

Article

Multi-Scale Geometric Feature Extraction and Global Transformer for Real-World Indoor Point Cloud Analysis

Yisheng Chen, Yu Xiao, Hui Wu, Chongcheng Chen and Ding Lin *

The Academy of Digital China, Fuzhou University, Fuzhou 350108, China; 235527006@fzu.edu.cn (Y.C.); 235527058@fzu.edu.cn (Y.X.); 235527056@fzu.edu.cn (H.W.); chencc@fzu.edu.cn (C.C.)

* Correspondence: linding@fzu.edu.cn

Abstract: Indoor point clouds often present significant challenges due to the complexity and variety of structures and high object similarity. The local geometric structure helps the model learn the shape features of objects at the detail level, while the global context provides overall scene semantics and spatial relationship information between objects. To address these challenges, we propose a novel network architecture, PointMSGT, which includes a multi-scale geometric feature extraction (MSGFE) module and a global Transformer (GT) module. The MSGFE module consists of a geometric feature extraction (GFE) module and a multi-scale attention (MSA) module. The GFE module reconstructs triangles through each point's two neighbors and extracts detailed local geometric relationships by the triangle's centroid, normal vector, and plane constant. The MSA module extracts features through multi-scale convolutions and adaptively aggregates features, focusing on both local geometric details and global semantic information at different scale levels, enhancing the understanding of complex scenes. The global Transformer employs a self-attention mechanism to capture long-range dependencies across the entire point cloud. The proposed method demonstrates competitive performance in real-world indoor scenarios, with a mIoU of 68.6% in semantic segmentation on S3DIS and OA of 86.4% in classification on ScanObjectNN.



Citation: Chen, Y.; Xiao, Y.; Wu, H.; Chen, C.; Lin, D. Multi-Scale Geometric Feature Extraction and Global Transformer for Real-World Indoor Point Cloud Analysis. *Mathematics* **2024**, *12*, 3827. <https://doi.org/10.3390/math12233827>

Academic Editors: Jian Luo, Zheming Gao and Xin Yan

Received: 2 November 2024

Revised: 24 November 2024

Accepted: 26 November 2024

Published: 3 December 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: point cloud analysis; geometric feature; multi-scale attention; transformer; real-world indoor scenario

MSC: 68T45

1. Introduction

The rapid development of 3D scanning technology has enabled the acquisition of 3D representations of real-world objects with unparalleled accuracy and speed. Rich in geometric and topological information, point clouds are extensively utilized across various fields, such as urban planning [1], building modeling [2], robot navigation [3], indoor reconstruction [4], and autonomous driving [5,6]. The classification and segmentation of point clouds represents a significant method for the effective comprehension of such data and has been the subject of considerable interest amongst researchers, offering considerable research value and a broad range of potential applications.

In indoor point cloud scenes, object structures are complex, and there are numerous objects with similar local structures but disparate semantics, such as tables and chairs. The scale of objects varies considerably, from small decorations to large furniture, encompassing a vast range of sizes for indoor objects. The presence of severe occlusion issues in indoor environments results in the loss of information in specific areas of the point cloud data, which in turn affects the ability to detect complete objects and leads to the blurring of adjacent objects. Additionally, the presence of noise and background in indoor scenes can influence the subsequent feature extraction and the effectiveness of classification and segmentation.

The extraction of effective global and local information and the achievement of effective information aggregation represent the key objectives of current research. Pointnet [7] has

pioneered an end-to-end network architecture based on MLP (Multi-Layer Perceptron), which learns high-dimensional spatial features individually for each input point and combines them with global features obtained from pooling operations, with the aim of predicting the class of each point. As PointNet is unable to perceive local information, PointNet++ [8] addresses this shortcoming by acquiring local features of the point cloud through multi-layer local feature abstraction. Subsequent studies have extended this approach in various ways, but these research methods have several limitations. They have a restricted receptive field, lack interaction with larger contextual information, fail to express different local geometrical shape information, and lack perception of the spatial topology of the point cloud.

The processing and analysis of point clouds, which are unstructured data, present a number of challenges. As the coordinate information of the point cloud cannot fully represent its surface structure, this results in the loss of structured features. In order to capture the local structure, the authors of [9] propose a new point cloud curvature estimation method, designated as “Umbrella curvature”. This method defines the curvature value by considering the projection of vectors between a point and its neighboring points on the normal vector, with the objective of reducing the computational cost and estimating the curvature of point cloud data. The Repsurf [10] algorithm introduces an explicit umbrella geometry, thereby enhancing the point cloud representation. The method demonstrates superior performance in point cloud classification, segmentation, and object detection, and it constructs a surface representation based on each point. However, this surface representation structure is susceptible to noise. GAF-Net [11] captures detailed local geometric structural relationships by utilizing the spatial relative positions between the center point and its neighboring points, orientation information, and normal vector information. However, GAF-Net mainly emphasizes extracting features from local regions, neglecting the extraction of global features. OPFR [12] designs the local reference constructor module by approximating the local coordinate system based on a set of triangles but requires relatively high computational resources.

Owing to its exceptional global feature learning capabilities and permutation-invariant operations, the Transformer introduced in [13] proves especially suitable for point cloud processing and analysis. Several studies [14–19] have investigated the application of Transformer architectures or modules in point cloud analysis. The local Transformer is intended to enable feature aggregation in local patches rather than in the entire point cloud. The Point Transformer adopts the hierarchical architecture of PointNet++ for point cloud classification and segmentation. It focuses on local patch processing and replaces the shared MLP modules in PointNet++ with local Transformer modules. The Stratified Transformer divides the point cloud into non-overlapping cubic windows using 3D voxelization and conducts local Transformer operations in each window. The aforementioned studies primarily concentrate on local feature learning and aggregation, and due to limited receptive fields, they largely fail to capture the long-range context from distant dependencies. They neglect global contextual information, which is crucial for distinguishing between semantically different yet locally similar categories, such as the distinctions among doors, windows, columns, and walls. The global Transformer module allows each output feature to connect with all input features. It is permutation-invariant to the input arrangement and capable of learning global contextual features.

To tackle the above challenges, we propose a novel point cloud analysis network, PointMGST, which integrates the multi-scale geometric feature extraction (MSGFE) module and the global Transformer (GT) module. The MSGFE module comprises the geometric feature extraction (GFE) module and the multi-scale attention (MSA) module. The GFE module is designed to extract local geometric details from triangles formed by the central point and its neighboring points, including the centroid, normal vector, and plane constant. However, anomalous points among the neighbors may interfere with the local perception of the centroid, such as background points in the ScanObjectNN dataset or structurally similar yet semantically distinct points in the S3DIS dataset. To address this, we propose the MSA module, which extracts geometric features at different scales using multi-scale convolution and dynamically adjusts weights based on the relative importance of each channel. This enables the model to highlight critical features, thereby improving the effec-

tiveness of feature aggregation. The significance of each local point in feature aggregation can be dynamically adjusted based on its specific attributes. The channel weighting mechanism emphasizes essential features while suppressing irrelevant or minor ones, thereby enhancing the model's robustness to noise and enabling more precise classification and segmentation in real-world scenarios. We utilize a global Transformer to improve the model's capacity for capturing global features by considering the interrelations among all positions in the input features, beyond just the local neighborhoods. By evaluating channel similarities, the model more effectively captures unique and semantically significant features while mitigating the influence of simple features (e.g., edges and corners), thereby improving its understanding of interrelationships within the data. Additionally, a multi-layer residual connection mechanism is incorporated, allowing the model to retain original information while integrating global features, thereby improving information flow.

Our network has shown notable performance in both classification and segmentation tasks. The primary contributions of our work can be summarized as follows:

- (1) We proposed a MSGFE module, including the geometric feature extraction (GFE) and multi-scale attention (MSA) modules. It explicitly builds the local geometric surface representation of point cloud data and leverages multi-scale attention to enhance the model's awareness of geometric structures at various scales.
- (2) We proposed a global Transformer module that captures global context information by computing the dot product between query and key, thereby capturing long-range dependencies between feature points in the point cloud.
- (3) We introduce an end-to-end network for point cloud analysis, designated PointMSGT, which is based on the MSGFE and global Transformer modules. The experimental results demonstrate the effectiveness of the method in understanding 3D point clouds.

2. Related Work

2.1. Point-Based Methods

PointNet [7] represents a unified architectural framework for point cloud processing, which directly uses the 3D coordinates of point cloud data as input. It can predict the category label for the entire point cloud in classification tasks and also output per-point semantic labels for tasks such as object part segmentation and scene semantic segmentation. Afterward, researchers investigated several approaches to improve the effectiveness of point cloud analysis. PointNet++ [8] further strengthened the integration of local information, and designed an encoder–decoder structure that uses skip connections to combine global and local features, greatly improving the accuracy of the algorithm. However, the interaction between points in the network is somewhat limited, overlooking the geometric relationships between point clouds, resulting in missing local features. PointCNN [20] handles unordered point clouds through hierarchical convolutions and X-Conv operators, with its core being the learning of parameters to achieve point cloud sorting consistency. KPConv [21] defines an explicit convolution kernel based on kernel points, using these kernel points as reference points for computing the weights, which are then used to update each point. Pointweb [22] seeks the interactions between points through an adaptive feature adjustment module, which effectively reflects the contextual information of local regions. PointMLP [23], building on the original PointNet++, employs a pure MLP network and residual connections, addressing the issues of large computational loads and performance saturation in geometric feature extraction modules. PointNeXt [24] leverages a series of efficient training strategies to enhance point cloud shape classification and semantic segmentation outcomes. Point-based approaches have advanced the aggregation of fine-grained geometric features in local space. These methods heavily depend on the learnability of feature extractors, which may miss inherent local shape information. They often face challenges in building correlations among local features, and many tend to overlook the capturing of global information. In this work, the method we propose not only leverages local receptive field learning features but is also more efficient in the parallel learning of global information.

2.2. Geometric Feature Representation-Based Methods

The construction of geometric relationships has long been a prominent research topic in point cloud analysis. Prior to the advent of PointNet, traditional convolution methods were primarily applied to structured grids or voxels. The primary reason is that 3D point clouds possess unstructured properties, while traditional convolution methods excel with structured data, such as 2D images. Although this geometric transformation is beneficial for the learning of convolutional neural networks, the significant computational cost associated with the initial construction phase also results in the loss of 3D information. Following the introduction of PointNet, deep learning methods began to operate directly on points, without the introduction of intermediate representations. The authors of [25] added a lightweight local dependency module based on PointNet++, effectively simulating local structures within point clouds. Drawing inspiration from the SIFT operator, PointSIFT [26] encodes features in the XYZ space along eight directions, thereby overcoming the limitation of PointNet++, which restricts the KNN to the same direction. Nevertheless, this approach is markedly susceptible to the orientation information of objects. The authors of [27] propose GBNet, enhancing input features by leveraging the nearest edges and normal vectors, enriching the features with local distances inside and outside planar and surface patch descriptors. The authors of [28] proposed SCF-Net, which learns a z-axis rotation-invariant representation using polar coordinates instead of Cartesian coordinates, capturing the local geometric context. CurveNet [29] aggregates continuous sequences of points in point clouds through guided walks, thereby facilitating the extraction of rich geometric information from the curves. This method effectively compensates for the deficiencies of traditional local and global feature extraction methods, thereby enabling the capture of the shape and geometric features of point cloud objects. Repsurf [12] improves upon PointNet++, offering two variants: Triangular RepSurf and Umbrella RepSurf, which enhance the representation of point clouds by introducing explicit local geometric structures, addressing PointNet++'s inability to capture complex local geometries. However, the umbrella structure is vulnerable to noise, leading to umbrella surface distortion. The authors of [30] proposed the SRX Net, which employs ellipsoidal and umbrella surface representations to model the local geometric structure of point clouds and establish robust surface features. Compared to RepSurf, SRX Net can handle complex surfaces more effectively and shows greater robustness in various scenarios. The authors of [31] proposed an explicit 3D structure modeling method (X-3D), which captures explicit local structural information within the input 3D space, and uses these data to generate dynamic kernels with shared weights for all neighboring points in the local region. The authors of [12] introduced the "On-the-fly Point Feature Representation (OPFR)", explicitly capturing local geometric features in point clouds such as position, orientation, and curvature, approximating local coordinate systems, and generating high-quality triangle sets, offering an efficient and robust feature representation method for point cloud analysis tasks. These methods are principally based on the K-nearest neighbor (KNN) approach, which is employed to capture local neighbors and thereby reinforce semantic context information. The researchers investigated a range of feature extraction modules for the purpose of capturing local contextual information from point clouds, with the result that excellent results were obtained in the field of point cloud analysis.

2.3. Attention-Based Methods

In recent times, the attention mechanism has emerged as a foundational technique for the extraction of global contextual features, with considerable implications for practical applications. For example, Hu et al. combined self-attention with a random sampling algorithm and proposed the RandLA-Net [32] network, gradually expanding the receptive field of each point to perceive and learn complex spatial geometric structures, achieving good segmentation results, though it easily loses local features in complex point cloud scenes. Feng et al. used a method called LAE-Conv [33], a local attention-edge convolution, which assigns different attention values to each edge in the local graph. They then used the weighted sum of neighboring points to assign each point an edge attention coefficient. They subsequently used the weighted sum of neighboring points to allocate an edge attention coefficient to each point. Wang et al. proposed

CSANet [34], which employs a cross self-attention module to facilitate the learning of point cloud coordinates and feature information, thereby integrating point cloud geometry and semantic information. Inspired by the noteworthy advancements made by Transformers in the domains of natural language processing and image processing, Zhao et al. proposed the pPoint Transformer [18], which introduced the self-attention network into the domain of 3D point cloud comprehension. A pPoint Transformer layer with high expressiveness was designed for point cloud processing. This layer computes attention scores between central and neighboring points for weighted summation. Additionally, position encoding was incorporated for scene understanding of point clouds. The authors of [19] introduced an enhanced method based on group vector attention, which was designed to address the limitations of the original model. However, these methods primarily concentrate on the extraction of local features for each point, while the global dependencies between all input points are largely overlooked. The failure to consider this aspect frequently results in errors in classification and segmentation, particularly when attempting to differentiate between small or partial three-dimensional objects within real-world scenes.

3. Methodology

3.1. Network Architecture

Our PointMSGT follows the classic PointNet++ architecture, using a hierarchical structure of encoder and decoder, as shown in Figure 1. It mainly consists of an encoder, MSGFE, classification, and segmentation. The original point cloud is used as input, and geometric features are extracted through the MSGFE layer. Four layers of Set Abstraction modules are set, mainly consisting of sampling, grouping, MLPs, a global Transformer, and reduction. The sampling layer uses the farthest point sampling (FPS) method to obtain the sampled center points. The grouping layer uses the k-nearest neighbors (kNN) approach to find a fixed number of points as a region. MLPs are used for transforming the dimensions of point cloud data in the local region. The global Transformer is used to capture the global context information between points, and the reduction layer uses max pooling to aggregate global features from multiple local regions. The decode module continues the post-processing process of PointNet++, split into two tasks: classification and segmentation. In classification tasks, the global feature vector is passed sequentially through an MLP and fully connected layers, with a dropout layer added to prevent overfitting. The output layer uses the softmax function to generate the probability distribution for each class. In segmentation tasks, the Feature Propagation (FP) layer is used to progressively restore the resolution of the point cloud. The FP layer fuses the features from the corresponding layers of the encoder with the current decoding layer's features through skip connections. Finally, a shared MLP predicts the semantic label for each point. The Feature Propagation layer adjusts input channels solely to match the outputs of Set Abstraction. Annotations for point cloud resolution and output channels are provided beneath each block.

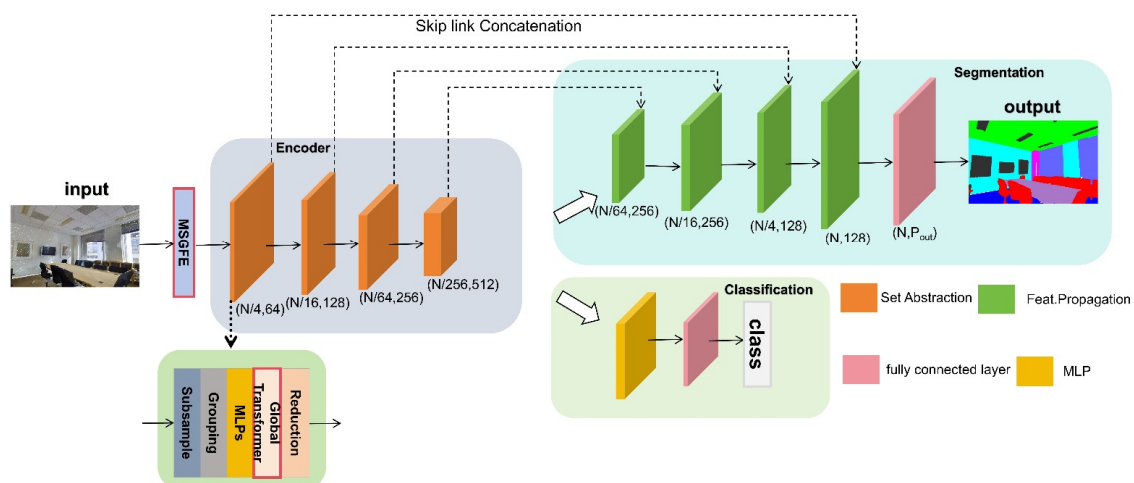


Figure 1. Architecture of the proposed network.

3.2. Multi-Scale Geometric Feature Extraction

Triangles are fundamental in 3D geometry as they are versatile in representing complex shapes, making them suitable for capturing detailed local structures in point clouds. Triangle meshes are a prevalent method of 3D model representation, typically composed of triangular planes with shared edges or vertices. Triangle meshes possess the capacity to reproduce continuous objects with complex three-dimensional shapes in a flexible manner, indicating that triangular surfaces are suitable for constructing spatial topological structures. Accordingly, we propose the multi-scale geometric feature extraction (MSGFE) module (Figure 2). We employed specific properties of triangles to represent the geometric features of points, thereby reconstructing the local point cloud structure, which consists of multiple triangular surfaces, through the geometric feature extraction (GFE) module. We utilized the multi-scale attention (MSA) module and pooling functions to learn and aggregate the local topological relationships of domain points. This geometric surface structure, as an extended form of triangular surfaces, expands the receptive field of geometric shape features in comparison to single triangular surface representation methods. It aggregates the features of multiple local triangles, thereby obtaining a richer local geometric description. By adaptively aggregating features at multiple scales within the local neighborhood, redundant information is further reduced, and the local details of the point cloud are effectively captured. Multi-scale attention and pooling layers are utilized to comprehensively learn the geometric shape features of multiple triangular surfaces within the central point neighborhood. Additionally, the feature channels are automatically adjusted to enhance the effective representation of local spatial features of the point cloud.

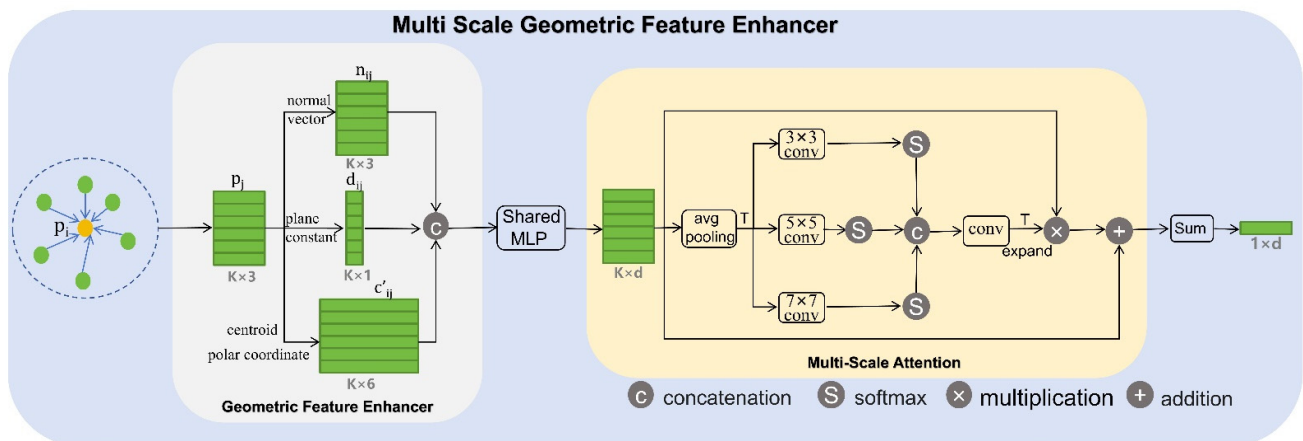


Figure 2. Multi-scale geometric feature extraction.

The MSGFE represents the local geometric structure of point clouds by means of a rotation and sorting of the neighboring points, with each pair of neighboring points connected to the central point. $P = \{p_i | i = 1, 2, \dots, n\}$ is defined as a set of point clouds, where n is the number of points in the point cloud, and each point has three-dimensional coordinates, i.e., $p_i \in \mathbb{R}^3$. When constructing geometric features, both the central point and its neighboring points are taken into account. The K nearest neighbors of the central point are identified through the application of the K -nearest neighbors (kNN) algorithm.

$$N_i = \text{KNN}(P, p_i) \quad (1)$$

The coordinate difference of each neighbor point relative to the centroid p_i is computed to form the neighborhood point set $P_i = \{p_j \in P | j \in N_i\}$, where N_i represents the K -nearest neighbor indexes of p_i . Subsequently, the polar angle θ_{ij} of each neighboring point p_j relative to the central point p_i is computed, and then sorted counterclockwise by the magnitude of the polar angle to ensure the order on the xy -plane from 0° to 359° along

the x-axis. (x_{ij}, y_{ij}, z_{ij}) represent the coordinates of the neighboring point p_j relative to the central point p_i .

$$\theta_{ij} = \arccos\left(\frac{z_{ij}}{\sqrt{x_{ij}^2 + y_{ij}^2 + z_{ij}^2}}\right) \quad (2)$$

Subsequently, each pair of adjacent points is selected from the sorted neighboring points to form a triangle with the central point, designated as p_i . The formation of these triangles with the central point p_i facilitates a more effective extraction and description of the local surface's geometric features. We calculate the centroid of p_{ij-1} and p_{ij} :

$$c_{ij} = \left(\frac{x_{ij} + x_{ij-1}}{2}, \frac{y_{ij} + y_{ij-1}}{2}, \frac{z_{ij} + z_{ij-1}}{2}\right) \quad (3)$$

Previous models that used Cartesian coordinates as input were unable to fully capture the complex geometric relationships between the central point and its neighboring points. In real-world scenes, objects belonging to different categories or representing different parts of the same object frequently display disparate orientations. The polar coordinate system represents point coordinates based on the origin through distance and angle, which adds sensitivity to distance and direction. Therefore, we incorporate polar coordinate features as part of the local structure to obtain a more comprehensive geometric representation. For the sorted neighboring points, we calculate their polar coordinates (radial distance, azimuth angle, and elevation angle) relative to the central point p_i .

The normalized centroid coordinates $(x'_{ij}, y'_{ij}, z'_{ij})$ of the neighboring points within the triangular structure are converted into polar coordinates using a spherical system. These are connected with the 3D coordinate features to form enhanced spatial relative coordinates, which facilitate the perception of the distance and direction of neighboring points. For the midpoint of the j th neighboring point of the central point p_i , we calculate its radial distance ρ_{ij} , azimuth angle θ_{ij} , and elevation angle ϕ_{ij} .

$$\rho_{ij} = \sqrt{x_{ij}^2 + y_{ij}^2 + z_{ij}^2} \quad (4)$$

$$\theta_{ij} = \arccos\frac{z'_{ij}}{\rho_{ij}} \quad (5)$$

$$\phi_{ij} = \arctan2(y'_{ij}, x'_{ij}) \quad (6)$$

To stabilize the training process, we normalize the polar coordinates, where the elevation angle t is normalized to $[0, 1]$ by dividing by π , and the azimuth angle is normalized to $[0, 1]$ by dividing by 2π and adding 0.5.

$$\theta'_{ij} = \frac{\theta_{ij}}{\pi} \quad (7)$$

$$\phi'_{ij} = \frac{\phi_{ij}}{2\pi} + 0.5 \quad (8)$$

Then, the calculated polar coordinate information is concatenated with the original Cartesian coordinate information to form an enhanced feature representation.

$$c'_{ij} = (x'_{ij}, y'_{ij}, z'_{ij}, \rho_{ij}, \theta'_{ij}, \phi'_{ij}) \quad (9)$$

For each pair of neighboring points p_{ij-1} and p_{ij} , we calculate the normal vector n_{ij} and the plane constant d_{ij} of the triangle they form with the central point p_i . The normal vector is computed by extracting the edge information between the central point and its

two neighboring points through a cross product, and the normal vector is normalized, where $\times$ represents the cross product of the vectors:

$$e_i = (e_i^1, e_i^2) = (p_{ij-1} - p_i, p_{ij} - p_i) \quad (10)$$

$$\eta_i = e_i^1 \times e_i^2 \quad (11)$$

$$n_{ij} = \frac{\eta_{ij}}{\|\eta_{ij}\|} \quad (12)$$

Using the centroid and the normal vector, we calculate the plane constant d_{ij} for the triangle, which is the constant term in the point cloud plane equation $n_{ij} \cdot (p_i - c_{ij}) = d_{ij}$. The formula for calculating the plane constant is as follows:

$$d_{ij} = x_{ij}x_{ij-1} + y_{ij}y_{ij-1} + z_{ij}z_{ij-1} \quad (13)$$

The normal vector n_{ij} of the triangle, the plane constant d_{ij} , and the normalized centroid c'_{ij} enhanced by polar coordinates are concatenated, and the features are further encoded through a shared MLP, resulting in the explicit geometric feature u_i of the triangle.

$$u_i = \text{MLPs}\left(\left[c'_{ij}; n_{ij}; d_{ij}\right]\right) \quad (14)$$

Finally, we obtain the geometric surface as the geometric features U_i of K triangles formed by p_i and its K neighboring points.

$$U_i = S(M(u_i)) \quad (15)$$

where S is the aggregated feature, and we use summation for aggregation. The features of the K triangles are processed using the multi-scale attention module (MSA) and the summation function.

3.3. Multi-Scale Attention Module

Following the construction of the local point cloud of the triangle geometric structure, the spatial features of the geometric structure are fed into the MSA for feature abstraction. The MSA module is a multi-scale channel attention mechanism that computes attention maps at multiple scales using convolution kernels of varying sizes, subsequently fusing these attention maps to enhance attention to channel features at different scales, thereby improving the model's capacity to process complex geometric structures in point clouds (Figure 3).

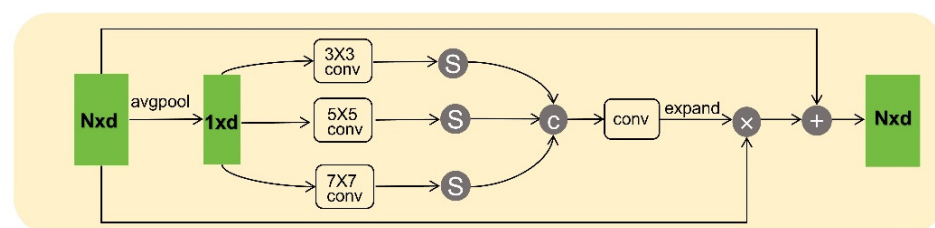


Figure 3. Multi-scale attention module.

The process of MSA is as follows:

(1) Initially, the input feature map is subjected to spatial feature compression. In the spatial dimension, average pooling is employed, resulting in a $1 \times d$ feature map.

$$Y = \text{AvgPool}(X) \quad (16)$$

(2) The channel feature information of the input feature map is captured at different scales, thereby enabling the model to handle geometric information at various levels. Through 3×3 , 5×5 , and 7×7 convolutions, the importance of different channels is learned

at multiple scales, with each feature map output still being $1 \times d$. Here, $i \in [1, \dots, m]$, where m is the number of convolution kernels, and this paper uses three convolutions.

$$Z_i = \text{Sigmoid}(\text{Conv1d}(Y)) \quad (17)$$

(3) Channel attention fusion. All channel attention feature maps are concatenated, and multi-scale information is fused through convolution. The result is then multiplied channel-wise with the original input feature map, and the final output is a feature map with channel attention. An adaptive weight is assigned to each channel to enhance important channel features and suppress less important ones.

$$Z_{\text{stack}} = \text{Concat}(Z_1, Z_2, \dots, Z_m) \quad (18)$$

$$X_{\text{out}} = \text{Conv}_{\text{fuse}}(Z_{\text{stack}}) \times Y \quad (19)$$

Multi-scale convolution enables the model to capture geometric information across a range of scales, while adaptive weighting allows for the allocation of weights based on the relative importance of the channels within these multi-scale features. This combination enables the model to simultaneously extract multi-level geometric information and dynamically adjust the weights to emphasize the most critical features. Through this channel weighting, the model is enabled to process features in a more intelligent manner, thereby facilitating the discernment of crucial geometric information. This results in a greater influence on the final feature aggregation. Consequently, the model's output contains more refined features, thereby enabling it to capture geometric surface characteristics at different scales and effectively enhancing its ability to perceive complex geometric structures. Furthermore, this reduces the impact of noise on the surface representation of geometric structures, making the representation of triangular geometric information more accurate and thus improving the aggregation of geometric features.

3.4. Global Transformer

Self-attention represents a specific form of attention mechanism, whereby the input sequence is utilized as the source of keys, values, and queries. Given an input sequence $X = \{x_1, x_2, \dots, x_n\}$, where x_i represents the feature representation of the i th element in the sequence, the self-attention mechanism treats each element x_i as the query q_i , key k_i , and value v_i . The attention output for element x_i is then computed as follows:

$$z_i = \sum_{j=1}^n \alpha_{ij} v_j \quad (20)$$

where α_{ij} represents the attention weight of element x_i on element x_j , which is computed as follows:

$$\alpha_{ij} = \frac{\exp(q_i^T k_j)}{\sum_{k=1}^n \exp(q_i^T k_k)} \quad (21)$$

It is important to note that in the self-attention mechanism, the values of q , k , and v are obtained through linear transformations.

$$q_i = x_i W_q, k_i = x_i W_k, v_i = x_i W_v \quad (22)$$

where W_q , W_k , and W_v are the learnable parameter matrices, respectively. The self-attention mechanism considers all elements in the sequence simultaneously, thereby conferring upon the network a global receptive field. Two major advantages are associated with self-attention: parallel computation and the shortest maximum path length. The self-attention mechanism captures the relationships between elements by constructing a similarity matrix, and it can adapt to the irregular nature of point clouds; thus, it is well-suited for point cloud data.

In the extraction of local features from point clouds in complex, large-scale scenes, existing deep learning-based point cloud semantic segmentation methods are capable of handling unordered point cloud data but may be unable to fully leverage global structural information. In order to enhance the network's ability to capture essential global details in real-world scenes and to more effectively capture the local structural information of point clouds in such scenarios, it is necessary to improve the model's processing capabilities for this type of data. In this paper, we propose a global Transformer inspired by non-local blocks and affinity matrices. This new approach emphasizes meaningful feature channels, helping the model capture semantic information from point clouds more effectively. It improves segmentation performance and reduces redundancy in feature representation. The global Transformer module is shown in Figure 4. It first computes feature correlations at the channel level, then uses non-local blocks to capture global context information, and finally combines the result with the original input through a residual connection.

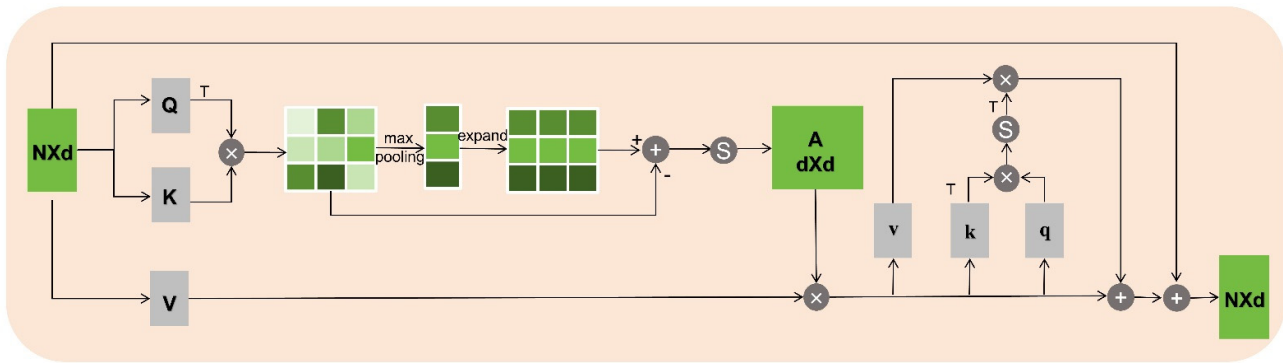


Figure 4. Global Transformer.

The input is a d -dimensional feature map $F_{N \times d}$, where N is the number of points and d is the feature dimension. Here, $F_{N \times d} = [c_1, c_2, \dots, c_d]$ and $c_i \in \mathbb{R}^N$ represent the feature vector for each channel. A shared MLP is used to simplify the contextual information in each channel, reducing computational cost. The shared MLP (W_{QC} and W_{KC}) is employed in the generation of the query matrix Q and the key matrix K , respectively. The shapes of Q and K are simplified by the weight matrices W_{QC} and W_{KC} to $(s/8) \times d$, thereby improving efficiency. In comparison to the use of all points or the selection of certain points based on additional metrics, this method effectively reduces the size while retaining the information in each channel. Subsequently, the corresponding channel similarity matrix S is estimated by applying the product of the transposed Q matrix and K matrix.

$$S_c = KQ^T = (F_{n \times d}W_{KC}) (F_{n \times d}W_{QC})^T \quad (23)$$

In this context, S_{ij} represents the degree of similarity between the i th and j th channels of the given feature map $F_{N \times d}$. The dissimilarity between channels, termed channel affinity, is defined and max pooling is performed along the columns of S to identify the maximum similarity, which is then expanded to the resulting matrix is of the same size as S . By subtracting the original S from the expanded matrix, channels with higher similarity will have lower dissimilarity. The channel correlation matrix A is computed through the similarity matrix S . The elements in A are inversely proportional to the similarity between the two corresponding channels, with larger elements indicating lower similarity.

$$A_c = \text{softmax}(\text{expand}(\text{maxpooling}(S)) - S) \quad (24)$$

Subsequently, the resulting channel correlation matrix A is multiplied by the V matrix, which was generated through the weight matrix W_{VC} , in order to obtain F_A .

$$F_A = VA \quad (25)$$

Subsequently, matrix multiplication is conducted utilizing the learnable parameter matrices q and k , which are of dimensions $N/2 \times d$. By employing a normalization function to obtain the weight coefficient matrix between individual point channels, the resulting weight coefficient matrix is multiplied by the v values in order to compute self-attention. The final output is combined through a residual connection, and learnable parameters γ are applied to control the weight of each part.

$$F_G = \gamma \left(\text{softmax}(q k^T) \right)^T \cdot v + F_A \quad (26)$$

Ultimately, the outcome is integrated with the initial input. Furthermore, residual connections with learnable weights α are employed to facilitate the training process.

$$F'_{N \times \lceil} = GLO(F) = F + \alpha \cdot F_G \quad (27)$$

The aforementioned operations can enhance the semantic relevance of feature representations, thereby enhancing the model's performance on semantic segmentation tasks. The model, through the global Transformer, is capable of learning long-range dependencies between all points in the point cloud, thereby enabling classification or segmentation tasks to capture more global context information.

4. Experiments

In this section, we perform an experimental validation to demonstrate the significant performance of our proposed method on the dataset. The validation is conducted from two main perspectives: a comparison of quantitative results for similar methods on public datasets and an evaluation of the proposed new modules' effectiveness through ablation experiments. The evaluation metrics include Overall Accuracy (OA), Mean Accuracy (mAcc), and mean Intersection-over-Union (mIoU). Let us consider a set of N points and K semantic categories. For a given category i , TP_i denotes the number of points correctly classified as category i , FP_i represents the number of non- i points incorrectly predicted as category i , FN_i denotes the number of points in category i incorrectly predicted as non- i , and TN_i represents the number of non- i points correctly classified as non- i . The formulas for these three metrics are as follows:

$$OA = \frac{1}{N} \sum_{i=1}^k TP_i \quad (28)$$

$$mIoU = \frac{1}{k} \sum_{i=1}^k \frac{TP_i}{TP_i + FP_i + FN_i} \quad (29)$$

$$mAcc = \frac{1}{k} \sum_{i=1}^k \frac{TP_i + TN_i}{TP_i + TN_i + FN_i + FP_i} \quad (30)$$

4.1. Experimental Settings

4.1.1. Dataset Description

We use the S3DIS dataset for semantic segmentation task and the ScanObjectNN dataset for classification task to verify the effectiveness of our PointMSGT. The S3DIS is an indoor scene point cloud semantic segmentation dataset used to validate the effectiveness of point cloud segmentation. The dataset includes six large indoor areas covering 6020 square meters and contains more than 215 million data points. It contains 13 semantic categories, including structural elements (ceiling, floor, wall, beam, column, window, door), common furniture (table, chair, sofa, bookshelf, board), and clutter. Since Area 5 is different and more complex than the rooms in other areas, making it suitable to evaluate the generalization ability of the model, we chose Area 5 as the test set and used the remaining five areas as the training set. The ScanObjectNN dataset comprises 15,000 objects, distributed across 15 categories, with 2902 unique instances observed in real-world scenes. The presence of background, noise, and occlusion in ScanObjectNN presents a significant challenge to existing methods. We validated

our PointMSGT in the most challenging variant (PB_T50_RS) of ScanObjectNN. Following standard procedures, 80% of the ScanObjectNN dataset was used for training and 20% for validation, with the sample rate of each point, reduced to 1024 points.

4.1.2. Train

The experimental platform used in this paper is a Linux system with a 64-bit operating system, the GPU used is an NVIDIA GeForce RTX 3090 with 24 GB of memory, and the network was constructed using the PyTorch framework. For the semantic segmentation task, the AdamW optimizer with default parameters was chosen to train our PointMSGT. The number of epochs is 100, the batch size is 2, and the initial learning rate is 0.006. A multi-step learning rate decay scheme was used, where the learning rate was decayed at epochs [60, 80] with a decay rate of 0.1. The performance of the point cloud segmentation algorithm is evaluated using three metrics: Overall Accuracy (OA), Mean Accuracy (mAcc) and Mean Intersection-over-Union (mIoU). For the shape classification task, the Adam optimizer with default parameters was chosen to train our PointMSGT. The number of epochs is 250, the batch size is 64, and the initial learning rate is set to 0.001, with a decay rate of 0.7 every 20 iterations. The classification performance of the point clouds is evaluated using Overall Accuracy (OA) and Mean Accuracy (mAcc).

4.2. Point Cloud Semantic Segmentation

Inspired by [24,35], we applied data augmentation during training, such as point cloud scaling, color contrast, color shift, and color jittering. Table 1 shows the comparison of test results on the S3DIS dataset between our PointMSGT and other advanced methods. Table 2 offers the specific results of S3DIS Area 5.

Table 1. Semantic segmentation results (%) of different methods on S3DIS (Area 5). The best performing results are shown in bold.

Model	OA	mACC	mIoU	Params
PointNet (2017) [7]	78.9	52.6	43.7	3.6M
PointNet++ (2017) [8]	86.4	61.2	56.0	1.0M
PointCNN (2019) [20]	—	—	56.9	—
RandLA-Net (2020) [32]	87.2	71.4	62.4	4.99M
PointNeXt-B (2022) [24]	89.0	73.0	66.5	3.8M
AGConv (2023) [36]	90.0	73.2	67.9	—
TCFAP-Net (2024) [37]	88.9	—	66.1	8.10M
LACV-Net (2024) [38]	—	—	66.4	5.67M
HADF-Net (2024) [39]	—	—	66.8	5.96M
DG-Net (2024) [40]	90.2	—	65.9	—
PointMSGT (ours)	90.1	76.3	68.6	2.03M

Table 2. Semantic segmentation results (%) of object categories using different methods on S3DIS (Area 5) (%). The best performing results are shown in bold.

Model	mIoU	Ceiling	Floor	Wall	Beam	Column	Window	Door	Chair	Table	Bookcase	Sofa	Board	Clutter
PointNet++ [8]	56.0	91.5	98.2	82.2	0	18.0	57.8	64.6	79.7	87.8	67.1	69.8	65.3	50.8
SAPointNet++ [41]	53.1	92.0	97.7	75.6	0	2.8	52.9	29.5	77.4	72.6	56.6	36.9	46.6	50.2
PointCNN [20]	57.3	92.3	98.2	79.4	0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
RandLA-Net [32]	62.4	91.1	95.6	80.2	0	24.7	62.3	47.7	76.2	83.7	60.2	71.1	65.7	53.8
PointGS [42]	61.7	92.2	97.9	82.6	0	25.9	53.4	69.8	74.9	80.3	38.6	66.2	69.6	51.1
DG-Net [40]	65.9	93.1	97.8	82.6	0	33.1	62.2	52.3	80.4	87.5	72.3	70.7	69.8	54.5
PointNeXt-B [24]	66.5	92.9	98.4	82.0	0	17.4	57.8	68.9	80.5	90.6	77.2	72.6	70.1	55.26
PointMSGT (Ours)	68.5	93.9	98.5	85.7	0	34.45	62.4	70.1	80.8	89.7	77.2	71.4	67.8	57.14

The results showed that our PointMSGT achieved the best results in terms of mIoU (76.3%) and mAcc (68.6%), although the OA was only 0.1% lower than the highest one of 90.2%. Compared to PointNet++, our OA increased by 3.7%, mAcc increased by 15.2%, and the most important metric, mIoU, increased by 12.6%. We also evaluated the network complexity of our method. While achieving the highest mIoU performance, our PointMSGT also features a

relatively low parameter count. Although the parameters of PointMSGT are slightly higher than those of the classic PointNet++, it remains competitive when compared to other advanced approaches. Furthermore, we analyzed the segmentation accuracy by category, where our PointMSGT demonstrated leading performance in ten out of thirteen categories, with significant improvements in five categories, notably pillars, doors, bookshelves, and clutter. Compared to other methods, PointMSGT also achieved higher segmentation performance in certain categories, including roof, floor, wall, pillar, door, window, chair, bookshelf, and clutter, while demonstrating close-to-highest performance in table, sofa, and board.

The intricate forms and multitude of indoor objects make the semantic segmentation of large indoor scenes challenging. To provide a clearer comparison among different methods, we qualitatively evaluated PointNet++, PointNeXt-B, Repsurf, and our PointMSGT by visualizing their results on the S3DIS dataset, as shown in Figure 5. PointNet++ and PointNeXt-B have a similar structure to our PointMSGT, but lack of sufficient geometric shape representation. Repsurf improves the surface representation with explicit geometric structures but suffers from noise interference, leading to misclassifications between similar categories. In Figure 5, the segmented objects in red boxes showed that our method clearly outperforms the others, especially in the column, clutter, and door categories. Our PointMSGT shows more complete and accurate shape segmentation of columns, proving that our network is more sensitive to geometric shapes, clearer at distinguishing boundary points, and less affected by noise interference, thus capturing local information in large-scale indoor scenes. We can more clearly distinguish clutter from semantically different categories such as bookshelves and boards, which have similar shapes. As shown in the top three rows, other methods tend to segment clutter as boards or bookshelves, whereas walls with hanging clutter are more clearly separated by our PointMSGT. Our PointMSGT demonstrates more precise segmentation performance and smoother object shapes compared to other methods, owing to the integration of MSGFE and GT. MSGFE represents local geometric features through triangles, constructing the spatial topological relationship between points and their neighboring points, allowing for detailed descriptions of local point cloud shapes. Other methods overlook complex local shapes, while MSGFE significantly enhances the ability to describe local shapes by incorporating geometric features such as polar coordinates, normal vectors, and triangle plane constants. The global Transformer leverages a self-attention mechanism to compute long-range dependencies between points, allowing for a more holistic capture of global features. Unlike conventional local feature extraction techniques, the global Transformer attends to the positional relationships of all points in the point cloud, offering a more accurate grasp of the overall structure in complex environments. MSGFE focuses on local geometric features, while the global Transformer emphasizes the global context. Their combination enables the model to comprehensively perceive complex indoor scenes in semantic segmentation tasks.

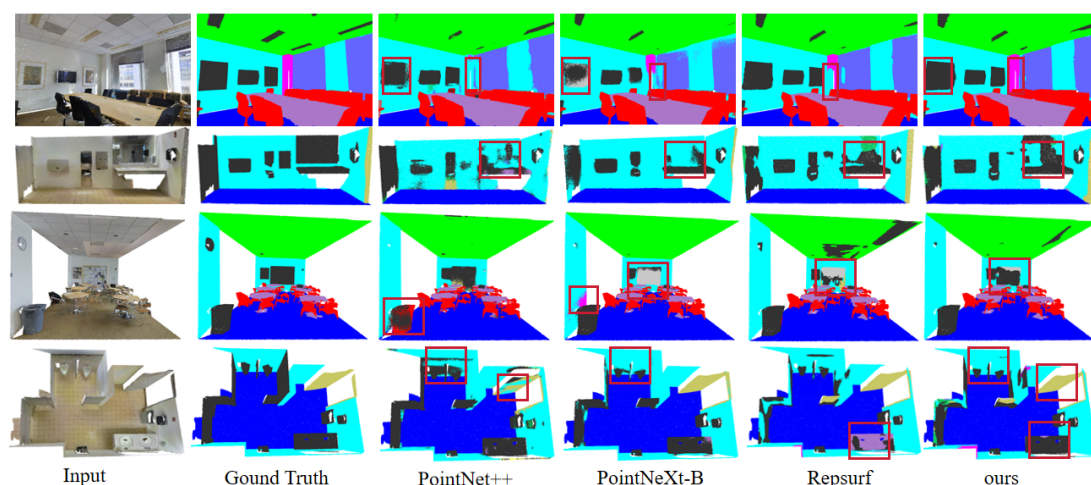


Figure 5. Visualization comparison of semantic segmentation results on S3DIS. The red rectangular boxes indicate the contrasting regions.

Point Cloud Object Classification

Due to the presence of a complex background, missing parts, and various deformations, the ScanObjectNN dataset presents a more practically challenging scenario than other point cloud classification datasets. This allows us to test the PointMSGT for a more robust and effective evaluation in complex real-world scenarios. The accuracy of competing methods on the same dataset is presented in Table 3.

Table 3. Object Classification results (%) of different methods on ScanObjectNN. The best performing results are shown in bold.

Model	Input Size	OA	mACC
PointNet++ [8]	1024	77.9	75.4
DGCNN [43]	1024	78.1	73.6
PointCNN [20]	1024	78.7	75.1
PRANet [44]	1024	82.1	79.1
PointMLP [23]	1024	85.7	84.4
PointTransformer [18]	1024	80.8	78.2
GBNet [27]	1024	80.5	77.8
Point-MAE [45]	1024	85.2	—
Point-TnT [46]	1024	83.5	81.0
Point-BERT [47]	1024	83.1	—
3DGTN [48]	1024	85.8	83.2
GuidedConv [49]	1024	84.0	79.4
PointMSGT (Ours)	1024	86.4	85.0

Under the setting of 1 k input points, our method demonstrates superior classification outcomes compared to most existing methods, with an OA of 86.4% and a mAcc of 85.0%. Experiments show that our PointMSGT outperforms other methods, with OA and mAcc being 3.1% and 4.0% higher than the transformer-based Point-TnT network, and 6.1% and 8.2% higher than the geometry descriptor-based GB-Net network. In addition, our model reduces the gap between OA and mAcc performance, showing less bias towards certain categories compared to existing works. Our model shows better performance in general and better adaption in each category's samples. Table 4 shows the classification results of different networks, with ours achieving the leading performance in most categories and significantly higher classification accuracy in bags and boxes. The proposed model effectively addresses the main difficulties of the ScanObjectNN dataset, such as confusing background points which are irrelevant to the shape structure. Our global Transformer module calculates the similarity between channels, which allows the model to more effectively capture distinctive and semantically significant features. This reduces the importance of background points, thereby improving the overall segmentation performance.

Table 4. Classification accuracy (%) of different methods on ScanObjectNN Benchmark. The best performing results are shown in bold.

Method	Bag	Bin	Box	Cabinet	Chair	Desk	Display	Door	Shelf	Table	Bed	Pillow	Sink	Sofa	Toilet	OA	mAcc
3DmFV [50]	39.8	62.8	15	65.1	84.4	36	62.3	85.2	60.6	66.7	51.8	61.9	46.7	72.4	61.2	63	58.1
PointNet [7]	36.1	69.8	10.5	62.6	89	50	73	93.8	72.6	67.8	61.8	67.6	64.2	76.7	55.3	68.2	63.4
SpiderCNN [51]	43.4	75.9	12.8	74.2	89	65.3	74.5	91.4	78	65.9	69.1	80	65.8	90.5	70.6	73.7	69.8
PointNet++ [8]	49.4	84.4	31.6	77.4	91.3	74	79.4	85.2	72.6	72.6	75.5	81	80.8	90.5	85.9	77.9	75.4
DGCNN [43]	49.4	82.4	33.1	83.9	91.8	63.3	77	89	79.3	77.4	64.5	77.1	75	91.4	69.4	78.1	73.6
PointCNN [20]	57.8	82.9	33.1	83.6	92.6	65.3	78.4	84.8	84.2	67.4	80	80	72.5	91.9	71.8	78.5	75.1
BGA-PN++ [52]	54.2	85.9	39.8	81.7	90.8	76	84.3	87.6	78.4	74.4	73.6	80	77.5	91.9	85.9	80.2	77.5
BGA-DGCNN [52]	48.2	81.9	30.1	84.4	92.6	77.3	80.4	92.4	80.5	74.1	72.7	78.1	79.2	91	72.9	79.7	75.7
GB-Net [27]	59.0	84.4	44.4	78.2	92.1	66	91.2	91	86.7	70.4	82.7	78.1	72.5	92.4	77.6	80.5	77.8
PointMSGT (ours)	71.1	85.43	69.9	86.3	94.4	72	91.7	95.7	89.6	74.4	84.6	82.9	83.3	95.7	89.4	86.4	85.0

We conducted performance test experiments to evaluate the robustness with different numbers of input points. The impact of different sampled point sets was tested in the

classification task with the number of sampled points set to 1024, 768, 512, and 256, as shown in Figure 6. It demonstrates relatively stable and robust performance under different sizes of input point settings, with the OA generally above 85%. When the input point count is reduced to 256, where much of the geometric information is lost, our model still achieves an OA of 83%. This illustrates that whatever variations in input counts and point cloud densities, along with differences in the distances between central and neighboring points, it is possible to obtain a balanced combination of local geometric and global contextual information to maintain the stability of the model's performance.

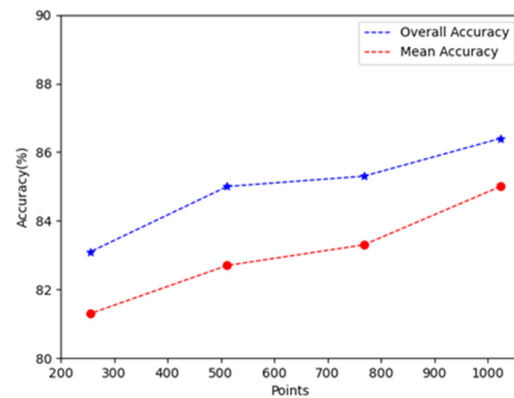


Figure 6. Performance of different numbers of input points.

4.3. Ablation Study

In this research, we conducted ablation studies on the core modules, core components, and the number (K) of neighborhood points to better demonstrate the effectiveness of PointMSGT.

4.3.1. Ablation Study of Core Modules

In this paper, the main contributions include the MSGFE and GT modules. The MSGFE module includes a multi-scale attention (MSA) module and a geometric feature extraction (GFE) module. To evaluate the effectiveness of these modules, we conducted experiments with different combinations. The experiments were validated on S3DIS Area 5, and the quantitative results are shown in Table 5. The baseline model performs the worst without the consideration of local geometric structure information. After introducing the MSGFE module, the mIoU increased by 4.2% and OA increased by 2.61%. The performance decreases when the MSA module is removed from the MSGFE module, which indicates that multi-scale attention was applied to the output features encoded by the topological relationships of geometric structures effectively. After introducing the GT module, the mIoU increased by 2.66% and OA increased by 1.1%, demonstrating the effectiveness of integrating global information into the model. The model achieved the highest performance when all three modules were introduced simultaneously. Compared to the baseline, our mIoU increased by 4.77%, OA by 2.12%, and mAcc by 4.47%. These results indicate that each module had a positive contribution.

Table 5. Ablation study of our core modules.

Model	MSGFE		GT	OA	mAcc	mIoU
	GFE	MSA				
1	×	×	×	87.99	71.80	63.84
2	✓	×	×	89.21	73.56	65.54
3	×	×	✓	89.09	74.12	66.50
4	✓	✓	×	89.6	75.53	68.04
5	✓	×	✓	89.79	75.54	68.02
6	✓	✓	✓	90.11	76.27	68.61

The confusion matrix shown in Figure 7 demonstrates the difference in segmentation accuracy between the baseline model and our model with clear outperformances.

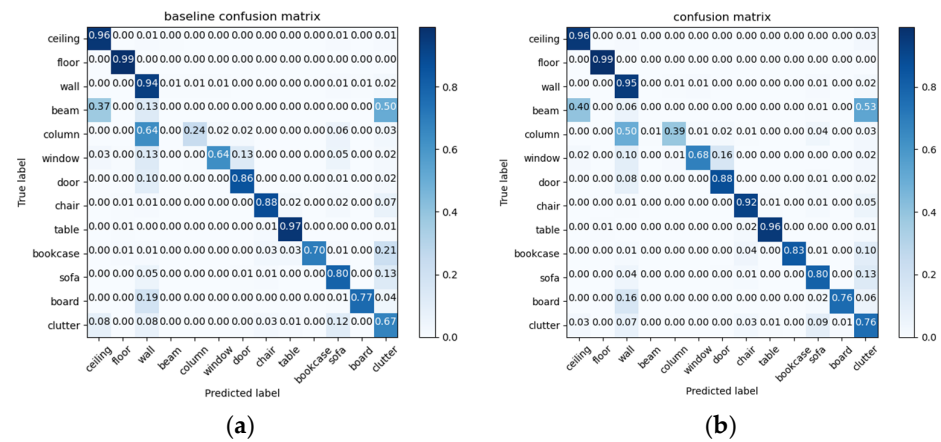


Figure 7. Confusion matrix on S3DIS Area 5 with the comparison of the baseline (a) and our PointMSGT (b). The confusion matrix indicates that the overall performance of our proposed method is superior to that of the baseline.

The results show that our model achieved significant improvements in recognizing categories like columns, bookshelves, and clutter, and it provides a clearer distinction between columns and walls. This indicates that our model can effectively reduce confusion from semantically different points when distinguishing structurally similar but semantically distinct objects. As shown in Figure 8, we highlighted the improved regions compared to the baseline in red, with the remaining regions highlighted in green. The improvements show that our model demonstrates enhanced performance in semantic segmentation compared to the baseline. As shown in the improvement column, our method achieves excellent segmentation results with clearer boundaries in highly similar areas (e.g., tables and chairs) and densely categorized areas (e.g., clutter, bookshelves, and walls). In contrast, the baseline produces visually blurred results in categories such as clutter, columns, and tables. Our model's enhanced ability to distinguish between semantically similar categories such as columns and walls is particularly noteworthy, showing superior performance in handling subtle differences in complex indoor scenes.

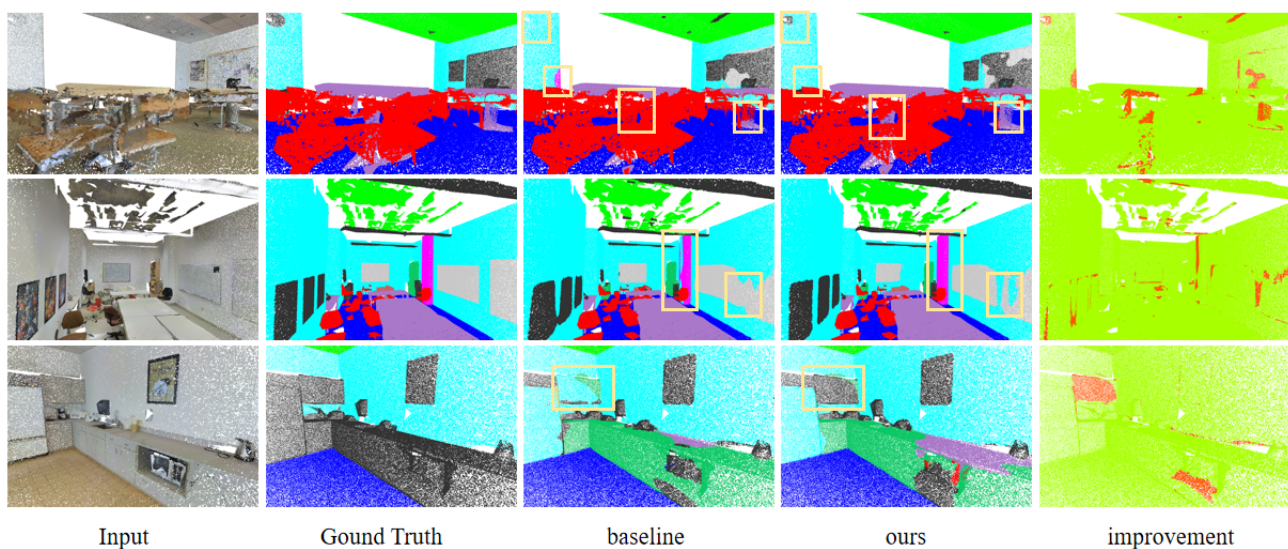


Figure 8. Visual comparison of baseline and our method on S3DIS. The yellow rectangular boxes indicate the contrasting regions.

4.3.2. Selection of the Number of Neighborhood Points K

This experiment analyzes the number of neighborhoods set in the MSGFE module. The number of neighborhoods utilized in the K-nearest neighbor (KNN) algorithm determines the range of the centroid's neighborhood. The results of this experiment are presented in Table 6, which illustrates the outcomes of tests conducted with varying K values on the S3DIS and ScanObjectNN datasets. The results indicate that $K = 8$ achieved the best performance in point cloud semantic segmentation tasks, while $K = 6$ performed best in shape classification tasks. When K is set to 4, the fewer neighborhood points result in less local point cloud spatial topology information being extracted, which reduces segmentation accuracy. As K increases to 10, the accuracy fluctuates in different directions. This is because as the number of neighborhood points increases, the information obtained also increases, but the feature differences between similar types of objects become less distinct, causing segmentation accuracy to fluctuate both upward and downward.

Table 6. Ablation study of our core modules.

k	S3DIS			ScanObjectNN	
	OA	mAcc	mIoU	OA	mAcc
4	89.74	75.38	67.99	85.88	84.42
6	89.59	74.77	67.21	86.4	85
8	90.11	76.27	68.61	86.09	84.49
10	89.67	75.07	67.69	85.39	83.48
12	89.71	75.14	67.67	85.57	83.92

4.3.3. Ablation of MSA Module

The purpose of our MSA module is to compute multi-scale attention maps using convolution kernels of different sizes, and then fuse these maps to strengthen attention to channel features at different scales, enhancing the model's ability to process complex geometric structures in point clouds. To examine the performance of our proposed MSA module, we performed ablation experiments on the kernel sizes and numbers: (a1) using one 3×3 kernel, (a2) using 1×1 , 3×3 and 5×5 kernels, and (a3) using 3×3 , 5×5 , and 7×7 kernels. As indicated in Table 7, a3 significantly outperforms both a1 and a2, and a2 performs better than a1. A visualization of models at various scales was conducted, as depicted in Figure 9. The geometric structure predicted by a3 is much more precise, whereas a1 and a2 show unclear edge segmentation, which makes them more prone to confusion with other categories and unable to demonstrate clear geometric structures. This advantage arises from the fact that multi-scale information assists the model in better understanding features at different levels. Compared to using a single convolution kernel, multiple convolutions expand the receptive field for triangular geometric features, aggregate features of multiple local triangles, and achieve richer local geometric descriptions. By adaptively aggregating features of different scales in the local neighborhood, it further reduces redundant information, thus enabling effective learning of the local details in point clouds. The combination of 3×3 , 5×5 , and 7×7 convolution kernels delivers the best performance, demonstrating that it spans different scales from small to large, enabling better capture of multi-level features in point cloud data.

Table 7. Ablation study of convolution kernels.

Model	OA	mAcc	mIoU
a1	89.49	74.35	66.97
a2	89.90	75.03	67.71
a3	89.60	75.53	68.04

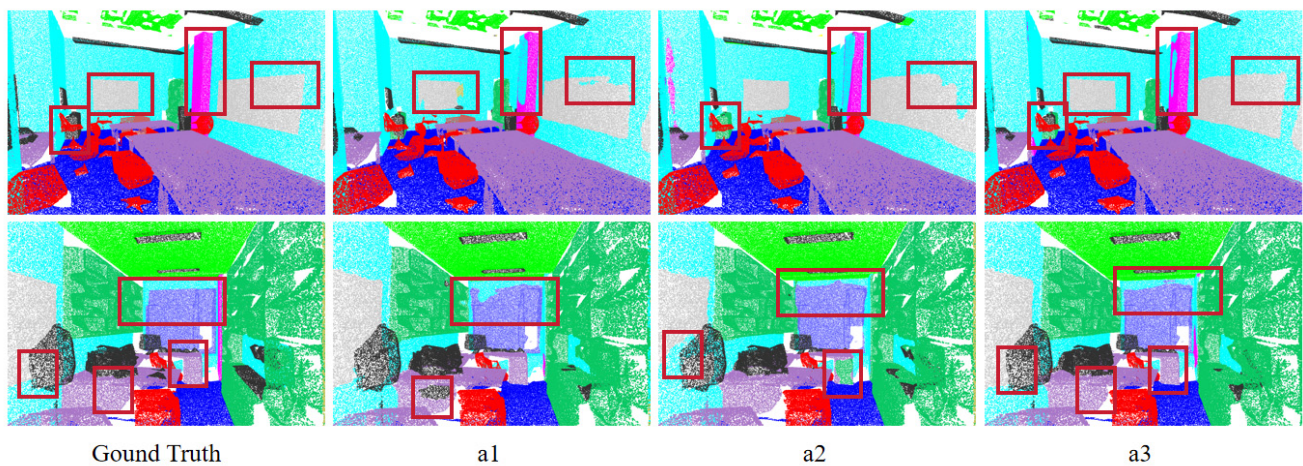


Figure 9. Ablation study of convolution kernels. The red rectangular boxes indicate the contrasting regions.

4.3.4. Ablation Studies of Different Pooling Methods

Our MSGFE module uses sum pooling to aggregate the features of neighboring points, forming a comprehensive local feature representation. This comprehensive feature representation can better capture the information of local geometric structures, thereby improving the model's performance. We conducted the following ablation experiments on feature aggregation using different pooling methods: max pooling, average pooling, and sum pooling. As shown in Table 8, sum pooling achieves the best performance. This may be because summing the features of neighboring points preserves all their information. This means that even neighboring points with smaller but important feature values are not ignored. In contrast, max pooling only retains the maximum values, while average pooling smooths out some feature values. Both methods may ignore important detailed information. Sum pooling retains more information, which helps the model learn more stably during training, thereby improving the model's generalization ability and robustness.

Table 8. Ablation studies of different pooling methods.

Pooling	OA	mACC	mIoU
max	90	75.36	68.22
avg	89.53	75.85	68.14
sum	90.11	76.27	68.61

5. Discussion

We propose PointMSGT, a method tailored for indoor point cloud processing, which achieves competitive results on the large-scale indoor dataset S3DIS and the real-world indoor dataset ScanObjectNN. Owing to the characteristics of indoor point clouds—high structural complexity and geometric similarity—the MSGFE module effectively extracts local geometric information of indoor objects, enabling the differentiation of geometrically similar objects, such as chairs and tables. The MSGFE module is specifically engineered to capture fine-grained geometric features, rendering it highly effective for semantic segmentation and object classification tasks. However, its dependence on triangular representations may necessitate adaptation for outdoor or highly sparse point clouds. Outdoor point clouds often feature richer object categories, significant density variations across regions, and larger data volumes. Sparse point clouds may distort triangular structures, hindering the MSGFE's ability to extract meaningful geometric features. Leveraging its adaptive feature aggregation via the MSA module, PointMSGT demonstrates scalability to larger-scale point clouds or datasets across various domains, effectively managing background points in real-world scenarios.

We assessed the computational complexity of our model compared to other advanced point cloud semantic segmentation methods on the S3DIS dataset. While our model contains fewer parameters than other advanced models, it has a substantially higher parameter count compared to the baseline. In future research, we aim to achieve higher performance with a reduced parameter count.

6. Conclusions

In this study, the PointMSGT model was introduced to improve indoor point cloud classification and segmentation by integrating local geometric information and global contextual information. The GFE module effectively captures local geometric structures by reconstructing the spatial topology of the point cloud through triangular surfaces. The MSA module improves the model's ability to handle complex geometric structures by selectively emphasizing key features at different scales. In addition, the global Transformer can capture long-range dependencies across the entire point cloud, overcoming the challenge of indoor point cloud objects with similar geometric structures but different semantics.

The PointMSGT has demonstrated superior performance on both point cloud classification and semantic segmentation tasks of indoor real scenes. Experimental results show that our model outperforms recent advanced models, achieving high accuracy and robustness in various categories. The ablation studies confirm the effectiveness of the GFE, MSA, and global Transformer modules, showing improvements in the model's capability to process both local and global features clearly.

Future work will emphasize optimizing computational efficiency, investigating the model's performance in weakly supervised learning for indoor point clouds, and extending its application to other environments, including outdoor scenes and urban roads.

Author Contributions: Conceptualization, Y.C.; Methodology, Y.C.; Validation, Y.C. and Y.X.; Investigation, Y.C.; Resources, Y.X., H.W. and D.L.; Data curation, H.W. and D.L.; Writing—original draft, Y.C.; Writing—review & editing, C.C. and D.L.; Visualization, Y.C.; Supervision, C.C. and D.L.; Project administration, D.L.; Funding acquisition, C.C. and D.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported in part by the project of State Grid Zhejiang Electric Power Company (5211HZ220007).

Data Availability Statement: The Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset can be found here: <http://buildingparser.stanford.edu/dataset.html> (accessed on 7 June 2024). The ScanObjectNN dataset can be found here: <https://hkust-vgd.github.io/scanobjectnn/> (accessed on 1 July 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, Y.; Zhang, S.; Han, T.; Du, Y.; Zhang, W.; Li, J. Chat3D: Interactive Understanding 3D Scene-Level Point Clouds by Chatting with Foundation Model for Urban Ecological Construction. *ISPRS J. Photogramm. Remote Sens.* **2024**, *212*, 181–192. [CrossRef]
2. Su, Y.; Liu, W.; Yuan, Z.; Cheng, M.; Zhang, Z.; Shen, X.; Wang, C. DLA-Net: Learning Dual Local Attention Features for Semantic Segmentation of Large-Scale Building Facade Point Clouds. *Pattern Recognit.* **2022**, *123*, 108372. [CrossRef]
3. Hu, D.; Gan, V.J.L.; Yin, C. Robot-Assisted Mobile Scanning for Automated 3D Reconstruction and Point Cloud Semantic Segmentation of Building Interiors. *Autom. Constr.* **2023**, *152*, 104949. [CrossRef]
4. Li, B.; Wang, X.; Gao, Q.; Song, Z.; Zou, C.; Liu, S. A 3D Scene Information Enhancement Method Applied in Augmented Reality. *Electronics* **2022**, *11*, 4123. [CrossRef]
5. Zhang, Z.; Liu, J.; Jiang, G. Spatial and Temporal Awareness Network for Semantic Segmentation on Automotive Radar Point Cloud. *IEEE Trans. Intell. Veh.* **2024**, *9*, 3520–3530. [CrossRef]
6. Xian, G.; Ji, C.; Zhou, L.; Chen, G.; Zhang, J.; Li, B.; Xue, X.; Pu, J. Location-Guided LiDAR-Based Panoptic Segmentation for Autonomous Driving. *IEEE Trans. Intell. Veh.* **2023**, *8*, 1473–1483. [CrossRef]
7. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
8. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; Volume 30.

9. Foorginejad, A.; Khalili, K. Umbrella Curvature: A New Curvature Estimation Method for Point Clouds. *Procedia Technol.* **2014**, *12*, 347–352. [\[CrossRef\]](#)
10. Ran, H.; Liu, J.; Wang, C. Surface representation for point clouds. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 18920–18930.
11. Zhou, C.; Ling, Q. GAF-Net: Geometric Contextual Feature Aggregation and Adaptive Fusion for Large-Scale Point Cloud Semantic Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5705815. [\[CrossRef\]](#)
12. Wang, J.; Cheng, Z.; Zhao, N.; Cheng, J.; Yang, X. On-the-fly Point Feature Representation for Point Clouds Analysis. In Proceedings of the 32nd ACM International Conference on Multimedia (MM '24), Melbourne, VIC, Australia, 28 October–1 November 2024; ACM: New York, NY, USA, 2024. 10p. [\[CrossRef\]](#)
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 15.
14. Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R.R.; Hu, S.-M. PCT: Point Cloud Transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [\[CrossRef\]](#)
15. Wang, P.-S. OctFormer: Octree-Based Transformers for 3D Point Clouds. *ACM Trans. Graph.* **2023**, *42*, 155. [\[CrossRef\]](#)
16. Li, S.; Li, H. Regional-to-Local Point-Voxel Transformer for Large-Scale Indoor 3D Point Cloud Semantic Segmentation. *Remote Sens.* **2023**, *15*, 4832. [\[CrossRef\]](#)
17. Lai, X.; Liu, J.; Jiang, L.; Wang, L.; Zhao, H.; Liu, S.; Qi, X.; Jia, J. Stratified Transformer for 3D Point Cloud Segmentation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 8490–8499. [\[CrossRef\]](#)
18. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.; Koltun, V. Point Transformer. *arXiv* **2021**, arXiv:2012.09164.
19. Wu, X.; Lao, Y.; Jiang, L.; Liu, X.; Zhao, H. Point Transformer V2: Grouped Vector Attention and Partition-Based Pooling. In *Advances in Neural Information Processing of the Systems 35, 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, New Orleans, LA, USA, 28 November–9 December 2022; Curran Associates, Inc.: Red Hook, NY, USA, 2023.
20. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution On X-Transformed Points. In *Advances in Neural Information Processing Systems 31, Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, Montreal, QC, Canada, 3–8 December 2018; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 31.
21. Thomas, H.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6411–6420.
22. Zhao, H.; Jiang, L.; Fu, C.-W.; Jia, J. PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; IEEE: Long Beach, CA, USA, 2019; pp. 5560–5568.
23. Ma, X.; Qin, C.; You, H.; Ran, H.; Fu, Y. Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework. *arXiv* **2022**, arXiv:2202.07123.
24. Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.A.A.K.; Elhoseiny, M.; Ghanem, B. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 23192–23204.
25. Huang, Q.; Wang, W.; Neumann, U. Recurrent Slice Networks for 3D Segmentation of Point Clouds. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2626–2635.
26. Jiang, M.; Wu, Y.; Zhao, T.; Zhao, Z.; Lu, C. PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation. *arXiv* **2018**, arXiv:1807.00652.
27. Qiu, S.; Anwar, S.; Barnes, N. Geometric Back-Projection Network for Point Cloud Classification. *IEEE Trans. Multimed.* **2022**, *24*, 1943–1955. [\[CrossRef\]](#)
28. Fan, S.; Dong, Q.; Zhu, F.; Lv, Y.; Ye, P.; Wang, F.-Y. SCFNet: Learning spatial contextual features for large-scale point cloud segmentation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 14499–14508.
29. Xiang, T.; Zhang, C.; Song, Y.; Yu, J.; Cai, W. Walk in the cloud: Learning curves for point clouds shape analysis. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 915–924.
30. Zhou, W.; Jiao, J.; Wei, M.; Nei, W.; Xu, H. SRX Net: A Point Cloud Segmentation Framework Based on Surface Representation and X-Net. *Comput. Graph.* **2024**, *120*, 103923. [\[CrossRef\]](#)
31. Sun, S.; Rao, Y.; Lu, J.; Yan, H. X-3D: Explicit 3D Structure Modeling for Point Cloud Recognition. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024.
32. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11108–11117.
33. Feng, M.; Zhang, L.; Lin, X.; Gilani, S.Z.; Mian, A. Point Attention Network for Semantic Segmentation of 3D Point Clouds. *Pattern Recognit.* **2020**, *107*, 107446. [\[CrossRef\]](#)
34. Wang, G.; Zhai, Q.; Liu, H. Cross Self-Attention Network for 3D Point Cloud. *Knowl.-Based Syst.* **2022**, *247*, 108769. [\[CrossRef\]](#)

35. Zhu, Q.; Fan, L.; Weng, N. Advancements in Point Cloud Data Augmentation for Deep Learning: A Survey. *Pattern Recognit.* **2024**, *153*, 110532. [\[CrossRef\]](#)
36. Wei, M.; Wei, Z.; Zhou, H.; Hu, F.; Si, H.; Chen, Z.; Zhu, Z.; Qiu, J.; Yan, X.; Guo, Y.; et al. AGConv: Adaptive Graph Convolution on 3D Point Clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 9374–9392. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Zhang, J.; Jiang, Z.; Qiu, Q.; Liu, Z. TCFAP-Net: Transformer-Based Cross-Feature Fusion and Adaptive Perception Network for Large-Scale Point Cloud Semantic Segmentation. *Pattern Recognit.* **2024**, *154*, 110630. [\[CrossRef\]](#)
38. Zeng, Z.; Xu, Y.; Xie, Z.; Tang, W.; Wan, J.; Wu, W. Large-Scale Point Cloud Semantic Segmentation via Local Perception and Global Descriptor Vector. *Expert Syst. Appl.* **2024**, *246*, 123269. [\[CrossRef\]](#)
39. Zhou, C.; Shu, Z.; Shi, L.; Ling, Q. Semantic Segmentation for Large-Scale Point Clouds Based on Hybrid Attention and Dynamic Fusion. *Pattern Recognit.* **2024**, *156*, 110798. [\[CrossRef\]](#)
40. Liu, T.; Ma, T.; Du, P.; Li, D. Semantic Segmentation of Large-Scale Point Cloud Scenes via Dual Neighborhood Feature and Global Spatial-Aware. *Int. J. Appl. Earth Obs. Geoinf.* **2024**, *129*, 103862. [\[CrossRef\]](#)
41. Wang, L.; Liu, Y.; Zhang, S.; Yan, J.; Tao, P. Structure-Aware Convolution for 3D Point Cloud Classification and Segmentation. *Remote. Sens.* **2020**, *12*, 634. [\[CrossRef\]](#)
42. Jiang, C.; Huang, K.; Wu, J.; Wang, X.; Xiao, J.; Hussain, A. PointGS: Bridging and Fusing Geometric and Semantic Space for 3D Point Cloud Analysis. *Inf. Fusion* **2023**, *91*, 316–326. [\[CrossRef\]](#)
43. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* **2019**, *38*, 146. [\[CrossRef\]](#)
44. Cheng, S.; Chen, X.; He, X.; Liu, Z.; Bai, X. PRA-Net: Point Relation-Aware Network for 3D Point Cloud Analysis. *IEEE Trans. on Image Process.* **2021**, *30*, 4436–4448. [\[CrossRef\]](#)
45. Pang, Y.; Wang, W.; Tay, F.E.; Liu, W.; Tian, Y.; Yuan, L. Masked Autoencoders for Point Cloud Self-supervised Learning. In *Computer Vision—ECCV 2022*; Springer: Cham, Switzerland, 2022.
46. Berg, A.; Oskarsson, M.; O'Connor, M. Points to Patches: Enabling the Use of Self-Attention for 3D Shape Recognition. In *Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR)*, Montreal, QC, Canada, 21–25 August 2022; pp. 528–534. [\[CrossRef\]](#)
47. Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; Lu, J. Point-BERT: Pre-training 3D Point Cloud Transformers with Masked Point Modeling. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 18–24 June 2022; pp. 19291–19300.
48. Lu, D.; Gao, K.; Xie, Q.; Xu, L.; Li, J. 3DGTN: 3-D Dual-Attention GLocal Transformer Network for Point Cloud Classification and Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5704313. [\[CrossRef\]](#)
49. Chu, X.; Zhao, S. Adaptive Guided Convolution Generated With Spatial Relationships for Point Clouds Analysis. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5700412. [\[CrossRef\]](#)
50. Ben-Shabat, Y.; Lindenbaum, M.; Fischer, A. 3DmFV: Three-Dimensional Point Cloud Classification in Real-Time Using Convolutional Neural Networks. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3145–3152. [\[CrossRef\]](#)
51. Xu, Y.; Fan, T.; Xu, M.; Zeng, L.; Qiao, Y. SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters. In *Proceedings of the Computer Vision—ECCV 2018*, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 90–105.
52. Uy, M.A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; Yeung, S.-K. Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Republic of Korea, 27 October–3 November 2019; pp. 1588–1597.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.