Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/13/20230731

PG-Net:3D point cloud completion based on graph convolutional network

Tiansheng Wang

East China Normal University, 3663 Zhongshan North Road, Shanghai, 200062, China

tiansheng-wang@foxmail.com

Abstract. With the advancement of autonomous driving technology, the problem of 3D point cloud completion has become increasingly important. Completing 3D point clouds can improve the accuracy of 3D object detection, which is crucial for the development of autonomous driving and other related fields. In this paper, we propose a new approach for 3D point cloud completion tasks using point cloud representation. We focuses on the point cloud completion problem using Graph Neural Network methods, which are known for their ability to capture topological features. Our approach utilizes key components extraction and learning from the point cloud, to constrain the output of the decoder, and thus enhance the performance of point cloud completion task. Our approach is able to overcome the limitations of traditional methods, such as memory consumption and computational burden, as well as the loss of detailed information caused by quantization operation in some sparse representation based methods. We conduct extensive experiments on several benchmark datasets to evaluate the performance of our approach and compare it to existing methods. Our experimental results demonstrate that our proposed method is competitive, achieving comparable or even better results compared to state-of-the-art models. In particular, we show that our method is able to improve upon the performance of earlier models and achieve results that are comparable to current state-of-the-art models. These results indicate that our approach is a promising solution for 3D point cloud completion tasks.

Keywords: Point Cloud Completion, Graph Convolutional Network, Part Learning, Feature Fusion.

1. Introduction

Real-world 3D data is often incomplete, resulting in loss of geometric and semantic information. For example, in LiDAR scans of cars, the data may be too sparse to identify the car accurately, due to limited sensor resolution and occlusion. The representation of 3D objects is a critical aspect in these tasks. Compared with current state-of-the-art models [1], there are many traditional models in early times. Traditional methods have used voxel-based representations, such as 3D-EPN [2], which are computationally expensive and have large memory requirements. On the other hand, direct processing of unstructured point cloud data is more memory-efficient and allows for more granular representation of the object. However, the usual convolution operators are not suitable for point clouds. PointNet and its variants were the first to handle unordered point cloud data directly, opening the path for further research in the area. Based on this breakthrough, PCN was the first learning-based architecture for point cloud completion, utilizing an encoder-decoder framework and FoldNet, which maps 2D points to 3D surfaces simulating 2D plane deformation. Subsequently, many studies have followed PCN, improving resolution and robustness in 3D completion tasks by using this architecture.

Point cloud completion is a challenging problem because the structural information required for the completion task is at odds with the unordered and unstructured nature of point cloud data. Thus, learning the structural features of point clouds is crucial for addressing this task. The current state-of-the-art solutions to this problem is to leverage the Transformer.

In this paper, we introduce a novel approach for completing partial 3D point clouds using a deep learning framework. Our approach utilizes an encoder-decoder network with the added capability of extracting topological features, and incorporating a part-based learning approach. Our proposed method is able to effectively complete the task of point cloud completion.

The proposed approach is inspired by recent advancements in the field and it aims to provide an effective solution for dealing with incomplete point clouds. [3-4].

Firstly, feature extraction on point clouds is conducted, in previous methods, Multi-layer Perceptron (MLP) is often used to extract features directly from point clouds, but it neglects the topological information contained in the point clouds. To address this issue, we propose a module that can perceive geometric information from point clouds and complete partial points. Graph Convolutional Networks (GCN) have natural topological characteristics that make them well-suited for 3D object recognition, particularly in dealing with point cloud data. However, point cloud data obtained from radar has a specific three-dimensional data structure and LiDAR scans only capture points on the surface of an object, resulting in relatively low-resolution point clouds. To overcome this, in our approach, each point in the point cloud is represented as a node and the relationships between nodes are constructed as edges, forming a large graph. The connections between graph nodes allow the model to learn global geometric features of the point cloud, enabling it to predict and complete the "partial" parts of the point cloud and improve its resolution.

The main contributions are as follows:

- We propose a new architecture that combines the use of Multi-Layer Perceptrons (MLP) and Graph Convolutional Networks (GCN) to extract and fuse point cloud features.
- We design a learnable weight module for edge features to increase the expressiveness of topological relations.
- We introduce a key component module that extracts important parts of the point cloud to optimize the output of the decoder.

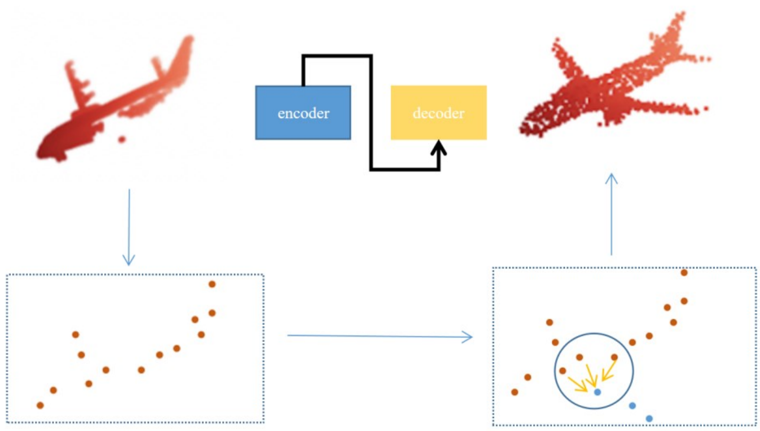

Figure 1. The illustration depicts the main ideas proposed in our paper.

DOI: 10.54254/2755-2721/13/20230731

2. Related Work

2.1. 3D Point Cloud Completion

In recent years, there has been an increasing interest in using unstructured point clouds as the representation of 3D objects for 3D shape completion tasks. Compared to traditional methods, which often adopt voxel grids or distance fields, point clouds have the advantage of low memory consumption and ability to preserve fine-grained details. However, the transition from structured 3D data understanding to point clouds analysis is non-trivial, as traditional convolution operators are not directly applicable to unordered point clouds.

The work of PointNet and its variants have been instrumental in advancing the field [5-7], as they introduced the ability to directly process 3D coordinates and inspired numerous downstream tasks. In the realm of point cloud completion, PCN was one of the first learning-based architectures to be proposed [8], introducing an Encoder-Decoder framework and a FoldingNet for mapping 2D points onto a 3D surface. Following PCN, there have been numerous other methods proposed [9-12], all with the goal of achieving point cloud completion at higher resolutions and with greater robustness.

Despite the advances made by these methods, there are still limitations to overcome. For example, memory consumption and computational burden remain a concern for methods that rely on voxel grids or distance fields. Additionally, quantization operation used in some sparse representation based methods also cause a significant loss in detailed information. In this paper, we propose a new approach to overcome these limitations and achieve state-of-the-art performance on 3D shape completion tasks using point clouds representation.

2.2. GCN

Graph Convolutional Networks (GCN) were first introduced by Thomas as a powerful tool for semi-supervised learning on graph-structured data [13]. It has been adapted to work on point clouds, by representing each point as a node in the graph, and creating edges based on the neighboring points. One of the unique characteristics of Graph Convolutional Networks (GCN) is that they are able to handle data that is represented in non-Euclidean space and is represented by graphs. The research on graph neural networks aims to generalize convolutional neural networks to work with graph representations. GCN updates vertex features based on the aggregation of edge features, typically determined through the use of a K-nearest neighbors approach to identify candidate neighboring points, and the features of the edges between each pair of points are used to update the vertex features. In the field of computer vision, some methods represent point clouds as graphs. The EdgeConv operation in DGCNN creates a graph in the feature space [14], which is updated after each layer of graph convolution operations. These methods typically operate convolutions on spatial neighbors, and use pooling to create a new, coarser graph by aggregating features from neighboring points. However, these methods are limited by their use of spatial constraints, which restrict their ability to extract key point features from the neighborhood.

To address this issue, SparseNet introduced the channel-attentive EdgeConv, which allows for the extraction of both local and global shape features from point clouds [15]. This increased representation of partial shapes allows the network to generate finer structures in the completed point clouds. However, simply using channel-attentive EdgeConv is not sufficient to extract global context features. In our work, we propose a novel adaptive neighborhood feature extraction module that dynamically selects points in the neighborhood based on the shape of the object. This allows us to extract more comprehensive features, further improve the performance and robustness of our approach.

3. Point Completion With GCN

In this section, we describe our proposed model. As shown in Figure 2, G-conv is a graph feature extractor based on the improved DGCNN [14]. It aims to capture the topological information of the point clouds while retaining the spatial information. The G-conv module performs graph convolution operation on the point clouds and updates the feature of each point by aggregating the features of its neighboring points. By using G-conv, the model is able to learn more robust and discriminative features of the point clouds, which is essential for the task of point cloud completion.

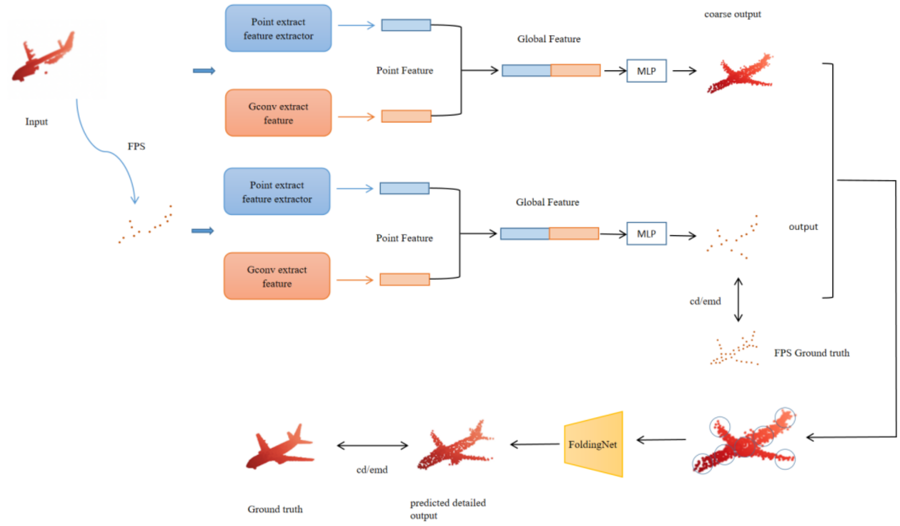

Figure 2. Shows the network structure of point cloud completion proposed in this paper.

Given an input point cloud of n*3, where n represents the number of points and 3 represents the (x, y, z) coordinates of each point. First, the variable structure of the point cloud is made robust by replacing the invariant structure. Then, the T-Net, through G-conv(64), transforms the n3 point cloud into n64. Next, G-conv(128) is used to expand the dimension of each point to 128 dimensions, and finally, the first three dimensions and the last n128 are concatenated to obtain a 2048-dimensional vector after convolution pooling. This vector is then passed through a layer of convolution pooling operation to reduce its dimension from 2048 to 1024. Finally, the 1024-dimensional feature is expanded to the same size of the ground-truth point cloud through the folding operation.

3.1. Extractor with GCN

In the task of point cloud completion, many classical studies have been based on point features [16-21], using PointNet to extract point features, and finally completing regression or classification tasks. In this paper, we propose a new method for topological information fusion that combines multiple sources of information to improve the accuracy and robustness of point cloud completion tasks. We use graph neural networks to extract features, the selection of nodes and the connections between nodes are crucial, so we designed a learnable edge feature weight module .

Topological information fusion refers to the process of integrating and combining information from different sources to gain a more comprehensive understanding of the underlying structure or relationship of a given system or data. This technique can be applied in a variety of fields, such as computer vision, image processing, and machine learning, to improve the performance of various tasks and algorithms. As shown in Figure 3, we compute a directed graph to represent the structure after extracting point cloud features with GCN, representing the set of vertices and edge features. In dynamic graph design, we use KNN to extract local set features, and use each point to calculate the nearest K points around it to construct vertex sets and edge sets. Specifically, our method allows the nodes in a graph to have different levels of influence on the edges they are connected to, through the use of learnable weights. This can help the model to better capture the underlying structure of the graph and improve its performance on tasks such as node classification and graph classification.

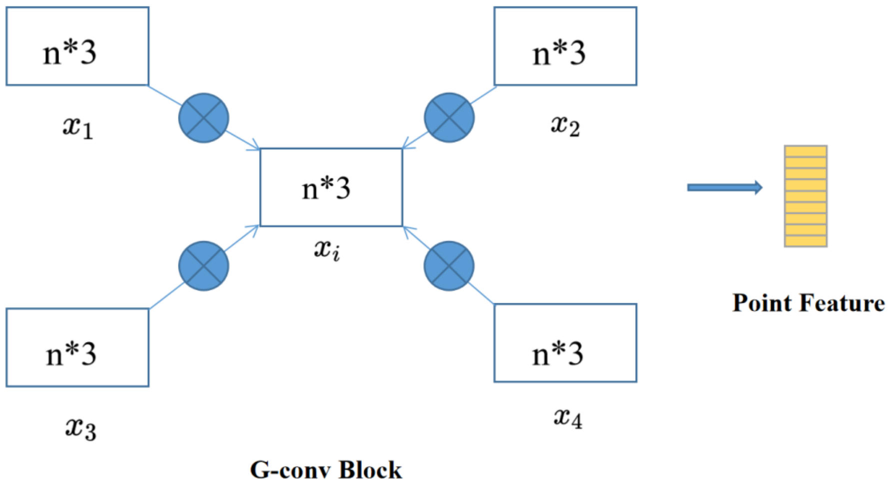

Figure 3. Utilizing a learnable module to extract edge features.

Assuming a point cloud containing n points, each point feature has F dimensions, we represent X = {x1, x2,...xn}, in the simplest mode, we take F as 3, that is, each point is (x, y, z) 3-dimensional coordinate information, on some specific data sets it is May also contain additional features such as rotation angles. In a deep neural network, each G-conv module will take the output as the input of the next layer.

3.2. Part Based Learning

In comparison to the complex input point clouds, the supervision of part keypoints is stronger in terms of topological information and has a crucial impact on the overall shape completion. We first extract features directly from the input partial point clouds, which we refer to as the L1 layer. Meanwhile, we obtain the part keypoints {k1, k2, ..., kn} through furthest point sample (FPS) sampling, and estimate the overall part keypoints through MLP, which we refer to as the L2 layer. By merging the L1 and L2 layers, we constrain the output in the decoder part for the point cloud completion task.

Downsample the point set using FPS (farthest point sampling), reducing the input point set from size N1 to a smaller size N2. FPS can be understood as making the sampling points as far apart as possible. Our method leverages a part-based learning approach, by obtaining key points of parts through the use of the FPS method. In terms of category classification, using a well-learned part keypoints can achieve high accuracy. By applying this concept to constrain the output of the decoder in the point cloud completion task, the output of the decoder can be made richer while also avoiding over-fitting under the constraint of parts.

The network for part-based learning and the feature extraction network for the input point cloud are similar, both combining MLP feature extraction and features extracted using GCN. Additionally, because the number of part keypoints is much smaller compared to the input point cloud, it is more conducive to expressing topological information and stronger semantics. G-conv maintains the local geometry by building a local neighbor graph, and then applies convolution-like ops on the edges connecting nodes to their neighbors. The neighbors of fixed nodes in each layer of G-conv change, so the graph structure of each layer is different, which also makes the algorithm have the property of non-local diffusion. The data structure of point cloud is discrete and lacks topological information, that is, the association between individual points and points is not explicitly established, but there should be practical significance between them. Naturally, if we can somehow establish topological relationships between points, it should enhance the power of representation.

G-conv dynamically constructs a graph structure on each layer of the network, uses each point as a center point to represent its edge features with each neighboring point, and then aggregates these features to obtain a new representation of the point. The actual implementation of G-conv is to construct a local neighborhood, which can be established in either the coordinate space or the feature space to characterize each point. From the above introduction, it is not difficult to imagine that the continuous stacking of multiple G-conv modules end to end is to obtain multi-level representations with richer semantics, that is, the input and output of G-conv.

3.3. Loss Function

In this paper, we propose an approach for optimizing the loss function in our point cloud completion model. Our method involves combining two different types of losses: one that is generated by the component learning module and another one that is generated by the final decoder. Specifically, we optimize the overall loss by summing up the component-wise loss and point cloud loss produced by the decoder. This enables our model to learn more detailed structural knowledge and improve the quality of the completed point clouds. We evaluate the performance of our approach on several benchmark datasets and demonstrate its effectiveness in achieving comparable results.

$$CD(S_1,S_2) = \frac{1}{|S_1|}\sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2 + \frac{1}{|S_2|}\sum_{y\in S_2}\min_{x\in S_1}\|y-x\|_2 \qquad (3-1)$$

CD distance: As shown in formula 3-1, the CD distance measures the closest distance between the average point of the output point cloud S1 and the real point cloud S2, where S1 and S2 must be the same size.

$$EMD(S_1,S_2) = \min_{\phi:S_1\to S_2}\frac{1}{|S_1|}\sum_{x\in S_1}\|x-\phi(x)\|_2 \qquad (3-2)$$

EMD distance: The EMD distance originally measured the distance between two distributions. Here, the input point cloud and the output point cloud are regarded as two distributions. As shown in Equation 3-2, S1 and S2 represent the complement point cloud and the real point cloud of the model prediction output, respectively.

In summary, the significance of using distance metrics in point cloud processing lies in its ability to effectively measure the dissimilarity between point clouds, making it suitable for various generative tasks such as upsampling, completion, and reconstruction. Additionally, using distance metrics that are invariant to point arrangement allows for robustness in the face of varying point cloud structures. This enables the implementation of shape and geometric constraints through the use of appropriate loss functions and backpropagation, leading to improved performance in deep learning tasks.

4. Experiments

This section mainly conducts experimental analysis of point cloud completion based on graph convolutional neural network, and compares it with mainstream completion algorithms. Then, we show the results of both our method and several baseline methods on ShpaeNet dataset.

4.1. Dataset

ShapeNet is a widely used dataset for point clouds. It includes two subsets, ShapeNetCore and ShapeNetSem, where ShapeNetCore has been updated with two versions, ShapeNetCore v1 and ShapeNetCore v2, covering 55 common object categories. ShapeNetCore and ShapeNetSem are commonly used in various research studies as benchmark datasets for evaluating the performance of point cloud-related tasks.

4.2. Model Training

This summary introduces the relevant details of experimental training and testing. In the model training phase, we use the classic Adam optimizer. Because the data structure of the graph is relatively large, the batch size in this paper is set to 8, the initial learning rate is set to 0.001, and each 40 epochs down 10%, on the GPU side, we use RTX 3090 24G model graphics card. The number after encoding the sparse point cloud is set to 1024, and the output number of the final dense point cloud is set to 16384.

We conducted experiments on the hyperparameters of KNN, and conducted control experiments on 5, 10, 15, and 20, respectively, to analyze the influence of the number of neighbor points on the extracted information when extracting neighbor points. The effect of using mlp for global information extraction and using G-conv module for local information extraction is compared with the effect of using G-conv alone to extract both global information and local information end-to-end.

4.3. Results on ShapeNet

Table 1. Comparative Results with KNN.

K	CD	EMD
K=5	0.01967	0.06987
K=10	**0.01681**	**0.06293**
K=15	0.01763	0.06783
K=20	0.01713	0.06881

We can see that after the selection of K nearest neighbors, when 10 neighbor points are taken, the model performs the best, the CD distance is 0.01681, and the EMD distance is 0.06293. When the selection of K is reduced, the CD distance increases significantly, increasing the 17%, when increasing the choice of K, the CD distance and EMD distance also increased slightly, increasing by 6% and 9%, respectively.

Table 2. Model parameter comparison.

method	3D-EPN	FC	Folding	PCN	PoinTr	Ours
Params	52.4M	53.2M	2.4M	6.85M	32.1M	7.59M

As can be seen from Table 2, our model size has obvious advantages, which makes it possible to deploy the model on the mobile terminal.

Table 3. CD and EMD result.

	PCN	FoldingNet	Ours
CD	0.0173	0.0232	**0.0091**
EMD	1.366	0.537	**0.531**

As can be observed from Table 5-3, the G-conv architecture proposed in this chapter demonstrates superior performance in terms of CD distance, with a value of 0.0091, and also achieves an EMD distribution score of 0.531. These results demonstrate the effectiveness of our proposed architecture in accurately completing point clouds.

Table 4. Results on ShapeNet.

	Table	Chair	Sofa	Bird house	Bag	CD-Avg
FoldingNet	2.53	2.81	2.48	4.71	2.79	3.12
PCN	2.13	2.29	2.06	4.50	2.86	2.66
PoinTr	0.93	**0.81**	**0.43**	1.86	**0.68**	**0.94**
Ours	**0.81**	0.95	0.79	**1.85**	0.93	1.09

As can be observed from Table 4, our proposed method outperforms the majority of conventional approaches. While the performance of our method may not be on par with the state-of-the-art (SOTA), the difference is relatively small, and our method has a significant advantage in terms of network size. These results demonstrate the effectiveness and efficiency of our proposed approach in point cloud completion.

Table 5. Ablation study.

Model	MLP	G-conv	Part based	CD	F-Score@1%
A				10.13	0.6612
B	√			10.09	0.7123
C	√	√		9.36	0.7232
D	√	√	√	**8.89**	**0.7318**

The ablation study presented in this chapter compares the performance of individual modules of our proposed method. As shown in the table 5, utilizing a multi-layer perceptron (MLP) for feature extraction, incorporating G-conv to capture local topological information, and constraining the decoder with a part-based approach yielded the best results. This analysis demonstrates the importance of each component in achieving high-quality point cloud completion.

We select some random samples from ShapeNet and visualize the results of the model in Figure 4.

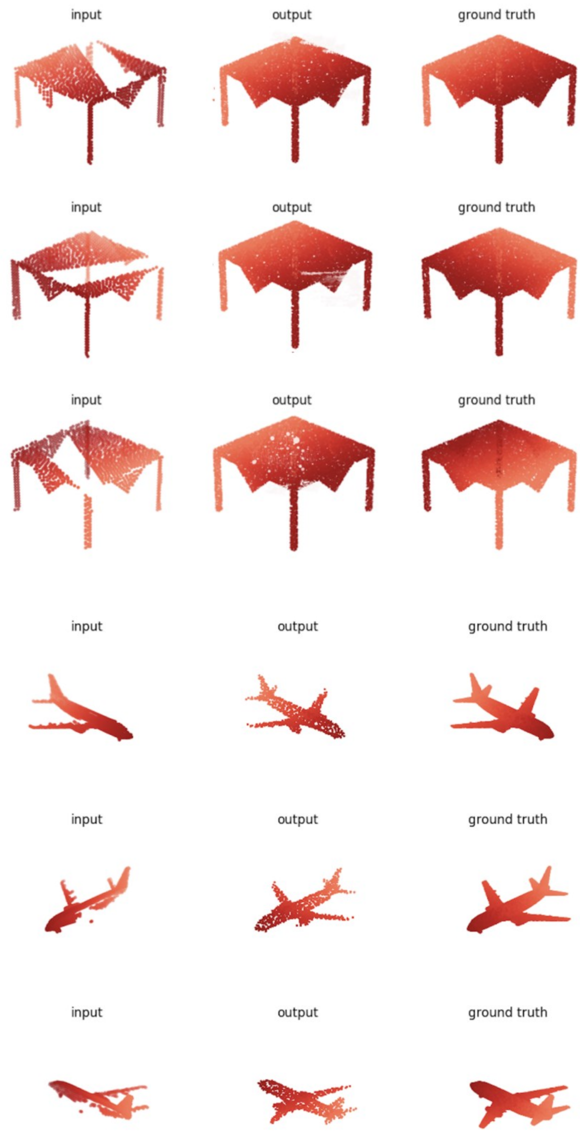

Figure 4. Point cloud completion results on ShapeNet. We display the input point cloud,the ground truth, and our model's predictions.

5. Conclusion

In this paper, we have proposed a novel lightweight architecture for point cloud completion tasks, which integrates both MLP and GCN methods to extract and fuse point cloud features. Our approach design a learnable weight block on the edge feature of graph to better capture underlying structure. Additionally, we focus on extracting key components on the input point clouds and incorporating them into the decoding process, which lead to comparable results on several benchmark datasets.

References

[1] Yu X, Rao Y, Wang Z, et al. Pointr: Diverse point cloud completion with geometry-aware transformers[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 12498-12507.
[2] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In Proc.IEEE Conf. on Computer Vision and Pattern Recognition(CVPR), volume 3, 2017. 1, 2, 5
[3] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), volume 3, 2017. 1, 2, 5
[4] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 190–198. IEEE,2017.1
[5] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.
[6] Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[J]. Advances in neural information processing systems, 2017, 30.
[7] Qi C R, Liu W, Wu C, et al. Frustum pointnets for 3d object detection from rgb-d data[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 918-927.
[8] Yuan W, Khot T, Held D, et al. Pcn: Point completion network[C]//2018 International Conference on 3D Vision (3DV). IEEE, 2018: 728-737.
[9] Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi,Ian D. Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In CVPR, pages 383–392, 2019. 2, 5, 6, 7, 8
[10] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le.Pf-net: Point fractal network for 3d point cloud completion.In CVPR, pages 7659–7667, 2020. 2, 4, 5, 6, 8
[11] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian.Foldingnet: Interpretable unsupervised learning on 3d point clouds. CoRR, abs/1712.07262, 2017. 4, 6, 7, 8
[12] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and ShiMin Hu. Morphing and sampling network for dense point cloud completion. In AAAI, pages 11596–11603, 2020. 2
[13] Kip, F.T.N., Welling, M.: Semi-Supervised Classification With Graph Convolutional Networks. arXiv preprint arXiv: 1609.02907 (2016)
[14] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma,Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. TOG, 38(5):1–12,2019. 3, 4
[15] Xie, C., et al.: Style-Based Point Generator With Adversarial Rendering for Point Cloud Completion. arXiv preprint arXiv: 2103.02535. (2021)
[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, NAACL, pages 4171–4186, 2019. 3
[17] Christoph L¨uscher, Eugen Beck, Kazuki Irie, Markus Kitza,Wilfried Michel, Albert Zeyer, Ralf Schl¨uter, and HermannNey. RWTH ASR systems for librispeech: Hybrid vs attention - w/o data augmentation. CoRR, abs/1905.03072, 2019.3

[18] Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Tatiana Likhomanenko, Edouard Grave, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. End-to-end ASR: from Supervised to Semi-Supervised Learning with Modern Architectures. arXiv e-prints, 2019. 3
[19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI blog, page 9, 2019. 3
[20] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, and Alexander Ku. Image transformer. CoRR, abs/1802.05751, 2018. 3
[21] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu,Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. arXiv preprint arXiv:2106.02034, 2021. 3