Dense Point Cloud Prediction from a Single RGB Image

Kevin Katzkowski
kevin.katzkowski@tum.de

Kartik Bali
Kartik.Bali@tum.de

Shichen Hu
shichen.hu@tum.de

Abstract

As 3D data is not abundantly available, a method that generates high-quality point clouds would be beneficial for many data-driven downstream tasks like shape classification or segmentation. Existing methods like PSGN construct a coarse point cloud given a single RGB image as input, but fail to capture fine-grained details. To address this problem, we propose a two-stage reconstruction pipeline that takes a single RGB image of an 3D object as input and predicts a high-quality point cloud representing the shape. Specifically, we first use a PSGN to predict a coarse point cloud from an input image and then apply a folding-based decoder to obtain a dense version with 16,384 points. In experiments, our approach shows strong performance for 3D reconstruction and improves the quality of point clouds over the baseline.

1. Introduction

Ever since the ground-breaking works of PointNet [10] and PointNet++ [11], point clouds as 3D representations have been increasingly attracting attention due to their simplicity, efficiency and ability to capture raw sensor data. Still, one of the open challenges in the research community is how to generate point cloud data at scale, such that it can be used for data-driven downstream tasks like classification and segmentation.

Given this challenge, [2] propose a Point Set Generation Network (PSGN) to construct a point cloud from a single RGB image, enabling 3D data synthesis at scale as high-resolution RGB images are abundantly available. However, the drawback is that the generated point cloud is coarse in resolution and fails to capture fine-grained details, which would be desirable for a 3D data representation.

Inspired by the recent adaptation of FoldingNet [13] in point cloud completion models like PCN [15] and PoinTr [14], we propose an approach that improves upon PSGN. First, we use a PSGN to generate a sparse point cloud given a single RGB input image. Then, a shared folding-based decoder is applied to densify the coarse output. As a result, a dense high-quality point cloud is generated. Our contributions can be summarized as following:

- We re-implement three variants of PSGN and examine them quantitatively and qualitatively.
- We propose a two-stage dense point cloud prediction pipeline which combines PSGN with a folding-based decoder and produces a high-quality point cloud from a single RGB image.

2. Related Work

Reconstructing the 3D shape of objects from a single RGB image is an ill-posed problem. Numerous different approaches have been developed to tackle the problem [2, 3, 6, 9]. Most proposed deep neural networks focus on using regular grid structures like voxels and extending 2D convolutions to 3D space. However, these volumetric representation-based methods are very inefficient, as they require cubic growth in compute and memory, while only containing relevant information near the object's surface. While several solutions reducing the computational effort have been presented [4, 12], methods that directly operate on the surface in form of irregular point clouds have proven to be more memory-efficient.

In contrast to volumetric representations which use convolutions as the de-facto operation, point cloud representations utilize fully-connected layers due to the missing grid structure. Points can be represented as sets by $N \times 3$ matrices [2, 3, 9], 3-channel grids in the form of $H \times W \times 3$ where each grid location encodes a point [2, 7], or as depth maps [6, 16].

Based on the point set representation, [9] use an autoencoder to learn a shape embedding and then learn a mapping from 2D images to the embeddings. [2] combine the point set and grid representation to generate point clouds from a single RGB image. They use an encoder to obtain a latent representation of the input image and then separately predict a coarse point set as well as a 3-channel point grid. Both representations are then fused to obtain the final point cloud, which is very coarse as it only consists of $N = 1024$ points.

Other approaches try to predict more detailed and dense point clouds by exploiting hierarchical learning. [3] applies

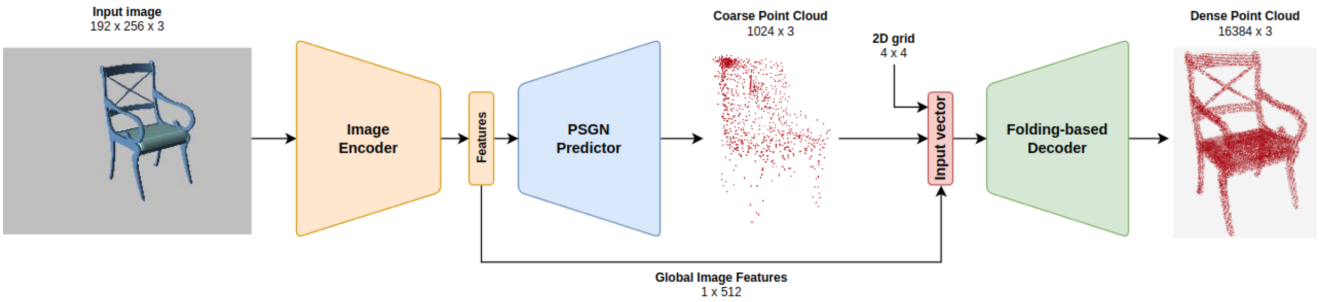

Figure 1. Our proposed two-stage pipeline for dense point cloud prediction from a single RGB input image.

1D convolutions to an ordered point cloud and predicts a dense point cloud by extracting local and global features at three resolutions. Using depth maps generated by a CNN from multiple pre-determined viewpoints using a single image only, [6] proposes a Grid Deformation Unit to predict a deformed depth map per viewpoint. With back-projection and viewpoint fusion, one dense point cloud is obtained from the different depth maps.

[13] uses a graph-based encoder to extract local and global features, and combines it with a novel folding-based decoder. The decoder implements a *folding operation* which deforms a 2D grid onto the object's surface to achieve higher-quality point cloud reconstructions. The approach can be used for point cloud completion, as in [15]. They propose an autoencoder to predict a dense, complete representation of a partial input point cloud. Similarly, [16] first predicts a depth map from a single RGB image, then uses it to calculate a partial point cloud based on the camera geometry and lastly predicts the full, dense point cloud with a folding-based encoder. [8] also predicts a coarse point cloud first, and then uses global and local features extracted at hierarchically increasing resolutions to obtain a dense point cloud.

Our proposed approach builds upon PSGN and combines it with a folding-based decoder to predict a dense point cloud from a single input image.

3. Method

Our method comprises two stages. In the first stage, a variant of PSGN generates a coarse point cloud $\{(x_i, y_i, z_i)\}_i^N$ given a single RGB image of an object. In the second stage, we apply a folding-based decoder to the coarse point cloud and the global feature vector to generate a denser point cloud.

3.1. Coarse Point Cloud Generation

To generate a coarse point cloud from a single RGB image of an object, we follow the architectures presented in [2]. We explore three kinds of PSGN architectures, namely Vanilla, Two-Branch and Hourglass (see Figure 2).

In the Vanilla architecture, the predictor consists of fully-connected (FC) layers and generates 1024 points. In the Two-Branch and Hourglass versions, we use a predictor network consisting of an FC branch and a parallel deconvolutional branch. They generate 256 points and a grid of 768 points respectively. Moreover, skip connections are used to enhance the information flow between the encoder and the predictor. The output of both branches is concatenated to form the final point cloud of 1024 points. The Hourglass network increases the depth of the network and utilizes more skip connections compared to the Two-Branch network, thus resulting in more representational power.

3.2. Folding Operation

In the second stage, we apply a *folding operation* to the coarse point cloud. We concatenate the point cloud with the global image feature vector extracted by the image encoder of PSGN and a uniform 2D grid of 4×4 points, which is then passed through a 3-layer MLP. This *folding* process deforms the 2D grid such that it approximates the local geometry around each point. This deformation of the 4×4 grid enables capturing the fine-grained details in the local neighborhood, while the coarse point cloud and the global image features encode the global shape structure. The output of the folding operation is added to the coordinate of each coarse point cloud to obtain the final dense output.

3.3. Loss and Evaluation Metric

Since point clouds are unordered, the loss function needs to be permutation invariant. Two popular losses for point cloud comparison are Chamfer distance (CD) and Earth Mover's distance (EMD). We use the CD loss, as it utilizes KD-trees to achieve a computational complexity of $O(\log(n))$, which is significantly faster than EMD ($O(n^2)$).

We define CD loss (L2) between two point sets S_1, S_2

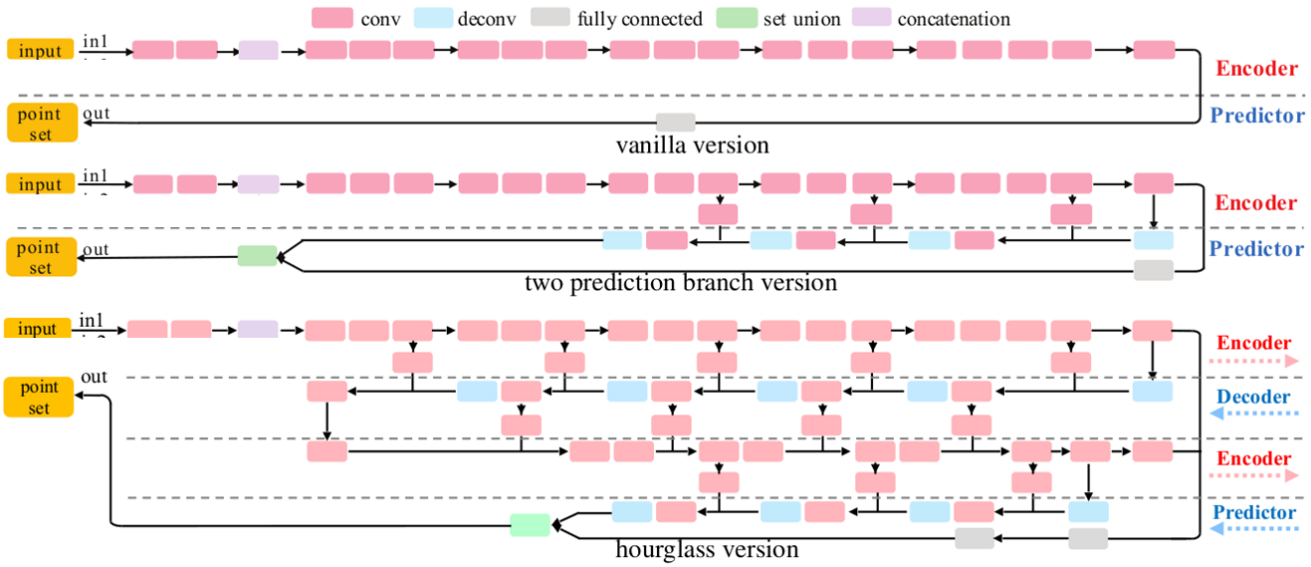

Figure 2. Overview of the three PSGN architectures, namely Vanilla, Two-Branch and Hourglass. Image taken from [2].

as:

$$CD(S_1, S_2) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} ||x - y||_2^2 + \frac{1}{|S_2|} \sum_{x \in S_2} \min_{y \in S_1} ||x - y||_2^2 \quad (1)$$

To evaluate our method, we use Chamfer distance (L1), which is defined analogously to CD (L2), but replaces squared L2 distance with L1 distance.

4. Training Details

To train our model, we use the shapes from the ShapeNet dataset [1] and the corresponding images of size 192×256 that were obtained by rendering different 2D views of the shapes. Specifically, we use a subset of the processed data provided by [2], which contains about 32,000 shapes with rendered images across all object categories of ShapeNet. The data split is set as 80/10/10 for training, validation and testing. During each training iteration, we compute the loss using 1024 points randomly sampled from the ground truth point cloud, which contains 16,384 points.

We implement our pipeline using PyTorch. For training our PSGN implementations, we use a batch size 64 paired with the Adam optimizer [5], with a learning rate of 0.0001 and a L2 weight decay factor of 0.000001. We train all networks until convergence using early stopping with a patience of 20. For training the folding-based decoder, we change the batch size to 16 and use a learning rate of 0.0003.

5. Results

We evaluate the performance of the implemented methods both quantitatively and qualitatively on a test set, which consists of 3198 images and their corresponding ground truth point clouds. Specifically, we evaluate the different PSGN architectures and our two-stage prediction pipeline that additionally applies the folding-based decoder.

5.1. Quantitative Results

5.1.1 Coarse Point Cloud Generation

We compare the implemented PSGN variations Vanilla, Two-Branch and Hourglass using Chamfer Distance (L1). A comparison of the results is in Table 1.

Model	Chamfer Distance (L1)
Vanilla PSGN (ours)	0.0765
Two-Branch PSGN (ours)	0.0743
Hourglass PSGN (ours)	**0.0737**

Table 1. Comparison of different PSGN architectures for coarse point cloud generation.

In our experiments, Hourglass PSGN produces the best result, while Vanilla PSGN produces a worse result than the other two models. The validation curves are recorded in Figure 3.

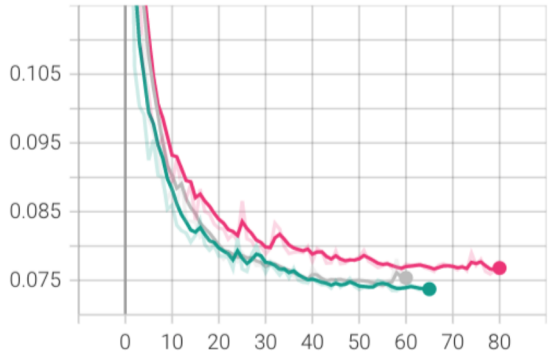

Figure 3. Chamfer distance (L1) for coarse point cloud prediction using different PSGN architectures, namely Vanilla (pink), Two-Branch (grey) and Hourglass (green).

5.1.2 Dense Point Cloud Generation

We further evaluate the results when applying our full pipeline to predict a dense point cloud. Instead of producing 1024 points, the dense output contains 16,384 points. A comparison of the results is in Table 2 and the validation curves are in Figure 4.

Model	Chamfer Distance (L1)
Hourglass PSGN (Coarse)	0.0737
Vanilla PSGN + Folding	0.0556
Two-Branch PSGN + Folding	0.0549
Hourglass PSGN + Folding	**0.0544**

Table 2. Dense point cloud prediction using our proposed method with different PSGN architectures, compared to coarse predicting using Hourglass PSGN.

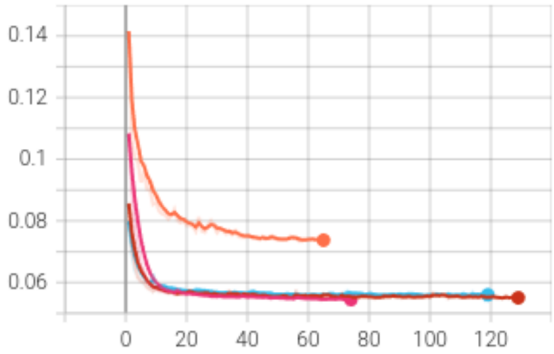

Figure 4. Chamfer distance (L1) for dense point cloud prediction using the folding-based decoder, pairs with Vanilla (blue), Two-Branch (red) and Hourglass PSGNs (pink)'s and compared to the coarse prediction of Hourglass PSGN (orange).

From our experiments, all variants of PSGN produce reasonably better results when the folding-based decoder is applied. Moreover, the Chamfer distance values for dense prediction are significantly lower than for the coarse predictions, as shown in 4.

5.2. Qualitative Results

The qualitative evaluation of Vanilla, Two-Branch and Hourglass PSGNs are recorded in Figure 5 in the supplementary material. From the visualization, we conclude that all PSGNs implemented have the capability to capture geometric information from the input images and are able to generalize to unseen inputs as well. Moreover, Hourglass PSGN produces slightly better results than the other two variants, which matches the quantitative evaluation in Section 5.1.

We also provide the qualitative results for dense point cloud prediction in Figure 6. From the visualization, we can conclude that the folding-based decoder is able to approximate the local geometry by deforming the input 4×4 grid.

6. Conclusion

In this work, we propose a two-stage pipeline for dense point cloud construction from a single RGB image. We quantitatively and qualitatively show that our method is able to produce dense, high-quality point clouds and outperforms PSGN. However, our proposed approach still leaves room for improvements, as it tends to predict over-smoothed local geometry. Future work can focus on tackling this problem, e.g. by using more folding operations in order to hierarchically increase the resolution of the point cloud in multiple steps. Finally, we hypothesize that re-training with the full ShapeNet dataset will lead to performance boosts.

References

[1] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 3

[2] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *CoRR*, abs/1612.00603, 2016. URL http://arxiv.org/abs/1612.00603. 1, 2, 3

[3] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. *CoRR*, abs/1807.03520, 2018. URL http://arxiv.org/abs/1807.03520. 1

[4] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *CoRR*, abs/1706.01307, 2017. URL http://arxiv.org/abs/1706.01307. 1

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980. 3

[6] Kejie Li, Trung Pham, Huangying Zhan, and Ian Reid. Efficient dense point cloud object reconstruction using deformation vector fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 497–513, 2018. 1, 2

[7] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. *CoRR*, abs/1706.07036, 2017. URL http://arxiv.org/abs/1706.07036. 1

[8] Priyanka Mandikal and R. Venkatesh Babu. Dense 3d point cloud reconstruction using a deep pyramid network. *CoRR*, abs/1901.08906, 2019. URL http://arxiv.org/abs/1901.08906. 2

[9] Priyanka Mandikal, Navaneet K. L., Mayank Agarwal, and R. Venkatesh Babu. 3d-lmnet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image. *CoRR*, abs/1807.07796, 2018. URL http://arxiv.org/abs/1807.07796. 1

[10] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016. URL http://arxiv.org/abs/1612.00593. 1

[11] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017. URL http://arxiv.org/abs/1706.02413. 1

[12] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. *CoRR*, abs/1611.05009, 2016. URL http://arxiv.org/abs/1611.05009. 1

[13] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Interpretable unsupervised learning on 3d point clouds. *CoRR*, abs/1712.07262, 2017. URL http://arxiv.org/abs/1712.07262. 1, 2

[14] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. *CoRR*, abs/2108.08839, 2021. URL https://arxiv.org/abs/2108.08839. 1

[15] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: point completion network. *CoRR*, abs/1808.00671, 2018. URL http://arxiv.org/abs/1808.00671. 1, 2

[16] Wei Zeng, Sezer Karaoglu, and Theo Gevers. Inferring point clouds from single monocular images by depth intermediation. *CoRR*, abs/1812.01402, 2018. URL http://arxiv.org/abs/1812.01402. 1, 2

Supplementary Material

A. Visualizations for Coarse Point Cloud Prediction

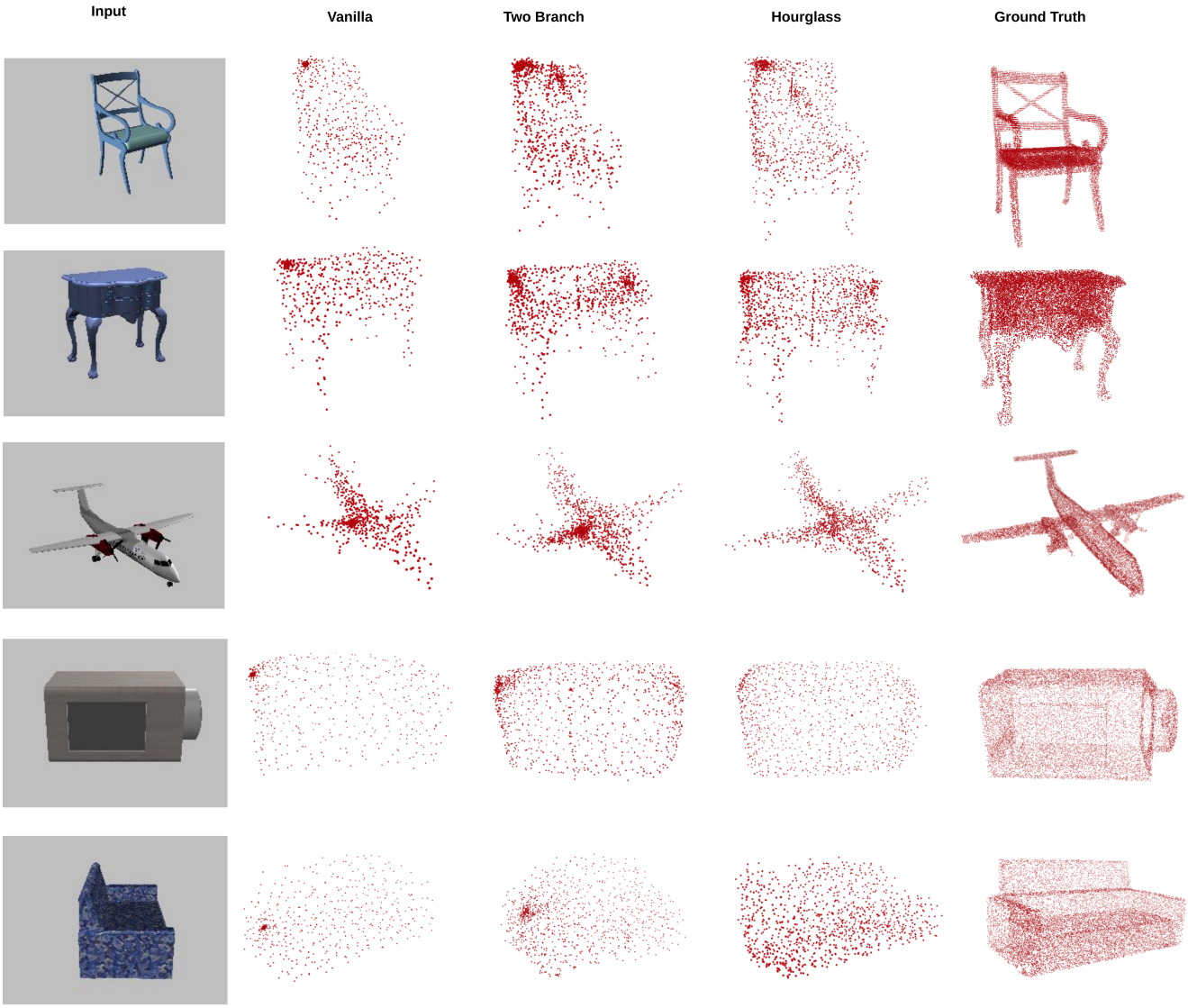

Figure 5. Qualitative results of various PSGN models.

B. Visualizations for Dense Point Cloud Prediction

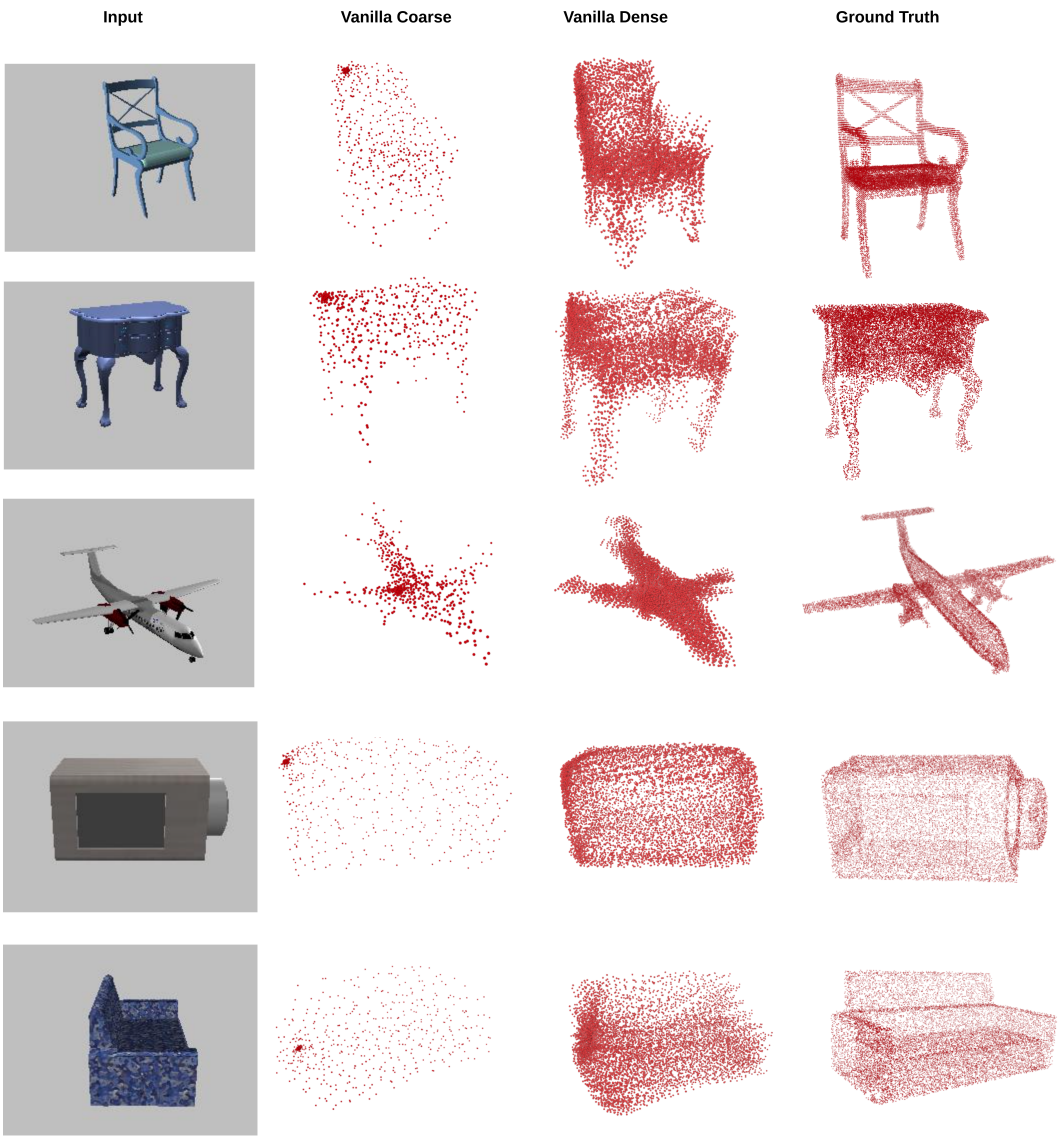

Figure 6. Qualitative results of dense point cloud prediction.