11-785 3D Object Inpainting Final Report

Project Team: ShallowLearners
{liuyuex, jinchenl, guilinz, zhitaow} @andrew.cmu.edu

1 Problem Statement

Inpainting is a process of restorative conservation where damaged, deteriorating, or missing parts of an artwork are reconstructed(1). In computer vision, most of the traditional methods focus on 2D image inpainting, where incomplete images are reconstructed. However, with the fast development of 3D perception technologies, demand for 3D inpainting has become more significant than ever. In this project, we propose a novel approach to perform 3D pointcloud object inpainting while leveraging Generative Adversarial Network (GAN) to repair incomplete 3D pointclouds. By improving conventional deep learning architectures, we have been able to achieve comparable performance against the state-of-the-art 3D pointcloud completion models. When fine tuned, the proposed model can potentially be applied to solve complex real-world problems in autonomous driving and city mapping. This work can be beneficial to research efforts in many fields such as self-driving, 3D mapping, and 3D data annotation.

2 Related Work

2.1 3D Shape Learning

Pointcloud is a lightweight and highly transferable 3D data representation that can be acquired directly from stereovision sensors. Its primary limitation, however, is that the acquired pointclouds are prone to partial incompletion at local regions where the light reflectively of the object surface is poor. For applications where dense pointcloud is required, the use of high resolution LiDAR sensor becomes a costly but inevitable solution. Deep learning models have shown promising results to augment the real-world corrupted 3D scans with synthetic data and produce high resolution shape prediction. Wang et al. (2) constructed a hybrid model with convolutional encoder generative adversarial network and long-term recurrent convolutional network for shape inpainting from corrupted 3D pointcloud model. Chen et al.(3) uses a implicit field encoder in replacement of a convolutional encoder for smooth 3D surface reconstruction. Park et al. (4) proposed a signed distance function (SDF) based approach to learn the implicit field and produce fully continuous object geometry without discretizing the SDF into regular grids. The CNN encoder networks typically require less training time to obtain the voxel model, whereas the implicit encoder networks require longer training time since it requires all points in a voxel grid be passed into the network before generating a model.

2.2 Generative Adversarial Network (GAN)

In recent works, generative adversarial networks (GAN) and their variants have been adopted to learn deep embeddings of target by training the discriminators and generators adversarially against each other. The applications of the architecture have shown motivating results in generating realistic images scenes, objects or portraits but GANs could produce unstable result and are prone to divergence issues. Radford et al.(5) proposed a set of constraints on the architectural topology of convolutional GANs which lead to more stable training convergence. In 3D shape learning domain, auto-encoders' ability to learn deep feature embedding has been recognized and adopted to compliment the GAN architecture. Variational Auto-encoders (VAE) learning scheme(6) perturbs the bottleneck features

with Gaussian noise to encourage smooth and complete latent space robust to noise or variations in the input space. Larson et al.(7) combined VAEs and GANs into an unsupervised generative model to simultaneously learn feature embedding, model representation and similarity measures. The promising result demonstrated by the coupled architecture encourages further investigation in its potential capabilities.

2.3 3D Object Detection

With an abundance of stereovision sensors, 3D object detection has been increasingly popular in recent years owing to its wide applications in various areas such as autonomous driving and robotics. Deep learning models have shown remarkable performances in 2D detection tasks, with average precision (AP) over 95%. Its performance in 3D applications is however not as comparable, with object detection average precision at around 80%(8). Since most of the recent 3D detection approaches make use of LiDAR pointclouds, we notice that one of the key factors leading to the huge difference between 2D and 3D detection precision is the incomplete nature of real-life acquired LiDAR data. Datasets such as ModelNet(9) provide 3D models of in-door objects and uniformly distributed high quality pointcloud data can be sampled from the models. However, it is impossible for real time systems such as self-driving cars to obtain perfect LiDAR scans of objects on roads. We have observed that most of the objects in the KITTI(8) dataset are incomplete, which may result in the lower precision of 3D object detection tasks. In light to overcome this difficulty, we attempt to leverage the theoretical foundation and practical capability of GAN to repair pointclouds suffering from incomplete shapes.

2.4 State-of-the-art (SOTA) Architechtures

- **L-GAN**: L-GAN(10) introduces the first deep-learning-based network for pointcloud completion by utilizing an Encoder-Decoder framework. Coupled with an adversarial player, the proposed Auto-encoder network is able to produce faithful samples and cover most of the ground truth distributions.
- **3D-Capsule**: 3D point capsule networks(11) surpasses the performance of other methods and becomes the state-of-the-art Auto-encoder in the domain of pointcloud completion. The architecture utilizes a dynamic routing scheme and its peculiar 2D latent space bring in improvements for several common pointcloud-related tasks, such as object classification, object reconstruction and part segmentation.
- **Point Completion Network (PCN)**: The PCN architecture is an encoder-decoder network which directly operates on raw pointclouds without any structural assumption or annotation about the underlying shape(12). The encoder takes the input pointcloud and outputs a feature vector, which is then taken by the decoder to produce two pointclouds with different resolutions. Such point-based completion method is proven to be more scalable and robust than voxel-based methods, which makes it a better candidate for repairing geometrically complex shapes.
- **Point Fractal Network (PF-Net)**: As the latest approach for pointcloud completion, PF-Net focuses on preserving the spatial arrangements of the incomplete pointcloud and figuring out the detailed geometrical structure of the missing region(s), instead of generating the overall shape of the pointcloud from the incomplete pointcloud(13). It estimates the missing pointcloud hierarchically by utilizing a pyramidal feature fusing network for generation. Such an architecture enables the network to generate the target pointcloud with both rich semantic profile and detailed contours while retaining the existing general shape.

2.5 Proposed Network

In this report we would like to present our proposed architecture for 3D pointcloud completion, as is shown in Figure 1. Pointcloud data can be seen as feature points that map out the general skeleton of an object. The input to the proposed network is incomplete pointcloud with part of the pointcloud randomly cropped from a complete pointcloud generated by sampling surface points from ShapeNet models.

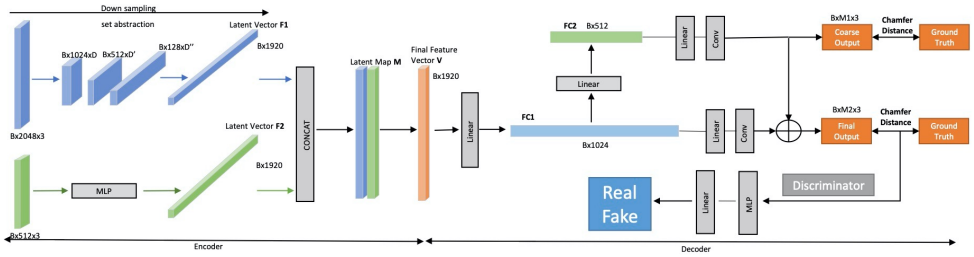


Figure 1: Architecture of Proposed Network

2.6 Multi-scale Encoding

The incomplete pointcloud is fed into the network through two distinct channels, each with a different resolution. The first channel would process the incomplete pointcloud without downsizing through a set abstraction network(14), originally proposed in the PointNet++ network. By doing this we are hypothesizing that the set abstraction network, with its increasingly large reception field, would be able to extract both global and local feature representative of the original pointcloud geometry. In addition to the set abstraction network, we are also incorporating a second channel with a set of convolutional multi-layer perceptrons to extract features from a lower resolution input. The iterative farthest point sampling method has been used to down-sample from the incomplete pointcloud and produce a lower resolution copy for the second channel. The feature vectors from the two channels are then concatenated at the end of the encoder network.

2.7 Multi-resolution Point Generation

Symmetric to the encoder network, the decoder also incorporates two channels for output at different resolutions. Both of the decoder channels consist of a series of multi-layer perceptrons. It is worth noting that the network would only produce points at the missing regions, in opposed to the mainstream approach that interpolates over the entire shape. This approach would in theory save us computation overhead that is often cumbersome and unnecessary for the shape completion task. We are also hypothesizing that this approach would not introduce additional noise to the resultant pointcloud and hence lead to smoother generated contour at a higher quality.

Autoencoders by themselves can typically reconstruct the learned distribution, with a lower resolution. In order to improve the performance of the network by preventing the autoencoder network from over-fitting to a flattened distribution, we are proposing to introduce a discriminator as part of the overall architecture. The discriminator learns the polynomial moments of the real distribution so the inclusion of the adversarial loss would make the interpolation closer to the distribution of a real pointcloud. The discriminator network is formulated to include a set of convolutional multi-layer perceptron similar to that used in the encoder structure.

2.8 Loss Functions

From each of the two decoder channels, the pointcloud generated are compared to the pointcloud cropped off from the original cloud through Chamfer loss. The Chamfer loss averages all of the nearest neighbor distance. It offers a mean to assess the generic L2 loss of the generated points against the ground truth points.

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

The two Chamfer loss values respectively from the fine and course channels are aggregated together to form the generator loss.

$$L_{gen} = d_{CD1}(Y_{detail}, Y_{gt}) + \lambda d_{CD2}(Y_{course}, Y'_{gt}) \sum_{x \in S_1} \tag{2}$$

The proposed adversarial loss is calculated based on the softmax outputs from the real cropped pointcloud and the interpolated pointcloud.

$$L_{adv} = \sum_{1 \leq i \leq S} log(D(y_i)) + \sum_{1 \leq j \leq S} log(1 - D(F(x_i))) \tag{3}$$

The weighted generator loss derived from the autoencoder network and the adversarial loss from the discriminator together optimizes the double resolution pointcloud generation network.

$$L_{total} = \alpha_{gen} L_{gen} + \alpha_{adv} L_{adv} \tag{4}$$

3 Dataset

For our project, we have identified three main data sources for model training:

- **ShapeNet**: ShapeNet is a richly-annotated, large-scale repository of shapes represented by 3D CAD models of objects. It contains 3D models from a multitude of semantic categories and organizes them under the WordNet taxonomy. It is a collection of datasets providing many semantic annotations for each 3D model. (15)
- **KITTI Vision Benchmark Suite**: KITTI contains a suite of vision tasks built using an autonomous driving platform. The full benchmark contains many tasks such as stereo, optical flow, visual odometry, etc. This dataset contains the object detection dataset, including the monocular images and bounding boxes. (8)
- **Argoverse**: The data in Argoverse comes from a subset of the area in which Argo AIs self-driving test vehicles are operating in Miami and Pittsburgh – two US cities with distinct urban driving challenges and local driving habits. The dataset contains 324,557 interesting vehicle trajectories extracted from over 1,000 driving hours. (16)

4 Experiments

4.1 Data Generation and Training Configurations

Over the past month, we have performed literature search for the topic of pointcloud completion. Based on the reviewed literature, we have identified PF-Net(13) to be an architecture that provides high quality results without imposing too much of computation overhead when compared to the other architectures. In order to further examine the underlying reasoning behind its design, we decided to implement the proposed network and attempt to reproduce the result.

In an attempt to arrive at similar results, we trained the network for 170 epochs on the ShapeNet dataset as benchmarked in the original paper. For each of the object in the training set, 2048 points are uniformly sampled from the mesh surface. In order to recreate a realistic representation of incomplete pointcloud data, the incomplete pointcloud is generated by randomly selecting a view point as the center and removing a pre-defined number of points within a certain radius from the complete pointcloud. In this way, each of the ShapeNet objects are randomly cropped before feeding into the network. The points cropped away from the complete pointcloud data are then used as part of the generator loss function to evaluate quality of the generated pointcloud set.

4.2 Implementation of Proposed Architecture

The network was implemented with Pytorch library and coded in Python. All of the three sub-networks are trained with ADAM optimizer since it has experimentally proven its effectiveness for training GAN networks in previous literatures. The initial learning rate was set to 0.0002 for fast convergence. The network was trained on a single Geforce GTX 1080Ti graphics card with batch size set to 10 to fit into the GPU memory. Batch normalization and ReLU activation units are deployed in the encoder and discriminator networks. Batch normalization was not incorporated in the decoder network until at the layer before output to avoid any loss of information.

4.3 Completion Results on ShapeNet

To empirically evaluate the performance of our approach on the ShapeNet dataset, we visualized both cropped and generated pointclouds with a subset of the objects to make a comparison. The generated points in the missing region are colored pink in the graphs.

Figure 2 shows the comparison of completion results between other aforementioned state-of-the-art methods and our approach. (left to right: input, L-GAN(10), PCN(12), 3D-Capsule(11), PF-Net(13), our method, and ground truth) It is obvious that our model can generate the missing point cloud accurately with less distortion and high-level restoration compared with other methods, especially L-GAN, PCN, and 3D-Capsule. Our results are also visually comparable to those of PF-Net.

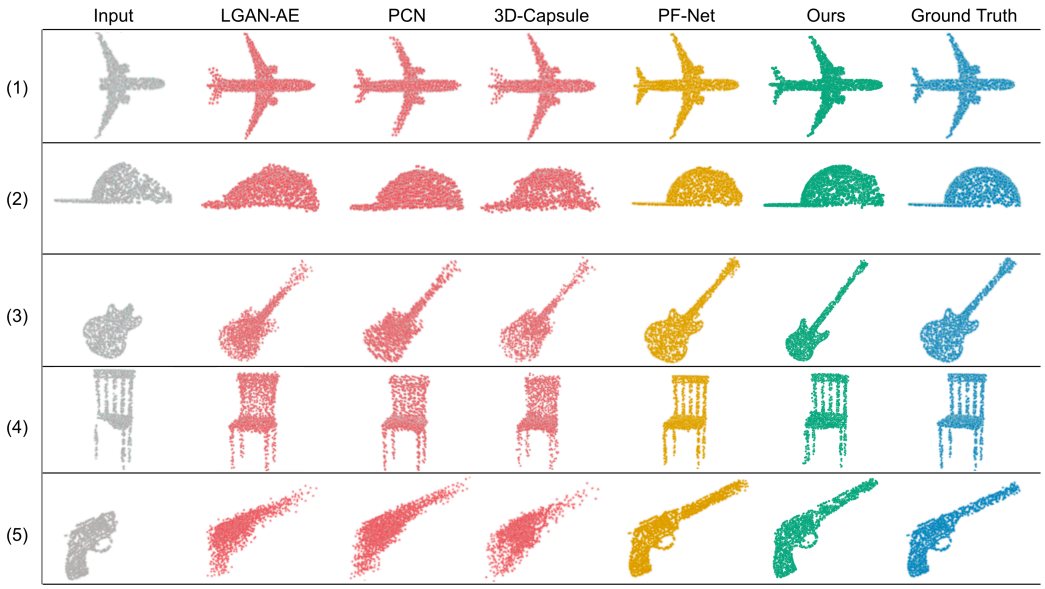
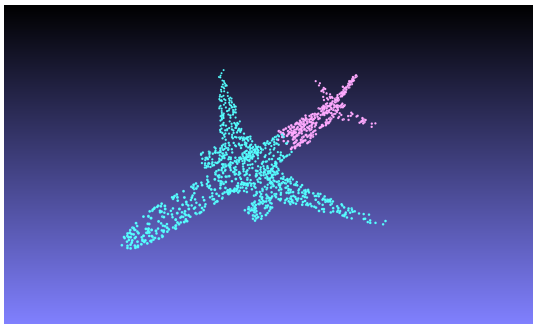
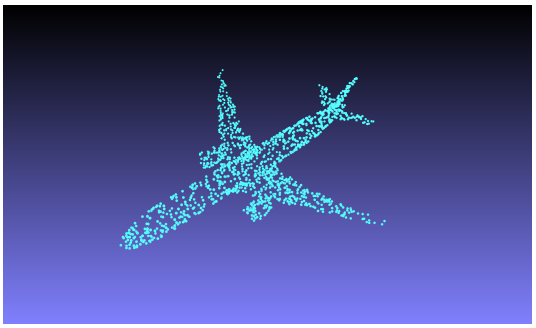


Figure 2: Cross-compared result of the networks

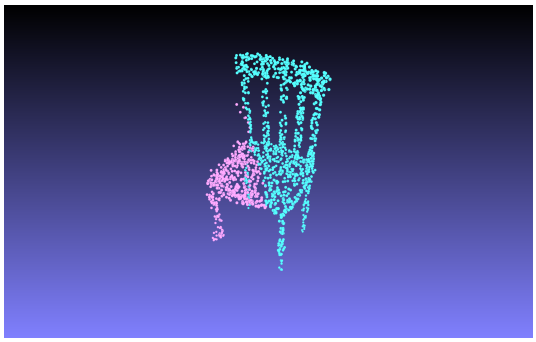
As we can see from the point cloud graphs in Figure 3, the generated points in the missing region well resembles the ground truth in terms of contour and distribution. With the idea of generating points only in the missing region, we preserved the raw distribution of the existing points so that our model output could match the ground truth to a maximum extent.



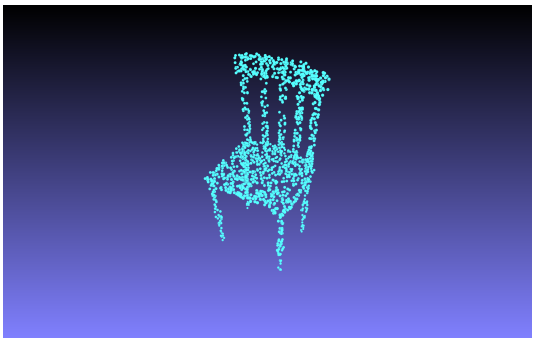
(1) Airplane (generated)



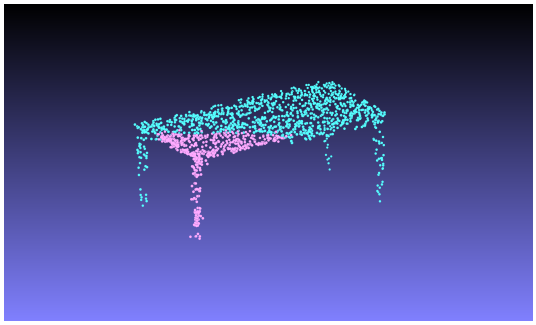
(2) Airplane (ground truth)



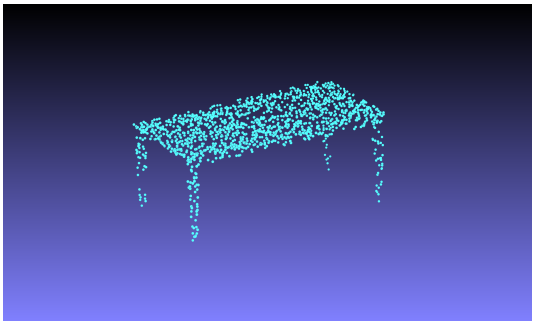
(3) Chair (generated)



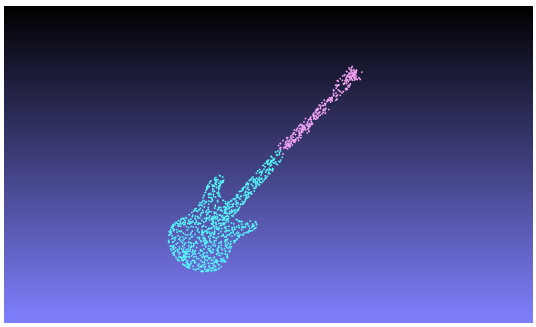
(4) Chair (ground truth)



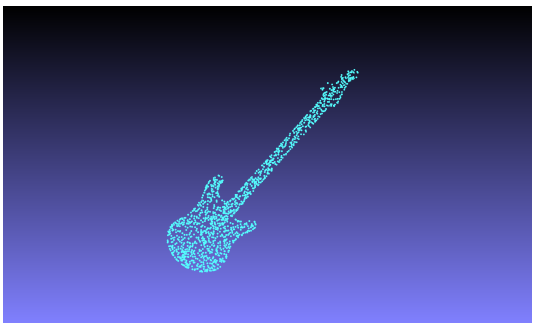
(5) Table (generated)



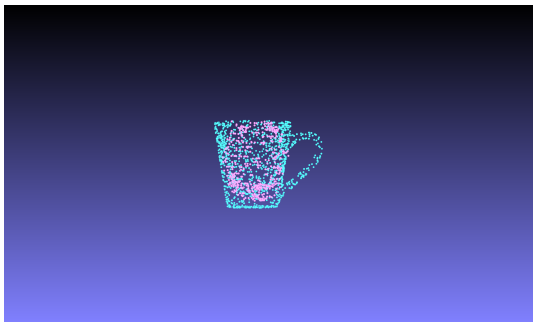
(6) Table (ground truth)



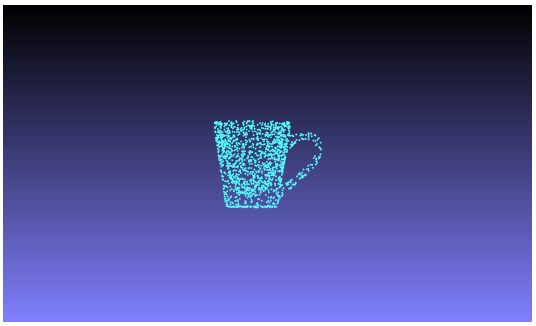
(7) Guitar (generated)



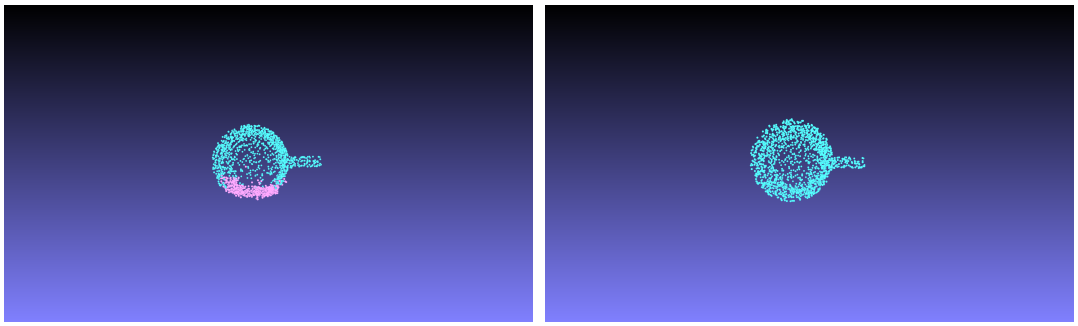
(8) Guitar (ground truth)



(9) Mug (generated; horizontal view)



(10) Mug (ground truth; horizontal view)



(11) Mug (generated; vertical view) (12) Mug (ground truth; vertical view)

Figure 3: Visualizations of Sample Results for Empirical Comparison

4.4 Quantitative Evaluation Result

In order to quantitatively evaluate our method, we tested its performance on the ShapeNet dataset to calculate the average Chamfer distance over 13 object categories and compared the results with some of the SOTA models, which are summarized in Table 1.

Object category	LGAN-AE	PCN	3D-Capsule	Our method
Plane	3.357	5.060	2.676	**2.319**
Bag	5.707	**3.251**	5.228	6.020
Cap	8.968	7.015	11.04	**6.724**
Car	4.531	2.741	5.944	**2.592**
Chair	7.359	3.952	**3.049**	3.262
Guitar	0.838	1.419	**0.625**	0.839
Lamp	8.464	11.61	9.912	**5.483**
Laptop	7.649	3.070	2.129	**2.085**
Motor	4.914	4.962	8.617	**2.759**
Mug	6.139	**3.590**	5.155	5.045
Pistol	3.944	4.484	5.980	**2.319**
Skate board	5.613	3.025	11.49	**1.986**
Table	2.658	**2.503**	3.929	3.984
Overall mean	5.395	4.360	5.829	**3.493**

Table 1: pointcloud completion results of overall pointcloud. The training data consists of 13 categories of different objects. The numbers shown are [prediction $\rightarrow$ ground truth error], scaled by 1,000. We compute the mean values across all categories and show them in the last row of the table.

As we can see from the results, our method outperforms the other methods, including LGAN-AE, PCN and 3D-Capsule, in most of these categories. In terms of the mean Chamfer distances across all 13 categories, our method also has considerable advantages over these methods.

4.5 Investigation on Robustness

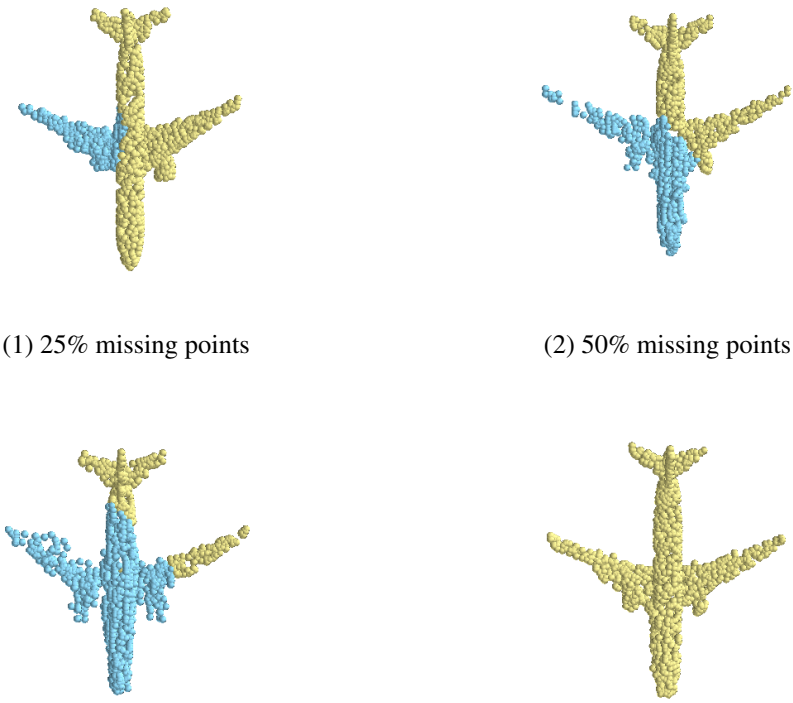
We conduct all robustness test experiments on the "Airplane" class since the airplane models carry both intricate and uniform contours. In the robustness test, we modified the number of missing points to be handled by the model and trained with this configuration. In this way, the trained model would be capable of repairing incomplete shapes with different extent of incompletion. The experimental results are shown in Table 2.

The percentages labeled in the table indicate that the input pointcloud loses 25%, 50%, and 75% of its original points. The prediction to ground truth and ground truth to prediction errors are the cross-compared Chamfer distance of the generated and original pointclouds. It can be seen that

Missig ratio	25%	50%	75%
Proposed network	2.183/1.531	1.961/1.537	2.514/2.001

Table 2: Robustness test on the proposed network. The numbers shown are [prediction → ground truth error/ground truth → prediction error], scaled by 1,000. The studied pointcloud loses 25%, 50%, and 75% of data from the original pointcloud.

our network has a strong capability of repairing incomplete pointclouds up to 50% of incompletion. Figure 4 shows the visualizations of our model's performance on the test set.



(1) 25% missing points

(2) 50% missing points

(3) 75% missing points

(4) Ground Truth

Figure 4: Visualizations of Robustness Test Results

It should be noted that our network is capable of retaining the geometric detail on the contour even with highly incomplete pointcloud. The slightly worse performance on the 75% case study can be explained by the fact that the network has not been fully trained due to time constraint and inevitable early stopping of the training.

5 Conclusion

To summarize, our proposed architecture is capable of completing shape from a partial pointcloud and make sure the completion only occurs in the missing region. Such a setting really preserves the original information of the pointcloud to a maximum extent. With a large enough dataset, our network is able to produce pointclouds with delicate details, and can also potentially improve the accuracy of 3D object recognition tasks. As the next steps, we would like to further improve this architecture to make it more generalizable to different object completion tasks.

Apart from summarizing our modeling process and experiments, we have also gained valuable experiences and insightful knowledge of 3D shape learning, which we would like to discuss in detail at the end of this report.

5.1 Power of Hierarchical Learning

Benefited from literature review and the exploration of various models, we find that hierarchical learning, which takes advantage of both global and local features, has great significance in ensuring effectiveness of 3D shape learning for feature extraction. With a nature of being disordered and sparse, pointcloud itself is where hierarchical learning tries to maximize the possible information obtained from scanning. It is the reason behind the fact that set abstraction method has been widely used in SOTA models since it was first proposed by PointNet++ in 2017(14). Further, we find that not only 3D shape learning benefits from hierarchical learning, other fields of deep learning also leverage this technique to meet their specific needs. It is notable that the pyramidal Bi-LSTM Encoder in our last homework assignment also implements a similar structure.

5.2 Significance of Keeping Feature Dimensions

While implementing our own architecture, we tried to preserve the dimension of points but meanwhile reducing the dimension of features during the process of hierarchical learning. The result was not as satisfactory as what we finally presented, which made us realize that it is important to keep the learned and encoded features. This concept is in some ways consistent with what we have learned in this course, such as utilizing features from the convolutional layers or MLP, or making use of hidden states output by the recurrent layers. Features are what are learned from the data instead of the number of points.

After all, the most important lesson we have learned, not only through this project, but also over this entire course, is keep trying, keep exploring, stay hungary, stay foolish, that's why we call us, Shallow Learners.

Our GitHub repository: `https://github.com/GuilinZ/Shallow-Learner-Project`

References

[1] Wikipedia, Inpainting, https://en.wikipedia.org/wiki/Inpainting, [Online; accessed 15-February-2020].

[2] W. Wang, Q. Huang, S. You, C. Yang, U. Neumann, Shape inpainting using 3d generative adversarial network and recurrent convolutional networks, 2017 IEEE International Conference on Computer Vision (ICCV) (2017) 2317–2325.

[3] Z. Chen, H. Zhang, Learning implicit fields for generative shape modeling, CoRR abs/1812.02822 (2018). arXiv:1812.02822.
URL http://arxiv.org/abs/1812.02822

[4] J. J. Park, P. Florence, J. Straub, R. A. Newcombe, S. Lovegrove, Deepsdf: Learning continuous signed distance functions for shape representation, CoRR abs/1901.05103 (2019). arXiv:1901.05103.
URL http://arxiv.org/abs/1901.05103

[5] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, CoRR abs/1511.06434 (2015).

[6] D. P. Kingma, M. Welling, Auto-encoding variational bayes, CoRR abs/1312.6114 (2013).

[7] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, O. Winther, Autoencoding beyond pixels using a learned similarity metric, in: ICML, 2015.

[8] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, International Journal of Robotics Research (IJRR) (2013).

[9] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[10] P. Achlioptas, O. Diamanti, I. Mitliagkas, L. Guibas, Learning representations and generative models for 3d point clouds, arXiv preprint arXiv:1707.02392 (2017).

[11] Y. Zhao, T. Birdal, H. Deng, F. Tombari, 3d point capsule networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1009–1018.

[12] W. Yuan, T. Khot, D. Held, C. Mertz, M. Hebert, Pcn: Point completion network, in: 2018 International Conference on 3D Vision (3DV), 2018, pp. 728–737.

[13] Z. Huang, Y. Yu, J. Xu, F. Ni, X. Le, Pf-net: Point fractal network for 3d point cloud completion (2020). arXiv:2003.00410.

[14] C. R. Qi, L. Yi, H. Su, L. J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, CoRR abs/1706.02413 (2017). arXiv:1706.02413.
URL http://arxiv.org/abs/1706.02413

[15] 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, F. Yu, Shapenet: An information-rich 3d model repository, CoRR abs/1512.03012 (2015).

[16] M.-F. Chang, J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, J. Hays, Argoverse: 3d tracking and forecasting with rich maps, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2019.