Analysis of 3d point cloud segmentation

ABSTRACT

Three-dimensional (3D) point cloud segmentation has emerged as a fundamental task in computer vision, enabling scene understanding for applications such as autonomous navigation, robotics, and augmented reality. Unlike structured image data, point clouds are unordered and irregular, making direct processing challenging. To address this, several deep learning architectures have been proposed to directly learn from raw point sets. In this paper, we present a comprehensive comparative analysis of four prominent neural architectures, PointNet, PointNet++, DGCNN, and Point Transformer for 3D point cloud segmentation. Each model embodies a distinct design philosophy, from global feature extraction to hierarchical local learning, graph-based feature aggregation, and transformer-driven self-attention mechanisms. All models were implemented under a unified experimental setting and evaluated ShapeNetpart [1] dataset. Our results demonstrate that transformer-based methods achieve superior segmentation accuracy and robustness compared to earlier point-based and graph-based approaches, albeit at a higher computational cost. The study highlights the progressive evolution of deep architectures for point cloud understanding and provides insights into the trade-offs between performance, complexity, and generalization in 3D vision.

Index Terms— Point Cloud Segmentation, Deep Learning, PointNet, DGCNN, Point Transformer, 3D Vision

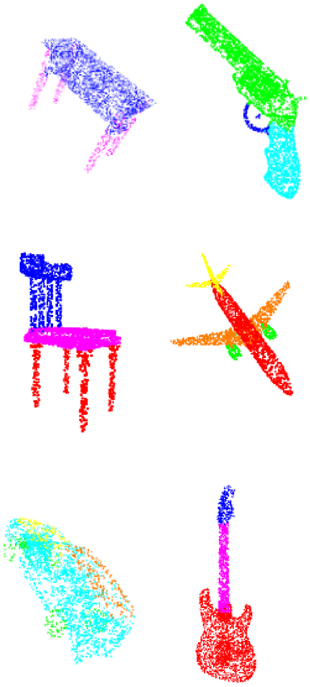

Fig. 1. 3D point cloud segmentation of some objects.

1. INTRODUCTION

Three-dimensional (3D) point clouds have become a fundamental data representation in computer vision, enabling machines to perceive and understand spatial geometry in real-world environments. With the increasing availability of 3D sensors such as LiDAR, depth cameras, and structured light scanners, point cloud data are now widely used in robotics, autonomous driving, augmented reality, and digital twin applications. Point cloud segmentation, which assigns semantic or instance-level labels to each point, is a crucial step toward comprehensive 3D scene understanding.

Unlike 2D images defined on regular grids, point clouds are irregular, unordered, and lack explicit topological connections, making it difficult to apply conventional convolutional neural networks (CNNs) directly. Early approaches relied on voxelization or multi-view projections, which introduce quantization artifacts and high computational cost. Thus, designing deep networks that can learn directly from raw point sets while preserving permutation invariance and geometric relationships remains an open challenge.

In this work, we present a unified comparative analysis of these four representative architectures — PointNet, PointNet++, DGCNN, and Point Transformer — under consistent experimental conditions. Each network is implemented and evaluated on the same dataset to investigate its segmentation performance, computational efficiency, and generalization capability. Our goal is to analyze the progression of architectural design in 3D point cloud learning and to highlight the trade-offs between accuracy, complexity, and robustness.

The remainder of this paper is organized as follows. Section 2 describe problem statement. Section III describes methods. Section IV presents the experimental results and performance analysis. Finally, Section V concludes the paper and discusses potential future research directions.

2. PROBLEM STATEMENT

Suppose that X = (M, d) is a discrete metric space whose metric is inherited from a Euclidean space Rn, where

$$M \subseteq \mathbb{R}^n$$

is the set of points and d is the distance metric. In addition, the density of M in the ambient Euclidean space may not be uniform everywhere. We have to design learning models that take such X(point order invariant) as the input (along with additional features for each point like its x,y,z coordinates) and produce information of semantic interest regrading X . For simplicity and clarity, unless otherwise noted, we only use the (x, y, z) coordinate as our point's channels. For semantic segmentation, the input can be a single object for part region segmentation, or a sub-volume from a 3D scene for object region segmentation. Our model will output n × m scores for each of the n points and each of the m semantic sub- categories.

3. METHODS

Point cloud semantic segmentation, which is a fundamen- tal task in 3D indoor scene understanding, aims to parti- tion a scene into multiple subsets. Based on the semantic meanings of the individual points, our objective is to assign each point in the scene to a specific category label.There are several supervised learning methods we implemented for point cloud segmentation are described here.

PointNet. Qi et al.[2] introduced the PointNet network architecture. This network comprises three key compo- nents: the multi-layer perceptron (MLP) module, the max pooling structure, and the feature fusion structure. The MLP module enables the extraction of point cloud fea tures through weight sharing. The max pooling structure,which employs a symmetric function, selects the maxi mum feature value within a group of points and serves as the global feature representation. This design addresses the problem of irregularity in the data. The feature fu- sion structure combines the local features and the global features obtained from the maximum pooling operation. These merged features are utilized as input, and the MLP predicts labels for each point. Moreover, PointNet incor- porates the T-Net structure, which facilitates the learning of an efficient rotation matrix. PointNet has demonstrated effectiveness in semantic segmentatation,making it a fundamental network ar- chitecture in this area.

Pointnet++. PointNet++ [3] introduces a set of abstraction structures consisting of sampling layers, grouping layers, and PointNet layers. This hierarchical design enables the extraction of multi-scale features from point clouds. By stacking multiple layers of this feature extraction structure, PointNet++ can be applied for tasks such as point cloud classification and segmentation.

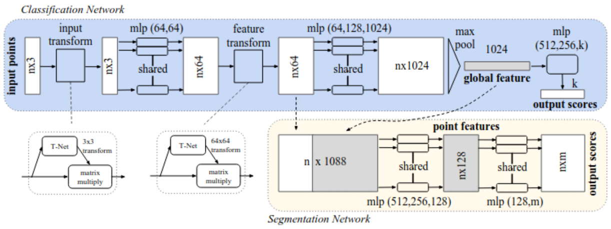

Fig. 2. *PointNet Architecture.* The network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.

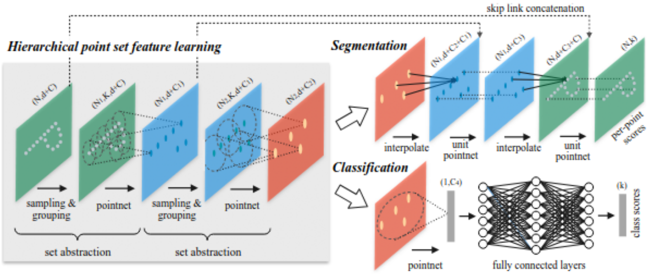

Fig. 3. llustration of our hierarchical feature learning architecture and its application for set segmentation and classification using points in 2D Euclidean space as an example. Single scale point grouping is visualized here.

DGCNN(Dynamic Graph CNN.DGCNN. Wang et al.[4] proposed the Dynamic Graph Convolutional Neural Network (DGCNN), which introduces a novel EdgeConv operation to capture local geometric relationships among points. Unlike PointNet and PointNet++, which process points independently or within fixed neighborhoods, DGCNN dynamically constructs a graph based on k-nearest neighbors in the feature space at each layer. The EdgeConv module aggregates edge features derived from point–neighbor pairs, enabling the network to learn both local and global contextual information. This dynamic updating of the graph during training allows DGCNN to adaptively refine neighborhood relations as feature representations evolve. Owing to its ability to effectively model local topology and spatial dependencies, DGCNN achieves superior performance in tasks such as point cloud classification and part segmentation compared to earlier point-based architectures.See figure 4

Attention based model Pointnet Transformer. While

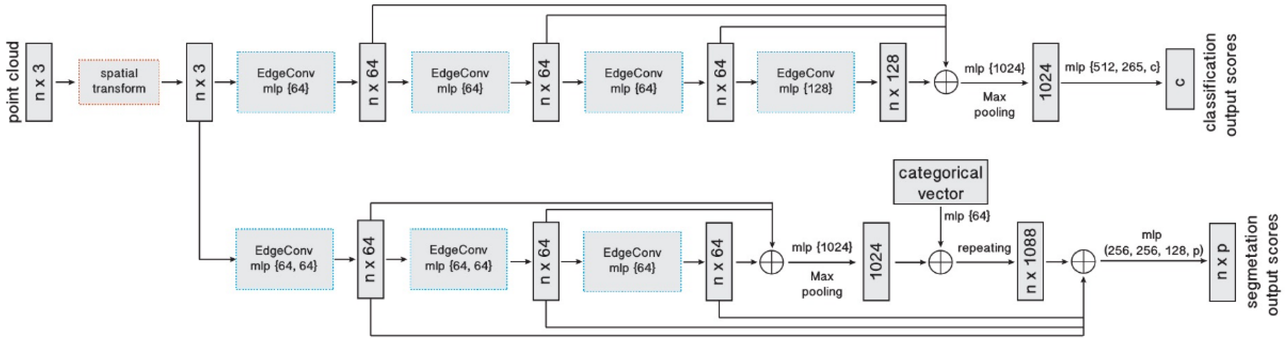

Fig. 4. Architecture of DGCNN.

PointNet effectively learns global features directly from unordered point sets, it lacks the ability to capture local geometric relationships among neighboring points. PointNet++ addresses this limitation by introducing a hierarchical feature extraction framework through sampling and grouping layers; however, it still suffers from sensitivity to non-uniform point densities and can incur significant computational cost when handling large-scale point clouds. DGCNN further improves local feature learning by employing a dynamic graph structure and EdgeConv operation, allowing adaptive neighborhood updates in the feature space. Nevertheless, DGCNN's reliance on k-nearest neighbor graph construction leads to high memory usage and limited global context modeling, restricting its efficiency in complex 3D scenes. To overcome these limitations, Point Transformer. Zhao et al. [5] introduced the Point Transformer network, which integrates the self-attention mechanism into point cloud learning. Unlike convolution- or graph-based approaches, Point Transformer models contextual relationships among points by adaptively weighing neighboring features through attention layers equipped with learnable positional encodings. Its hierarchical encoder–decoder structure employs Transition Down layers for point sampling and feature aggregation, enabling progressive abstraction of geometric details, and Transition Up layers for feature propagation and resolution recovery. This design allows the network to effectively fuse local geometric cues with global contextual information. By capturing long-range dependencies and spatial semantics, Point Transformer achieves superior performance in both 3D point cloud segmentation tasks.

4. EXPERIMENTS AND RESULTS

For 3d point cloud segmentation, we are provided with benchmark dataset named Shapenetpart. ShapeNetPart is a 3D point cloud dataset derived from the ShapeNet repository. It's widely used for part segmentation tasks that is, identifying which part of an object each point belongs to (like chair legs, seat, and back).Each object in dataset has its category label(e.g. table,aeroplane) and have 2048 cloud points to represent their object with their coodinates (x,y,z) and each point among 2048 cloud points has a segmentation label(e.g.table legs,table seat,aeroplane body).

Property	Description
Total Shapes	16,881 models
Categories	16 (e.g., airplane, chair, car, lamp, etc.)
Part Labels	50 total part types
Input Format	3D point clouds (2,048points per shape)
Task	Part segmentation
Source	Subset of ShapeNetCore

Table 1. Summary of the ShapeNetPart Dataset

we started our experiments with a baseline network.To establish a baseline, we first implemented PointNet without the input and feature transformation networks. Although the simplified model produced acceptable performance (overall accuracy : 0.8533), it lacked the ability to learn transformation invariance. The input transformation network normally aligns the raw point cloud into a canonical coordinate space, while the feature transformation network refines learned feature representations by correcting local misalignments in higher-dimensional space. Excluding these components made the network more sensitive to object orientation and geometric variations, which resulted in less stable segmentation boundaries and marginally lower accuracy. We,then implemented T-NET architecture and introduce random rotation,scaling and zittering and our overall accuracy increased to 0.8893 showing importance of these transformations and preprocessing methods.

Pointnet has a drawback that it treats all points independently. It doesn't care about local interactions among points.So,PointNet doesn't naturally capture local structures or patterns that may be important in the data.Pointnet++ try to overcome this issue by proposing sampling and grouping layers in their network.In sampling layer , it selects a subset of points from input points,which defines the centroids of local regions. In grouping layer it uses FPS(Farthest Point sampling) to select k number of points around centroid within

Table 2. Comparisons of methods on Shapenetpart dataset

Method	**Overall Acc.**	**Inst. mIoU**	**Class mIoU**	**Inf. Time (s)**	**Memory (MB)**	**Model (MB)**
PointNet	0.8893	0.7678	0.6555	3.35	2028.27	4.37
PointNet++	0.9019	0.7824	0.6714	9.39	2840.62	4.67
DGCNN	0.9098	0.7899	0.6822	21.52	2201.44	5.34
Point Transformer	0.9216	0.8023	0.6910	41.52	2151.44	11.70

a giving radius.From here onwards,We use U-Net [6] style local and global features concatenation on baseline pointnet to produce segmentation labels.First We implemented pointnet++ architecture without this U-Net style concatenation of features, this gives worse performance(overall accuracy:0.8767)than Pointnet although we performed sampling and grouping. After that when we implemented U-Net style concatenation , We got stronger result than pointnet(overall accuracy:0.9019).

Although PointNet++ improved upon PointNet by introducing hierarchical feature learning on local regions, it still relied on static neighborhood structures defined in the input space. This limited its ability to capture dynamic relationships between points as features evolved during training. DGCNN overcomes these limitations by constructing dynamic graphs in the feature space and applying EdgeConv operations to explicitly model interactions among neighboring points. This dynamic neighborhood update allows the network to learn richer local geometric dependencies and yields more discriminative features, resulting in improved segmentation and classification performance compared to PointNet++. On implemented this dgcnn architecture , it improved overall accuracy to 0.9098.

After evaluating DGCNN, we further extended our experiments to the Point Transformer architecture to incorporate attention-based feature learning. While PointNet and PointNet++ rely on local aggregation and struggle to model long-range dependencies, and DGCNN captures only limited relational context through edge convolutions, Point Transformer introduces a self-attention mechanism that dynamically learns the relevance between all points in the cloud. This allows the network to integrate both local and global geometric information adaptively, rather than relying on fixed or feature-space neighborhoods. Consequently, it mitigates issues such as loss of global context, sensitivity to point density, and limited feature interaction seen in earlier models. Owing to these advantages, Point Transformer achieves state-of-the-art performance and serves as a new benchmark on the ShapeNetPart dataset.When we implemented this on our dataset, it gave best overall accuracy 0.9216.

5. CONCLUSION AND FUTURE SCOPE

In this work, we conducted a comprehensive comparison of various point cloud segmentation architectures, including PointNet, PointNet++, DGCNN, and Point Transformer, on the ShapeNetPart dataset. Among all, the Point Transformer achieved the highest overall accuracy and segmentation quality, demonstrating its superior ability to model both local and global geometric dependencies through dynamic self-attention mechanisms.

In future work, we plan to extend this study by exploring unsupervised [7] and self-supervised learning approaches for point cloud segmentation. This will enable the model to learn more generalized and robust representations without relying heavily on labeled data, potentially improving scalability and performance on real-world datasets.

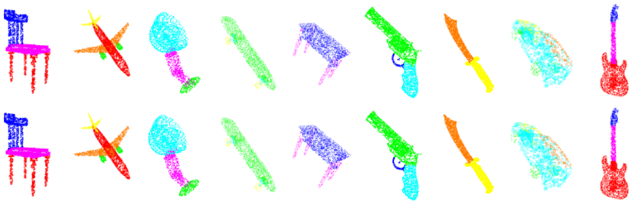

Fig. 5. True segmentation above and model segmentation below

.

6. REFERENCES

[1] Kaggle. (2016) Shapenetpart dataset. Accessed: 2025-10-13. [Online]. Available: https://www.kaggle.com/datasets/lokisilvres/shapenetpart

[2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.

[3] 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 (NeurIPS)*, 2017, pp. 5105–5114.

[4] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, pp. 1–12, 2019.

[5] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 16 259–16 268.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds. Springer International Publishing, 2015, pp. 234–241.

[7] "Dcpoint-unsupervised method," Point Paper for the Sponsoring Office/Command. [Online]. Available: https://arxiv.org/abs/2304.08965