CS230

CS230 Project Final Report
3D Point Cloud Completion

Chen Wang
SCPD
Stanford University
cwang48@stanford.edu

Abstract

3D point cloud is one significant 3D representation, which contains the xyz coordinates of points on object surface that sampled by 3D sensors. The point cloud completion is a key task in the 3D world. In this project, encoder-decoder architectures are studied. The local feature extraction capability of encoder and local feature reconstruction capability of decoder are the key factors to impact the completion performance.

1 Introduction

The 3D point is represented by the xyz coordinates of the point. The point cloud is the set of 3D points sampled around the surface of objects in the scene. With increasing improvement and availability of 3D acquisition technologies, 3D sensors, and measurement, such as 3D scanner, RGB-D cameras, and LiDAR, 3D point cloud becomes one significant 3D representation. Because of its rich geometric, shape, and scale information, it is very convenient to represent the real-world, and be applied to autonomous vehicles, robotics, augmented and virtual reality area [1, 2].

Applying deep learning on 3D point cloud data has many challenges. 1. Irregularity: the points are not evenly sampled on the surface of the object; 2. Unstructured: the point is not on a regular grid, unlike pixel on a 2D grid; 3. Unorderdness: the order of the points does not change the object represented.

The real-world 3D point cloud is often incomplete, which may lose geometric and semantic information. For example, the LiDAR on a vehicle or the 3D sensor on a robot can only sense the object's surface that faces it. Like the image completion task for the 2D world, the point cloud completion is a key task in the 3D world. In this project, I studied the 3D object shape completion based on **PointNet** [3, 4] and **encoder-decoder architecture**. The **input** of the model is a partial 3D point cloud, and the **output** is expected to have the same shape as the ground-truth completed point cloud.

2 Related Work

The pioneering works, PointNet [3] and PointNet++ [4], published in 2017. They combine point-wise multilayer perceptrons (MLP) with a symmetric aggregation function (maxpool) that achieve invariance to permutation and robustness to perturbation, which are essential for direct and effective feature learning on raw 3D point clouds.

In the past two years, several advanced works have been done to explore different encoder and decoder networks to achieve better 3D point cloud completion.

CS230: Deep Learning, Fall 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Encoders: Based on PointNet's good feature extraction performance, many works designed encoder networks based on it and its variants. Point Completion Network (PCN) [5] equips with an encoder that consists of two stacked PointNet layers, which can mix the local and global geometry information. PF-Net [6] goes further. It utilizes Combined Multi-Layer Perception (CMLP) to extract multi-layer features contain both local and global features, also low-level and high-level features, enhancing the ability of the network to extract semantic and geometric information.

Decoders: TopNet [7] proposes a decoder modeled as a hierarchical rooted tree in which each node of the tree represents a subset of the point cloud and the root of the tree represents the entire point cloud. It has the ability to represent arbitrary topologies and structure on discrete point sets, and it is a more efficient and compact representation. PF-Net [6] also presents a hierarchical decoder structure: Point Pyramid Decoder (PPD), which is a multi-scale generating network based on multi-scale feature vectors. Each feature vector is responsible for predicting point cloud in different resolutions. By predicting primary, secondary, and detailed points, it preserves multi-scale geometry information.

Other works, such as Gridding Residual Network (GRNet) [8] and Cascaded Refinement Network [9], also tried different strategies to push this task forward.

3 Dataset

In this project, Completion3D benchmark [7] is used as the dataset. The Completion3D benchmark is a subset of ShapeNet dataset [10]. The data is uniformly sampled on synthetic CAD models. There are 8 categories: Plane, Cabinet, Car, Chair, Lamp, Couch, Table, Watercraft.

The data is composed of 28974 **training** samples and 800 **validation** samples. The ground-truth point cloud data is 2048 points. The input partial point cloud data is back-projecting 2.5D depth images into 3D, which also has 2048 points, but all within the partial shape. **Sample point clouds** will be shown in Figure 3.

The date is **augmented** by 1) random rotating around the z-axis, 2) mirroring about x- or y- axes with 50% probability. The raw 3D point cloud is processed by the network directly.

4 Model

This project employed the encoder-decoder architecture to complete 3D point cloud. The encoder encodes the input partial 3D point cloud into an embedding (feature vector), and a decoder generates the complete 3D point cloud from the embedding. Unlike an auto-encoder, it is not explicitly enforces the network to retain the input points in its output. Instead, it learns a projection from the space of partial observations to the space of complete shapes by minimizing the Chamfer distance (CD).

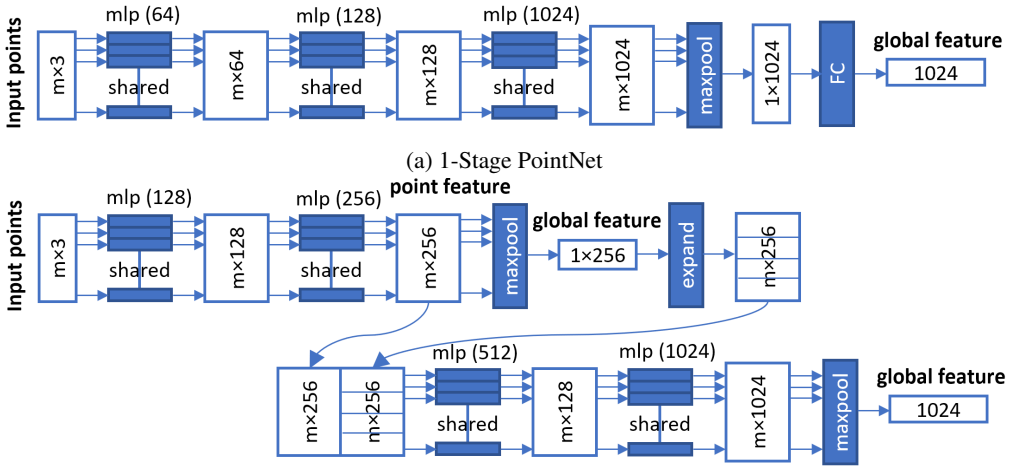
4.1 Encoder

The encoder is in charge of summarizing the geometric information in the input point cloud as a feature vector.

Two encoder networks are studied in this project: PointNet [3] and 2-stage PointNet [5]. PointNet is a network to learn the pointwise features directly from 3D point cloud, then uses the learned features to perform 3D object classification and segmentation, which has been proven to have a good feature extraction performance. According to transfer learning, part of PointNet (the network before global feature) is used as the encoder (Figure 1a). The PointNet consumes m 3D points as input. The input matrix is $m \times 3$ where each row is the 3D coordinate of a point (x, y, z). It uses the shared pointwise Multi-Layer Perceptrons (MLP) to convert the points from the initial 3D dimension to a higher dimension [64, 128, 1024], then uses the max-pooling to aggregates the point features. This architecture achieves invariance to permutation and robustness to perturbation, which are essential for effective feature learning on point clouds.

A 2-stage PointNet is also studied, which stacks two PointNets (Figure 1b). The first stage of PointNet converts the points from the initial 3D dimension to a higher dimension [128, 256], then a max-pooling layer aggregates the global feature. Instead of passing this global feature to the decoder, it concatenates the global feature with the point feature right before the max-pooling, and passes the combined feature to the second stage of PointNet. Similarly, the second stage converts the combined

feature to a higher dimension [512, 1024], then generates the final feature. This is a good way to preserve both local and global information in the final feature [5].



(a) 1-Stage PointNet

(b) 2-Stage PointNet

Figure 1: Encoder architectures: (a) 1-stage PointNet , (b) 2-stage PointNet.

4.2 Decoder

Two decoder networks are studied in this project: fully-connected decoder and Rooted Tree Decoder [7]. A 4-layer fully-connected decoder serves as the baseline of this project. Each fully-connected layer has 256, 512, 1024, and 2048 $\times$ 3 units. The fully-connected decoder is good at predicting the global geometry of the point cloud. Nevertheless, it causes loss of local geometric information since it only uses the final layer to predict the shape [5].

The structure of rooted tree decoder is as Figure 2 shows. The root node processes the global feature vector from encoder, and learns the local feature for its child nodes. The child node concatenates the global feature from the encoder and local feature from its parent and learns even lower-level features. To the last level of the tree, each leaf node generates a subset of the entire point cloud. In this project, a 4-level tree configuration is used. From top to bottom, the nodes at each level have [4, 8, 8, 8] children. Each node generates [8,] local feature vector, and then concatenated with [1024,] global feature vector. Each MLP block represents several MLP layers. The same colored MLP share the same parameters.

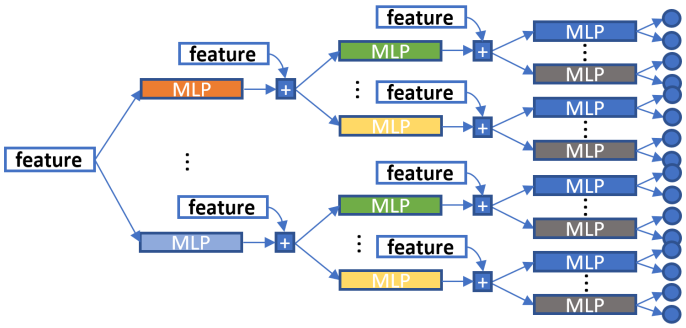


Figure 2: Rooted Tree Decoder architectures. This is for the illustration of the topology. It is not the exact configuration used in this project.

4.3 Loss Function

The loss function measures the average difference between the output point clouds and the ground truth point clouds. The Chamfer distance [11] is used as the metric. For each point, the Chamfer

distance finds the nearest neighbor in the other set and computes their squared distances which are summed over both generated S and ground-truth S_G sets.

$$d_{CD}\left(S, S_G\right) = \sum_{x \in S} \min_{y \in S_G} \|x - y\|^2 + \sum_{y \in S_G} \min_{x \in S} \|x - y\|^2 \tag{1}$$

$$\text{Loss} = \frac{1}{m} \sum_{S \in \text{batch}} d_{CD}\left(S, S_G\right) \tag{2}$$

5 Experiments

All combinations of 2 encoders and 2 decoders (total 4 model) are evaluated quantitatively and qualitatively on the task of 3D point cloud completion.

5.1 Training Setup

Adam and Adagrad **optimizers** with **learning rate** of 10^{-2}, 10^{-3}, and 5×10^{-4} are tried to train the models. Learning rate 10^{-2}, 10^{-3} are too large. The loss stuck at certain level and oscillate, cannot converge to the optima. I finally choose Adam optimizer with initial learning rate of 5×10^{-4}, and learning rate decay 0.707 every 30 epochs. The initial learning rate is large enough to learn rapidly at the beginning. Decayed learning decay helps it continuously learn and reach to the optimal. Other hyperparameters are 200 epochs and a batch size of 32.

The best model is chosen based on the validation set.

5.2 Evaluation

The four models, PointNet encoder + fully-connected decoder (PN+FC), PointNet encoder + Rooted Tree decoder (PN+RT), 2-stage PointNet encoder + fully-connected decoder (2PN+FC), and 2-stage PointNet encoder + Rooted Tree decoder (2PN+RT), are evaluated across 8 categories. For each class, the mean Chamfer Distance is computed across all class instances. The **final metric** for the models is the mean Chamfer Distance across all classes. The results of the evaluation of all models are as Table 1 shows.

Model /Dev.	Chamfer Distance (CD) [10^{-4}]								
	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Watercraft	Mean
PN+FC	7.44	24.79	10.26	24.85	31.43	18.67	26.96	14.82	19.90
PN+RT	7.60	25.04	10.79	27.87	24.48	19.45	**22.05**	13.24	18.81
2PN+FC	7.21	25.11	**10.22**	24.99	27.20	**18.55**	25.07	14.35	19.09
2PN+RT	**6.86**	**23.60**	10.23	**23.50**	**19.07**	18.56	22.36	**11.61**	**16.97**

Table 1: Point cloud completion quantitative results on ShapeNet: comparison of four networks.

From the results, if we fixed the encoder, the networks with rooted tree decoder outperform those with fully-connected decoder. This is because that rooted tree tends to generate structured point cloud by generating a point cloud as a collection of its subsets. However, the fully-connected decoder tends to predict the global geometry of the point cloud and lost local geometric information. If we fixed the decoder, the networks with the 2-stage PointNet encoder outperform those with 1-stage PointNet encoder. This is because the 2-stage mixes local and global features and provides more information for the decoder. However, with the FC decoder, 2PN is not much better than PN, which may because the FC is the bottleneck and has limited capability to process the local information. Overall, 2PN+RT is the best model with the lowest Chamfer Distance.

Figure 3 shows the qualitative comparisons on shapes. As can be seen, the outputs of models with FC are accurate but the points are overly concentrated in certain regions, the boundary of local geometry is blurry and contains more floating points. The outputs of models with 2PN have more accurate local geometry details and local point density. The overall difference is subtle, may needs more varied training samples, and longer training time to have a distinct difference.

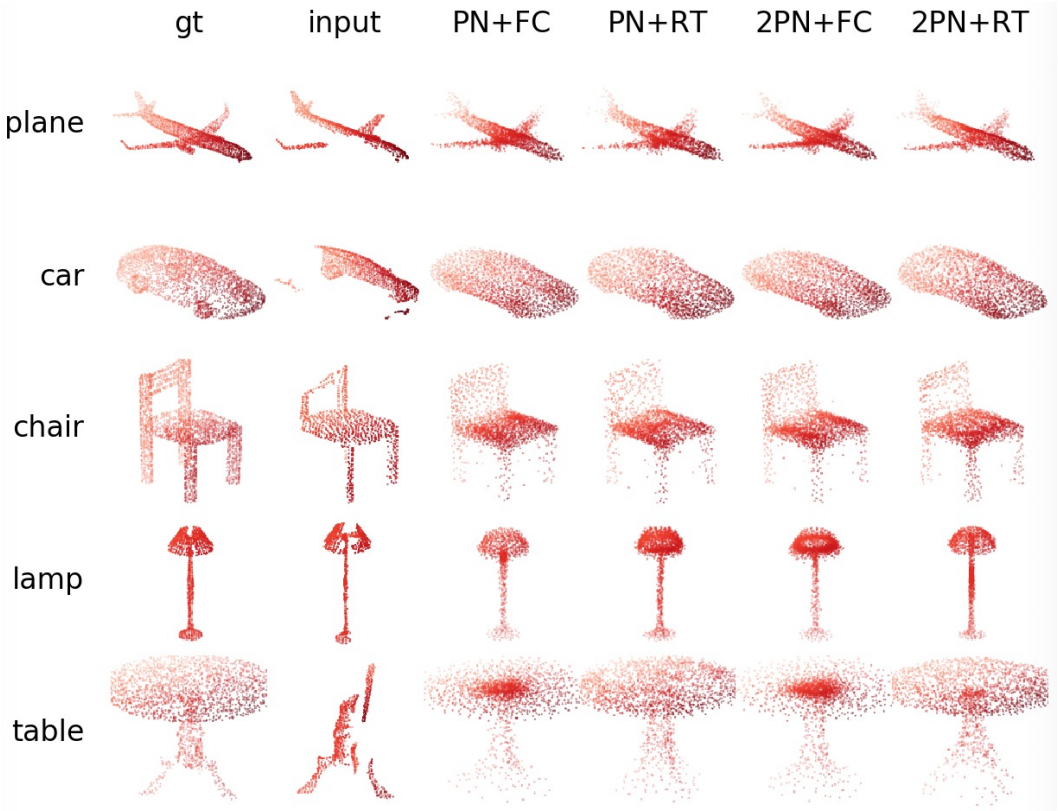


Figure 3: Qualitative completion results on ShapeNet. Top rows are the ground truth, middle rows are partial point clouds as input, bottom rows are completion results as output.

5.3 Number of Parameters

The number of parameters of each model is analyzed since it can influence the performance. Table. 2 lists the number of parameters of each encoder and decoder. The total number of parameters of one model is the sum of one encoder and one decoder. 2-stage PointNet encoder and rooted tree decoder lead to better performance, but have fewer parameters, which suggests that the number of parameters is not the primary reason for the good performance.

Nets	PN	2PN	FC	RT
# Params	1.2M	0.8M	7.2M	2.4M

Table 2: Number of trainable model parameters

6 Conclusion and Future Work

In this project, I studied the 3D point cloud completion performance of different encoder-decoder architectures. The 2-stage PointNet encoder + rooted tree decoder has the highest performance. For better performance, the encoder should have the capability to extract both global and local geometry features, and the decoder should the capability to distinguish the global and local information and reconstruct them using different strategies. For future work, within each encoder/decoder network, there are hyperparameters like the number of layers, and the dimension of features could be analyzed.

References

[1] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. PP, 2020.

[2] S. A. Bello, S. Yu, and C. Wang, "Review: deep learning on 3d point clouds," *Remote. Sens.*, vol. 12, p. 1729, 2020.

[3] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017.

[4] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 5099–5108, Curran Associates, Inc., 2017.

[5] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in *3D Vision (3DV), 2018 International Conference on*, 2018.

[6] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "Pf-net: Point fractal network for 3d point cloud completion," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7659–7667, 2020.

[7] L. P. Tchapmi, V. Kosaraju, S. H. Rezatofighi, I. Reid, and S. Savarese, "Topnet: Structural point cloud decoder," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[8] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "Grnet: Gridding residual network for dense point cloud completion," in *ECCV*, 2020.

[9] X. Wang, M. H. Ang Jr, and G. H. Lee, "Cascaded refinement network for point cloud completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 790–799, 2020.

[10] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *CoRR*, vol. abs/1512.03012, 2015.

[11] H. Fan, H. Su, and L. Guibas, "A point set generation network for 3d object reconstruction from a single image," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2463–2471, 2017.