remote sensing

MDPI

Article

Multilevel Geometric Feature Embedding in Transformer Network for ALS Point Cloud Semantic Segmentation

Zhuanxin Liang and Xudong Lai *

School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430070, China; gisleung@whu.edu.cn
* Correspondence: laixudong@whu.edu.cn

Abstract: Effective semantic segmentation of Airborne Laser Scanning (ALS) point clouds is a crucial field of study and influences subsequent point cloud application tasks. Transformer networks have made significant progress in 2D/3D computer vision tasks, exhibiting superior performance. We propose a multilevel geometric feature embedding transformer network (MGFE-T), which aims to fully utilize the three-dimensional structural information carried by point clouds and enhance transformer performance in ALS point cloud semantic segmentation. In the encoding stage, compute the geometric features surrounding tee sampling points at each layer and embed them into the transformer workflow. To ensure that the receptive field of the self-attention mechanism and the geometric computation domain can maintain a consistent scale at each layer, we propose a fixed-radius dilated KNN (FR-DKNN) search method to address the limitation of traditional KNN search methods in considering domain radius. In the decoding stage, we aggregate prediction deviations at each level into a unified loss value, enabling multilevel supervision to improve the network's feature learning ability at different levels. The MGFE-T network can predict the class label of each point in an end-to-end manner. Experiments were conducted on three widely used benchmark datasets. The results indicate that the MGFE-T network achieves superior OA and mF1 scores on the LASDU and DFC2019 datasets and performs well on the ISPRS dataset with imbalanced classes.

Keywords: self-attention mechanism; geometric feature embedding; fixed-radius dilated KNN search; multilevel loss aggregation; point cloud semantic segmentation

check for updates

Citation: Liang, Z.; Lai, X. Multilevel Geometric Feature Embedding in Transformer Network for ALS Point Cloud Semantic Segmentation. *Remote Sens.* **2024**, *16*, 3386. https://doi.org/10.3390/rs16183386

Academic Editor: Sander Oude Elberink

Received: 8 July 2024
Revised: 30 August 2024
Accepted: 10 September 2024
Published: 12 September 2024

1. Introduction

As a form of 3D data representation, point clouds precisely convey the spatial location and three-dimensional structure of objects. Airborne LiDAR is an efficient method for acquiring large-scale surface point cloud data, significantly contributing to fields such as real-world 3D modeling, topographic surveying, urban planning, and design. Achieving high-precision semantic segmentation of point cloud data is crucial in the point cloud data processing workflow, as it directly influences the effectiveness of downstream tasks in leveraging the value of point cloud data.

Semantic segmentation techniques based on deep learning have seen significant advancements in the field of computer vision. However, unlike rasterized image data, point clouds are irregular, unordered, and unstructured, meaning that semantic segmentation frameworks popular in the image domain cannot be directly applied to point clouds. Early scholars primarily focused on converting irregular point clouds into regular data structures to facilitate the use of Convolutional Neural Networks (CNNs) for deep learning tasks. The multi-view CNN [1,2] projects 3D point clouds or shapes onto multiple 2D images, integrates multiple-view information into a single and compact shape descriptor, and then applies a 2D convolutional network for classification. Beyond projection to 2D images, some scholars have also applied 3D convolutional networks to process data through point cloud voxelization [3,4]. While converting point clouds to regular structures successfully enables the application of convolutional networks to point cloud tasks, the conversion into 2D images results in the loss of 3D information, and the voxelization approach is limited by its 3D volumetric resolution and the computational cost of 3D convolutions.

To achieve point-based semantic segmentation, it is essential for the network model to exhibit both permutation invariance and rotation invariance with respect to the set of points. The PointNet [5] network innovatively employs symmetric functions, such as max and sum functions, to address the challenge of the disordered nature of point cloud data. To tackle the limitation of PointNet's disregard for local feature information, the subsequently proposed PointNet++ [6] further improves network performance by aggregating local point features while progressively downsampling the point cloud at the set abstraction layer. PointNet++ marks a significant maturation in point-based deep learning tasks. So far, network architectures for point cloud learning tasks can be broadly categorized into four models: the MLP network [6–8], the point convolutional network [9–12], the graph convolutional network [13–15], and the transformer network [16,17]. These network structures share a common approach to semantic segmentation tasks. In the network encoding stage, point-wise domain information is comprehensively extracted and aggregated in high-dimensional space. In the network decoding stage, the high-dimensional feature information is restored to the original point positions via interpolation. Notably, the residual connection [18] has become an indispensable module in deep learning networks, directly adding the inputs of the modular units with the outputs in the form of a skip connection, effectively solving the vanishing gradient problem in deep neural networks.

Unlike other point cloud data, ALS point clouds exhibit distinct characteristics. Firstly, objects are horizontally distributed with a significant variation in size. Secondly, there is a minimal overlap in the vertical direction, though elevation changes between points are significant. Lastly, the point cloud is characterized by sparsity and non-uniform density while encompassing a wide area. To cope with the new challenges posed by large-scale point clouds, scholars have re-examined the limitations of existing networks in outdoor scenes and made improvements. LGENet [19] resampled the point cloud on a 0.24 m grid during the data loading stage to mitigate the negative effects of the inhomogeneous densities of the ALS point cloud. Several studies [20–23] have added spectral information to the initial input point features to enrich the initial point features. To adapt the network to the complex structure of objects in the ALS point cloud, additional features based on geometric computation, such as structural features [22], height above ground [24], and normal vectors [25], were added to the network, yielding significant improvements. Additionally, some attention mechanisms have also been added to segmentation networks to improve the accuracy of point-by-point classification. GADH-Net [26] designed an elevation attention module to adjust the final category probability maps, which improves the recognition of categories with relatively stable elevation distributions. Jiang [27] used FCN [28] to separate ground and non-ground points and designed a Ground-Aware Attention module to improve the segmentation performance of small, sparse urban objects. GraNet [29] integrated Attention Pooling and a Global Relation-Aware Attention module to capture global attention from the structural relations of all points and channels, positioned at the network's end to enhance high-dimensional features. Additionally, the network structure based on the idea of dense connectivity [26,30] has also been applied to the ALS point cloud task, yielding better results.

Inspired by the self-attention mechanism, we propose a novel ALS semantic segmentation network named Multilevel Geometric Feature Embedding Transformer (MGFE-T). The main contributions of this paper are as follows:

(1) The GFE-T module is specifically designed to enhance the network's ability to learn and capture local geometric features. By embedding these geometric features into the point transformer, the network effectively learns local geometric structure features, enhancing its classification capability.

(2) We propose the FR-DKNN method, which effectively addresses the issue of inconsistent neighborhood ranges in KNN due to uneven point cloud distributions by

using dilated K-nearest neighbor queries within a fixed radius. This ensures that the network retains robust discriminative capability when learning neighborhood features of the points.

(3) Based on the proposed GFE-T module and FR-DKNN method, we design MGFE-T, a transformer ALS point cloud semantic segmentation network with multilevel geometric feature embedding and multilevel loss aggregation (M-Loss) supervising the network at each level.

(4) We conducted experiments on the LASDU, DFC2019, and ISPRS datasets, demonstrating the excellent performance of the proposed method. Ablation experiments were conducted to verify the effectiveness of each module in the network. Cross-validation across datasets demonstrated the reliable generalization ability of the network.

The remainder of this paper is organized as follows: Section 2 presents the overall network architecture and elaborates on the GFE-T module and FR-DKNN method. Section 3 describes the dataset, evaluation metrics, and experimental results. Section 4 presents ablation experiments to verify the effectiveness of the proposed module. Section 5 examines the generalization capabilities of the proposed method. Section 6 concludes the paper with a summary.

2. Methods

In this section, we first describe the overall semantic segmentation network structure, highlighting its backbone network and the application of new modules. Then, we discuss the embedding of geometric features in Section 2.2 and propose a fixed-radius KNN query method in Section 2.3. Finally, we detail the loss function used for network training in Section 2.4.

2.1. Overall Architecture

Following Point Transformer [31], we have developed a new network architecture named MGFE-T, as illustrated in Figure 1. The MGFE-T network consists of standard encoding and decoding stages. The input is segmented point cloud data containing N points, each with five attributes: X, Y, Z (the coordinates of the points normalized to the origin), reflection intensity, and height. In the downsampling stage, we used the Farthest Point Sampling [6] (FPS) method to gradually reduce the number of training points and minimize memory usage. In the upsampling stage, we applied spatial interpolation to combine features from the downsampling stage and gradually restore feature dimensionality. The outputs of each network layer were aggregated using multilevel loss, and a fully connected layer was employed to map high-dimensional features to their corresponding semantic classes.

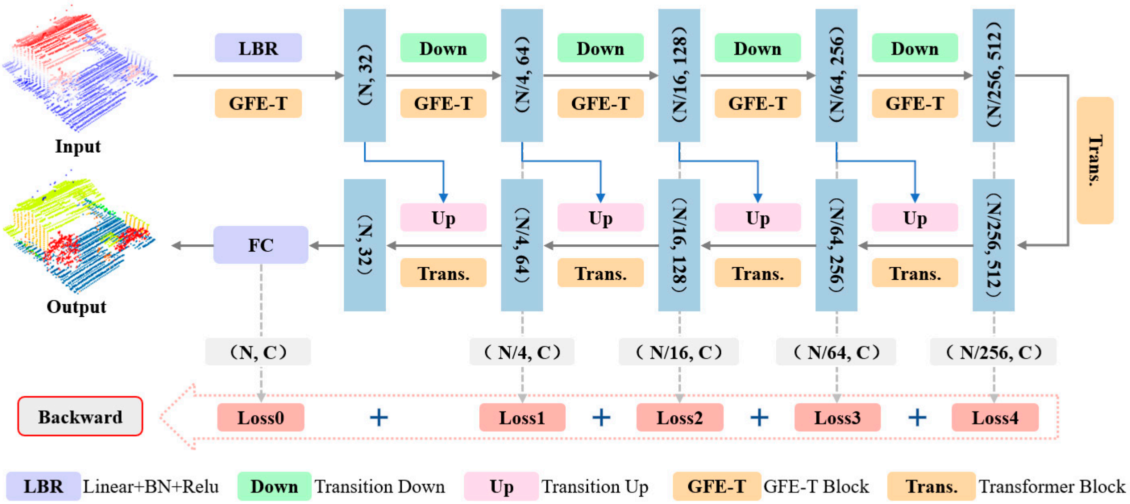


Figure 1. MGFE-T Semantic Segmentation Network Architecture.

To enable efficient extraction of high-dimensional and geometric features, we designed the GFE-T block and embedded it into the encoding stage of the network. The structure of the GFE-T block is depicted in Figure 2. To ensure a stable receptive field for the sampling points in each layer when computing local features, we used FR-DKNN for domain search. The residual connection was also used to transform the GFE-T Block's focus from learning the mapping function to learning the residual term, improving the deep network's ability to learn features.

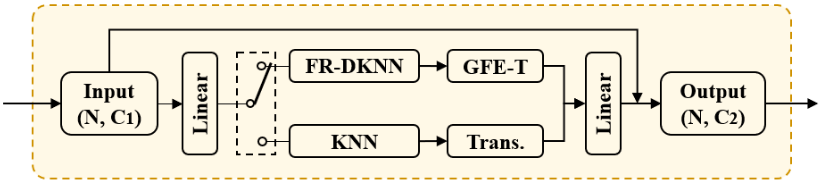


Figure 2. GFE-T/Transformer Block with Residual Connection.

2.2. Geometric Feature Embedding Transformer

2.2.1. Point Transformer

Self-attention networks have achieved notable success in natural language processing [32–34] and 2D image analysis [35–37]. Since the self-attention operator is insensitive to the arrangement and cardinality of the input elements, it is also suitable for point cloud data structures with inherently disordered nature.

In the self-attention operator of Point Transformer, the initialization of the query (Q), key (K), and value (V) matrices is crucial. The query and key matrices calculate the attention score for each point, which is then used to compute a weighted sum with the value matrix, ultimately producing the operator's output. To allow the network to understand the positional relationship between the points, positional encoding was integrated into the computation of the score and value. For the point set $\mathcal{X}(i) = \{P_j\}_{j=1}^{k}$ consisting of the center point P_i and its k nearest points, the center point feature x_i is transformed by the self-attention operator into the feature y_i^T, which can be expressed as follows:

$$y_i^T = \sum_{x_j \in \mathcal{X}(i)} \left(mlp\left(\varphi(x_j) - \psi(x_i) + \delta \right) \right) \odot \left(\alpha(x_j) + \delta \right) \tag{1}$$

$$\delta = mlp(p_j - p_i) \tag{2}$$

where x_j represents the features of the neighborhood points, φ, ψ, and α denote the linear mapping or multilayer perceptron used to initialize the Q, K, and V, δ is the positional encoding, and p_i and p_j are the 3D spatial coordinates.

2.2.2. Geometric Feature Embedding

ALS point cloud contains various objects that have unique geometric and structural differences. For instance, points on roofs and floors have planar properties, while points on trees have significant elevation differences and uneven distributions. Therefore, it is plausible that prior computational geometric properties can provide essential reference information for point cloud segmentation tasks.

We computed a set of low-dimensional geometric features within the point domain while extracting local features based on the Transformer. Specifically, we derived the low-dimensional features that characterize the geometric structure of the point set $\mathcal{X}(i)$ based on its covariance matrix eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$, including neighborhood linearity L_λ, planarity P_λ, sphericity S_λ, omnivariance O_λ, anisotropy A_λ, and change of curvature C_λ, defined as follows:

$$\begin{array}{ll} L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1} & P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1} \\ S_\lambda = \frac{\lambda_3}{\lambda_1} & O_\lambda = \sqrt[3]{\lambda_1 \lambda_2 \lambda_3} \\ A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1} & C_\lambda = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \end{array} \tag{3}$$

In addition to the aforementioned features, we computed two additional attributes based on the absolute elevation value of each point in the point set $\mathcal{X}(i)$, the elevation range $R_z = z_{max} - z_{min}$ and the elevation variance Var_z. The final geometrically computed low-dimensional feature was defined as an 8-dimensional vector $[L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, C_\lambda, R_z, Var_z]$.

To tackle the challenge of integrating high-dimensional abstract features with low-dimensional geometric features, we propose a module called the Geometric Feature Embedding Transformer (GFE-T). Specifically, the geometric computational features are transformed into high-dimensional features with an equal number of channels as y_i^T, employing a sequential combination of a linear layer, a batch normalization layer, and a ReLU activation layer. Subsequently, the two sets of features are concatenated, and the final output is projected to match the original input features' channel count through the MLP (Multi-Layer Perceptron). After the GFE-T module transforms the input features, its output feature y_i can be expressed as:

$$y_i = mlp\left(cat\left(y_i^T, LBR([L_\lambda \cdots Var_z])\right)\right) \tag{4}$$

The structure of GFE-T is illustrated in Figure 3. After passing through the GFE-T module, the points not only carry high-dimensional features derived from the learning mechanism but also improve the ability of the features to characterize geometric structures. As a result, the network can better focus on the geometric structure differences between different objects more effectively.

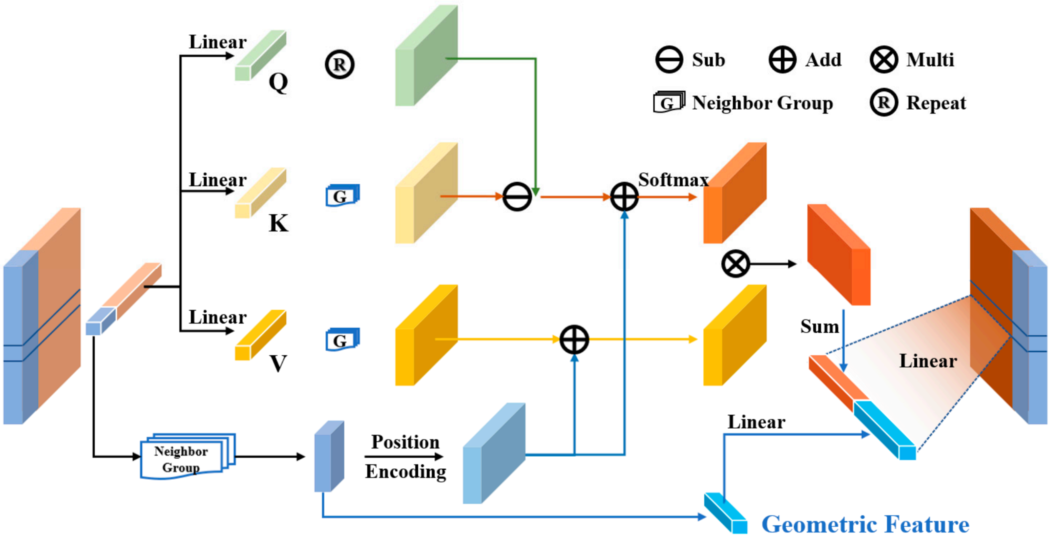


Figure 3. GFE-T Module Architecture.

2.3. *Fixed-Radius Dilated KNN*

The main methods for querying adjacent points in point cloud segmentation tasks include KNN, Ball Query, and Dilated KNN [9,24,38]. Due to the embedding of low-dimensional geometric features in the GFE-T module, new requirements have been introduced for domain search. Firstly, the k neighboring points of all sampling points at any level should form a region with a consistent spatial scale, enabling the calculation of comparable geometric features. Secondly, k neighboring points should provide sufficient receptive fields so that centroid points can gather more domain information.

We propose a Fixed-radius Dilated KNN Search (FR-DKNN) method, which builds upon the Dilated KNN [24]. As illustrated in Figure 4, compared to other methods, FR-DKNN not only effectively expands the receptive field but also ensures the spatial scale consistency of domain search results. FR-DKNN is highly effective at extracting stable local features within a domain, particularly when processing point cloud data characterized by significant density variations or uneven distributions.

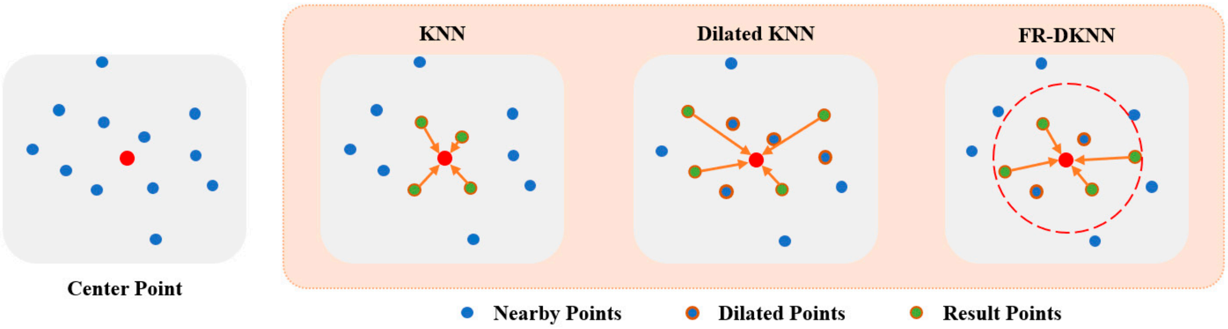


Figure 4. Comparison of FR-DKNN with other methods (k = 4, d = 2).

During the process of querying neighboring points for sampling point P, our method initially selects $d \times k$ points, with d representing the dilation rate. These points are then filtered using the constraint radius R_l. The value of R_l is determined by the hierarchical level at which the current point is situated. After filtering, k points are randomly selected from the remaining points to obtain the final query result. The FR-DKNN method can be formalized as follows:

$$FR\_DKNN_P = Random_k\left(\mathcal{F}_{R_l}\left\{P_j \,\middle|\, \forall P_j \in \mathcal{N}_0^{dk}(P)\right\}\right) \tag{5}$$

where $\mathcal{N}_0^{dk}(P)$ denotes the domain comprising $d \times k$ neighboring points around the center point P, $\mathcal{F}_{R_l}$ represents the radius constraint, and $Random_k$ indicates the final random sampling process. Notably, when d is sufficiently large, FR-DKNN becomes equivalent to Ball Query, and when d is set to 1, it is consistent with KNN query results.

During implementation, the radius constraint for each network layer is pre-evaluated to ensure optimal performance. For a given dataset, the first step involves downsampling it L times. This generates a series of point clouds, denoted as $\{S_0, S_1, S_2, \ldots, S_l\}$, where S_l represents the point cloud at each network layer. A point-by-point dilated KNN query is conducted to calculate the neighborhood radius. The neighborhood radii of all points are sorted, and the median radius is chosen as the constraint radius for the FR-DKNN method at the current layer. To enhance efficiency, random sampling may be used when S_l contains numerous points, selecting only a portion of the points for domain radius estimation in order to avoid time-consuming operations on all points.

2.4. Multilevel Loss Aggregation

Semantic segmentation networks usually rely solely on the final layer's output to compute the prediction error, overlooking the downsampled features from intermediate layers. However, the downsampled point cloud in intermediate layers still provides valuable semantic signals that can optimize the output of each layer in the decoding phase. Inspired by RFFS-Net [30], as illustrated in Figure 1, we employed a multilevel loss aggregation (M-loss) approach. Specifically, during the downsampling stage, we recorded the true labels of each sampling point, mapped the decoding stage outputs at corresponding levels to the predicted labels, computed the losses at each level, and represented the final loss of the network as the sum of these losses, denoted as $Loss = \sum_{l=0}^{L} \lambda_l \mathcal{L}_l$. The loss value at each level is calculated as follows:

$$\mathcal{L}_l = -\frac{1}{N}\sum_{i=1}^{N}\left(w_c \cdot y_{i,c} \cdot \log \frac{\exp(x_{n,c})}{\sum_{j=1}^{C}\exp(x_{n,j})}\right) \tag{6}$$

where $y_{i,c}$ denotes the value of the c-th element in the one-hot encoded label of the sample point. $\exp(x_{n,j})$ represents the predicted probability for the j-th class output by the network.

w_c represents the weight assigned to each class label during loss computation. If the number of points in the j-th class is M_c, then the weight w_c is computed as follows:

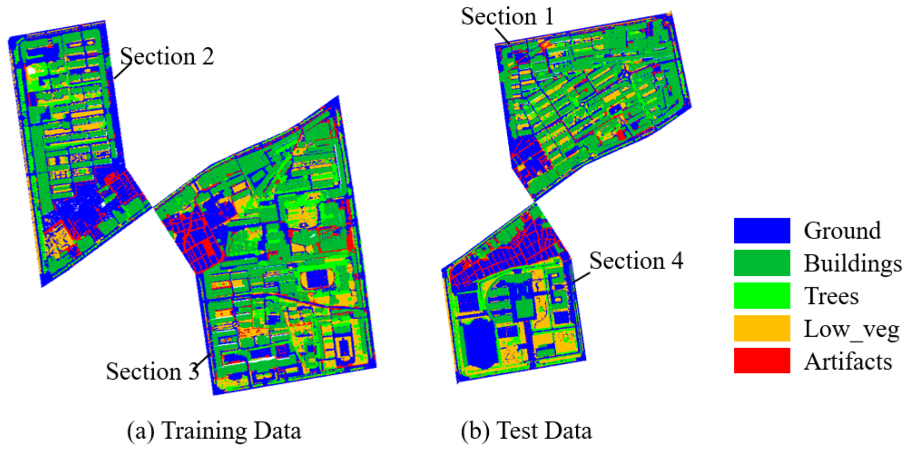
$$w_c = \frac{1/\sqrt{M_c}}{\sum_j^C 1/M_j} \tag{7}$$

3. Results

3.1. Datasets

To assess the effectiveness of the proposed method, we conducted experiments on three widely recognized benchmark datasets: LASDU [39], DFC2019 [40], and ISPRS [41].

(1) LASDU: The LASDU dataset is a large ALS point cloud data collected at an altitude of about 1200 m in the Heihe River Basin in northwest China. As shown in Figure 5, the points in the dataset are labeled into a total of five classes: ground, buildings, trees, low vegetation, and artifacts. Considering the balanced distribution of the labels of each category in each section, the publisher suggests using Sections 2 and 3 as training data and Sections 1 and 4 as test data.

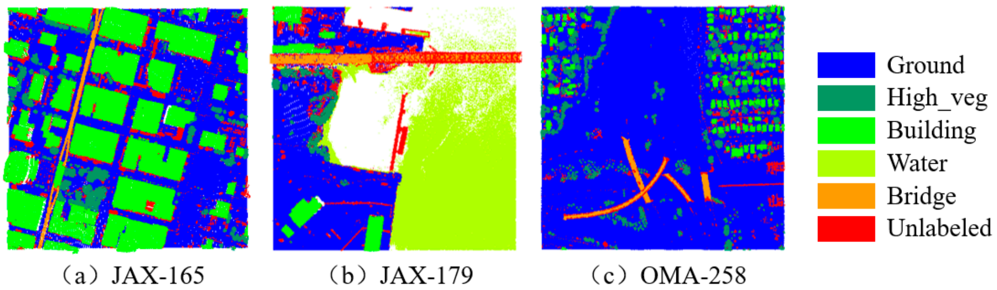


(a) Training Data

(b) Test Data

Figure 5. Preview of the LASDU dataset.

(2) DFC 2019: The DFC2019 dataset is the ALS point cloud data provided by the Data Fusion Contest 2019, collected from the urban areas of Jacksonville, Florida and Omaha, Nebraska, USA. As shown in Figure 6, the 110 files in the dataset are regular independent regions in which each point is labeled into six classes: ground, high vegetation, building, water, bridge deck, and unlabeled.



（a）JAX-165

（b）JAX-179

（c）OMA-258

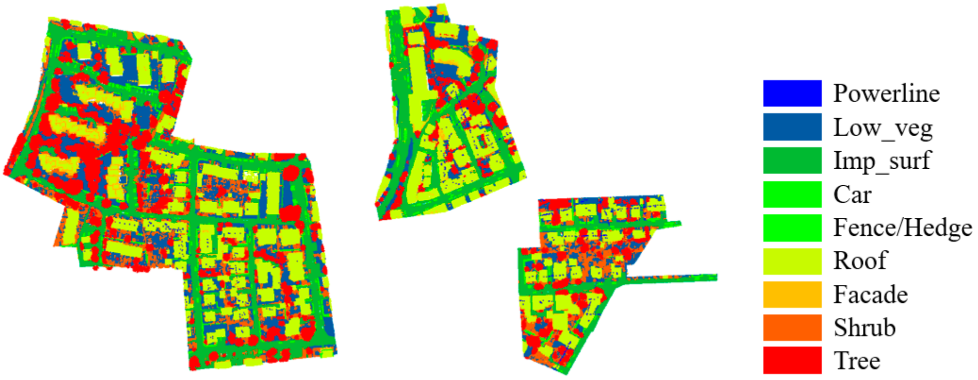
Figure 6. Preview of the DFC2019 dataset (3 of 110 files).

Consistent with existing research work, we divided the 110 files into two groups, one containing 100 files for the training set and the other containing 10 files as the test set. The distribution of each category in the training and test sets is shown in Table 1.

Table 1. Class distribution statistics for the training and test sets in the DFC2019 dataset.

Class	**Training Data**		**Test Data**	
	Number	**Ratio**	**Number**	**Ratio**
Ground	49,517,558	64.63%	4,701,150	66.78%
High_veg	11,152,546	14.56%	944,234	13.41%
Building	10,136,608	13.23%	915,618	13.01%
Water	1,295,778	1.69%	80,387	1.14%
Bridge	869,547	1.13%	90,032	1.28%
Unlabeled	3,642,710	4.75%	308,205	4.38%
Sum	76,614,747		7,039,626	

(3) ISPRS: The ISPRS dataset was obtained by airborne LiDAR scanning in Vaihingen, Germany. As shown in Figure 7, the ISPRS dataset was divided into two parts: a training set and a test set. The points in the dataset were labeled into a total of nine classes: powerline, low vegetation, impervious surfaces, car, fence/hedge, roof, facade, shrub, and tree.



(a) Training Data (b) Test Data

Figure 7. Preview of the ISPRS dataset.

3.2. Implementation Details

To optimize the efficiency and performance of the experimental process, we partitioned the large-scale scenes into regular data blocks and downsampled them according to the density distribution of the points. Specifically, the LASDU dataset was divided into 40 m × 40 m blocks, the DFC2019 dataset into 80 m × 80 m blocks with 0.5 m voxel downsampling, and the ISPRS dataset into 24 m × 24 m blocks with 0.24 m voxel downsampling. The overlap of neighboring blocks was set to 10 m across all datasets.

Before feeding the block data into the network for training, we translated the point coordinates to the origin of the 3D coordinate system to stabilize the coordinates in the horizontal and vertical directions. To further enhance the training data, we performed data augmentation and generalization on each block, including random rotation around the Z-axis, random jitter of point positions, and scaling adjustments. When input into the network, each point is represented as a five-dimensional row vector [X, Y, Z, I, H], where X, Y, and Z are the coordinates of the points after translation, I denotes intensity, and H represents relative height. During testing, we applied the same data partitioning strategy as used in training but without augmentation.

All experiments were conducted using the PyTorch framework on a GPU 3060 machine for all datasets. The dilation rate was configured to $d = 2$, and the number of neighboring points was set to $k = [8, 16, 16, 16, 16]$. For each layer, the program automatically evaluated the constraint radius. Additionally, the loss coefficient for each level in the M-loss module was set to $\lambda = [1.0, 0.2, 0.2, 0.2, 0.2]$.

3.3. Evaluation Metrics

Commonly used metrics in the semantic segmentation task of ALS point clouds include Intersection over Union (IoU) and its mean value (mIoU), the F1 score and its mean value (mF1), as well as overall accuracy (OA). The F1 score offers a comprehensive evaluation of precision and recall for each category, which is particularly useful when dealing with imbalanced sample distributions, effectively assessing the classification performance of individual categories.

In this study, as in most research works, the F1 score, mean F1 score (mF1), and overall accuracy (OA) were selected as evaluation indices. The formulas for these metrics are calculated as follows:

$$Precision_c = \frac{TP_c}{TP_c + FP_c} \tag{8}$$

$$Recall_c = \frac{TP_c}{TP_c + FN_c} \tag{9}$$

$$F1_c = \frac{precision_c \times recall_c}{precision_c + recall_c} \tag{10}$$

$$OA = \frac{\sum_{c=1}^{C} TP_c}{Number\ of\ all\ points} \tag{11}$$

where *TP* represents true positives, *FP* denotes false positives, and *FN* indicates false negatives.

3.4. Experimental Results

3.4.1. Result of the LASDU

The segmentation results for the LASDU dataset are summarized in Table 2, which include results from other ALS point cloud segmentation networks for comparison. The MGFE-T network demonstrates a commendable F1 score across all categories, particularly excelling in ground, low vegetation, and artifacts. The overall accuracy (OA) and mean F1 score (mF1) achieved the highest levels reported to date, significantly outperforming other methods with scores of 89.1% and 80.1, respectively. Although the F1 score for tree segmentation is 1.1 lower than that of VD-LAB [42], the score for low vegetation is 0.9 higher, indicating that our model tends to confuse some trees with low vegetation. Figure 8 illustrates the segmentation results for the MGFE-T network and the baseline (Point Transformer), highlighting that the incorporation of the GFE-T module notably enhanced the segmentation results for building and artifact categories. This improvement is attributed to the distinct geometric features, such as flat-roof structures, which are more easily distinguishable.

Table 2. Comparison of results between MGFE-T and other ALS point cloud semantic segmentation methods on the LASDU dataset (bold indicates the best results, underlining indicates the second-best results).

Method	Ground	Building	Trees	Low_veg	Artifacts	OA	mF1
GraNet [29]	89.9	95.8	86.1	64.7	42.4	86.2	75.8
VD-LAB [42]	91.2	95.5	**87.2**	73.5	44.6	88.0	78.4
RFFS [30]	90.9	95.4	86.8	71.0	44.4	87.1	77.7
RRDAN [43]	91.6	96.6	84.1	66.3	48.3	87.7	77.4
MCFN [44]	91.6	**96.7**	85.9	67.1	43.8	88.0	77.0
IPCONV [45]	90.5	96.3	85.8	59.6	46.3	86.7	75.7
Ours	**92.6**	96.6	86.1	**74.4**	**50.7**	**89.1**	**80.1**

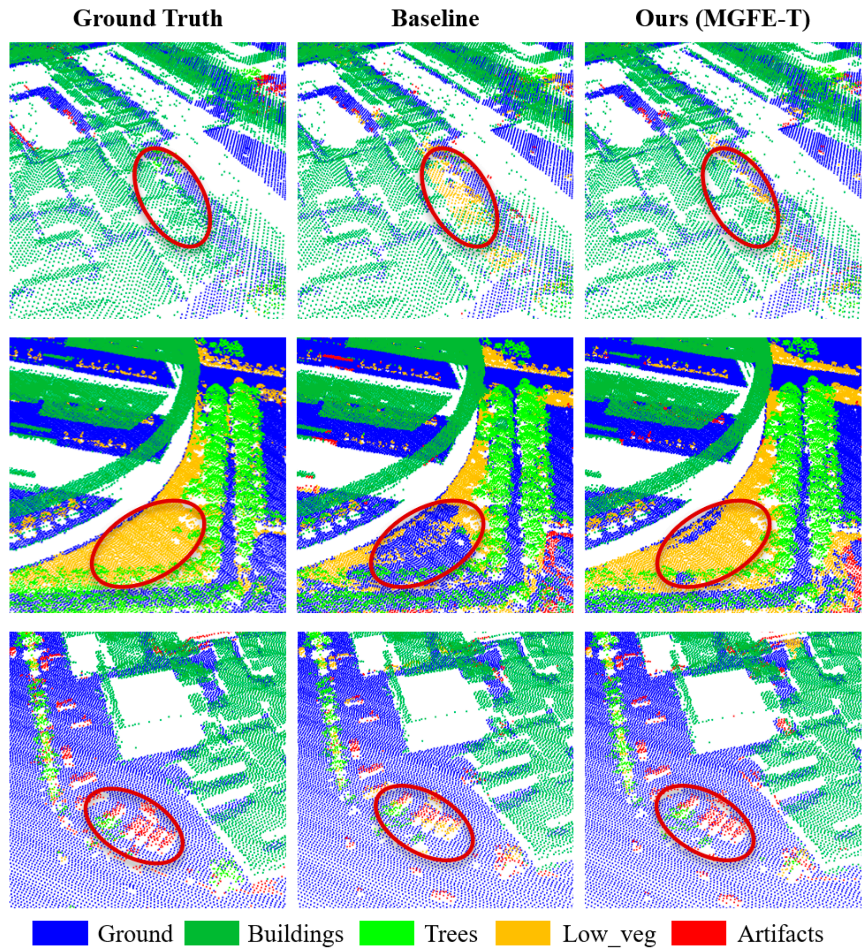


Figure 8. Visualization of semantic segmentation results for some regions of the LASDU dataset (the first, second, and third columns are the ground truth, the results of the baseline, and the results of MGFE-T, respectively).

3.4.2. Result of the DFC2019

The semantic segmentation results of our method on the DFC2019 dataset are presented in Table 3. The F1 score of the MGFE-T network exceeds 90 across all categories, with notable enhancements in the water and bridge categories. The overall accuracy (OA) of the entire test set is 98.5%. Although this is only 0.1 higher than the second-ranked LGENet [19], the mean F1 (mF1) score of 95.7 represents a substantial improvement. We visualized several segmentation results, and Figure 9 demonstrates that, compared to the baseline, the MGFE-T network significantly improves the classification of high vegetation and mitigates the issue of the baseline network confusing building roofs with bridge decks.

Table 3. Comparison of results between MGFE-T and other ALS point cloud semantic segmentation methods on the DFC2019 dataset (bold indicates the best results, underlining indicates the second-best results).

Method	Ground	High_veg	Building	Water	Bridge	OA	mF1
LGENet [19]	99.3	**98.3**	92.8	47.4	79.1	98.4	83.4
DA-Net [21]	99.3	97.6	92.7	41.6	85.1	98.3	83.3
Local and [27]	98.9	96.1	90.2	41.6	83.7	94.8	81.4
RFFS-Net [30]	96.6	96.1	88.7	77.8	81.0	94.3	88.0
RRDAN [43]	99.1	98.1	95.8	62.8	82.3	98.1	87.6
IPCONV [45]	98.8	97.3	92.9	92.1	58.2	97.1	87.9
Ours	**99.6**	96.6	**95.0**	**94.0**	**93.3**	**98.5**	**95.7**

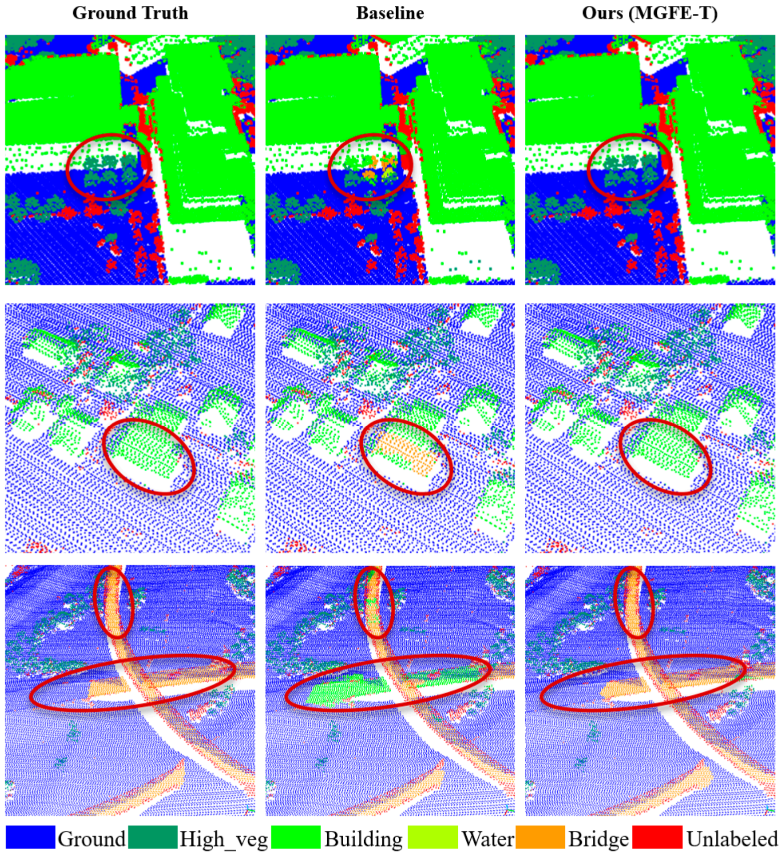


Figure 9. Visualization of semantic segmentation results for some regions of the DFC2019 dataset (the first, second, and third columns are the ground truth, the results of the baseline, and the results of MGFE-T, respectively).

3.4.3. Result of the ISPRS

The segmentation results of the ISPRS dataset are shown in Table 4. Compared with other methods, the MGFE-T network demonstrates a better performance in low_veg, imp_surf, roof, facade, and tree. The OA of our method is 85.2%, which achieves the best result. Disappointingly, the F1 score for fence/hedge was not good, resulting in an mF1 score of only 71.3. This is due to two reasons. Firstly, there are fewer samples of fence/hedge in the dataset, resulting in the network not being able to learn its features adequately. Second, fence/hedge has a complex interleaved spatial distribution with shrub and their elevations are similar, resulting in a large number of fence/hedge being incorrectly classified into shrub. The visualization of the segmentation results of the MGFE-T network and the baseline on the ISPRS dataset is given in Figure 10, which shows that our proposed method has a significant improvement in the classification performance.

Table 4. Comparison of results between MGFE-T and other ALS point cloud semantic segmentation methods on the ISPRS dataset (bold indicates the best results, underlining indicates the second-best results).

Method	Power	Low_veg	Imp_surf	Car	Fence/Hedge	Roof	Facade	Shrub	Tree	OA	mF1
GraNet [29]	67.7	82.7	91.7	80.9	**51.1**	94.5	62.0	49.9	82.0	84.5	73.6
VD-LAB [42]	69.3	80.5	90.4	79.4	38.3	89.5	59.7	47.5	77.2	81.4	70.2
RFFS [30]	**75.5**	80.0	90.5	78.5	45.5	92.7	57.9	48.3	75.7	82.1	71.6
RRDAN [43]	72.2	81.7	91.2	**84.6**	44.8	94.7	**65.2**	**52.0**	**85.3**	84.9	**74.6**
MCFN [44]	74.5	82.3	**91.8**	79.0	37.5	94.7	61.7	48.7	83.3	84.4	72.6
IPCONV [45]	66.8	82.1	91.4	74.3	36.8	94.8	**65.2**	42.3	82.7	84.5	70.7
Ours	70.7	**84.0**	**91.8**	79.6	23.6	**95.0**	63.4	49.5	84.3	**85.2**	71.3

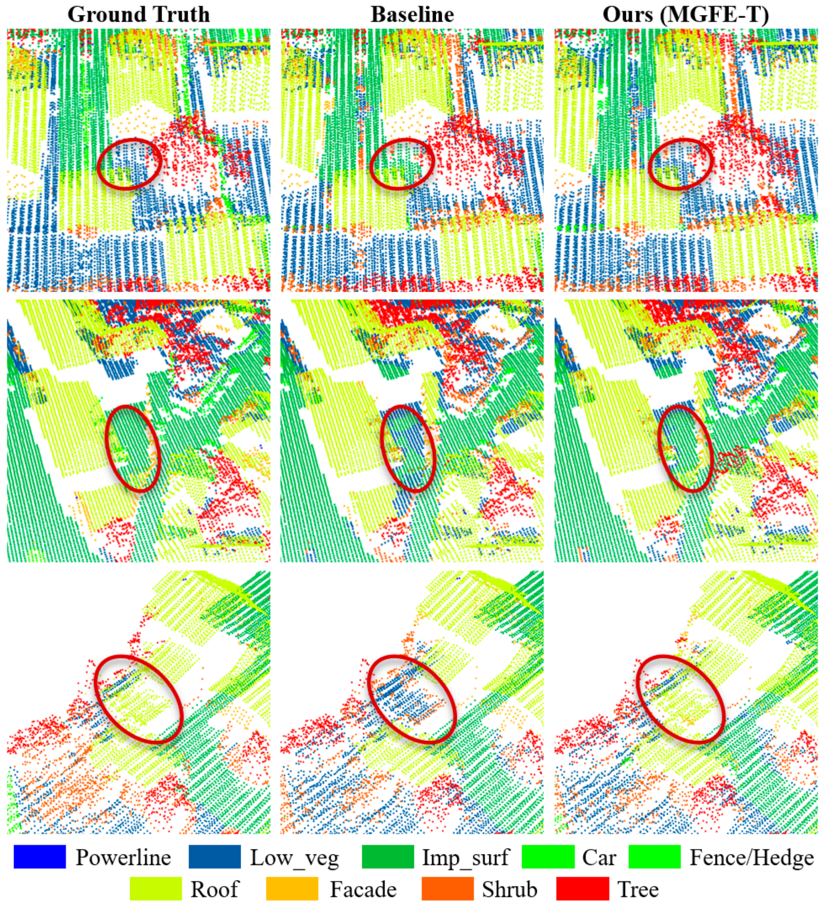


Figure 10. Visualization of semantic segmentation results for some regions of the ISPRS dataset (the first, second, and third columns are the ground truth, the results of the baseline, and the results of MGFE-T, respectively).

The ISPRS benchmark datasets have attracted numerous researchers to conduct experiments since their release. These experiments encompass machine learning methods, deep learning, and a combination of both. Pirotti et al. [46] enabled the Random Forest (RF) method to achieve favorable classification results by testing various combinations of feature numbers and decision trees. Atik et al. [47] evaluated the performance of various machine learning algorithms on the dataset, including RF and Support Vector Machines (SVMs). Additionally, there are also studies [23,48] that use RF for feature optimization, subsequently employing these features to train deep learning models. The results of the aforementioned methods are presented in Table 5. In addition, to better illustrate the comparison, our method and the RRDAN method with the best-combined performance are also added. Since these methods focus on different categories, we show F1 scores and their averages for only three categories that are addressed in all methods. From the results, it can be seen that the Random Forest method achieves the best results; however, in the experiments by Atik et al. [47], SVM results are better than RF, but the results achieved still lag behind the other methods. This suggests that the performance of machine learning methods depends on the selection of initial features and that higher results are achieved when appropriate features and parameter settings are selected. Comparing the two listed methods that combine machine learning and deep learning, OFFS-Net [23] achieves the best results and outperforms all methods in the categories Imp_surf and Roof, while the method of H-MLP [48] does not highlight the advantages of combining the two methods. From the results in Tables 4 and 5, it can be concluded that the deep learning-based methods are more stable, which is, of course, due to the well-designed network model architecture and the long time of training on large-scale datasets. In addition, the comparison results also illustrate that if additional effective features can be input before training, this will improve the performance of the network.

Table 5. Experimental results of machine learning and deep learning methods on the ISPRS dataset. OFFS-Net(S) is a deep learning method that does not use optimal features.

	Method	Imp_surf	Roof	Tree	mF1
Machine Learning	RF (Pirotti et al. [46])	92.6	96.2	84.1	91.0
	SVM (Atik et al. [47])	87.7	74.6	67.8	76.7
Deep Learning	RRDAN [43]	91.2	94.7	85.3	90.4
	Ours	91.8	95.0	84.3	90.4
	OFFS-Net(S) [23]	90.2	94.6	82.3	89.0
Combined ML and DL	H-MLP [48]	83.3	94.7	-	-
	OFFS-Net [23]	92.4	95.3	83.6	90.4

4. Ablation Study

4.1. Impact of Query Radius on Performance

In the FR-DKNN method, radius selection varies with point density across datasets, making a uniform radius unsuitable for all datasets. Therefore, pre-evaluating the radius is essential. Specifically, before training, the 2 × k nearest points for each point are queried to form a local spherical neighborhood, and the local radius for each point is computed. This process results in a sorted list of N radius values, $R_{sort} = [r_1, r_2, r_3, \ldots, r_n(max)]$. To further explore the impact of radius size, nine comparative experiments are conducted using radii located at the 10–90% positions of the sorted results.

As shown in Figure 11, while the overall accuracy (OA) value does not exhibit significant variation, the average F1 score displays a clear trend of initially increasing and then decreasing. The best result is achieved when the radius is set at the 50th percentile. According to these comparative experiment findings, we establish the query radius for the FR-DKNN method as the 50th percentile of the radius sorted list for each dataset before network training. It is worth noting that the radius is computed based on the 2 × k nearest neighbors. Even if the chosen radius considers only 50% of the points, most points still have at least k neighbors for feature extraction.

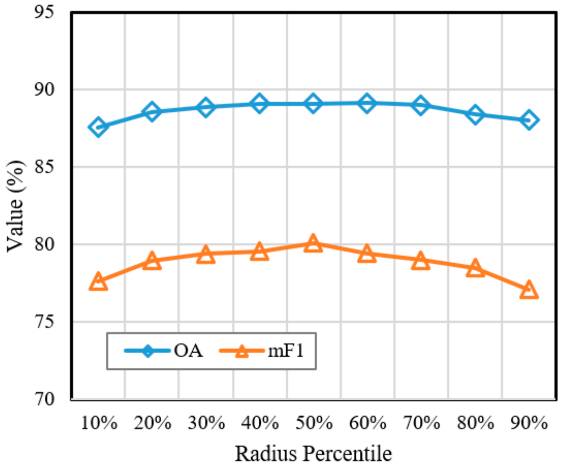


Figure 11. Comparison of experimental results for different radius percentiles.

4.2. Impact of Network Depth on Performance

We conducted comparative experiments on the LASDU dataset to validate the rationale for configuring the MGFE-T network with four layers. Two control experiments were conducted, involving three and five downsampling operations on the input data, respectively. As shown in Table 6, with a network depth of four layers, the model achieved an overall accuracy (OA) of 80.1% and an average F1 score (mF1) of 89.1, demonstrating the best performance. These results indicate that increasing the number of layers to four yields the optimal model performance, striking a balance between accuracy and the F1 score. Fewer than four layers may limit the network's learning capacity, while more than four layers may result in overfitting or diminishing returns in performance.

Table 6. Experimental results with varying network depths.

Layers	OA (%)	mF1
3	79.3	88.9
4	80.1	89.1
5	79.0	88.8

4.3. The Effectiveness of the Proposed Module

Compared with the baseline network Point Transformer, the MGFE-T network has three improvements: the GFE-T module, the FR-DKNN method, and the M-Loss strategy. We performed ablation experiments on the LASDU dataset, and the experimental results are shown in Table 7, which shows that our proposed semantic segmentation network MGFE-T (Model A) improves by 2.5 in mF1 and by 1.1% in overall OA compared to the baseline.

Table 7. Ablation experiments on the LASDU dataset (bold indicates the best values, wavy lines indicate the worst values). Model A refers to our proposed MGFE-T method. Model B does not embed geometric features, Model C uses KNN for neighborhood queries instead of FR-DKNN, and Model D does not utilize multilevel loss aggregation.

Model	GFE-T	FR-DKNN	M-Loss	Ground	Building	Trees	Low_veg	Artifacts	OA	mF1
baseline				91.7	96.1	86.5	70.6	43.1	88.0	77.6
A	√	√	√	**92.6**	**96.6**	86.1	**74.4**	**50.7**	**89.1**	**80.1**
B		√	√	92.0	96.4	86.4	73.3	49.2	88.5	79.5
C	√		√	92.0	96.4	86.2	72.0	48.6	88.4	79.0
D	√	√		92.2	**96.6**	**86.3**	72.1	49.7	88.7	79.4

(1) Efficacy of GFE-T: Compared with Model A, Model B removes the GFE-T module. From the experimental results, it can be seen that the addition of GFE-T has a stable improvement effect on the classification accuracy of all five categories, indicating that geometric features play an effective role in ALS point cloud semantic segmentation.

(2) Efficacy of FR-DKNN: Compared with Model A, Model C removes the FR-DKNN module. From the experimental results, it can be seen that the FR-DKNN module effectively improves the classification ability of the network for trees and low_veg. This is due to the uneven distribution of the points of this type of object, and the addition of the radius constraint can effectively extract the local high-dimensional features and geometric features and enhance the network's recognition ability.

(3) Efficacy of M-Loss: From the experimental results of Model A and Model D, it can be seen that the addition of the M-Loss strategy improves both the overall OA and mF1, indicating that it is effective in supervising the network across multiple levels.

4.4. Complexity and Runtime Analysis

This study investigates the influence of the different modules of the MGFE-T network on its complexity and running time. Table 8 presents the differences in parameter counts, complexity, and execution time among various models trained over 200 epochs with the same input data. Relative to the baseline network (Point Transformer), our method (Model A) exhibits a 14.57% increase in parameter counts and a 10.65% increase in complexity. This increase is primarily due to the integration of geometric features from point clouds, which are subsequently fused with self-learned features. The experimental results of Model A and Model B indicate that embedding geometric features increases the model's parameters by 1.1 M and computational complexity by 1 G. A comparison between Model A and Model D reveals that the inclusion of M-Loss does not significantly increase the parameter count. The experimental results for Model C demonstrate that the FR-DKNN method does not significantly impact the parameter count or complexity of the model. This is because the FR-DKNN module operates independently of convolutional operators and does not involve the computation of model parameter weights.

Table 8. Parameters, complexity, and runtime of different models (200 epochs). The "M" stands for million, and the "G" stands for billion. Model A refers to our proposed MGFE-T method. Model B does not embed geometric features, Model C uses KNN for neighborhood queries instead of FR-DKNN, and Model D does not utilize multilevel loss aggregation.

Model	#Params	#FLOPs	#Time
baseline	7.41 M	9.39 G	2 h 48 m
A (Ours)	8.49 M	10.39 G	3 h 29 m
B (No GFE-T)	7.59 M	9.61 G	3 h 11 m
C (No FR-DKNN)	8.49 M	10.39 G	2 h 55 m
D (No M-Loss)	8.31 M	9.84 G	3 h 27 m

Custom operators are extensively used in point cloud deep learning tasks. Therefore, in addition to comparing model parameters, it is essential to compare the total program execution times of different models. Table 8 illustrates the training time for the ISPRS dataset over 200 epochs, including all steps from the initial data input, partitioning, augmentation to the final output of model training results. Compared to the baseline model, the runtime of MGFE-T has increased by 41 min. The FR-DKNN method, which applies a fixed-radius constraint to KNN queries, is mainly responsible for this increase in processing time. In addition, comparing Model A and Model B, the GFE-T module only added 18 min (9.4%) to the training time, thanks to our streamlined approach to geometric feature computation. Meanwhile, it is worth noting that the M-Loss has a negligible impact on the additional training and testing time required for the network. This is due to the fact that it predominantly relies on CUDA-based dense computations.

5. Generalization Performance

To further validate the generalization capability of our model, two experiments were conducted. The first experiment involved training the model on the LASDU dataset and then testing it on the LASDU test set. The second experiment involved training the model on the DFC2019 dataset and then applying it directly to the LASDU test set without retraining. Due to differences in semantic categories between the two datasets, performance was evaluated on the three common categories: Ground, Building, and Trees. Precision, as defined by Equation (8), was used as the evaluation metric.

Table 9 presents the results of the generalization experiments. The increase in Precision for Ground in Experiment 2 by 6.2 can be attributed to the DFC2019 dataset providing more training data, which allowed the network to learn more representative ground features. Given the similarity of ground features across datasets, the higher precision in Experiment 2 compared to Experiment 1 is reasonable. Conversely, the accuracy for Building and Trees decreased slightly due to differences in building types and tree species between the LASDU and DFC2019 datasets. This is because the features learned from DFC2019 were insufficient for accurately representing Building and Trees in LASDU. This highlights the importance of dataset diversity and representativeness for model generalization. Overall, the results presented in Table 9 demonstrate that our method exhibits reliable generalization capability.

Table 9. Generalization performance results on the LASDU dataset.

Experiment	Ground	Building	Trees
exp. 1 Train on LASDU, Test on LASDU	92.6	96.2	84.1
exp. 2 Train on DFC2019, Test on LASDU	98.8	95.6	82.2
(Δ)	(+6.2)	(−0.6)	(−1.9)

6. Conclusions

In this paper, we proposed a multilevel geometric feature embedding transformer for airborne point cloud semantic segmentation networks, embedding low-dimensional geometric features to enhance the network's ability to learn local structural features during feature extraction via the self-attention mechanism. To address the issue of KNN queries providing scale-inconsistent neighborhood ranges, we introduced the FR-DKNN method, extending KNN neighborhood point queries within a fixed-radius range to tackle the challenge of uneven point cloud density. Finally, we employed a multilevel loss aggregation strategy to achieve multilevel supervised learning in the network. Experiments conducted on three popular benchmark datasets demonstrate the outstanding performance of the proposed method. Additionally, we conducted generalization experiments and ablation studies on the DFC2019 and LASDU datasets, further validating the effectiveness of the proposed method.

The transformer self-attention mechanism performs well in urban scenes and shows potential for other point cloud processing tasks. In future research, we aim to explore its performance in weakly supervised semantic segmentation and instance segmentation and extend its application to other environments, such as railways and forests.

Author Contributions: Data curation, Z.L.; Funding acquisition, X.L.; Methodology, Z.L.; Validation, Z.L.; Writing—original draft, Z.L.; Writing—review and editing, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China (No. 42130105), the Science and Technology Research and Development Program of China State Railway Group Co., Ltd. (No. L2023G016), the Fundamental Research Funds for the Central Universities (No. 2042024kf0029), and the project of Basic Geographic Information Application by the Hubei Provincial Geographical National Condition Monitoring Center (No. 2023-2-06).

Data Availability Statement: The data supporting the reported results in this work are available as open datasets. The three datasets used (LASDU [39], DFC2019 [40], and ISPRS [41]) can be accessed through the sources provided in the references or by contacting the respective authors.

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

References

1. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-View Convolutional Neural Networks for 3D Shape Recognition. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 945–953.
2. Qi, C.R.; Su, H.; Niessner, M.; Dai, A.; Yan, M.; Guibas, L.J. Volumetric and Multi-View CNNs for Object Classification on 3D Data. In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition, (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 5648–5656.
3. Maturana, D.; Scherer, S. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
4. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
5. 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), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85.

6. 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 30, Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017*; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 30.
7. 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.
8. Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.; Elhoseiny, M.; Ghanem, B. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 23192–23204.
9. 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.
10. 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.
11. Wu, W.; Qi, Z.; Fuxin, L. PointConv: Deep Convolutional Networks on 3D Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9621–9630.
12. 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.
13. Simonovsky, M.; Komodakis, N. Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 29–38.
14. 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]
15. Liu, Y.; Fan, B.; Xiang, S.; Pan, C. Relation-Shape Convolutional Neural Network for Point Cloud Analysis. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 8887–8896.
16. 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 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.
17. 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]
18. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
19. Lin, Y.; Vosselman, G.; Cao, Y.; Yang, M.Y. Local and Global Encoder Network for Semantic Segmentation of Airborne Laser Scanning Point Clouds. *ISPRS J. Photogramm. Remote Sens.* **2021**, *176*, 151–168. [CrossRef]
20. Yousefhussien, M.; Kelbe, D.J.; Ientilucci, E.J.; Salvaggio, C. A Multi-Scale Fully Convolutional Network for Semantic Labeling of 3D Point Clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 191–204. [CrossRef]
21. Zhang, K.; Ye, L.; Xiao, W.; Sheng, Y.; Zhang, S.; Tao, X.; Zhou, Y. A Dual Attention Neural Network for Airborne LiDAR Point Cloud Semantic Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5704617. [CrossRef]
22. Lai, X.; Pian, W.; BO, L.; He, L. A Building Extraction Method Based on IGA That Fuses Point Cloud and Image Data. *J. Infrared Millim. Waves* **2023**, *43*, 116–125.
23. He, P.; Gao, K.; Liu, W.; Song, W.; Hu, Q.; Cheng, X.; Li, S. OFFS-Net: Optimal Feature Fusion-Based Spectral Information Network for Airborne Point Cloud Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 141–152. [CrossRef]
24. Yang, Y.; Tang, R.; Wang, J.; Xia, M. A Hierarchical Deep Neural Network with Iterative Features for Semantic Labeling of Airborne LiDAR Point Clouds. *Comput. Geosci.* **2021**, *157*, 104932. [CrossRef]
25. Ma, L.; Li, J.; Guan, H.; Yu, Y.; Chen, Y. STN: Saliency-Guided Transformer Network for Point-Wise Semantic Segmentation of Urban Scenes. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 7004405. [CrossRef]
26. Li, W.; Wang, F.-D.; Xia, G.-S. A Geometry-Attentional Network for ALS Point Cloud Classification. *ISPRS J. Photogramm. Remote Sens.* **2020**, *164*, 26–40. [CrossRef]
27. Jiang, T.; Wang, Y.; Liu, S.; Cong, Y.; Dai, L.; Sun, J. Local and Global Structure for Urban ALS Point Cloud Semantic Segmentation With Ground-Aware Attention. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5702615. [CrossRef]
28. Jin, S.; Su, Y.; Zhao, X.; Hu, T.; Guo, Q. A Point-Based Fully Convolutional Neural Network for Airborne LiDAR Ground Point Filtering in Forested Environments. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 3958–3974. [CrossRef]
29. Huang, R.; Xu, Y.; Stilla, U. GraNet: Global Relation-Aware Attentional Network for Semantic Segmentation of ALS Point Clouds. *ISPRS J. Photogramm. Remote Sens.* **2021**, *177*, 1–20. [CrossRef]
30. Mao, Y.; Chen, K.; Diao, W.; Sun, X.; Lu, X.; Fu, K.; Weinmann, M. Beyond Single Receptive Field: A Receptive Field Fusion-and-Stratification Network for Airborne Laser Scanning Point Cloud Classification. *ISPRS J. Photogramm. Remote Sens.* **2022**, *188*, 45–61. [CrossRef]
31. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.; Koltun, V. Point Transformer. *arXiv* **2021**, arXiv:2012.09164. [CrossRef]

32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.
33. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Minneapolis, MN, USA, 2–7 June 2019.
34. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *arXiv* **2019**, arXiv:1901.02860.
35. Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; Shlens, J. Stand-Alone Self-Attention in Vision Models. In *Advances in Neural Information Processing Systems 32, Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019) Vancouver, BC, Canada, 8–14 December 2019*; Curran Associates, Inc.: New York, NY, USA, 2020; Volume 32.
36. Zhao, H.; Jia, J.; Koltun, V. Exploring Self-Attention for Image Recognition. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 10073–10082.
37. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
38. Wang, L.; Wu, J.; Liu, X.; Ma, X.; Cheng, J. Semantic Segmentation of Large-Scale Point Clouds Based on Dilated Nearest Neighbors Graph. *Complex Intell. Syst.* **2022**, *8*, 3833–3845. [CrossRef]
39. Ye, Z.; Xu, Y.; Huang, R.; Tong, X.; Li, X.; Liu, X.; Luan, K.; Hoegner, L.; Stilla, U. LASDU: A Large-Scale Aerial LiDAR Dataset for Semantic Labeling in Dense Urban Areas. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 450. [CrossRef]
40. Le Saux, B.; Yokoya, N.; Haensch, R.; Brown, M. 2019 IEEE GRSS Data Fusion Contest: Large-Scale Semantic 3D Reconstruction [Technical Committees]. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 33–36. [CrossRef]
41. Niemeyer, J.; Rottensteiner, F.; Soergel, U. Contextual Classification of Lidar Data and Building Object Detection in Urban Areas. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 152–165. [CrossRef]
42. Li, J.; Weinmann, M.; Sun, X.; Diao, W.; Feng, Y.; Hinz, S.; Fu, K. VD-LAB: A View-Decoupled Network with Local-Global Aggregation Bridge for Airborne Laser Scanning Point Cloud Classification. *ISPRS J. Photogramm. Remote Sens.* **2022**, *186*, 19–33. [CrossRef]
43. Zeng, T.; Luo, F.; Guo, T.; Gong, X.; Xue, J.; Li, H. Recurrent Residual Dual Attention Network for Airborne Laser Scanning Point Cloud Semantic Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5702614. [CrossRef]
44. Zeng, T.; Luo, F.; Guo, T.; Gong, X.; Xue, J.; Li, H. Multilevel Context Feature Fusion for Semantic Segmentation of ALS Point Cloud. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 5506605. [CrossRef]
45. Zhang, R.; Chen, S.; Wang, X.; Zhang, Y. IPCONV: Convolution with Multiple Different Kernels for Point Cloud Semantic Segmentation. *Remote Sens.* **2023**, *15*, 5136. [CrossRef]
46. Pirotti, F.; Tonion, F. Classification of aerial laser scanning point clouds using machine learning: A comparison between random forest and tensorflow. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *XLII-2-W13*, 1105–1111. [CrossRef]
47. Atik, M.E.; Duran, Z.; Seker, D.Z. Machine Learning-Based Supervised Classification of Point Clouds Using Multiscale Geometric Features. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 187. [CrossRef]
48. Feng, C.-C.; Guo, Z. A Hierarchical Approach for Point Cloud Classification With 3D Contextual Features. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 5036–5048. [CrossRef]

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.