International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Attention Mechanisms in PointNet++ for Effective Object Classification in 3D Point Clouds

Perfilev Dmitrii

School of Computer and Communication, Lanzhou University of Technology, Qilihe District, Lanzhou, 730050, Gansu, China

Abstract: ***In this research, an attention mechanism is integrated to enhance object classification in the processing of 3D point clouds. Point clouds obtained from LiDAR sensors are crucial for robotics and autonomous driving, as they provide detailed spatial data. However, large data volumes and noise often challenge traditional processing methods. To address this, the improved PointNet++ neural network is employed with an embedded attention mechanism, allowing it to focus on the most relevant portions of the input for object classification. PointNet++'s hierarchical structure, combined with attention layers, enables effective classification of complex scenes by prioritizing key features in the point cloud data. Tests on the KITTI dataset demonstrate that the attention-based approach boosts classification accuracy and reduces processing time. This method shows promise for building more reliable and efficient perception systems for self-driving vehicles and other 3D data analysis applications. By leveraging attention mechanisms within PointNet++, this study underscores their potential to enhance processing speed and accuracy, addressing critical challenges in the management of large-scale 3D point cloud data and supporting the development of faster, more accurate neural network-based systems for real-world applications.***

Keywords: point cloud, cluster, attention, pointnet, lidar

1. Introduction

With the rapid growth of 3D sensing technologies, such as LiDAR [1], [2] and RGB-D cameras, 3D point cloud data has become essential in various fields, including robotics, autonomous driving, augmented reality, and geospatial analysis. Point clouds provide rich spatial information that is crucial for understanding and interacting with complex environments. However, processing point cloud data presents unique challenges due to its unordered, irregular structure and high dimensionality, which make it difficult to apply traditional deep learning methods directly. To address these challenges, specialized neural network architectures have been developed to enable efficient and accurate processing of point clouds. Recent advancements, such as PointNet and its extension, PointNet++ [3], [4], [5], [6], have demonstrated the potential for deep learning models to extract meaningful features from 3D point clouds [7]. However, these models sometimes struggle to capture intricate spatial relationships within complex scenes. Attention mechanisms, widely adopted in natural language processing and image recognition, have emerged as a promising solution for focusing on the most relevant information in large datasets. By incorporating attention mechanisms into point cloud processing networks, researchers aim to improve feature extraction and enhance the accuracy and efficiency of 3D object classification and segmentation tasks [8].

2. Problem Statement

Despite the progress made in 3D point cloud processing, achieving reliable feature representation for complex, large-scale data remains challenging. Traditional methods often lack the capacity to capture subtle relationships between points or focus on key structural elements within the data. This limitation can reduce the performance of point cloud processing systems, especially in real-world scenarios where capturing spatial relationships is critical. The primary issue lies in the need for an approach that enables a neural network to selectively emphasize important points while preserving local and global geometric structures in 3D space. The attention mechanism offers a potential solution by allowing the network to assess the relative importance of each point in the context of the entire point cloud. This study seeks to integrate an attention mechanism within a neural network architecture to address these challenges, aiming to enhance feature learning and improve object classification accuracy within 3D point cloud.

3. Method

Our work introduces a novel hierarchical structure to PointNet++ and Point Cloud Transformer [9], [10], [11]. We begin with a review of K-means clustering and then proceed with an introduction to the basic hierarchical PCA (Point Cloud Attention). This approach enables the model to effectively learn features, even in non-uniformly sampled point sets, by leveraging attention mechanisms to focus on relevant points within the 3D data.

3.1 Related Work

We review past work on clustering, segmentation, and classification of point clouds.

3.1.1 Pointnet++

PointNet++, an advanced version of PointNet, employs hierarchical feature learning to better handle large-scale 3D point cloud data. Unlike PointNet, which processes points independently, PointNet++ captures local geometric structures through a hierarchical approach, allowing it to handle more complex 3D shapes. The architecture of PointNet++ consists of three main components: PointNet-based feature extraction, grouping, and sampling. This hierarchical design enables the network to capture both local and global features of the point cloud. By building on PointNet’s foundational concepts, PointNet++ enhances feature learning by extracting abstract global features at higher levels and detailed local features at lower levels. This hierarchical structure provides PointNet++ with a more comprehensive understanding of spatial relationships and geometric structures in 3D point clouds.

3.1.2 Point Cloud Transformer

Point Cloud Transformer (PCT) is a neural network architecture designed to process and analyze 3D point cloud data using self-attention techniques. This study offers a concise overview of PCT, focusing on its structure and its contributions to advancements in 3D point cloud processing. Based on the Transformer model, PCT is specifically adapted to handle the unordered and irregular characteristics of point cloud data. At the core of PCT is the self-attention mechanism, which enables the model to assess the significance of each point in relation to others. The attention score between two points, i and j, is calculated as follows:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where Q (query), K (key), and V (value) are linear projections of the input features, and d_k is the dimension of the key vectors.

3.2 Point Cloud Attention

Despite having a simpler architecture, Point Cloud Attention offers higher outcomes. The clustering module is not utilized in this model.

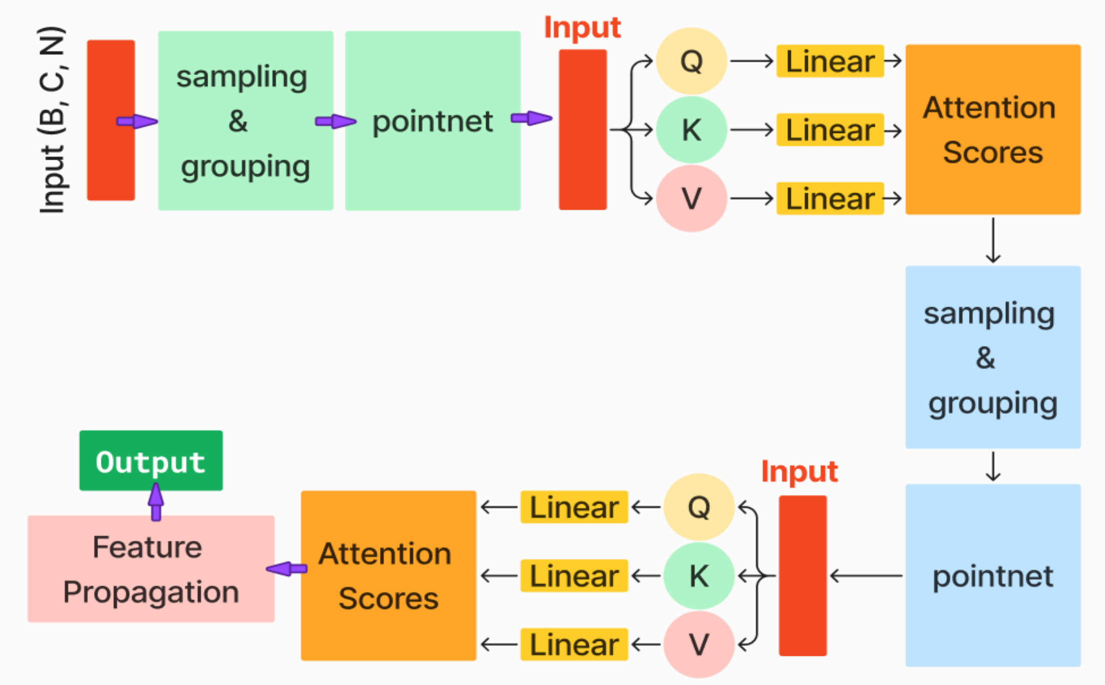

Figure 1: Point Cloud Attention Architecture. This attention method is integrated using PointNet++.

3.3.1 PointNet module

- **Input Transformation:** A mini-network (T-Net) is used to learn an affine transformation matrix that aligns the input point cloud into a canonical space.
- **Feature Extraction:** The aligned points are passed through a series of Multi-Layer Perceptron's (MLPs) to extract local features for each point.
- **Max Pooling:** A symmetric function (max pooling) is applied to aggregate the point features into a global feature vector.
- **Output Transformation:** The global feature vector is further processed by another T-Net to perform a feature transformation.
- **Classification/Segmentation:** For classification tasks, the transformed global feature is passed through fully connected layers to produce class scores. For segmentation tasks, the global feature is concatenated back with local features, followed by MLPs to predict point-wise labels.

3.3.2 Multi-Head Attention

Multi-head attention is applied with 8 heads:

$$Attention(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

where d_k is the dimension of the keys.

3.3.3 Attention Scores.

The attention mechanism calculates attention scores to assess the significance of each point relative to others. These scores are determined through the query *(Q)*, key *(K)*, and value *(V)* matrices, as shown in formula 3, where d_k represents the dimension of the keys. The softmax function ensures that the attention scores add up to one, enabling a probabilistic interpretation. Within the PointNet module, these attention scores are used to weigh the influence of different points during the aggregation of feature representations. This allows the network to concentrate on the most relevant points, enhancing its ability to capture essential structures in point cloud data. The process can be outlined in the following steps:

1) Linear Transformations: The input features are linearly transformed to obtain the query, key, and value matrices:
$$Q = W_q F_{pooled}, K = W_K F_{pooled}, V = W_V F_{pooled} \quad (3)$$
2) Score Computation: The attention scores are computed by taking the dot product of the query and key matrices, scaled by the square root of the key dimension:
$$S = \frac{QK^T}{\sqrt{d_k}} \quad (4)$$
3) Softmax Normalization: The scores are normalized using the softmax function to obtain the attention weights:
$$A = softmax(S) \quad (5)$$

4) Weighted Sum: The value matrix is weighted by the attention scores to produce the output features:

$$F_{attn} = AV \qquad (6)$$

These attention scores allow the model to dynamically adjust the focus on different points in the point cloud, thereby improving the overall feature representation and aiding in tasks such as classification and segmentation.

3.3.4 Feature Propagation

Feature propagation is a crucial step in point cloud processing that aims to up-sample the points and propagate features from the subsampled points back to the original resolution. This process helps in refining and combining local features to produce a more comprehensive global representation.

4. Experiments

4.1 Segmentation and Classification on KITTI dataset

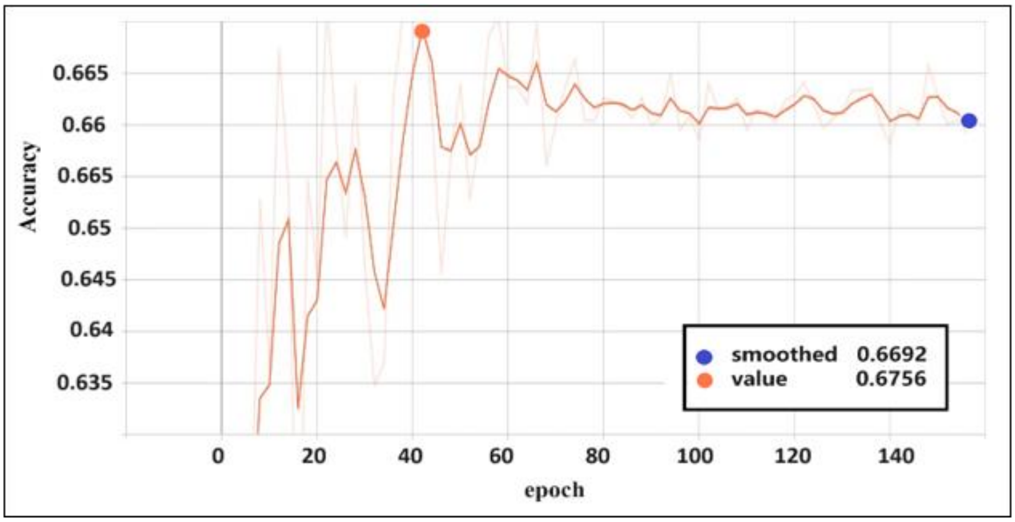

Figure 2: Point Cloud Attention. The accuracy result is 0.6692

Table 1: Clusters details

Model name for KITTI	Accuracy
PointNet++	0.6575
PCA (Fig. 2)	**0.6756**

This architecture integrates point sampling, grouping, attention mechanisms, and clustering to process point cloud data, enabling accurate classification and feature extraction. The neural network architecture for point cloud processing includes multiple set abstraction (SA) modules and feature propagation (FP) modules. Emphasis is placed on the SA modules, which incorporate attention mechanisms and clustering (Fig. 1, PCA Structure). The input data comprises point clouds represented as $N \times D$, where N is the number of points and D is the feature dimension. The first module, SA Module 1, follows these steps:

Point Sampling: Farthest Point Sampling is used to reduce the number of points to M.

Point Grouping: k-Nearest Neighbors (k-NN) is used for grouping with group size K.

MLP Layers: Three MLP layers for local features with 64, 128, and 256 neurons, respectively, each activated by ReLU.

Max-Pooling: Max-pooling is performed over K points. The output of SA Module 1 is processed by the first attention mechanism (Attention Mechanism 1).

Linear Layers: Transform keys, queries, and values to a M × 256 dimension.

Multi-Head Attention: Applied with 8 heads.

Output: Resulting in a M × 256 dimension, followed by a linear layer with ReLU activation. The results are clustered using K-means into C clusters, producing outputs of dimension C × D. This process is repeated in SA Module 2, resulting in an output of dimension M ′ × 512.

The second attention mechanism (Attention Mechanism 2) follows the same process as the first:

Linear Layers: Transform keys, queries, and values to a M ′ × 512 dimension.

Multi-Head Attention: Applied with 8 heads.

Output: Resulting in a M ′ × 512 dimension, followed by a linear layer with ReLU activation. A second clustering step results in C′ clusters. The final module, the FP Module, upscales the points using linear interpolation to a N × 512 dimension, followed by: **MLP Layers:** Three MLP layers for global features with 256, 128, and 64 neurons respectively, each activated by ReLU. The network concludes with an output layer: **Linear Layer:** The neuron count matches the number of classes.

Activation: Softmax activation is used for classification. In the first experiment, I trained the PCA (Point Cloud Attention) model with a batch size of 16 over 148 epochs.

This led to a **1.78%** improvement in performance, as demonstrated in Table 1 (Cluster Details) and the training graph in TensorBoard, Fig. 2. Multi-Scale Grouping Module

(SA Module): The SA Module performs point sampling at various scales and uses convolutional layers to extract features. These extracted features are then processed through the attention mechanism and clustered. Attention Mechanism: In this architecture, the attention mechanism is implemented using linear layers to generate queries (Q), keys (K), and values (V). Attention is calculated as a weighted sum of the values, with the weights determined by the similarity between the queries and keys.

Conclusion

In this study, we explored the integration of attention mechanisms within the processing of 3D point clouds to enhance feature extraction and object classification. By leveraging the attention mechanism, we demonstrated the ability to focus on the most relevant points in a point cloud, improving the model's ability to capture key spatial relationships and geometric structures. Our approach, which incorporates multi-scale grouping and hierarchical feature learning, was shown to provide significant performance improvements, as evidenced by the experimental results. The inclusion of attention mechanisms, particularly in the context of PointNet++ and similar architectures, allows for more effective handling of complex 3D point cloud data, reducing noise and improving the model’s accuracy and efficiency. The results from our experiments indicate that incorporating attention mechanisms can lead to better classification performance, even with non-uniformly sampled point clouds. Overall, the findings highlight the potential of attention-based techniques in 3D point cloud processing and their importance in advancing applications in robotics, autonomous driving, and other areas that rely on high-quality 3D data. Future work will focus on further optimizing these attention-based approaches and exploring their integration with other cutting-edge techniques to enhance the reliability and scalability of 3D point cloud processing systems.

References

[1] R. Abbasi, A. K. Bashir, H. J. Alyamani, F. Amin, J. Doh, and J. Chen, “Lidar point cloud compression, processing and learning for autonomous driving,” *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 962–979, 2022.

[2] Chunxiao Wang *et al.*, “An Improved DBSCAN Method for LiDAR Data Segmentation with Automatic Eps Estimation.,” *Sensors*, vol. 19, no. 1, p. 172, Jan. 2019, doi: 10.3390/s19010172.

[3] Charles R. Qi *et al.*, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” *ArXiv Comput. Vis. Pattern Recognit.*, Dec. 2016, doi: 10.1109/cvpr.2017.16.

[4] Charles R. Qi *et al.*, “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,” *Neural Inf. Process. Syst.*, vol. 30, pp. 5099–5108, Jun. 2017.

[5] A. Fusco *et al.*, “RadarSleepNet: Sleep Pose Classification via PointNet++ and 5D Radar Point Clouds,” *2023 IEEE Microw. Antennas Propag. Conf. MAPCON*, 2023, doi: 10.1109/mapcon58678.2023.10463831.

[6] Bülent Haznedar, Rabia Bayraktar, Ali Emre Ozturk, and Yusuf Arayıcı, “Implementing PointNet for point cloud segmentation in the heritage context,” *Herit. Sci.*, vol. 11, no. 1, Jan. 2023, doi: 10.1186/s40494-022-00844-w.

[7] Alex H. Lang *et al.*, “PointPillars: Fast Encoders for Object Detection from Point Clouds,” *ArXiv Learn.*, Dec. 2018, doi: 10.1109/cvpr.2019.01298.

[8] Bingjie Liu, Huaguo Huang, Shuxin Chen, Xin Tian, and Min Ren, “Tree Species Classification of Point Clouds from Different Laser Sensors Using the PointNet++ Deep Learning Method,” *IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2023, doi: 10.1109/igarss52108.2023.10282354.

[9] M.-H. Guo *et al.*, “PCT: Point cloud transformer,” *ArXiv Comput. Vis. Pattern Recognit.*, 2020, doi: 10.1007/s41095-021-0229-5.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *North Am. Chapter Assoc. Comput. Linguist.*, 2019, doi: 10.18653/v1/n19-1423.

[11] Ashish Vaswani *et al.*, “Attention Is All You Need,” *ArXiv Comput. Lang.*, Jun. 2017.