Supplementary Material for SVDFormer: Complementing Point Cloud via Self-view Augmentation and Self-structure Dual-generator

Zhe Zhu[1], Honghua Chen[1], Xing He[1], Weiming Wang[2†], Jing Qin[3], Mingqiang Wei[1†]
[1]Nanjing University of Aeronautics and Astronautics
[2]Hong Kong Metropolitan University
[3]The Hong Kong Polytechnic University
zhuzhe0619@nuaa.edu.com; chenhonghuacn@gmail.com; hexing@nuaa.edu.cn;
wmwang@hkmu.edu.hk; harry.qin@polyu.edu.hk; mqwei@nuaa.edu.cn

In this supplementary material, we provide more detailed information to complement the main manuscript. Specifically, we first introduce the implementation details, including network architecture details and experimental settings. Then, we conduct more ablation studies to analyze our method. Next, we provide some failure cases and a discussion on the limitations of our work. Finally, we present additional quantitative and qualitative results.

A. Detailed Settings

Network implementation details. We apply perspective projection to get the depth maps with the resolution of 224 × 224 from three orthogonal views. We directly feed the projected depth maps to the network without applying any color mapping enhancement. In SVFNet, we use PointNet++ [5] to extract features from point clouds. The detailed architecture is: $SA(C = [3, 64, 128], N = 512, K = 16) \rightarrow SA(C = [128, 256], N = 128, K = 16) \rightarrow SA(C = [512, 256])$. The final feature dimension of ResNet18 [2] is set to 256. The dimension of the embed query, key, and value in View Augment is set to 256. After concatenation, we get the shape descriptor F_g with 512 channels. We use a self-attention layer of 512 hidden feature dimensions followed by an MLP to regress the coarse points P_C. The merged point cloud P_0 has 512 and 1024 points for PCN and ShapeNet-55, respectively.

During refinement, we set the upsampling rates $\{r_1, r_2\}$ of the two SDGs as {4,8} and {2,4} for PCN and ShapeNet-55, respectively. We adopt EdgeConv [7] to extract local features from P_{in}. The detailed architecture is: $EdgeConv(C = [3, 64], K = 16) \rightarrow FPS(2048, 512) \rightarrow EdgeConv(C = [64, 256], K = 8)$. We use a shared-weights architecture above in the two SDGs. After obtaining F_Q and F_H, we use a decoder composed of two self-attention layers (one in the ShapeNet-55 experiments) to further analyze the coarse shapes. The hidden feature dimensions of self-attention layers are set as [768, $128r_1$] and [512, $128r_2$] in the two SDGs, thus producing $F_l \subseteq \mathbb{R}^{N \times 256r}$. F_l is then passed to an MLP and reshaped to $rN \times 128$. Finally, the coordinates offset is predicted by an MLP with feature dimensions of [128, 64, 3].

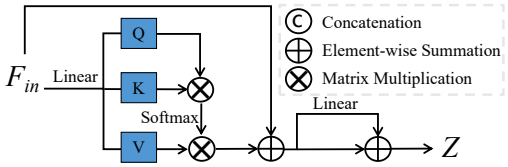

Figure 1. The calculation process of Self-Attention.

Usage of attention. In our method, the self-attention layer is used to generate P_c in SVFNet and decode F_Q and F_H in SDG. We also use a cross-attention layer to find the geometric similarity. In our experiments, we implement the self-attention module and the cross-attention module following the same transformer architecture [6]. The point-wise features are regarded as sequence data. The calculation procedure is illustrated in Figure 1. Given the input feature $F_{in} = \{f_i\}_{i=1}^{N_{l-1}}$, the output feature matrix $Z = \{z_i\}_{i=1}^{N_{l-1}}$ is calculated as :

$$\begin{aligned} z_i &= h_i + Linear(h_i) \\ h_i &= b_i + f_i \\ b_i &= \sum_{j=1}^{N_{l-1}} a_{i,j}(f_j W_V) \\ a_{i,j} &= Softmax((f_i W_Q)(f_j W_K)^T) \end{aligned} \quad , \tag{1}$$

Experiment and training settings. The network is imple-

†Co-corresponding authors

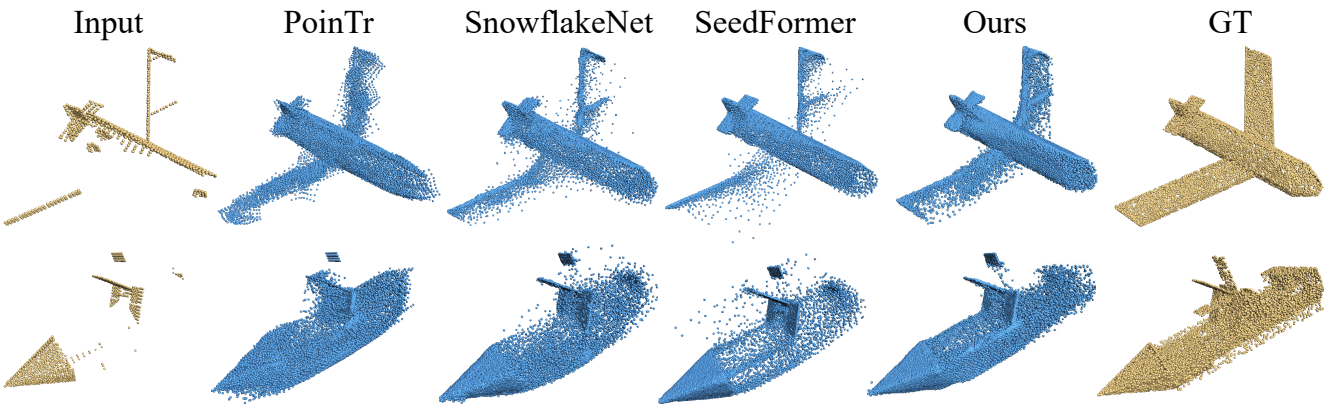

Figure 2. Example of failure cases.

Table 1. Results and inference time of more ablation variants on PCN. (ℓ^2 CD $\times 10^3$ and F1-Score@1%)

Variants	CD↓	DCD↓	F1↑	Time
1 View	6.58	0.538	0.835	24.86ms
3 Views (Ours)	6.54	0.536	**0.841**	26.55ms
6 Views	6.55	0.536	0.840	27.11ms
Random Projection (inference)	6.58	0.537	0.838	26.25ms
Encoder in SpareNet [11]	6.66	0.551	0.825	37.26ms
ResNet-50 [2] as the 2D backbone	**6.52**	**0.535**	**0.841**	31.37ms
Vit-B/16 [1] as the 2D backbone	6.56	0.543	0.837	34.16ms

mented using PyTorch [4] and trained with the Adam optimizer [3] on NVIDIA 3090 GPUs.

For training on the PCN dataset [14], the initial learning rate is set to 0.0001 and decayed by 0.7 for every 40 epochs. The batch size is set to 12. It takes 400 epochs for convergence. Since the point coordinates in PCN are normalized to [-0.5, 0.5], the depth maps are projected at a distance of 0.7 in order to observe the whole shape. To ensure that the input point cloud contains exactly 2048 points, we take a subset for point clouds with more than 2048 points and randomly duplicate points for those with less than 2048 points.

For training on ShapeNet-55/34 [13], the number of missing points is randomly selected from 2048 to 6192. The initial learning rate is set to 0.0001 and decayed by 0.98 for every 2 epochs. The batch size is set to 16. It takes 300 epochs for convergence. The point coordinates in ShapeNet-55 are normalized to [-1.0, 1.0]. Therefore, D are projected at a distance of 1.5. Following [10, 15], We use a partial matching strategy, which includes setting a larger resolution for P_0 and adding a partial matching loss [8].

B. Ablation Studies

Ablation on the number of projections. We conduct an ablation experiment on the number of depth maps D in SVFNet. The depth maps D are projected from 1, 3, and 6 orthogonal views, respectively. The results on PCN are shown in Table 1. To balance the trade-off between effectiveness and computational consumption, we conduct all experiments using three views. This choice allows us to capture sufficient information from the point clouds while keeping the computational cost manageable.

Ablation on choice of coordinate systems. To testify the robustness of our method, during inference, we introduced random variations to the projection, including camera view angle offsets ranging from 0 to 10 degrees and observation distance displacements ranging from 0 to 0.1. The result reported in the 5th row of Table 1 shows that the performance will not significantly drop with random projections.

Ablation of different encoders. We testify the design of encoder to further demonstrate the effect of our self-view fusion feature extractor. We first replace the SVFNet with the encoder in SpareNet [11], which contains layers of channel-attentive EdgeConv, to re-produce the shape descriptor F_g. We report the new results in Table 1, which demonstrates that our self-view fusion feature extractor achieves better performance than existing encoder while having a tolerable computation cost. In addition, we ablate the choice of 2D backbone in the SVFNet. To be specific, We replace it with ResNet-50 [2] and the vision transformer (ViT-B/16) [1], respectively. We find that a larger CNN-based 2D backbone can slightly improve the performance while introducing more computation cost. Moreover, using the ViT results in unsatisfactory performance. This could be attributed to the fact that the entire model was trained from scratch, and larger models may not perform optimally with a limited amount of 3D training data.

C. Failure Cases and Limitations

Figure 2 displays the failure cases we observed. It's worth noting that in cases where input shapes lack irregular structures that are uncommon in the training dataset (such as the water wheel of a watercraft), the network may

Table 2. DCD results on the PCN dataset. Lower is better.

Methods	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat	Avg
GRNet [12]	0.688	0.582	0.610	0.607	0.644	0.622	0.578	0.642	0.622
PoinTr [13]	0.574	0.611	0.630	0.603	0.628	0.669	0.556	0.614	0.611
SnowflakeNet [10]	0.560	0.597	0.603	0.582	0.598	0.633	0.521	0.583	0.585
PMP-Net++ [9]	0.600	0.605	0.614	0.613	0.610	0.647	0.577	0.622	0.611
Seedformer [15]	0.557	0.592	0.598	0.579	0.585	0.626	0.520	0.605	0.583
Ours	**0.506**	**0.549**	**0.559**	**0.524**	**0.535**	**0.579**	**0.472**	**0.562**	**0.536**

Table 3. F1-Score@1% on the PCN dataset. Higher is better.

Methods	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat	Avg
GRNet [12]	0.843	0.618	0.682	0.673	0.761	0.605	0.751	0.750	0.708
PoinTr [13]	0.915	0.665	0.718	0.710	0.798	0.632	0.796	0.797	0.754
SnowflakeNet [10]	0.941	0.695	0.745	0.776	0.858	0.691	0.867	0.834	0.801
PMP-Net++ [9]	0.941	0.660	0.721	0.754	0.860	0.657	0.822	0.830	0.781
Seedformer [15]	0.950	0.700	0.753	0.803	0.885	0.712	0.884	0.850	0.818
Ours	**0.962**	**0.738**	**0.792**	**0.833**	**0.897**	**0.746**	**0.901**	**0.863**	**0.841**

not be capable of producing satisfactory results. Nevertheless, our method still outperforms state-of-the-art (SOTA) methods [13, 10, 15] when dealing with simple geometric structures, like the body of the watercraft. Our SDG incorporates a Structure Analysis unit that leverages learned priors to complete shapes. However, its effectiveness may be constrained by the limited amount of available training data. Given that transformers have demonstrated effectiveness in scenarios with abundant training data, pretraining with large-scale 2/3D datasets could be a promising approach to address this limitation.

D. Additional Results

More detailed quantitative results for individual cases are available in Tables 2 and 3. Our method achieves the best DCD and F1-score on each category of the PCN dataset. In addition, we show more visual results in Figures 3, 4, and 5. In Figure 3, we present two partial point clouds on each category of the PCN dataset. In Figure 4, we present six partial point clouds of ShapeNet-55 and compare the results with two representative approaches [13, 15]. Our method generates more compact overall shapes and richer details. Also, we visualize more results in Figure 5, where the partial shapes are generated from two different viewpoints.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 2

[5] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[7] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 1

[8] Xin Wen, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13080–13089, 2021. 2

[9] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net++: Point cloud completion by transformer-enhanced multi-step point moving paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[10] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5499–5509, 2021. 2, 3

[11] Chulin Xie, Chuxin Wang, Bo Zhang, Hao Yang, Dong Chen, and Fang Wen. Style-based point generator with adversarial rendering for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4619–4628, 2021. 2

[12] 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. 3

[13] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12498–12507, 2021. 2, 3, 6

[14] 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. 2

[15] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *European Conference on Computer Vision*, pages 416–432. Springer, 2022. 2, 3, 6

Figure 3. Visual results on the PCN dataset.

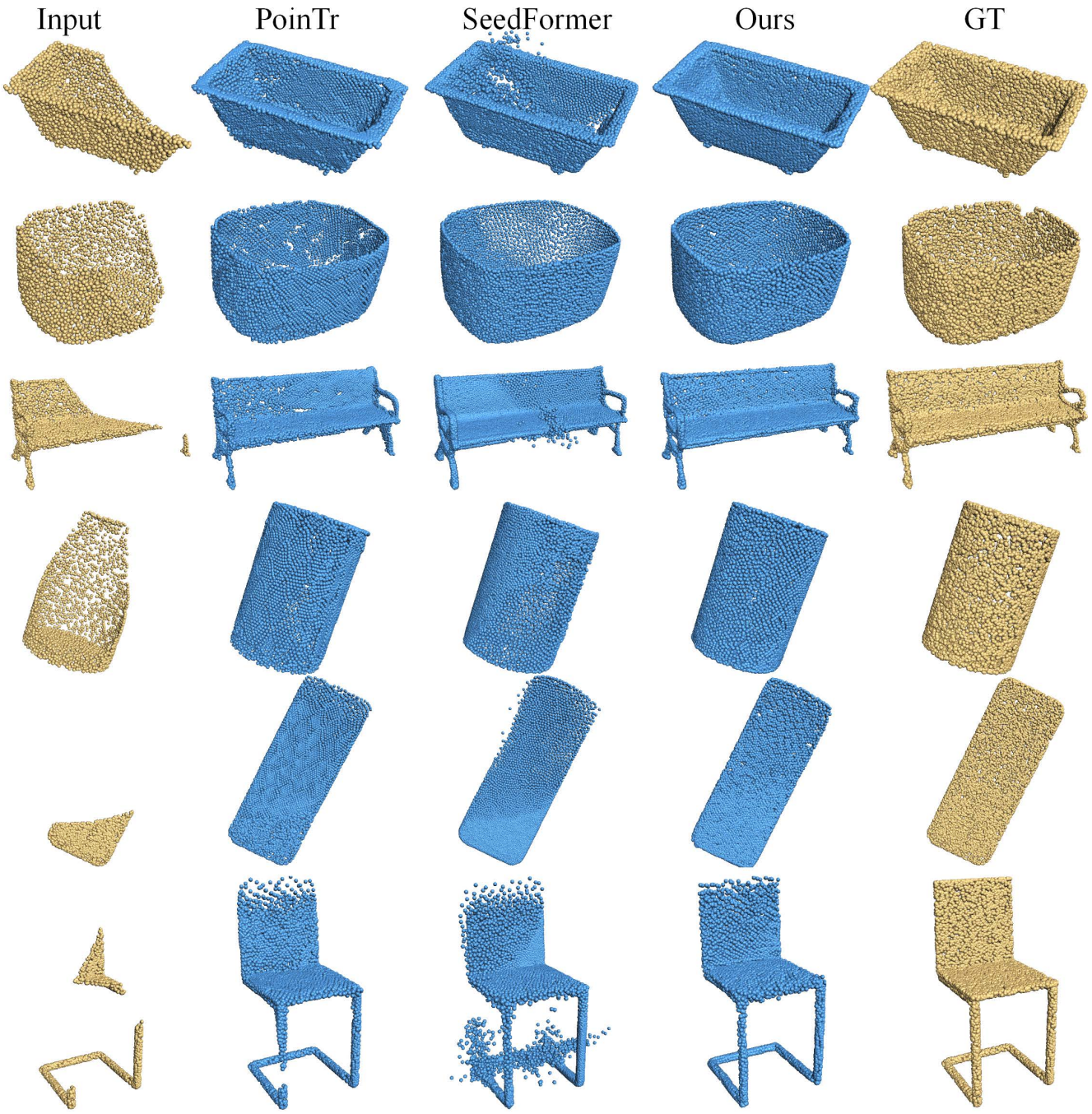

Figure 4. Visual Comparison with two representative approaches [13, 15] on ShapeNet-55.

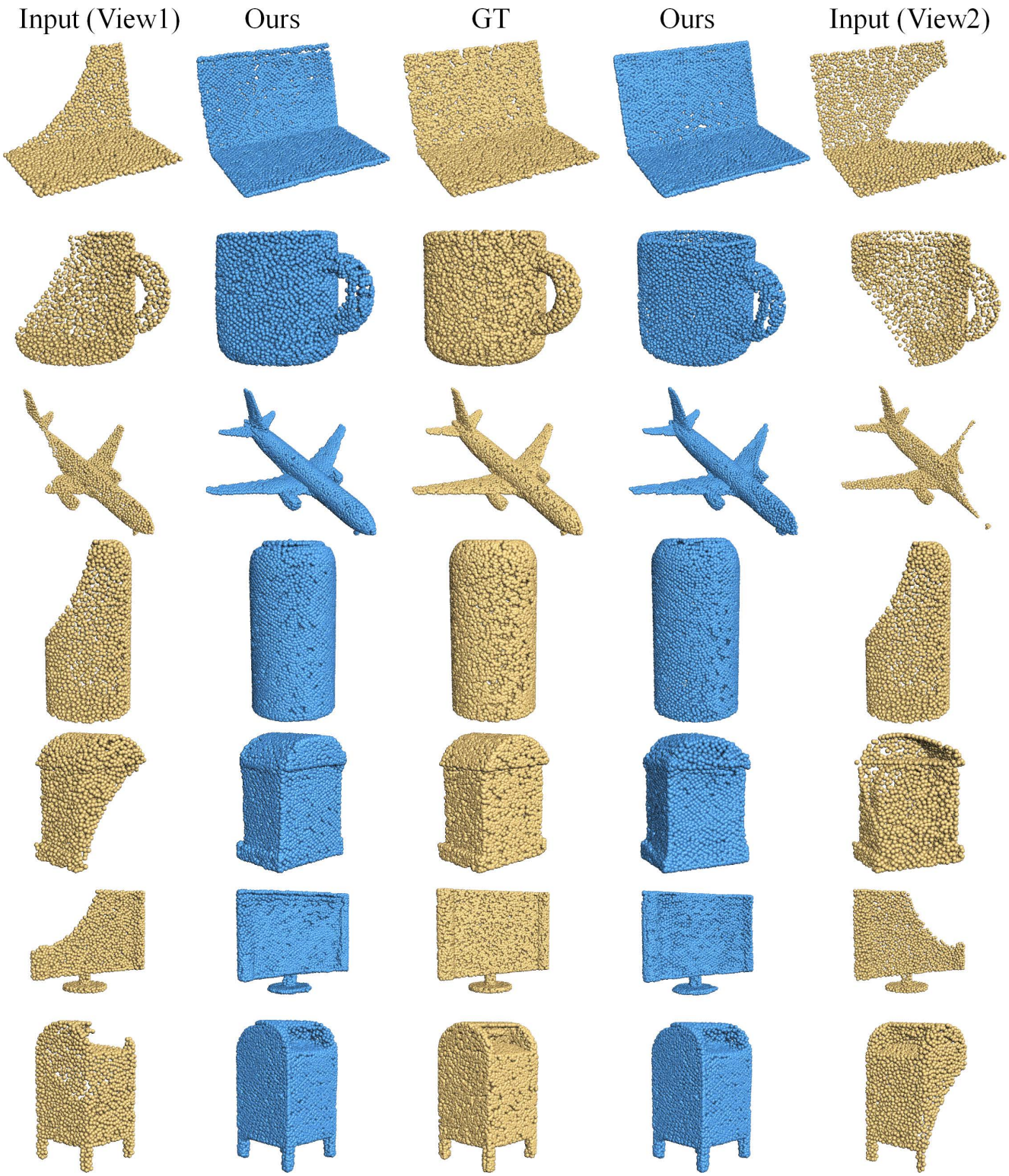

Figure 5. More visual results on ShapeNet-55. We show results when the partial input are generated from two viewpoints.