Article

Large-Scale Point Cloud Semantic Segmentation with Density-Based Grid Decimation

Liangcun Jiang , Jiacheng Ma , Han Zhou , Boyi Shangguan * , Hongyu Xiao , Zeqiang Chen

Posted Date: 12 May 2025

doi: 10.20944/preprints202505.0798.v1

Keywords: Deep learning; point cloud compression; semantic segmentation

Article

Large-Scale Point Cloud Semantic Segmentation with Density-Based Grid Decimation

Liangcun Jiang [1], Jiacheng Ma [1], Han Zhou [1], Boyi Shangguan [2,*], Hongyu Xiao [3] and Zeqiang Chen [4]

[1] Resources and Environmental Engineering, Wuhan University of Technology, Wuhan 430070, China.
[2] The State Key Laboratory of Space-Ground Integrated Information Technology, Space Star Technology Co., Ltd., Beijing, China.
[3] Changjiang Schinta Software Technology Co.Ltd., Wuhan 430010, China.
[4] The National Engineering Research Center of Geographic Information System, China University of Geosciences, Wuhan 430074, China.
* Correspondence: boyishangguan_cast@163.com.

Abstract: Precise segmentation of point clouds into categories such as roads, buildings, and trees is critical for applications in 3D reconstruction and autonomous driving. However, large-scale point cloud segmentation faces challenges such as uneven density distribution, sampling inefficiencies, and limited feature extraction capabilities. To address these issues, this paper proposes RT-Net, a novel framework that incorporates a density-based grid decimation algorithm for efficient preprocessing of outdoor point clouds. This algorithm mitigates uneven density distribution and enhances computational efficiency. RT-Net also introduces two innovative modules: Local Attention Aggregation, which captures fine-grained local features, and Attention Residual, which improves feature propagation. These modules boost segmentation performance, particularly for small objects like poles and signs. Experimental results on the Toronto3D, Semantic3D, and SemanticKITTI datasets demonstrate the superiority of RT-Net, achieving state-of-the-art mean Intersection over Union (mIoU) scores of 86.79% on Toronto3D and 79.88% on Semantic3D, while also showing substantial improvements in small object segmentation.

Keywords: Deep learning; point cloud compression; semantic segmentation

1. Introduction

Point clouds, serving as a fundamental geospatial data structure [1], play an import role in 3D perception and understanding. Accurate semantic segmentation of point clouds into distinct entities like roads, buildings, and trees is vital for applications such as 3D reconstruction or autonomous driving. Traditional techniques for the semantic segmentation of point clouds include region growth-based segmentation [2], model fitting-based segmentation [3], graph optimization-based segmentation [4], and edge information-based segmentation [5]. While these methods are reliable and widely used in commercial applications, they often encounter challenges when processing extensive point cloud data.

The past several years have witnessed significant advancements in semantic segmentation algorithms rooted in deep learning, offering promising solutions for handling extensive earth observation data. Notable developments in point cloud semantic segmentation domain include Point-Net [1], Point-Net++ [6], Point-CNN [7], DGCNN [8], Point-RNN [9], Point Transformer [10]. However, the scope of many such methods is confined to smaller point cloud datasets, a limitation partly attributed to their dependence on time-consuming or inefficient point sampling methods. For example, using the farthest point sampling (FPS) algorithm for sampling 10% of points from a 1 million point cloud takes over 200 seconds [6]. In contrast, RandLA-Net introduced by [11], incorporates the Random Sampling (RS) algorithm, which exhibited remarkable efficiency, completing the same sampling task in just 0.004 seconds. Subsequently, many works [13,14,15] have adopted RS algorithm during the down-sampling processes in their networks. This substantial decrease in sampling time highlights the potential of the RS algorithm in optimizing semantic segmentation networks for large-scale outdoor point cloud data.

Large-scale outdoor point cloud datasets often contain millions or even billions of individual points, making it computationally infeasible to process them in their entirety. Therefore, preprocessing before network training is crucial to reduce the computational burden by selecting representative subsets of points for analysis. While grid-based decimation is a commonly employed preprocessing technique [13,14,15,16,17], it fails to address the uneven distribution of object categories within large-scale point clouds, primarily due to its use of a fixed grid size. This limitation often leads to unsatisfactory segmentation results, especially for small-sized objects. To address variations in point cloud density across different outdoor scenes, we propose a density-based grid decimation algorithm that dynamically adjusts grid size based on the density of input point clouds.

Accurate semantic segmentation of point clouds also depends on robust feature extraction from their complex structures. This involves developing innovative architectures that can effectively down-sample large-scale point clouds while preserving important spatial structures and semantic information. Recent studies, such as those by [10,18,19], have shown that Transformer-based models excel in point cloud semantic segmentation, highlighting their remarkable feature recognition and extraction capabilities. However, these Transformer-based methods can result in significant computational overhead and require considerable memory resources, especially when processing large point batches. Thus, developing a network architecture that balances feature extraction capabilities with GPU constraints remains a key focus.

To address the challenges of uneven point density, imbalanced sampling, and limited feature extraction capabilities in large-scale outdoor point clouds, we propose a novel framework, RT-Net, that leverages a density-based grid decimation algorithm as its foundation. This decimation algorithm dynamically adjusts grid sizes based on point cloud density, ensuring balanced representation of object categories and improving computational efficiency. Further, RT-Net incorporates advanced attention-based modules to enhance feature extraction from both local and global structures. These innovations enable RT-Net to achieve superior segmentation performance, particularly for small-sized object categories, while maintaining computational feasibility on modern GPUs. The main contributions of this paper are as follows:

- **Density-Based Grid Decimation Algorithm:** A novel preprocessing method that dynamically adjusts grid sizes based on point density, addressing imbalanced sampling and improving computational efficiency compared to traditional grid-based approaches.
- **Attention-Based Modules:** Two new modules—Local Attention Aggregation (LAA) and Attention Residual (AR)—are designed to efficiently capture both local and global features, reducing memory consumption and computational overhead.
- **RT-Net Architecture:** A new segmentation network that integrates the density-based grid decimation algorithm with the proposed attention-based modules. RT-Net outperforms existing benchmarks, especially in segmenting small-sized object categories, as demonstrated on large-scale datasets.

2. Related Works

2.1. Deep Learning Methods for Extracting Point Cloud Features

Deep learning-based methods for extracting features from point clouds are mainly categorized into projection-based, voxel-based, and point-based strategies. The projection-based method converts 3D point clouds into multiple 2D views to leverage established 2D convolutional neural networks [15,20,21,22,23], but they face challenges like occlusion and overlap. Voxel-based approaches involves converting point clouds into 3D voxel grids, allowing for the application of 3D convolutions for feature extraction [24,25,26,27,28], yet they often lead to data redundancy and increased computational demands, making them less fitting for processing large-scale outdoor point cloud datasets. In contrast, point-based approaches have become increasingly popular for their adaptability to point cloud structures and flexibility for adjustment, as evidenced by the works of [1,6,7,10,12,13,14,16,17,18,19,29,30,31].

The success of Transformer models in both natural language processing and computer vision has spurred their increasing adoption in the domain of point cloud semantic segmentation. Point Transformer, introduced by Zhao [10] is a pioneering study which explores the implementation of Transformer architecture within point cloud semantic segmentation. However, as the model goes deeper, it tends to cause overfitting in the network's feature extraction ability on the majority of point categories. Recognizing these limitations, [18,19] improved the network by optimizing feature extraction modules using various techniques. PReFormer [17] further enhanced self-attention calculations in Transformer, thereby improving memory efficiency and accuracy in segmenting point clouds. These innovations underscore the potential of attention to balance feature extraction capabilities with computational feasibility, making them promising solutions for the semantic segmentation of extensive point cloud datasets.

2.2. Preprocessing of Point Clouds Before Training

Point cloud preprocessing involves a range of processing steps aimed at preparing point clouds for integration into training networks. Given current hardware constraints, direct training on all points within large-scale point clouds is impractical. Consequently, thinning processing is typically employed as a necessary step. Many preprocessing techniques for vast point cloud datasets avoid heuristic or complex mathematical procedures due to their high memory and time consumption. Instead, spatial region filtering [32] and grid-based decimation [11,15,33,34] are preferred alternatives for preprocessing point clouds. Since spatial region filtering relies heavily on empirical knowledge, grid-based decimation is gaining popularity. However, when utilized in large-scale complex point cloud scenes, grid-based decimation often results in uneven point cloud distribution. KPFCNN [35] leverages density-based kernels for convolution processing in point cloud segmentation, but it still faces the challenge of introducing redundant data. To address this, we propose using density-based grid decimation to mitigate the uneven distribution of object categories.

3. Methodology

3.1. Network Architecture

Our network's design follows an encoder-decoder pattern (see Figure 1.). Initially, a fully connected (FC) layer is used to process the input point clouds, allowing the extraction of features for each point. During the encoding phase, every encoding layer incorporates a Local Attention Aggregation (LAA) module along with a RS module. The network uses four encoding layers to progressive down-sample the point size in a sequential manner (from N to N/4, then N/16, N/64, and finally N/256), with N denoting the initial point count. Concurrently, the feature dimensions for each point are expanded following the sequence. The architecture bridges the encoder and decoder with a shared MLP (Multi-Layer Perceptron) featuring a 1x1 convolutional kernel, an activation function, and a normalization layer to encapsulate the contextual information of the point clouds. In the decoding phrase, each decoder layer consists of an Attention Residual (AR) module and an interpolation Up-sampling (US) module. The shape transformations of the point clouds are reversed compared to the encoder. Finally, a pair of FC layers is employed to produce the semantic label predictions.

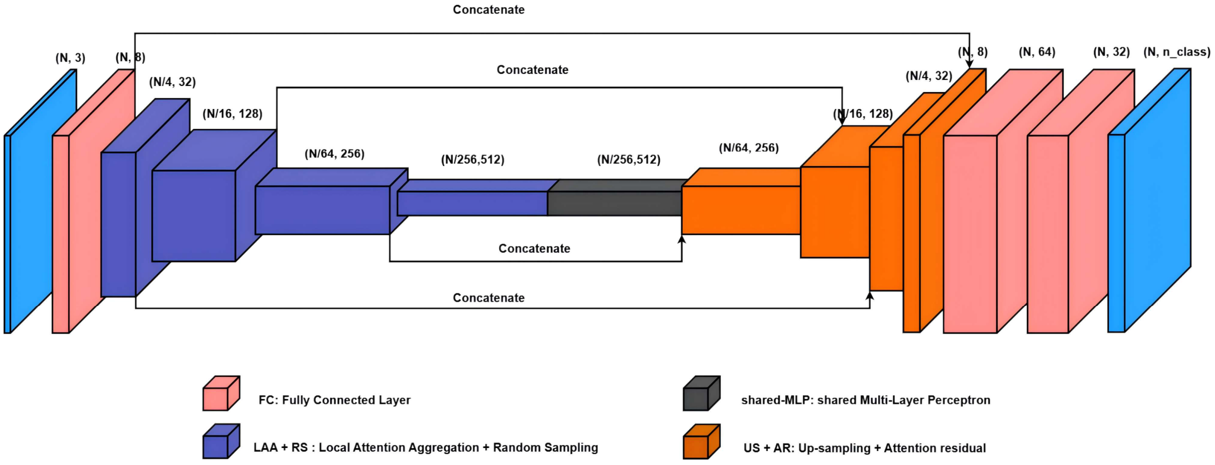

Figure 1. Network architecture of RT-Net. The notation signifies the count of points and feature dimensions, respectively.

3.2. Local Attention Aggregation

Figure 2. details the LAA module. It consists of two units: Local Spatial Encoding and Attention Aggregation. The former unit integrates each point's positional context in relation to others, which enables the latter unit to consider spatial relationships among points during self-attention operations, rather than solely relying on feature similarity to calculate attention coefficients. The latter unit employs a self-attention mechanism to extract internal features from each block of the K-nearest neighbor points. Unlike the LFA module introduced by Hu [30], we have replaced its Attentive Pooling unit with our Attention Aggregation unit, as shown on the right side of Fig.2,

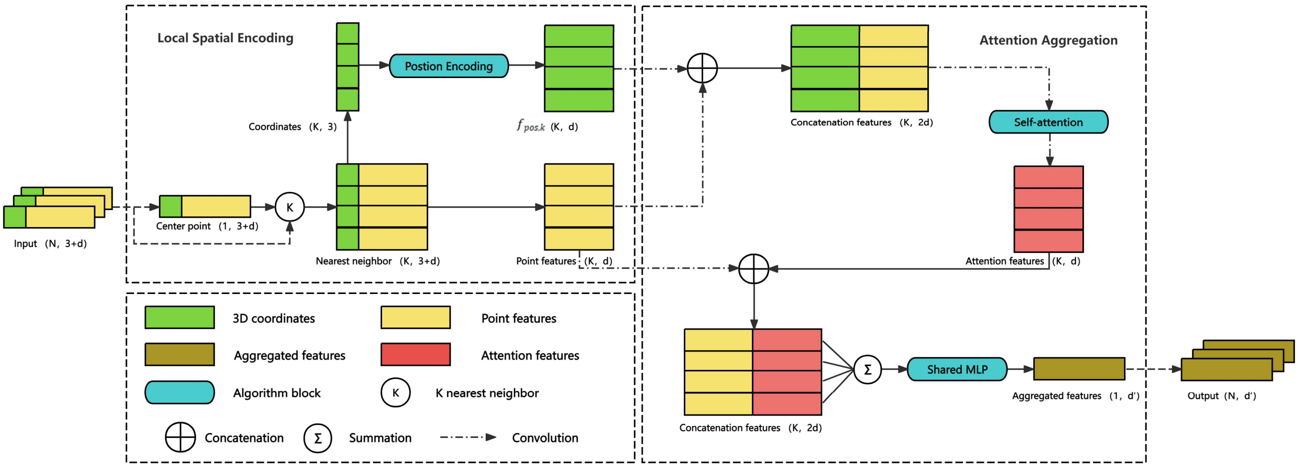

Figure 2. Local Attention Aggregation module.

$$\mathrm{p_{rel,k}} = \mathrm{p_{coor,k}} - \mathrm{p_{coor,i}}, \tag{1}$$

$$\mathrm{dist_k} = ||\mathrm{p_{coor,k}} - \mathrm{p_{coor,i}}||, \tag{2}$$

$$\mathrm{r_k} = \mathrm{p_{coor,k}} \oplus \mathrm{p_{coor,i}} \oplus \mathrm{p_{rel,k}} \oplus \mathrm{dist_k}, \tag{3}$$

$$\mathrm{f_{pos,k}} = \mathrm{Conv}(\mathrm{r_k}). \tag{4}$$

Here, computes the Euclidean distance, represents the operation of concatenation and signifies the convolution operation. In the above equations, is the relative position bias computed from the center point coordinates and the feature point coordinates, and is the corresponding absolute distance. The set combines these values and is passed through a convolutional layer to obtain the local spatial encoding.

The Attention Aggregation unit merges the collection of neighboring features, to derive aggregated features for each center point. It utilizes a self-attention block to expand the receptive field of each point, capturing long-range contexts for better generalization. First, the collection of neighboring features undergo standard self-attention operations, producing attention features of. These attention features are subsequently merged with the neighboring point features of. This operation facilitates a comprehensive capture of local features within point clouds by supplementing them with self-attention features, thus enhancing the overall representation. Finally, the concatenated features are summed and passed through a MLP layer with shared parameters to derive the aggregated features of. Through Local Spatial Encoding and Attention Aggregation units, the input point clouds transform into their corresponding aggregated features that effectively encode local contextual information.

3.3. Attention Residual

As depicted in Fig.1 and Figure 3., the AR module receives inputs from both the preceding layer's output features and the aggregated features generated by the LAA module. First, it conducts up-sampling on the previous layer's output. The up-sampled features are then fed into the Attention block as the Query input. The Key and Value inputs for the Attention block are extracted from the LAA module's aggregated features using a 1×1 convolutional layer. Subsequently, attention features are computed in the Attention block, and their feature dimensions are restored through concatenation with the up-sampled point cloud features. This procedure is devised to accelerate computations and enhance the precision of point cloud feature extraction. Further elaboration on this topic are provided in the Ablation of AR in Section IV. Finally, the concatenated result is transformed by a shared MLP into attention residual features, which serves as the output of the AR module.

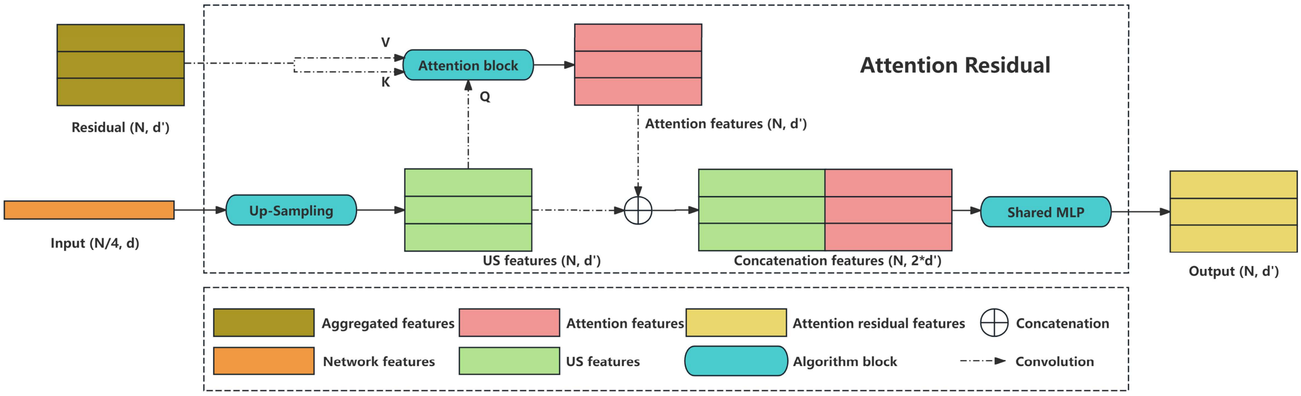

Figure 3. The proposed Attention Residual module. Accept aggregated features and network features as inputs. Attention residual features as module output.

The equations used in this module are represented as follows:

$$\mathrm{AR} = \mathrm{Concatenate}\left(\mathrm{res.}, \mathrm{Soft\ max}\left(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{\mathrm{d}}}\right)\mathrm{V}\right), \quad (1)$$

$$\mathrm{Q}, \mathrm{K}, \mathrm{V} = \mathrm{XW}_{\mathrm{q}}^{\frac{\mathrm{d}}{2}}, \mathrm{XW}_{\mathrm{k}}^{\frac{\mathrm{d}}{2}}, \mathrm{XW}_{\mathrm{v}}^{\frac{\mathrm{d}}{2}}, \quad (1)$$

Here, is the feature of the residual layer. Q stands for the query matrix, K for the key matrix, and V for the value matrix. X represents the point set matrix, while,, are the linear transformation matrices.

3.4. Density-Based Grid Decimation

Given the extensive volume of points within raw point clouds, direct utilization of the original data for model training is considered impractical. Data preprocessing becomes a necessary step before model training. In the original point cloud set, each element is delineated by its three-dimensional coordinates. The sequence of steps involved in density-based grid decimation for point clouds is as follows:

1. Compute the upper and lower bounds of the 3D coordinates in the original point cloud set.
2. Specify the grid size and calculate the grid count per dimension.
3. Determine the three-dimensional grid indices for every point within the point set by their grid index.
4. Classify points in based on the indices calculated in the previous step. Points sharing the same index are grouped into a grid. Unlike traditional grid-based decimation, where the grid size is fixed, our approach considers the density of points and dynamically adjusts the grid size accordingly. If the point count in a grid exceeds a preset threshold, the grid is subdivided into four equal-sized sub-grids. This process repeats until the number of points in all grids falls below the preset threshold. Subsequently, each grid randomly retains one point while discarding the rest, completing the density-based grid decimation process, and obtaining

For the sparse point clouds (m is the number of sub-point clouds, automatically generated according to grid-based decimation) resulting from density-based grid decimation processing, they serve as the basis for model training. Utilizing a sampling method based on probability ensures that the data used for training in each epoch is not fixed. This variability arises because the data sampled in each epoch differs, aiding the model in comprehending more intricate scenes. In the experiments section, the data preprocessing process involves simultaneous density-based grid decimation and probability sampling operations on both the training and testing datasets. Additionally, for every point in the original point clouds, the model identifies its closest neighbor in the thinned point clouds. This process allows mapping predictions from the thinned point clouds back to the original for precise performance assessment. Fig.5 shows that, under identical sampling settings, the points sampled using our method are more concentrated and primarily distributed in high-density regions, such as those near roads and buildings. In contrast, conventional grid-based decimation results in more dispersed sampling points.

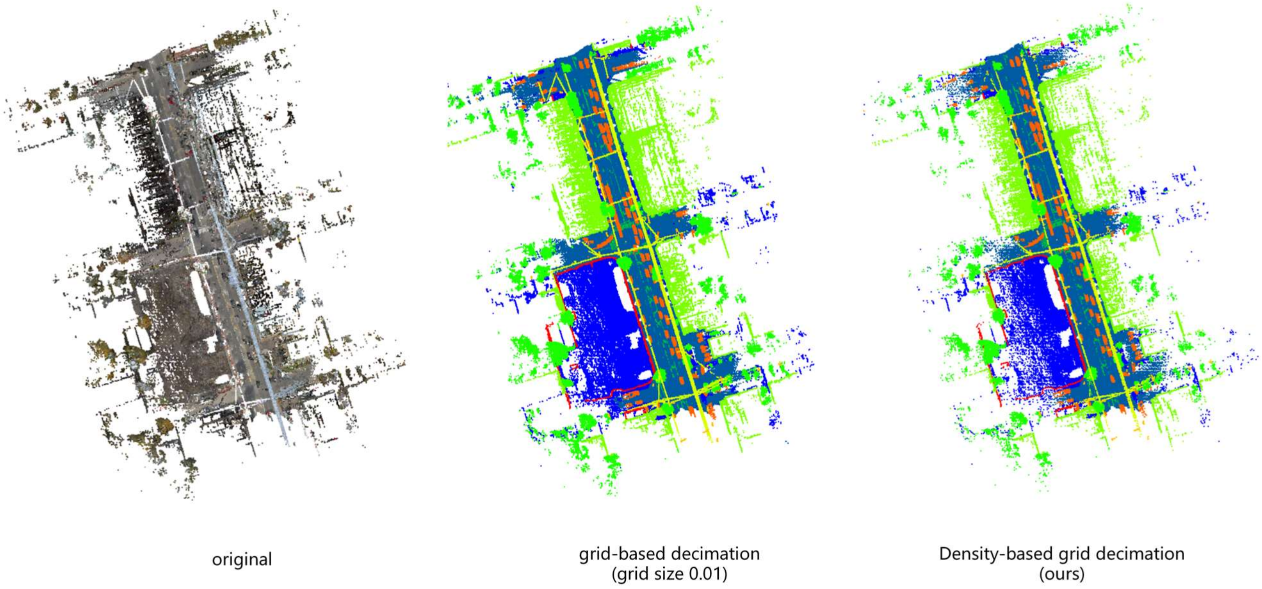

Figure 4. Grid-based decimation and density-based grid decimation on the L001 region of Toronto3D.

4. Experiments and Results

4.1. Density-Based Grid Decimation

In the following experiments, we used three benchmark datasets: Toronto3D [36], Semantic3D [37], and SemanticKITTI [38], applying the same network architecture to them. The initial grid size in the preprocessing phrase was configured at 0.1. An Adam optimizer was employed for training with its default parameter settings. To reduce the influence of class imbalance, we opted for a weighted cross-entropy loss, with class weights calculated as the inverse of their frequency in the training samples. The model was trained for 100 epochs. We initiated the learning rate at 0.01, with each subsequent epoch's rate set to 95% of its predecessor. For the KNN algorithm, we utilized the Kd-Tree module from the Scikit-learn package, setting it to find 16 closest neighbors. All algorithms in this section were implemented using PyTorch 1.13.1 and CUDA 12.4 on Ubuntu 22.04. Our experiments were performed on a NVIDIA GeForce RTX 3090 24G GPU.

4.2. Density-Based Grid Decimation

In this section, we assess the performance of our RT-Net architecture on large-scale point clouds and compare its results with those of state-of-the-art semantic segmentation algorithms. For the Toronto3D dataset, following the practice described by Tan [36], we designated the L002 region as the test subset, with the remaining regions form the training subset. Performance assessment was conducted using eight categories of mean Intersection over Union (mIoU) along with their respective IoU metrics. For the Semantic3D dataset, following the guidelines set by Thomas [35], we assigned two regions as the test subset, with the rest of the regions allocated for training and validation purposes, and applied the same evaluation metrics. For the SemanticKITTI dataset, sequences 00 to 10 are used as the training set, sequence 08 as the validation set, and sequences 11 to 21 as the test set. During the preprocessing stage, infrequent categories are discarded, resulting in 19 categories being used for training and evaluation.

Table 1. Quantitative results on Toronto3D. Bold numbers mark the highest column values.

Methods	mIoU	road	rd mrk.	natural	building	util. line	pole	car	fence
PointNet++ [6]	41.81	89.27	0.00	69.06	54.16	43.78	23.30	52.00	2.95
DGCNN [8]	61.79	93.88	0.00	91.25	80.39	62.40	62.32	88.26	15.81
KPFCNN [35]	69.11	94.62	0.06	96.07	91.51	87.68	81.56	85.66	15.81
TGNet [39]	61.34	93.54	0.00	90.93	81.57	65.26	62.98	88.73	7.85
RandLA-Net [11]	81.77	96.69	64.21	96.62	94.24	88.06	77.84	93.37	42.86
PointNest [40]	74.7	91.0	27.9	96.2	89.5	88.3	78.6	91.1	35.1
MVPNet [12]	84.14	**98.00**	76.36	**97.34**	94.77	**87.69**	84.61	**94.63**	39.74
EyeNet [41]	81.13	96.98	65.02	97.83	93.51	86.77	84.86	94.02	30.01
PReFormer [17]	75.8	96.8	65.4	92.4	84.6	82.0	68.3	85.5	31.2
DG-Net [13]	82.1	97.1	65.3	97.2	92.6	88.1	84.2	93.6	38.7
RandLA-Net (Ours rep.)	76.64	93.10	55.23	94.45	93.35	76.21	73.37	80.24	47.18
RandLA-Net (Ours w/ density-grid)	81.55	94.77	60.85	96.25	**95.31**	80.66	79.28	86.99	54.64
Ours (w/ RGB w/o density-grid)	80.85	94.95	63.72	96.00	95.01	81.30	80.34	86.06	49.42
Ours (w/ RGB and density-grid)	**86.79**	92.28	**81.22**	95.03	89.96	86.97	**90.45**	88.06	**70.29**

Table 1. summarizes the quantitative results for the Toronto3D dataset, presenting mIoU and IoU values for each category. The data from the table indicates that RT-Net, integrated with a density-based grid decimation approach for semantic segmentation on the Toronto3D dataset, surpasses current state-of-the-art models in mIoU. Observations also reveal that density-based grid decimation significantly enhances the segmentation performance of RT-Net across various small-sized object categories, as shown in Table 1. Among these categories, fences demonstrate the most significant improvement, with a 20.87% increase from 49.42% to 70.29%. Road markings, poles, and utility lines have also seen substantial improvements, with increases of 17.5%, 10.11%, and 5.67%, respectively. However, there are slight reductions in the IoU scores for certain categories: roads see a decrease from 94.69% to 92.28%, natural areas from 96.62% to 95.03%, utility lines from 88.06% to 86.97%, and cars from 93.37% to 88.06%, respectively.

It should be noted that our rigorous replication of RandLA-Net's experiment yielded lower mIoU value (76.64%) than the authors' reported figures (81.77%). Despite ensuring consistency in parameter settings according to the original study, discrepancies likely arose due to variations in experimental conditions. In our replication of RandLA-Net, employing the density-based grid decimation method, we realized enhanced outcomes as well, with an increase in performance from 76.64% to 81.55%. As shown in Fig.6, L001 demonstrates the strong segmentation performance of our method on edge points, while L002 highlights its effectiveness in segmenting pole point clouds. Additionally, L003 and L004 illustrate the method's ability to accurately identify ground marks.

Under identical experimental conditions, we conducted semantic segmentation experiments on the Semantic3D dataset. The proposed RT-Net outperforms other semantic segmentation models, as shown in Table 2., reaching a state-of-the-art mIoU score of 79.88%. Specifically, our model achieves remarkable results in the segmentation of high vegetation, hardscape, and cars categories, with IoU scores of 90.35%, 60.48%, and 67.55%, respectively. These results highlight our model's robust segmentation capabilities, particularly in challenging terrains and diverse landscape categories, and indicate its strong generalization potential.

Table 2. Quantitative results on Semantic3D (Semantic-8). Bold numbers mark the highest column values.

Methods	mIoU	man-made.	natural.	high veg.	low veg.	buildings	hard scape	scanning art.	cars
PointNet++ [6]	63.1	81.9	78.1	64.3	51.7	75.9	36.4	43.7	72.6

SPGraph [15]	76.2	91.5	75.6	78.3	71.7	94.4	56.8	52.9	88.4
ConvPoint [42]	76.5	92.1	80.6	76.0	**71.9**	95.6	47.3	61.1	87.7
EdgeConv [43]	64.4	91.1	69.5	65.0	56.0	89.7	30.0	43.8	69.7
RGNet [44]	72.0	86.4	70.3	69.5	68.0	**96.9**	43.4	52.3	89.5
RandLA-Net [11]	77.8	97.4	93.0	70.2	65.2	94.4	49.0	44.7	**92.7**
SCF-Net [9]	77.6	97.1	91.8	86.3	51.2	95.3	50.5	67.9	80.7
CAN [45]	74.7	97.9	**94.1**	70.8	64.3	94.0	48.5	38.8	89.2
LEARD-Net [46]	74.5	97.5	92.7	74.6	61.0	93.2	40.2	44.2	92.2
DCNet [47]	74.1	**97.9**	86.5	72.9	64.6	96.2	48.7	35.3	90.4
LSGRNet [48]	77.5	97.2	91.2	84.4	52.2	94.8	51.6	**70.1**	78.5
RandLA-Net (Ours rep.)	71.80	91.71	86.81	87.51	55.07	91.93	31.26	54.07	76.03
RandLA-Net (Ours w/ density-grid)	76.22	90.13	87.65	87.29	57.96	93.52	55.75	62.84	74.59
Ours (w/ RGB w/o density-grid)	74.58	90.20	86.67	83.34	61.70	92.44	52.76	59.30	70.28
Ours (w/ RGB and density-grid)	**79.88**	92.41	86.91	**90.35**	63.32	95.04	**60.48**	67.55	82.96

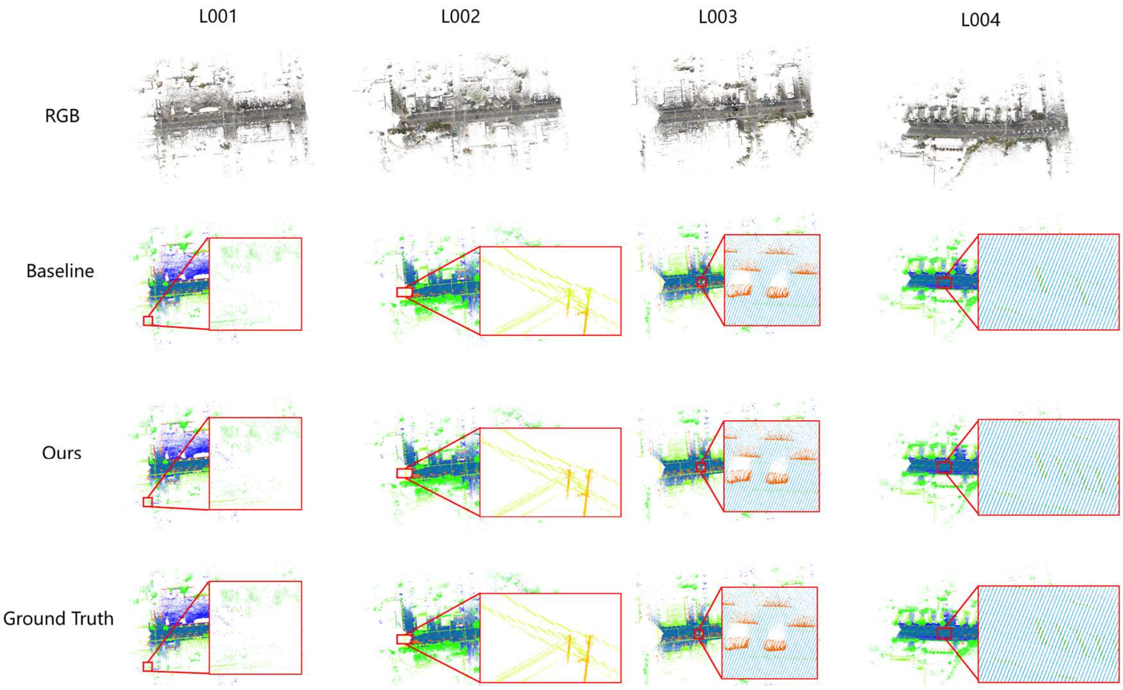

Figure 5. Semantic segmentation results on Toronto3D datasets, compared with RandLA-Net.

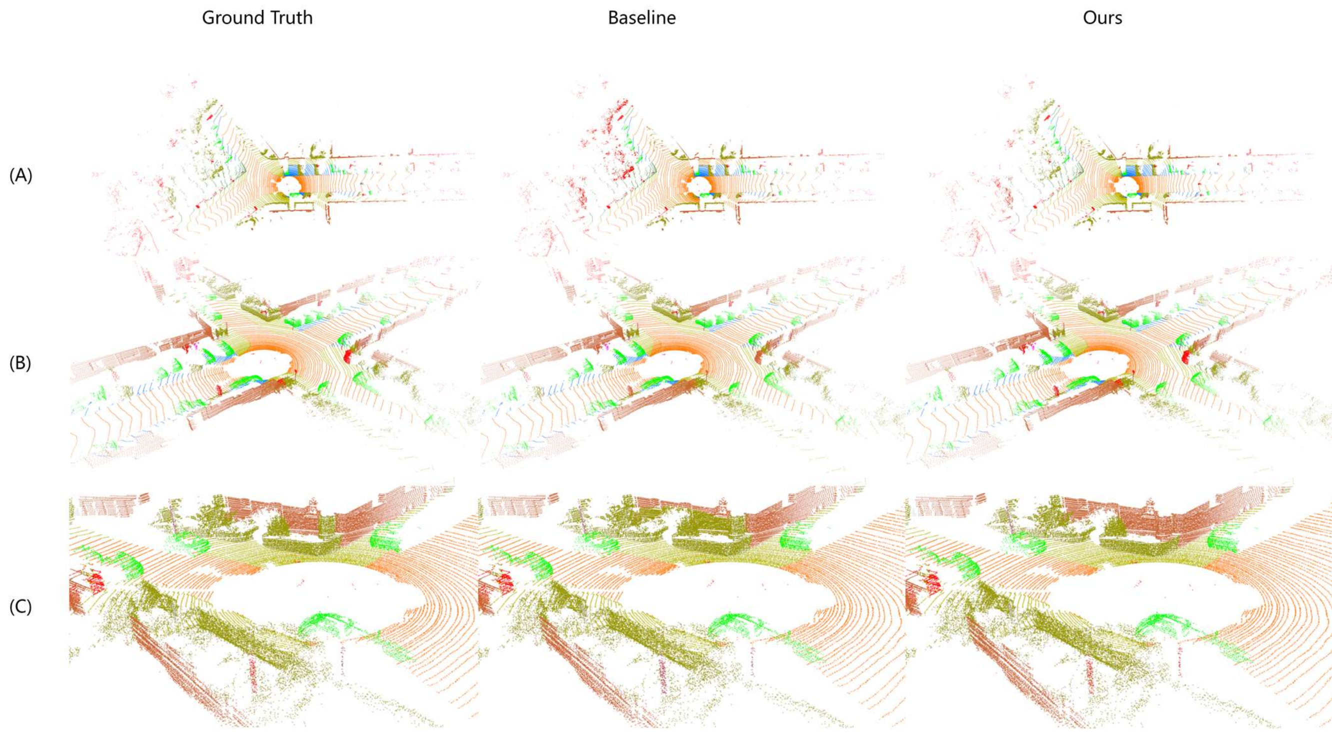

Figure 6. Semantic segmentation results on SemanticKITTI.

Table 3. Quantitative results on SemanticKITTI. Bold numbers mark the highest column values.

Methods	mIoU	Road	Sidewalk	Parking	Other-gro.	Building	Car	Truck.	Bicycle	Motorcycle	Other-veh.	Vegetation	Trunk	Terrain	Person	Bicyclist	Motorcyclist	Fence	Pole	Traffic sign
PointNet++ [6]	20.1	72.0	41.8	18.7	5.6	62.3	53.7	0.9	1.9	0.2	0.2	46.5	13.8	30.0	0.9	1.0	0.0	16.9	6.0	8.9
KPConv [35]	58.8	90.3	72.7	61.3	31.5	90.5	95.0	33.4	30.2	42.5	44.3	84.8	69.2	69.1	61.5	61.6	11.8	64.2	56.4	47.4
RandLA-Net [11]	55.9	90.5	74.0	61.8	24.5	89.7	94.2	43.9	29.8	32.2	39.1	83.8	63.6	68.6	48.4	47.4	9.4	60.4	51.0	50.7
RPVNet [49]	70.3	93.4	80.7	70.3	33.3	93.5	97.6	44.2	68.4	68.7	61.1	86.5	75.1	71.7	75.9	74.4	43.4	72.1	64.8	61.4
PVKD [50]	71.2	91.8	70.9	77.5	41.0	92.4	97.0	67.9	69.3	53.5	60.2	86.5	73.8	71.9	75.1	73.5	50.5	69.4	64.9	65.8
RandLA-Net (Ours rep.)	51.2	88.7	72.4	62.1	22.1	85.1	89.7	38.9	27.6	33.0	33.0	81.1	63.2	66.8	44.6	42.1	8.3	54.8	47.5	45.2
RandLA-Net (Ours	53.6	84.3	80.2	63.3	37.6	91.3	91.7	41.8	45.6	59.2	32.8	84.9	68.5	70.5	64.5	49.9	20.8	68.2	60.4	59.3

w/ density-grid)																				
Ours (w/ RGB w/o density-grid)	70.2	89.9	76.8	59.3	40.1	91.6	96.8	57.9	43.5	53.5	58.8	80.2	72.8	70.9	60.5	66.4	47.3	69.8	61.5	59.8
Ours (w/ RGB and density-grid)	69.9	88.9	86.9	67.4	50.8	93.4	97.4	57.6	72.6	70.2	60.4	83.4	71.9	60.4	52.4	60.8	46.9	80.8	72.8	74.1

Our comparative analysis across different datasets indicates that the results can be attributed to two main factors: the robust feature extraction capabilities of the attention modules and the density-based grid decimation method's proficiency in evenly distributing point cloud categories and preserving fine details.

4.3. Efficiency of Density-Based Grid Decimation

In this section, the efficiency of our density-based grid decimation is assessed against the traditional grid-based method. For density-based grid decimation, the initial grid can choose as large a value as possible according to the span of the actual space coordinates to ensure that there is enough space for calculation, but at the same time, too large a value will lead to a waste of computational efficiency. We conducted a series of four experiments to evaluate their impact on semantic segmentation performance. All the preprocessing approaches were followed by the full RT-Net semantic segmentation network.

Table 4. The mIoU result of RT-Net with different grid size.

Model	mIoU (%)
Full RT-Net with 0.01 grid-based decimation	76.66
Full RT-Net with 0.06 grid-based decimation	80.85
Full RT-Net with 1.0 initial grid density-based grid decimation	86.91
Full RT-Net with 10 initial grid density-based grid decimation	86.79

Table 4. demonstrates our density-based grid decimation method's advantage, with higher mIoU scores over the grid-based method. It is noteworthy that initializing the grid size at 1.0 yielded the best outcomes, with a mIoU score of 86.91%. However, we ultimately opted for density-based grid decimation starting at a grid size of 10. This choice was influenced by the substantial time expenditure of the former solution, which is almost 120 times that of the latter, for only a marginal enhancement in mIoU.

It is worth mentioning that our preprocessing strategy is not limited to RT-Net; it is versatile and can be applied to other networks. This adaptability is evident from successful experiments conducted with RandLA-Net, as indicated in Table 1. The integration of density-based grid decimation into the RandLA-Net methodology also resulted in notable enhancements, particularly for small-sized object categories such as road markings, utility lines, poles, cars, and fences. These enhancements are attributed to the ability of density-based grid decimation to effectively recognize and preserve intricate details within the scene. Consequently, it emerges as an effective preprocessing technique for large-scale point clouds, offering efficiency gains across different semantic segmentation networks.

To validate the efficacy of the modules in RT-Net, we carried out a set of ablation studies concentrating on the LAA and AR modules. These studies were all conducted on the Toronto3D dataset applying density-based grid decimation, with evaluations focused on the L002 region.

4.4. Ablation of RT-Net Framework

The core components of RT-Net framework are the LAA and AR modules, while the remainder of the framework adheres to the design put forward by Hu [11]. To showcase the effectiveness of each component, we designed two key ablation experiments:

- Removal of Self-Attention Pooling: This structure facilitates the aggregation of features from neighboring points in the point clouds. Upon its removal, we replaced it with standard max/mean/sum pooling for the local feature encoding.
- Removal of Attention Residual: This structure enhances the effectiveness of residual adversarial networks by emphasizing feature values through attention mechanisms. Upon its removal, we utilized an original residual connection.

Table 5. presents the mIoU results derived from our network ablation studies. The results underscore the crucial function of the self-attention pooling module in enhancing the network's overall performance. Its absence leads to a notable 21.13% drop in mIoU, stemming from the substantial loss of detailed features during random sampling phrase. On the other hand, the Attention Residual (AR) module is also crucial for the network's overall performance. Its removal is associated with a significant drop in mIoU, amounting to an 11.98% decrease. This indicates that while its impact may be less dramatic than the self-attention pooling module, the AR module is nonetheless a key contributor to the network's effectiveness.

Table 5. The mIoU results of network ablation experiments.

Model	mIoU (%)
Removing self-attention pooling	65.66
Removing attention residual	74.81
The full framework (RT-Net)	86.79

4.5. Ablation of AR

As described in Section III, we formulated attention residuals and subsequently calculated attention residual features. To assess the impact of different residual block structures, we conducted additional ablation experiments, summarized in Table 6.:

- RT-Net with Residual: Utilizes a standard residual connections module.
- RT-Net with Attention Residual: Uses an attention residual connections module with addition.
- RT-Net with Attention Residual (Concatenation): Employs our complete attention residual module with concatenation.

Table 6. The mIoU results of ablation experiments with different residual configurations.

Model	mIoU(%)
RT-Net with residual	74.81
RT-Net with attention residual	79.96
RT-Net with attention residual (concatenation)	83.00

Table 6. displays the impact of different residual configurations on RT-Net's performance. Implementing the attention residual module in RT-Net led to a 5.15% improvement in mIoU scores over the use of the standard residual structure. Furthermore, utilizing the attention residual module with concatenation in RT-Net yielded an additional 3.04% increase in mIoU.

5. Conclusions

This paper presents RT-Net, a novel semantic segmentation framework for large-scale point clouds, harnessing the power of random sampling, attention mechanisms, and density-based grid decimation. RT-Net features innovative local attention aggregation and attention residual modules designed to capture a comprehensive range of features within point clouds, including both local and global characteristics. The integration of a density-based grid decimation algorithm for preprocessing large-scale point clouds addresses the issue of imbalanced sampling categories encountered with traditional preprocessing methods. These innovations enable RT-Net to outperform existing methods, particularly in the segmentation of small-sized object categories such as road markings, utility lines, poles, and fences. Our approach has been rigorously evaluated on the Toronto3D, Semantic3D, and SemanticKITTI datasets, achieving state-of-the-art mIoU scores of 86.79% on Toronto3D and 79.88% on Semantic3D, while demonstrating significant improvements in small object categories across all three benchmark datasets. These results highlight RT-Net's robustness and adaptability to various datasets and scene types.

While RT-Net has demonstrated outstanding performance, there are several promising areas for future work. One such area is the implementation of domain-adaptive point cloud transfer learning. Considering the labor-intensive process of annotating point cloud data, exploring domain-adaptive transfer learning is a crucial next step. This approach, by leveraging existing labeled datasets, could significantly enhance segmentation performance and streamline the learning process across various scenes. Additionally, achieving real-time capabilities in semantic segmentation of point clouds is vital for applications that require rapid analytical and decision-making processes. Further research in this area will not only reinforce RT-Net's practicality in real-world applications but also ensure the network's ongoing adaptability and responsiveness to new data.

Author Contributions: Conceptualization, L.J., J.M. and B.S.; methodology, J.M. and L.J.; validation, J.M., H.Z. and B.S.; formal analysis, J.M. and B.S.; investigation, L.J., H.X., and Z.C.; writing—original draft, L.J., J.M., B.S., Z.C.; visualization, J.M., H.Z. and H.X.; supervision, L.J. and H.Z.; funding acquisition, L.J. and Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Key R&D Program of China (No. 2024YFB3909604) and the National Natural Science Foundation of China (No. 42471446); in part by the Open Fund of National Engineering Research Center of Geographic Information System, China University of Geosciences, Wuhan 430074, China (No. 2023KFJJ10).

Data Availability Statement: Data from Toronto3D can be accessed through https://github.com/WeikaiTan/Toronto-3D?tab=readme-ov-file#download (accessed on 26 April 2025). For access to the data collected from Semantic3D, visit https://www.semantic3d.net/ (accessed on 26 April 2025). While the SemanticKITTI dataset is available for download at https://semantic-kitti.org/ (accessed on 26 April 2025).

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

Abbreviations

The following abbreviations are used in this manuscript:

GPU	Graphics processing unit
LAA	Local attention aggregation

AR	Attention residual
FC	Fully connected
US	Up sampling
MLP	Multi-layer perceptron
KNN	K-nearest neighbor
IoU	Intersection over union
mIoU	Mean intersection over union

References

1. Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Honolulu, HI, July 2017; pp. 77–85, doi: 10.1109/CVPR.2017.16.
2. Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic Point Cloud Interpretation Based on Optimal Neighborhoods, Relevant Features and Efficient Classifiers. ISPRS Journal of Photogrammetry and Remote Sensing 2015, 105, 286–304, doi:10.1016/j.isprsjprs.2015.01.016.
3. Schnabel, R.; Wahl, R.; Klein, R. Efficient RANSAC for Point-Cloud Shape Detection. Computer Graphics Forum 2007, 26, 214–226, doi:10.1111/j.1467-8659.2007.01016.x.
4. Strom, J.; Richardson, A.; Olson, E. Graph-Based Segmentation for Colored 3D Laser Point Clouds. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems; IEEE: Taipei, October 2010; pp. 2131–2136, doi: 10.1109/IROS.2010.5650459.
5. Jiang, X.Y.; Meier, U.; Bunke, H. Fast Range Image Segmentation Using High-Level Segmentation Primitives. In Proceedings of the Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV'96; IEEE Comput. Soc. Press: Sarasota, FL, USA, 1996; pp. 83–88, doi: 10.1109/ACV.1996.572006.
6. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017); Long Beach, CA, USA, 2017; Vol. 30, doi:10.48550/arXiv.1706.02413.
7. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution On X-Transformed Points. In Proceedings of the Advances in Neural Information Processing Systems 31 (NeurIPS 2018); Montréal, Canada, 2018; Vol. 31, doi:10.48550/arXiv.1801.07791.
8. 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, 1–12, doi:10.1145/3326362.
9. Fan, S.; Dong, Q.; Zhu, F.; Lv, Y.; Ye, P.; Wang, F.-Y. SCF-Net: 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); IEEE: Nashville, TN, USA, June 2021; pp. 14499–14508, doi: 10.1109/CVPR46437.2021.01427.
10. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.; Koltun, V. Point Transformer. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV); IEEE: Montreal, QC, Canada, October 2021; pp. 16239–16248, doi: 10.1109/ICCV48922.2021.01595.
11. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. Learning Semantic Segmentation of Large-Scale Point Clouds with Random Sampling. IEEE Trans. Pattern Anal. Mach. Intell. 2021, 1–1, doi:10.1109/TPAMI.2021.3083288.
12. Li, H.; Guan, H.; Ma, L.; Lei, X.; Yu, Y.; Wang, H.; Delavar, M.R.; Li, J. MVPNet: A Multi-Scale Voxel-Point Adaptive Fusion Network for Point Cloud Semantic Segmentation in Urban Scenes. International Journal of Applied Earth Observation and Geoinformation 2023, 122, 103391, doi:10.1016/j.jag.2023.103391.
13. Liu, T.; Ma, T.; Du, P.; Li, D. Semantic Segmentation of Large-Scale Point Cloud Scenes via Dual Neighborhood Feature and Global Spatial-Aware. International Journal of Applied Earth Observation and Geoinformation 2024, 129, 103862, doi:10.1016/j.jag.2024.103862.
14. 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 Systems with Applications 2024, 246, 123269, doi:10.1016/j.eswa.2024.123269.

15. Landrieu, L.; Simonovsky, M. Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; IEEE: Salt Lake City, UT, June 2018; pp. 4558–4567, doi: 10.1109/CVPR.2018.00479.
16. Park, J.; Kim, C.; Kim, S.; Jo, K. PCSCNet: Fast 3D Semantic Segmentation of LiDAR Point Cloud for Autonomous Car Using Point Convolution and Sparse Convolution Network. Expert Systems with Applications 2023, 212, 118815, doi:10.1016/j.eswa.2022.118815.
17. Akwensi, P.H.; Wang, R.; Guo, B. PReFormer: A Memory-Efficient Transformer for Point Cloud Semantic Segmentation. International Journal of Applied Earth Observation and Geoinformation 2024, 128, 103730, doi:10.1016/j.jag.2024.103730.
18. Wu, X.; Lao, Y.; Jiang, L.; Liu, X.; Zhao, H. Point Transformer V2: Grouped Vector Attention and Improved Sampling – Supplementary Material. In Proceedings of the Advances in Neural Information Processing Systems 35 (NeurIPS 2022); 2022; Vol. 35, pp. 33330–33342, doi:10.48550/arXiv.2210.05666.
19. Wu, X.; Jiang, L.; Wang, P.-S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; Zhao, H. Point Transformer V3: Simpler, Faster, Stronger 2024, doi: 10.48550/arXiv.2312.10035.
20. Boulch, A.; Guerry, J.; Saux, B.L.; Audebert, N. SnapNet: 3D Point Cloud Semantic Labeling with 2D Deep Segmentation Networks. Computers & Graphics 2018, 71, 189–198, doi:10.1016/j.cag.2017.11.010.
21. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. PointPillars: Fast Encoders for Object Detection From Point Clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Long Beach, CA, USA, June 2019; pp. 12689–12697, doi: 10.1109/CVPR.2019.01298.
22. Yang, B.; Luo, W.; Urtasun, R. PIXOR: Real-Time 3D Object Detection from Point Clouds. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; IEEE: Salt Lake City, UT, USA, June 2018; pp. 7652–7660, doi: 10.1109/CVPR.2018.00798.
23. Lyu, Y.; Huang, X.; Zhang, Z. EllipsoidNet: Ellipsoid Representation for Point Cloud Classification and Segmentation. In Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); IEEE: Waikoloa, HI, USA, January 2022; pp. 256–266, doi: 10.1109/WACV51458.2022.00033.
24. Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. SEGCloud: Semantic Segmentation of 3D Point Clouds. In Proceedings of the 2017 International Conference on 3D Vision (3DV); IEEE: Qingdao, October 2017; pp. 537–547, doi: 10.1109/3DV.2017.00067.
25. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; IEEE: Salt Lake City, UT, USA, June 2018; pp. 4490–4499, doi: 10.1109/CVPR.2018.00472.
26. Meng, H.-Y.; Gao, L.; Lai, Y.-K.; Manocha, D. VV-Net: Voxel VAE Net With Group Convolutions for Point Cloud Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); IEEE: Seoul, Korea (South), October 2019; pp. 8499–8507, doi: 10.1109/ICCV.2019.00859.
27. Liu, Z.; Tang, H.; Lin, Y.; Han, S. Point-Voxel CNN for Efficient 3D Deep Learning. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019); Vancouver, Canada, 2019; Vol. 32, doi:10.48550/arXiv.1907.03739.
28. Chen, Y.; Liu, S.; Shen, X.; Jia, J. Fast Point R-CNN. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); IEEE: Seoul, Korea (South), October 2019; pp. 9774–9783, doi: 10.1109/ICCV.2019.00987.
29. Fan, H.; Yang, Y. PointRNN: Point Recurrent Neural Network for Moving Point Cloud Processing 2019, doi: 10.48550/arXiv.1910.08287.
30. 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 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Seattle, WA, USA, June 2020; pp. 11105–11114, doi: 10.1109/CVPR42600.2020.01112.
31. Xu, Y.; Tang, W.; Zeng, Z.; Wu, W.; Wan, J.; Guo, H.; Xie, Z. NeiEA-NET: Semantic Segmentation of Large-Scale Point Cloud Scene via Neighbor Enhancement and Aggregation. International Journal of Applied Earth Observation and Geoinformation 2023, 119, 103285, doi:10.1016/j.jag.2023.103285.

32. Rethage, D.; Wald, J.; Sturm, J.; Navab, N.; Tombari, F. Fully-Convolutional Point Networks for Large-Scale Point Clouds. In Computer Vision – ECCV 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2018; Vol. 11208, pp. 625–640 ISBN 978-3-030-01224-3, doi: 10.1007/978-3-030-01225-0_37.
33. Tatarchenko, M.; Park, J.; Koltun, V.; Zhou, Q.-Y. Tangent Convolutions for Dense Prediction in 3D. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; IEEE: Salt Lake City, UT, USA, June 2018; pp. 3887–3896, doi: 10.1109/CVPR.2018.00409.
34. Chen, S.; Niu, S.; Lan, T.; Liu, B. PCT: Large-Scale 3d Point Cloud Representations Via Graph Inception Networks with Applications to Autonomous Driving. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP); IEEE: Taipei, Taiwan, September 2019; pp. 4395–4399, doi: 10.1109/ICIP.2019.8803525.
35. Thomas, H.; Qi, C.R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; Guibas, L. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); IEEE: Seoul, Korea (South), October 2019; pp. 6410–6419, doi: 10.1109/ICCV.2019.00651.
36. Tan, W.; Qin, N.; Ma, L.; Li, Y.; Du, J.; Cai, G.; Yang, K.; Li, J. Toronto-3D: A Large-Scale Mobile LiDAR Dataset for Semantic Segmentation of Urban Roadways. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); June 2020; pp. 797–806, doi: 10.1109/CVPRW50498.2020.00109.
37. Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.D.; Schindler, K.; Pollefeys, M. SEMANTIC3D.NET: A NEW LARGE-SCALE POINT CLOUD CLASSIFICATION BENCHMARK. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. 2017, IV-1/W1, 91–98, doi:10.5194/isprs-annals-IV-1-W1-91-2017.
38. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); IEEE: Seoul, Korea (South), October 2019; pp. 9296–9306, doi: 10.1109/ICCV.2019.00939.
39. Li, Y.; Ma, L.; Zhong, Z.; Cao, D.; Li, J. TGNet: Geometric Graph CNN on 3-D Point Cloud Segmentation. IEEE Trans. Geosci. Remote Sensing 2020, 58, 3588–3600, doi:10.1109/TGRS.2019.2958517.
40. Wan, J.; Zeng, Z.; Qiu, Q.; Xie, Z.; Xu, Y. PointNest: Learning Deep Multiscale Nested Feature Propagation for Semantic Segmentation of 3-D Point Clouds. IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing 2023, 16, 9051–9066, doi:10.1109/JSTARS.2023.3315557.
41. Yoo, S.; Jeong, Y.; Jameela, M.; Sohn, G. Human Vision Based 3D Point Cloud Semantic Segmentation of Large-Scale Outdoor Scenes. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); IEEE: Vancouver, BC, Canada, June 2023; pp. 6577–6586, doi: 10.1109/CVPRW59228.2023.00699.
42. Boulch, A.; Saux, B.L.; Audebert, N. Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks. 3dor@ eurographics 2017, 3, 1–8, doi:10.2312/3dor.20171047.
43. Contreras, J.; Denzler, J. Edge-Convolution Point Net for Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium; IEEE: Yokohama, Japan, July 2019; pp. 5236–5239, doi: 10.1109/IGARSS.2019.8899303.
44. Truong, G.; Gilani, S.Z.; Islam, S.M.S.; Suter, D. Fast Point Cloud Registration Using Semantic Segmentation. In Proceedings of the 2019 Digital Image Computing: Techniques and Applications (DICTA); IEEE: Perth, Australia, December 2019; pp. 1–8, doi: 10.1109/DICTA47822.2019.8945870.
45. Liu, C.; Zeng, D.; Akbar, A.; Wu, H.; Jia, S.; Xu, Z.; Yue, H. Context-Aware Network for Semantic Segmentation Toward Large-Scale Point Clouds in Urban Environments. IEEE Trans. Geosci. Remote Sensing 2022, 60, 1–15, doi:10.1109/TGRS.2022.3182776.
46. Zeng, Z.; Xu, Y.; Xie, Z.; Tang, W.; Wan, J.; Wu, W. LEARD-Net: Semantic Segmentation for Large-Scale Point Cloud Scene. International Journal of Applied Earth Observation and Geoinformation 2022, 112, 102953, doi:10.1016/j.jag.2022.102953.

47. Yin, F.; Huang, Z.; Chen, T.; Luo, G.; Yu, G.; Fu, B. DCNet: Large-Scale Point Cloud Semantic Segmentation With Discriminative and Efficient Feature Aggregation. IEEE Trans. Circuits Syst. Video Technol. 2023, 33, 4083–4095, doi:10.1109/TCSVT.2023.3239541.
48. Luo, L.; Lu, J.; Chen, X.; Zhang, K.; Zhou, J. LSGRNet: Local Spatial Latent Geometric Relation Learning Network for 3D Point Cloud Semantic Segmentation. Computers & Graphics 2024, 124, 104053, doi:10.1016/j.cag.2024.104053.
49. Xu, J.; Zhang, R.; Dou, J.; Zhu, Y.; Sun, J.; Pu, S. RPVNet: A Deep and Efficient Range-Point-Voxel Fusion Network for LiDAR Point Cloud Segmentation, doi: 10.1109/ICCV48922.2021.01572.
50. Hou, Y.; Zhu, X.; Ma, Y.; Loy, C.C.; Li, Y. Point-to-Voxel Knowledge Distillation for LiDAR Semantic Segmentation, doi: 10.1109/CVPR52688.2022.00829.

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.