1024	
Max-pool	1024		1024	
Concatenate	2048			
Output	Global Feature Vector g^*			

Table 2: PCN+ray $\mathcal{R}^*_{ray}$ Encoder

The core part of Feature Encoding (FE) layers is 3 separate Multilayer Perceptron Layers (MLP), which transform input points $\{q_i | q_i \in \mathbf{Q}\}$ or ray samples $\{\mathbf{r} | \mathbf{r} \in \mathcal{R}^*_{ray}\}$ into point features. A point-wise max-pooling is respectively performed on point features to obtain global features. Lastly, global features are concatenated together to form a single global feature vector.

PCN [10] has a deeper encoder by appending extra FE layers to learn a more abstract and latent global feature. To keep consistent with its original network, we construct an encoder for PCN+ray as shown in Table 2.

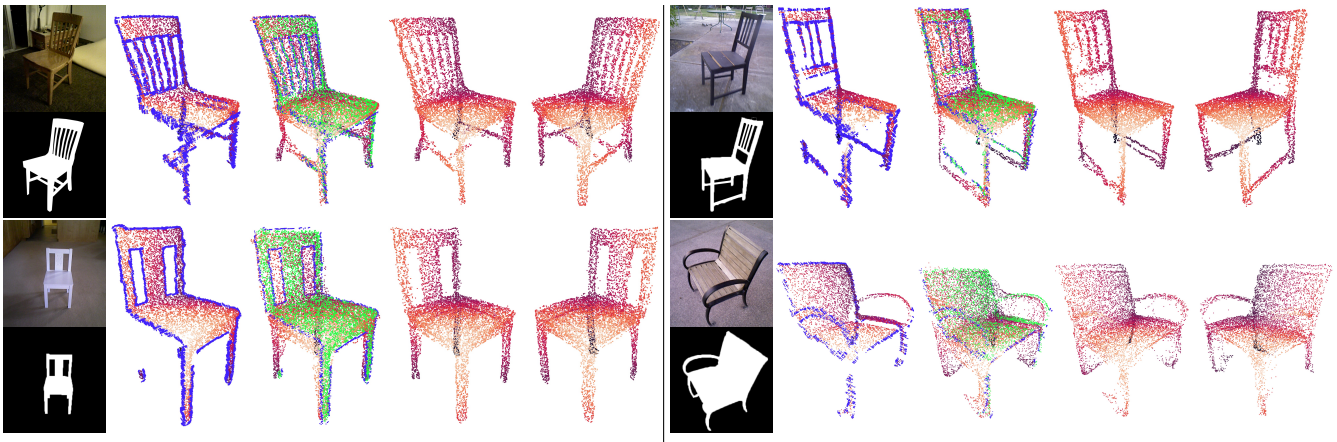
It first concatenates the global feature g' to each f_i and g_i to obtain augmented point feature matrices $\widetilde{F}$ and $\widetilde{G}$ whose rows are the concatenated feature vectors $[f_i, g']$ and $[g_i, g']$. Then, $\widetilde{F}$ and $\widetilde{G}$ are respectively passed through another two-layer MLP followed by point-wise max-pooling (as the first FE layer). The updated global feature vector g^* is outputted by concatenating the pooling results.

Emptiness $\mathcal{R}^d_{ray1}$ and $\mathcal{R}^d_{ray2}$ Encoder Rays $\mathcal{R}^d_{ray1}$ and $\mathcal{R}^d_{ray2}$ are sampled using the coarse points $\mathbf{P}^c$. The sampled empty rays $\mathcal{R}^d_{ray1} \in \mathbf{R}^{N_c \times K \times 9}$ tell whether a coarse point in $\mathbf{P}^c$ is in an 'empty' region while $\mathcal{R}^d_{ray2} \in \mathbf{R}^{N_c \times K \times 15}$ represents the boundaries of real shapes. For the sampling method, we refer readers to our paper. In our experiments, $N_c = 8192$ for our method and vanilla MSN+ray, and $N_c = 1024$ for PCN+ray. Both networks use the same emptiness encoder for local feature extraction, as shown in Table 3.

Input	Ray Samples $\mathcal{R}^d_{ray1}$	Ray Samples $\mathcal{R}^d_{ray2}$
3 MLPs	16, 32, 32	16, 32, 32
Grouping	$N_c \times 32$	$N_c \times 32$
Concatenate	$N_c \times 64$	
Output	Local Feature Vector	

Table 3: Local Emptiness Encoder for ray $\mathcal{R}^d_{ray1}$ and $\mathcal{R}^d_{ray2}$

Rays in $\mathcal{R}^d_{ray1}$ represent empty space neighboring to coarse points, while $\mathcal{R}^d_{ray2}$ informs the coarse points with the real shape boundary. Two FE layers are respectively used to encode $\mathcal{R}^d_{ray1}$ and $\mathcal{R}^d_{ray2}$. Grouping operation is a PointConv-style aggregation defined in [7]. We train the entire network end-to-end with the loss function in Equation (10) of the paper with $\lambda_1 = 1.0$ and $\lambda_2 = 0.1$.



(a) Scans (b) Input (c) Coarse (d) Output (e) Output (f) Scans (g) Input (h) Coarse (i) Output (j) Output

Figure 1: Tests on real scans. From left to right: image and mask of the target object; input partial scan; predicted coarse point cloud; predicted final point cloud from two different viewpoints. Note that blue points in (b),(f)/(c),(g) are sampled from the empty rays $\mathcal{R}_{ray}$/$\mathcal{R}^*_{ray}$. Green points in (c), (g) are sampled from real points.

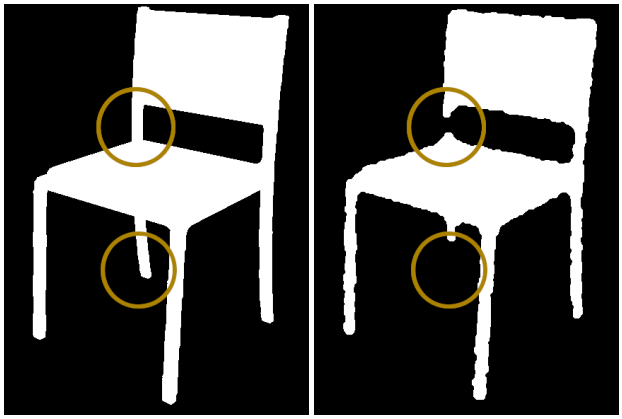
2. Tests on Real Scans

We train our network with ShapeNet and test it on real scans to investigate its generalization ability. Four real scans are taken from [4] for the testing, where the 3D points are back-projected from a single depth map and aligned to a canonical system. The corresponding 2D masks are extracted from RGB images.

From the results in Figure 1, we can see that our method can achieve plausible results 'in the wild'. The ray feature works quite well even though the depth scans are captured by users without training.

3. Robustness of Encoding Emptiness

In real-world applications, the input point clouds are usually involved with background points. For such a scenario, former works usually adopt a binary mask on the image plane to filter background points [9, 4]. This kind of masks can be directly used in our cases to learn emptiness. Since the mask should be given for most methods in real applications, in this section, we would like to explore the robustness of our method to imperfect masks. We simulate the masks extracted from real-world depth/RGB data, and add random segmentation errors (see Figure 2) to the boundaries of masks. We fine-tune our model on all categories using the noisy mask for 2 epochs. Test results are shown in Table 6. From the results we can see that the performance measured in both EMD and CD is only degraded slightly, which verifies the robustness of our method. It further demonstrates that, for synthesized depth scans, we can obtain masks by thresholding the depth maps; for real scans, we can use segmentation methods to extract masks from RGB/RGBD images. For both cases, our method can deliver faithful results.



(a) Before (b) After

Figure 2: Adding noise to simulate the mask from real world data: a) mask of chair back without noise; b) mask with noises on boundaries.

4. More Qualitative Comparisons

We list more qualitative comparisons with previous methods in Figure 3.

5. More Quantitative Comparisons

5.1. Comparison with Existing Methods

In this section, we report our quantitative evaluation on all 14 categories: faucet, cabinet, table, chair, vase, lamp, bottle, clock, display, knife, mug, fridge, scissors and trashcan. Some methods (PCN [10], CRN [5], GR-Net [8], and MSN [3]) support upsampling to recover higher resolution of outputs, we compare our methods with them under the

methods	faucet	cabinet	table	chair	vase	lamp	bottle	clock	display	knife	mug	fridge	scissors	trashcan	average
PCN	16.49	9.34	11.34	10.94	12.43	16.10	8.07	8.48	10.24	8.26	9.49	9.21	10.98	10.50	10.85
PCN+Ray	13.63	8.25	10.79	9.74	11.10	14.30	5.91	6.51	8.21	6.88	7.02	6.97	9.95	7.89	9.08
CRN	13.43	9.85	7.93	8.67	12.49	11.38	10.23	7.76	8.47	5.47	12.16	11.25	7.22	12.61	9.92
GRNet	10.36	7.75	7.50	7.74	11.21	10.74	9.11	7.52	7.18	8.53	9.46	8.62	7.64	9.76	8.79
MSN	7.71	6.70	6.52	6.57	6.89	7.55	5.17	5.77	6.06	4.51	5.63	6.68	4.29	6.26	6.17
Ours	**6.31**	**6.14**	**5.33**	**5.12**	**5.93**	**6.76**	**4.06**	**4.45**	**4.46**	**3.65**	**4.29**	**4.97**	**3.69**	**4.92**	**5.01**

(a) Evaluation on EMD ($\times 10^2$) with Res.=8,192

methods	faucet	cabinet	table	chair	vase	lamp	bottle	clock	display	knife	mug	fridge	scissors	trashcan	average
PCN	4.17	4.67	3.82	4.01	6.31	3.73	3.75	4.67	4.15	1.82	5.93	4.65	2.33	5.33	4.24
PCN+Ray	2.80	4.55	3.57	3.81	5.80	3.12	3.24	3.67	**2.97**	1.57	4.34	**3.51**	**1.26**	4.50	3.48
CRN	3.67	**4.49**	**3.44**	3.81	5.49	3.19	3.35	4.32	4.00	1.64	5.58	4.52	2.06	5.08	3.90
GRNet	3.28	4.66	3.73	3.94	5.53	3.52	4.51	4.77	4.08	2.03	6.17	4.99	2.15	5.37	4.20
MSN	4.02	5.75	4.61	4.81	5.71	4.34	4.55	4.86	4.45	1.89	5.42	5.25	2.04	5.49	4.51
Ours	**2.62**	4.72	3.76	**3.62**	**4.54**	**3.02**	**3.11**	**3.59**	3.52	**1.46**	**4.28**	4.17	1.51	**4.48**	**3.46**

(b) Evaluation on CD ($\times 10^2$) with Res.=8,192

Table 4: Comparison with Existing Methods. Evaluation with Res.=8,192

methods	faucet	cabinet	table	chair	vase	lamp	bottle	clock	display	knife	mug	fridge	scissors	trashcan	average
PCN	16.81	10.47	12.22	11.81	13.25	16.67	8.62	9.63	11.07	8.64	10.83	10.47	11.58	11.64	11.69
PCN+Ray	16.13	10.18	11.68	10.61	11.13	14.90	7.02	8.16	9.40	7.23	8.90	8.93	9.79	9.63	10.26
PF-Net	16.11	10.04	9.97	10.61	11.50	14.07	9.17	10.96	9.55	10.04	10.21	10.55	11.02	9.79	10.97
P2P-Net	16.09	11.64	10.73	12.29	16.36	13.52	18.10	11.95	11.00	13.28	20.55	15.63	8.78	16.59	14.04
SoftPoolNet	15.03	14.30	11.28	14.05	17.63	15.89	18.35	13.05	10.52	10.66	17.58	17.34	11.47	18.87	14.72
CRN	14.00	11.00	9.09	9.70	13.32	12.09	11.02	9.01	9.39	5.84	13.07	12.41	7.69	13.18	10.77
GRNet	11.30	9.16	8.61	8.82	12.27	11.28	9.98	8.83	8.24	9.07	11.06	9.91	7.70	11.12	9.81
MSN	8.52	8.19	7.82	7.82	8.36	8.51	6.43	7.41	7.14	4.92	7.84	8.28	4.94	8.29	7.46
Ours	**6.89**	**7.48**	**6.63**	**6.63**	**7.16**	**7.48**	**5.53**	**6.19**	**6.02**	**4.44**	**6.67**	**6.87**	**4.00**	**7.04**	**6.36**

(a) Evaluation on EMD ($\times 10^2$) with Res.=2,048

methods	faucet	cabinet	table	chair	vase	lamp	bottle	clock	display	knife	mug	fridge	scissors	trashcan	average
PCN	5.62	7.28	5.95	6.14	8.71	5.15	5.53	6.97	6.29	2.64	8.79	7.38	3.26	8.09	6.27
PCN+Ray	4.35	7.14	5.19	5.98	7.19	4.61	4.45	6.14	**5.23**	2.35	7.21	**6.33**	**2.17**	7.28	5.40
PF-Net	8.96	8.15	6.94	7.48	10.10	7.56	6.96	8.67	7.16	4.12	9.80	8.54	5.24	9.08	7.77
P2P-Net	4.47	7.21	**5.49**	5.92	7.62	4.41	7.01	6.79	6.45	2.77	8.71	8.22	2.47	8.42	6.14
SoftPoolNet	5.54	7.85	6.41	6.59	8.27	5.56	6.67	7.63	6.59	3.05	8.90	8.24	3.45	8.57	6.67
CRN	5.14	7.13	5.59	5.94	7.96	4.63	5.28	6.72	6.12	2.48	8.49	7.30	2.96	7.94	5.98
GRNet	4.72	7.21	5.77	6.00	7.90	4.92	6.24	7.06	6.17	2.89	8.86	7.62	3.03	8.14	6.18
MSN	5.25	8.06	6.50	6.70	7.92	5.66	6.16	7.01	6.36	2.67	8.16	7.62	2.88	8.07	6.36
Ours	**3.90**	**7.01**	5.65	**5.61**	**6.68**	**4.26**	**4.92**	**5.88**	5.55	**2.25**	**7.19**	6.73	2.34	**7.13**	**5.36**

(b) Evaluation on CD ($\times 10^2$) with Res.=2,048

Table 5: Comparison with existing methods. Evaluation with Res.=2,048

resolution of 8,192. We report the comparisons using EMD [3] and CD scores [1] in Table 4a and Table 4b respectively. For the other methods that do not support upsampling (including PF-Net [2], P2P-Net[9], SoftPoolNet[6]), we downsample the output of all the methods to the resolution of 2,048 to enable a fair comparison. The quantitative scores on EMD and CD are respectively listed in Table 5a and Table 5b.

5.2. More Ablation Studies on Ray Encoding

In our network, empty rays are feed into network together with real/coarse points. The most significant differ-

Category	only pts in rays	remove $\mathcal{R}^d_{ray2}$	remove $\mathcal{R}^d_{ray1}$	noisy boundary	MSN	Ours
Faucet	6.54 / 2.61	7.88 / 4.34	6.88 / 2.82	6.94 / 2.90	7.71 / 4.02	6.31 / 2.62
Cabinet	6.35 / 4.72	6.55 / 5.52	6.72 / 5.02	6.71 / 4.99	6.70 / 5.75	6.14 / 4.72
Table	6.24 / 4.78	6.12 / 4.71	5.91 / 3.96	5.89 / 3.95	6.52 / 4.61	5.33 / 3.76
Chair	7.43 / 5.71	6.07 / 4.73	5.33 / 4.02	5.15 / 3.97	6.57 / 4.81	5.12 / 3.62
Vase	6.01 / 4.54	7.18 / 5.28	6.62 / 4.83	6.68 / 4.94	6.89 / 5.71	5.93 / 4.54
Lamp	7.14 / 2.95	8.71 / 4.19	7.30 / 3.23	7.21 / 3.27	7.55 / 4.34	6.76 / 3.02
Average	6.62 / 4.22	7.09 / 4.80	6.46 / 3.98	6.43 / 4.00	6.99 / 4.87	5.93 / 3.71

Table 6: Ablation study on different of configurations (EMD / CD $\times 10^2$)

ence between rays and points is that each ray has a directional vector and an offset vector to a neighboring point. In encoder-decoder stage, the input rays are:

$$\mathcal{R}^*_{ray} = \{\{p^e_k\}, \{D_k\}, \{v_k\}\} \in \mathbf{R}^{M\times K\times 9} \quad (1)$$

While in the refining stage, we revisit the emptiness information by two set of rays:

$$\mathcal{R}^d_{ray1} = \{\{p^e_k\}, \{D_k\}, \{v_k\}\} \in \mathbf{R}^{N_c\times K\times 9}. \quad (2)$$

$$\mathcal{R}^d_{ray2} = \{\{p^{c,e}_k\}, \{D^c_k\}, \{\mathbf{r}^c\}\} \in \mathbf{R}^{N_c\times K\times 15} \quad (3)$$

We design an ablation study to explore the effectiveness of directional vector and offset vector in ray representation. In the ablated version, we remove D_k, v_k and $\mathbf{r}^c$ in ray representation. As a result, the rays inputted to our network only contains p^e_k or $p^{c,e}_k$. The quantitative results is shown in Table 6 as '**only pts in rays**'. The results demonstrate that D_k, v_k and $\mathbf{r}^c$ are significant for our method to learn emptiness.

In our paper, the first $\mathcal{R}^d_{ray1}$ informs our decoder with the emptiness information, telling our decoder *'whether the coarse points are in empty regions'*. The second $\mathcal{R}^d_{ray2}$ informs our decoder with the shape information, telling our decoder *'what the real surface looks like'*. To exam the necessity of the two rays in the refining stage, we design two additional ablation studies: '**remove** $\mathcal{R}^d_{ray2}$' and '**remove** $\mathcal{R}^d_{ray1}$'. The first one means the local feature vector is computed only from $\mathcal{R}^d_{ray1}$. Similarly, the second one means the local feature vector is computed only from $\mathcal{R}^d_{ray2}$. The quantitative results are shown in Table 6. The results prove that both $\mathcal{R}^d_{ray1}$ and $\mathcal{R}^d_{ray2}$ are important in refining stage.

References

[1] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 3

[2] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7662–7670, 2020. 3

[3] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11596–11603, Apr. 2020. 1, 2, 3

[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. 2

[5] Xiaogang Wang, Marcelo H. Ang Jr. , and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 790–799, June 2020. 2

[6] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Softpoolnet: Shape descriptor for point cloud completion and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 70–85, August 2020. 3

[7] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 1

[8] 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. 2

[9] Kangxue Yin, Hui Huang, Daniel Cohen-Or, and Hao Zhang. P2p-net: Bidirectional point displacement net for shape transform. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 2, 3

[10] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018. 1, 2

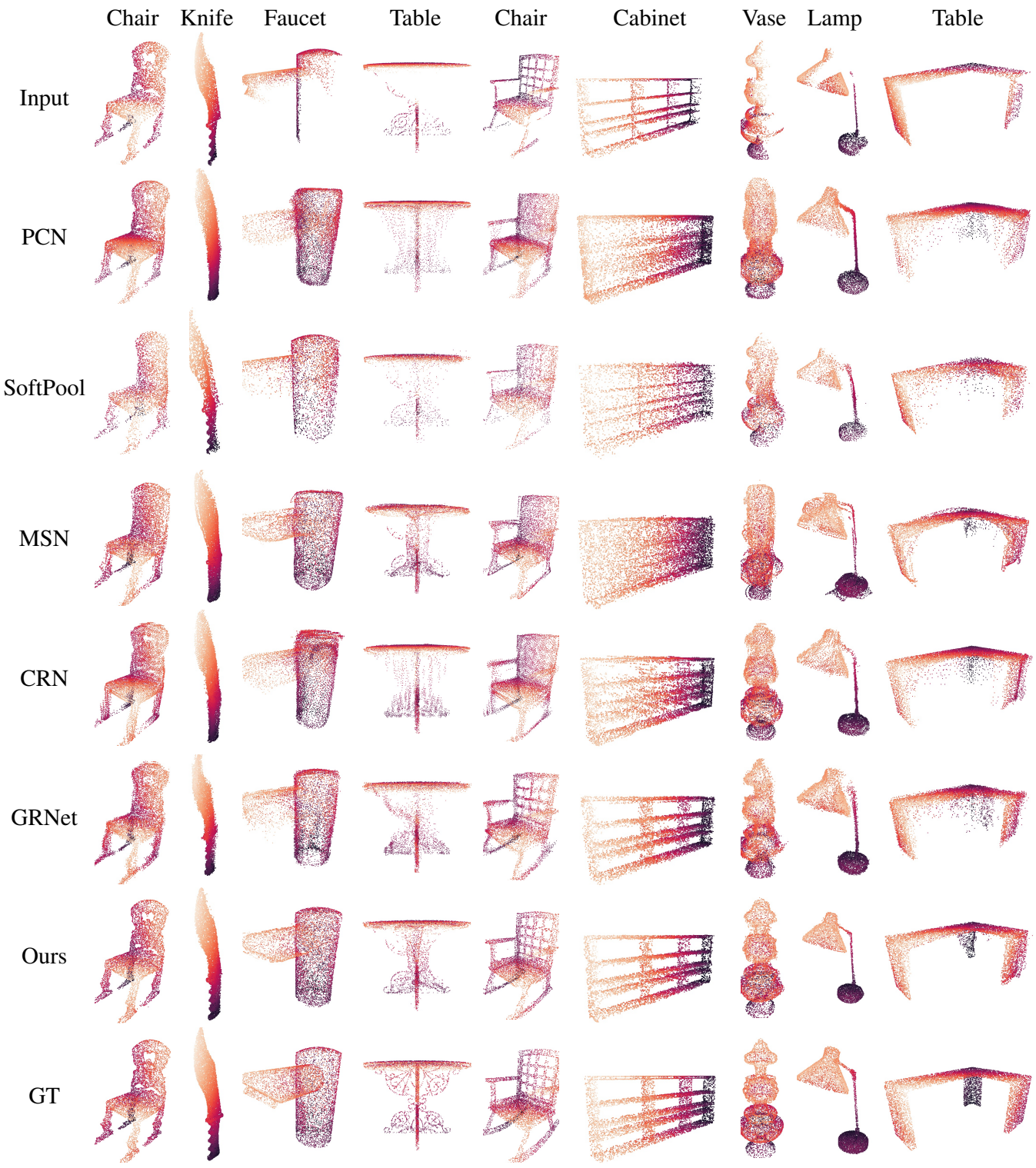


Figure 3: Comparisons of different methods on point cloud completion. Note that 8,192 points are exported from each method for comparison, except SoftPoolNet (4,096 points) due to its network specification.