

Article

Research on Point Cloud Segmentation Method for Ancient Building Interior Scenes Based on Geometric Feature Multi-Scale Network

Jian Ma and Dong Wu *

School of Management, Shenyang Jianzhu University, Shenyang 110168, China; majian88@sjzu.edu.cn

* Correspondence: cehui0129@126.com

Abstract

Aiming at addressing the problems of complex geometric features, high inter-class similarity, and insufficient single-scale information in semantic segmentation of point clouds for ancient building interior components, this paper takes the Ba Wang Academy of Shenyang Jianzhu University as the research object and proposes a Geometric Feature Multi-scale Network (GFMN). First, a point cloud dataset containing five types of components—windows, beams, walls, roofs, and columns—was collected and constructed using a FARO Focus3D X330 terrestrial laser scanner (FARO Technologies, Lake Mary, Florida, USA). Second, 46-dimensional handcrafted geometric descriptors were extracted for each discrete point from four aspects: basic point attributes, local geometric features, density and scale features, and multi-scale fusion. On this basis, features were grouped according to semantics and fed into independent encoding branches, where a gated adaptive fusion mechanism was employed to dynamically adjust the contribution of each branch, and optimization was performed in combination with a prototype classification head and a joint loss function. Experimental results show that the proposed method achieved an overall accuracy of 93.17% on the test set, significantly outperforming state-of-the-art methods such as PointNet, PointNet++, Point Transformer, and Point Cloud Transformer. This study provides an effective solution for high-precision semantic segmentation of ancient building interior components.

Keywords: ancient buildings; point cloud semantic segmentation; geometric features; multi-branch network

Academic Editor: Beiwen Li

Received: 27 June 2026

Revised: 20 July 2026

Accepted: 31 July 2026

Published: 1 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Ancient buildings, as precious carriers of human civilization, embody profound historical, cultural, and artistic values and represent important manifestations of collective memory and human wisdom. Traditional Chinese timber architecture, in particular, is renowned for its unique construction techniques, sophisticated mortise-and-tenon joinery, and rich decorative arts, occupying a distinctive position in the history of world architecture [1]. However, due to natural weathering, fires, earthquakes, and human-induced damage, a large number of ancient buildings are facing severe risks of deterioration, destruction, and even disappearance [2]. Therefore, systematic documentation, scientific conservation, and digital archiving of ancient buildings have become increasingly urgent.

In recent years, the rapid development of three-dimensional laser scanning technology has provided a new technical pathway for the digital preservation of ancient buildings. Compared with traditional close-range photogrammetry and manual surveying methods, Terrestrial Laser Scanning (TLS) can acquire three-dimensional geometric information of building surfaces in a non-contact, high-accuracy, and high-density manner, comprehensively recording the spatial locations, boundary contours, and local details of both interior and exterior architectural components [3,4]. However, raw point cloud data are characterized by massive volume, discreteness, irregularity, and a lack of structured information, making them difficult to directly utilize for parametric modeling and semantic analysis of ancient buildings [5,6]. Consequently, how to efficiently and accurately identify and extract semantically meaningful architectural components from raw point clouds has become a key scientific challenge in the fields of digital heritage preservation and Heritage Building Information Modeling (HBIM) reconstruction [7,8].

Point cloud semantic segmentation, as a core technology for three-dimensional scene understanding, aims to assign a semantic label to each individual point and partition the raw point cloud into subsets with distinct semantic meanings. It serves as a bridge between low-level geometric data and high-level architectural semantics [9]. Over the years, extensive research has been conducted on point cloud segmentation and classification. According to the technological development trajectory, point cloud semantic segmentation methods can generally be categorized into three groups: traditional segmentation methods, machine learning-based methods, and deep learning-based methods.

Traditional segmentation methods mainly rely on geometric constraints and statistical rules, partitioning point clouds into regions with homogeneous geometric and radiometric characteristics through manually designed features [10,11]. These methods primarily include edge detection-based approaches, region growing-based approaches, model fitting-based approaches, and feature clustering-based approaches. In practical applications, traditional methods are sensitive to noise and outliers, often resulting in over-segmentation or under-segmentation. Moreover, they heavily depend on expert knowledge for parameter tuning and exhibit limited generalization capability when confronted with complex and diverse ancient architectural scenes [12].

Machine learning-based methods perform semantic labeling by first manually extracting geometric, radiometric, and multi-scale features and then employing classifiers such as Random Forests and Support Vector Machines. Grilli et al. [13] systematically analyzed the classification performance of covariance-based geometric features under different neighborhood radii, providing important insights for machine learning classification of cultural heritage point clouds. Teruggi et al. [14] proposed a Multi-Level and Multi-Resolution (MLMR) Random Forest classification framework that achieved hierarchical semantic segmentation from primary structural elements to fine architectural details. Croce et al. [15] applied a Random Forest classifier to semantically segment the point cloud of the Santa Caterina Church in Pisa and further utilized the segmentation results for HBIM reconstruction. Matrone et al. [16] systematically compared various machine learning and deep learning approaches and proposed the DGCNN-Mod + 3Dfeat framework, which combines the advantages of both paradigms. Nevertheless, the performance of machine learning methods is highly dependent on the design and selection of feature descriptors, limiting their generalization ability in complex scenarios [17].

Benefiting from their powerful automatic feature learning capabilities, deep learning-based methods have become the state-of-the-art direction for point cloud semantic segmentation research. According to how raw data are processed, these methods can be further divided into projection-based methods [18–21], voxel-based methods [22–26], and point-based methods.

Projection-based methods transform three-dimensional point clouds into multi-view two-dimensional images or range images, perform segmentation using mature two-dimensional convolutional neural networks, and subsequently map the results back to three-dimensional space. However, the projection process inevitably causes the loss of three-dimensional geometric information, and segmentation performance is highly sensitive to the selection and number of viewpoints, making it difficult to preserve the fine geometric details of complex ancient architectural structures [21].

Voxel-based methods convert irregular point clouds into regular three-dimensional voxel grids and employ three-dimensional convolutional neural networks for feature extraction and segmentation. Maturana et al. [22] proposed VoxNet, the first framework to input voxelized point clouds into a 3D CNN for real-time object recognition. Tchapmi et al. [23] introduced SEGCloud, which combines a 3D fully convolutional network with tri-linear interpolation to achieve point-wise semantic segmentation and further refines the results using Conditional Random Fields. However, voxel-based methods suffer from resolution limitations, leading to the loss of fine-grained details. Moreover, their computational and memory requirements increase cubically with voxel resolution, posing significant challenges when processing large-scale and high-density point clouds of ancient buildings [26].

Point-based methods directly operate on raw point clouds, avoiding information loss caused by data format conversion and preserving the spatial geometric structure of point clouds to the greatest extent. According to network architecture, these methods can be further categorized into Multi-Layer Perceptron (MLP)-based methods [27–30], Graph Convolutional Network (GCN)-based methods [31–34], and Transformer-based methods [35–37]. PointNet, proposed by Qi et al. [27], was the first framework to directly process raw point clouds in an end-to-end manner, effectively addressing the challenges of permutation invariance and unordered point sets. However, PointNet neglects local neighborhood information and thus struggles to capture fine-grained geometric structures. To address this limitation, Qi et al. [28] proposed PointNet++, which introduces a hierarchical local feature extraction architecture and significantly improves the perception of local geometric details. Nevertheless, pure MLP-based architectures remain limited in modeling long-range dependencies within complex geometric structures. GCN-based methods represent point clouds as graph structures and employ graph convolution or edge convolution operations to capture topological relationships and local geometric features among points. However, GCN methods typically require reconstructing graph adjacency relationships for each batch of data, resulting in high computational complexity and efficiency bottlenecks when processing large-scale point clouds [34]. Transformer-based methods introduce self-attention mechanisms into point cloud processing, effectively modeling long-range dependencies and global contextual information. Zhao et al. [35] proposed Point Transformer, which employs vector self-attention and positional encoding to facilitate efficient local feature interaction through residual point transformer blocks, achieving state-of-the-art performance on multiple point cloud segmentation benchmarks. Nevertheless, the computational and memory costs of Transformer-based methods grow quadratically with point cloud size, limiting their applicability in large-scale ancient architectural scenes.

Although significant progress has been achieved in point cloud semantic segmentation, directly applying existing methods to indoor scenes of ancient buildings still faces several challenges. First, ancient architectural components exhibit complex geometric shapes and high inter-class similarity, making it difficult to effectively distinguish different component categories using geometric information at a single scale alone [38]. Second, significant scale variations exist among indoor architectural components, making a single

neighborhood scale insufficient to meet the feature extraction requirements of components with different sizes. Third, existing deep learning methods often underutilize manually designed geometric descriptors and therefore fail to fully exploit the subtle yet critical geometric differences among ancient architectural components.

To address these challenges, this study takes the Ba Wang Academy of Shenyang Jianzhu University as the research object and proposes a Geometric Feature Multi-scale Network (GFMN). Reconstructed using preserved components from the Ming Dynasty Ba Wang Temple, the academy serves both cultural heritage preservation and educational purposes. Its interior space contains beams, columns, windows, walls, and roof structures with complex geometric forms, spatial arrangements, and local details, making it a representative case for semantic segmentation research in ancient building interior scenes. The main contributions of this study are summarized as follows:

1. A point cloud dataset of ancient building interiors was collected and constructed using a FARO Focus3D X330 terrestrial laser scanner. The dataset includes five categories of architectural components: windows, beams, walls, roofs, and columns, providing a reliable foundation for subsequent research.
2. A comprehensive 46-dimensional handcrafted geometric descriptor was extracted for each point from four aspects: basic point attributes, local geometric features, density and scale features, and multi-scale fusion features. This descriptor provides a comprehensive representation of local geometric information within the point cloud.
3. The extracted features were grouped according to semantic characteristics and fed into independent encoding branches. A gated adaptive fusion mechanism was designed to dynamically adjust the contribution of each branch, while a prototype-based classification head and a joint loss function were employed for optimization. This framework effectively addresses the complementarity and fusion of heterogeneous geometric features and significantly improves the semantic segmentation accuracy of ancient building interior components.

2. Research Object and Data Acquisition

This study takes the Ba Wang Academy of Shenyang Jianzhu University as the research object. Located within the campus, the Ba Wang Academy is a reconstructed ancient building site that integrates historical and cultural inheritance with educational functions. It was rebuilt on the basis of surviving components from the Ming Dynasty Ba Wang Temple, and thus possesses both significant heritage conservation value and strong educational and research significance. Compared with ordinary modern buildings, the beams, columns, windows, walls, and roof structures in the interior space of the Ba Wang Academy exhibit more complex geometric forms, spatial layouts, and local details. Moreover, different components often show high visual similarity, making it a representative object for semantic segmentation research on point clouds of ancient building interior scenes.

In terms of data acquisition, a FARO Focus3D X330 terrestrial laser scanner was used to collect point cloud data of the interior space of the Ba Wang Academy (as shown in Figure 1), based on which a three-dimensional point cloud model of the indoor scene was constructed (as shown in Figure 2). Three-dimensional laser scanning can comprehensively record the spatial positions, boundary contours, and local geometric details of interior components in ancient buildings. Compared with traditional two-dimensional imaging methods, it is more suitable for representing the three-dimensional spatial relationships in the complex structures of ancient architecture. The point cloud data acquired in this way provide a reliable raw data foundation for subsequent component recognition, feature extraction, and semantic segmentation.

In terms of data content, the point cloud dataset constructed in this study mainly covers five categories of interior components: windows, beams, walls, roofs, and columns.

These categories essentially cover the main enclosure, load-bearing, and opening structures in ancient building interior spaces, and also reflect the common structural variations and semantic boundary characteristics in such scenes.



Figure 1. Data Acquisition.

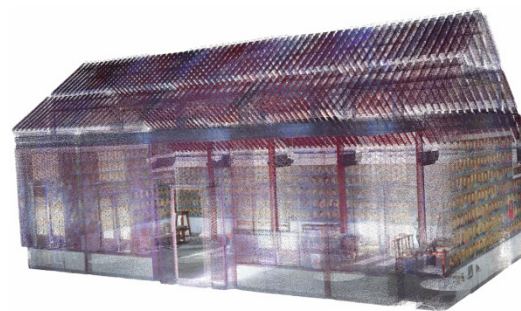


Figure 2. 3D Point Cloud Model.

3. Local Geometric Feature Extraction of Point Clouds

To effectively distinguish different categories of interior component point clouds, such as windows, beams, walls, roofs, and columns, this study no longer relies solely on the absolute coordinates of discrete points. Instead, it further exploits the geometric structural features embedded in the local neighborhood surrounding each point. For point cloud semantic classification and segmentation tasks, a single point typically lacks sufficient discriminative information. In contrast, the geometric properties of its local neighborhood, such as shape, orientation, planarity, and scale distribution, can more fully reflect the spatial structural differences among architectural components. Therefore, this study constructs point-level descriptors from four aspects: basic point attributes, local geometric features, density and scale features, and multi-scale features, which are used for subsequent model training.

Let the input point cloud be $\mathbf{P} = \{p_i \mid p_i = (x_i, y_i, z_i)^T, i = 1, 2, \dots, N\}$, where N denotes the number of points. For any point p_i , its local neighborhood can be defined as:

$$\Omega_i = \{p_j \in \mathbf{P} \mid \|p_j - p_i\|^2 \leq r\} \quad (1)$$

In this study, the neighborhood is constructed using a fixed radius r to accommodate feature extraction under different point-density conditions.

3.1. Basic Point Attribute Features

First, the basic spatial attributes of each point are extracted, including its coordinates (x_i, y_i, z_i) and normalized height. The normalized height is calculated as follows:

$$\hat{z}_i = \frac{z_i - z_{\min}}{z_{\max} - z_{\min}} \quad (2)$$

where $z_{\min}$ and $z_{\max}$ denote the minimum and maximum heights in the current point cloud scene, respectively.

3.2. Local Geometric Features

Local geometric features are the core for distinguishing component categories. Specifically, the indicators include linearity L , planarity P , sphericity S , anisotropy A , curvature C , omnivariance O , and eigenentropy E . The calculation procedure is as follows.

For the neighborhood Ω_i of point p_i , the centroid is first computed:

$$\bar{p}_i = \frac{1}{|N_{\Omega_i}|} \sum_{p_j \in \Omega_i} p_j \quad (3)$$

Then, the covariance matrix is constructed:

$$C_i = \frac{1}{|N_{\Omega_i}|} \sum_{p_j \in \Omega_i} (p_j - \bar{p}_i)(p_j - \bar{p}_i)^T \quad (4)$$

In the formula, N_{Ω_i} denotes the number of points within the neighborhood Ω_i .

By performing eigen decomposition on C_i , three eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ are obtained. These three eigenvalues describe the distribution pattern of the local point set along the principal directions and form the basis for constructing structural geometric descriptors.

Based on the eigenvalues, the linearity L , planarity P , sphericity S , anisotropy A , curvature C , omnivariance O , and eigenentropy E are further computed:

$$L = \frac{\lambda_1 - \lambda_2}{\lambda_1} \quad (5)$$

$$P = \frac{\lambda_2 - \lambda_3}{\lambda_1} \quad (6)$$

$$S = \frac{\lambda_3}{\lambda_1} \quad (7)$$

$$A = \frac{\lambda_1 - \lambda_3}{\lambda_1} \quad (8)$$

$$C = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad (9)$$

$$O = (\lambda_1 \lambda_2 \lambda_3)^{1/3} \quad (10)$$

$$E = -\sum_{m=1}^3 \frac{\lambda_m}{\lambda_1 + \lambda_2 + \lambda_3} \ln \left(\frac{\lambda_m}{\lambda_1 + \lambda_2 + \lambda_3} \right) \quad (11)$$

In addition, the local normal vector can be derived from the eigenvector corresponding to the smallest eigenvalue, namely:

$$\mathbf{n} = \mathbf{v}_3 \quad (12)$$

where $\mathbf{v}_3$ is the corresponding eigenvector.

The normal vector can further be used to derive the verticality feature V , which is used to characterize the orientation difference of a point cloud patch relative to the vertical

direction. For structural elements such as walls, roofs, beams, and columns, this type of directional feature is highly discriminative.

$$\mathbf{V} = 1 - \left| \mathbf{n}^T \mathbf{e}_z \right| \quad (13)$$

where $\mathbf{e}_z = (0, 0, 1)^T$ represents the unit vector in the vertical direction. The symbol $|\cdot|$ denotes the absolute value operation; $\cdot$ stands for the inner product of two vectors; $V \in [0, 1]$.

3.3. Density and Scale Features

In addition to geometric shape itself, the distribution density and scale information of points within the local neighborhood are also important for discrimination. These features are especially effective for distinguishing large planar components, elongated components, and sparse boundary regions.

Let the set of distances from the points in Ω_i to the central point p_i be given; then, the local mean distance, variance, minimum value, and maximum value can be defined as:

$$\mu_i = \frac{1}{|N_{\Omega_i}|} \sum_{\mathbf{p}_j \in \Omega_i} d_{ij} \quad (14)$$

$$\sigma_i^2 = \frac{1}{|N_{\Omega_i}|} \sum_{\mathbf{p}_j \in \Omega_i} (d_{ij} - \mu_i)^2 \quad (15)$$

$$d_i^{\min} = \min_{\mathbf{p}_j \in \Omega_i} d_{ij} \quad (16)$$

$$d_i^{\max} = \max_{\mathbf{p}_j \in \Omega_i} d_{ij} \quad (17)$$

where N_{Ω_i} denotes the number of points within the neighborhood Ω_i . d_{ij} denotes the Euclidean distance from the central point p_i to the j -th discrete point within its neighborhood Ω_i .

A density surrogate measure can then be constructed to characterize the degree of point aggregation in local space, with its formula given as:

$$\rho_i = \frac{1}{\mu_i} \quad (18)$$

Meanwhile, along the height direction, neighborhood height variations can also be summarized, including the height standard deviation and the range, with the corresponding formulas:

$$\sigma_{z,i} = \text{std} \{ z_j \mid \mathbf{p}_j \in \Omega_i \} \quad (19)$$

In this formula, the vertical bar $|$ serves as a set constraint separator. This equation calculates the standard deviation of the vertical coordinate z_j of all points within the neighborhood Ω_i of the current point p_i , which characterizes the dispersion degree of height fluctuation of the local point cloud.

$$R_{z,i} = \max \{ z_j \} - \min \{ z_j \}, \mathbf{p}_j \in \Omega_i \quad (20)$$

This formula first extracts the maximum and minimum values of the vertical coordinates of points in the neighborhood Ω_i of p_i , and subtracts the two values to obtain the neighborhood height range, so as to quantify the overall vertical span of local structures.

3.4. Multi-Scale Fusion Features

Considering the significant differences in spatial scale among different interior components, a single neighborhood scale is insufficient to simultaneously capture large planar structures such as walls and roofs as well as local detail structures such as windows and columns. Therefore, this study further extracts the same type of features at multiple scales. These multi-scale features capture local edges, component surfaces, and broader spatial layout information at the same time.

Let multiple neighborhood scales be denoted as s_m , where s_m may represent different radii r_m or different numbers of nearest neighbors k_m . For each scale s_m , the above local geometric features, density features, and directional features can be computed, and the multi-scale features are then concatenated:

$$f_i = [f_i^{(s_1)} \| f_i^{(s_2)} \| \dots \| f_i^{(s_M)}] \quad (21)$$

where $f_i^{(s_m)}$ denotes the feature vector of point p_i at scale s_m .

3.5. Feature Representation

To address the challenging semantic segmentation task characterized by drastically varying geometric shapes and low inter-class feature distinguishability of indoor ancient building components, this paper constructs a complete hierarchical point-level hand-crafted geometric descriptor, forming a 46-dimensional independent feature vector that is solely computed from the 3D coordinates of point clouds and multi-radius neighborhood statistics. The full feature set is divided into four modules with detailed dimension allocation and feature specifications as follows: The first module consists of 4 basic point attribute features, including the raw horizontal coordinate x_i vertical coordinate y_i , vertical coordinate z_i , and normalized height of each point, which provide a global spatial reference for the entire scene. The second module contains 9 single-scale local geometric features derived from the eigenvalue decomposition of the neighborhood covariance matrix, namely linearity, planarity, sphericity, anisotropy, curvature, volume metric, feature entropy, normal verticality, and 3D unit normal vector. These features are adopted to distinguish geometric shapes of planar, linear and curved components. The third module covers 8 single-scale density and height statistical features: neighborhood mean distance, distance variance, minimum distance, maximum distance, local density proxy, neighborhood height standard deviation, neighborhood height range, and the number of neighborhood points. They characterize differences in local point cloud density, vertical undulations and structural boundaries. The fourth module is composed of 25 multi-scale fused features. Three distinct neighborhood scales s_1 , s_2 , s_3 are defined, and eight statistical metrics (linearity, planarity, sphericity, verticality, mean distance, density proxy, height standard deviation, height range) are extracted repeatedly for each scale, yielding a total of 24 dimensions across the three scales. An additional 1 dimension of multi-scale fusion entropy is supplemented to compensate for the limitation that a single neighborhood radius cannot simultaneously capture large planar components and fine-grained structural details.

In summary, for each discrete point p_i , the final feature vector can be expressed as:

$$\mathbf{x}_i = [\mathbf{x}_i^{\text{basic}} \| \mathbf{x}_i^{\text{geo}} \| \mathbf{x}_i^{\text{density}} \| \mathbf{x}_i^{\text{multi-scale}}] \quad (22)$$

where $\mathbf{x}_i^{\text{basic}}$ denotes the basic point attributes; $\mathbf{x}_i^{\text{geo}}$ denotes the local geometric features; $\mathbf{x}_i^{\text{density}}$ denotes the density and scale statistical features; $\mathbf{x}_i^{\text{multi-scale}}$ denotes the multi-scale fusion features; The symbol $\|$ denotes feature concatenation.

This feature representation jointly captures point-level positional prior, contextual local geometric shape, neighborhood distribution patterns, and cross-scale structural information, and can therefore better support classification and segmentation tasks for indoor component point clouds.

4. Point Cloud Semantic Segmentation Based on the GFMN

To address the semantic segmentation task of interior architectural components such as windows, beams, walls, roofs, and columns, this study proposes a Geometric Feature Multi-scale Network (GFMN). The core idea of the proposed method is as follows: first, handcrafted features with explicit geometric significance are constructed for each discrete point in the point cloud. These features are then divided into different feature groups according to their semantic meaning and statistical scale, and subsequently fed into independent encoding branches for representation learning. Unlike approaches that simply concatenate all features and directly input them into a classifier, the proposed method progressively refines feature information through a grouped encoding–gated fusion–prototype classification framework, thereby more effectively exploiting the complementary relationships among different geometric descriptors. The overall network architecture is illustrated in Figure 3.

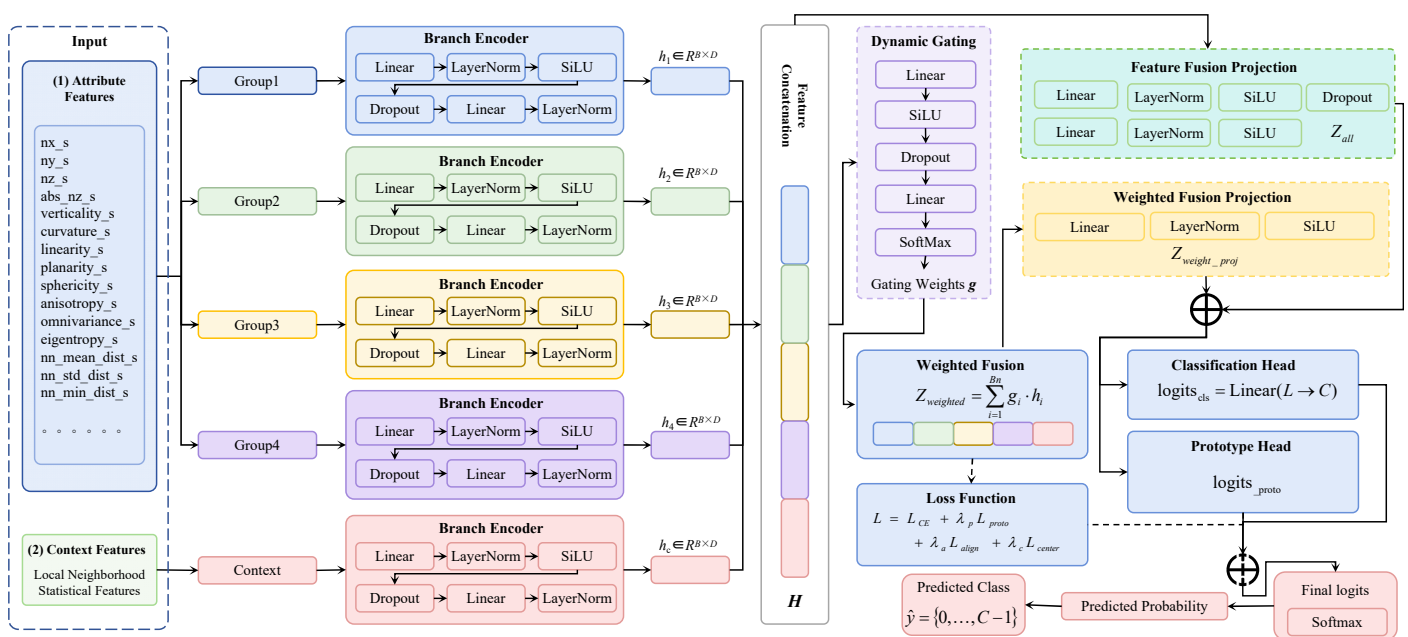


Figure 3. Architecture of the GFMN Model.

In indoor scenes, different architectural components often exhibit distinct local geometric patterns. For example, walls and roofs generally present strong planar structures, whereas beams and columns exhibit pronounced linear extension characteristics. Windows, on the other hand, are more frequently characterized by boundaries, openings, and local structural discontinuities. Therefore, if the model relies solely on features from a single scale or a single feature type, it is often difficult to comprehensively capture these structural differences. To address this issue, this study not only preserves the original point-level handcrafted features but also introduces local neighborhood contextual statistical features to enhance the model's perception of the surrounding spatial environment. Ultimately, the network outputs the class probability for each discrete point, enabling point-wise classification of interior architectural components and thereby achieving point cloud segmentation.

Assume that, after preprocessing, the input point cloud consists of N discrete points. The original feature representation of each point is denoted as $x_i \in R^{46}, i = 1, 2, \dots, N$, and the corresponding class label is denoted as $y_i \in \{0, 1, \dots, C-1\}$. Let C represent the number of interior component categories.

Meanwhile, a local contextual feature $c_i \in R^{12}$ is constructed for each point to describe the spatial distribution of its surrounding neighborhood. The model then outputs the predicted class label for that point.

4.1. Multi-Branch Encoder Design

Since different feature groups describe different types of information—for example, basic positional features, geometric structural features, and local statistical features exhibit significantly different distributions—an independent branch encoder is designed for each feature group.

All encoders adopt the same bottleneck architecture, consisting of two fully connected layers. After each layer, Layer Normalization (LayerNorm), the SiLU activation function, and Dropout regularization are applied sequentially. Taking the k -th branch as an example, the encoding process can be formulated as follows.

Let the input vector of the k -th branch be $\mathbf{g}_k$. The first linear layer projects it to a higher-dimensional space of dimension d_b , followed by layer normalization and nonlinear activation:

$$\mathbf{u}_k^{(1)} = \text{SiLU}(\text{LayerNorm}(\mathbf{W}_k^{(1)}\mathbf{g}_k + \mathbf{b}_k^{(1)})) \quad (23)$$

where $\mathbf{W}_k^{(1)}$ and $\mathbf{b}_k^{(1)}$ are learnable parameters, $\text{LayerNorm}(\cdot)$ denotes layer normalization, and $\text{SiLU}(x) = x \cdot \text{sigmoid}(x)$ (with $\text{sigmoid}(x)$ being the sigmoid function) is the activation function.

A dropout operation is then applied to the intermediate representation:

$$\mathbf{u}_{k \text{ drop}}^{(1)} = \text{Dropout}(\mathbf{u}_k^{(1)}, p) \quad (24)$$

The second linear layer keeps the feature dimension unchanged and again applies layer normalization and SiLU activation:

$$\mathbf{u}_k^{(2)} = \text{SiLU}(\text{LayerNorm}(\mathbf{W}_k^{(2)}\mathbf{g}_k + \mathbf{b}_k^{(2)})) \quad (25)$$

where $\mathbf{W}_k^{(2)}$ and $\mathbf{b}_k^{(2)}$ are also learnable parameters.

Finally, another dropout operation is applied to obtain the final encoded output of the branch:

$$\mathbf{h}_k = \text{Dropout}(\mathbf{u}_k^{(2)}, p) \quad (26)$$

In the experiments, the dropout rate is set to $p = 0.20$, and no additional residual connections are introduced between the two layers.

4.2. Gated Adaptive Fusion Mechanism

To adaptively adjust the contribution of different branches according to the local structure of each point, a gated fusion mechanism is designed.

First, the outputs of the five branches are concatenated into a joint feature vector $\mathbf{h}_{\text{joint}}$:

$$\mathbf{h}_{\text{joint}} = [\mathbf{h}^{(1)} \parallel \mathbf{h}^{(2)} \parallel \mathbf{h}^{(3)} \parallel \mathbf{h}^{(4)} \parallel \mathbf{h}^{(5)}] \quad (27)$$

where the symbol $\parallel$ denotes feature concatenation.

The gating network consists of two fully connected layers. Taking $\mathbf{h}_{\text{joint}}$ as input, it first generates a five-dimensional unnormalized score vector through a linear transformation, which is then converted into a probability distribution α by the softmax function to serve as the fusion weights of the branches:

$$\alpha = \text{softmax} \left(\mathbf{W}_{\text{gate}}^{(2)} \cdot \text{SiLU} \left(\mathbf{W}_{\text{gate}}^{(1)} \mathbf{h}_{\text{joint}} + \mathbf{b}_{\text{gate}}^{(1)} \right) + \mathbf{b}_{\text{gate}}^{(2)} \right) \quad (28)$$

where $\mathbf{W}_{\text{gate}}^{(1)}$ and $\mathbf{W}_{\text{gate}}^{(2)}$ are learnable weight matrices, $\mathbf{b}_{\text{gate}}^{(1)}$ and $\mathbf{b}_{\text{gate}}^{(2)}$ are bias vectors, $\text{SiLU}(x) = x \cdot \sigma(x)$ is the activation function, and $\text{Softmax}(\cdot)$ normalizes the five-dimensional vector into positive values summing to one.

The branch weights α are then used to compute a weighted sum of the branch encoding vectors, yielding the weighted fusion feature:

$$\mathbf{f}_w = \sum_{k=1}^5 \alpha_k \mathbf{h}_k \quad (29)$$

where α_k represents the contribution of the k -th branch to the local geometric structure of the current point.

To match the dimensionality of another path, the feature is further projected to a 128-dimensional space through a projection network Ψ_{proj} , which consists of a single fully connected layer:

$$\mathbf{f}'_w = \Psi_{\text{proj}}(\mathbf{f}_w) \quad (30)$$

To fully capture nonlinear high-order interactions among the five branch encodings, a parallel fusion path denoted as Φ_{fuse} is introduced. This path takes $\mathbf{h}_{\text{joint}}$ as input and outputs a fusion feature $\mathbf{f}_c$ through a three-layer fully connected subnetwork:

$$\begin{cases} \mathbf{f}_c = \varphi_3 \circ \varphi_2 \circ \varphi_1(\mathbf{h}_{\text{joint}}) \\ \varphi_1(\mathbf{x}) = \text{Dropout}(\text{SiLU}(\text{LN}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}))) \\ \varphi_2(\mathbf{y}) = \text{SiLU}(\text{LN}(\mathbf{W}^{(2)}\mathbf{y} + \mathbf{b}^{(2)})) \\ \varphi_3(\mathbf{z}) = \text{SiLU}(\text{LN}(\mathbf{W}^{(3)}\mathbf{z} + \mathbf{b}^{(3)})) \end{cases} \quad (31)$$

where $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, and $\mathbf{W}^{(3)}$ denote learnable weight matrices; $\mathbf{b}^{(1)}$, $\mathbf{b}^{(2)}$, and $\mathbf{b}^{(3)}$ are bias vectors; and $\text{LN}(\cdot)$ denotes layer normalization.

Finally, the outputs of the two paths are added element-wise to obtain the global embedding vector $\mathbf{z}$:

$$\mathbf{z} = \mathbf{f}_c + \mathbf{f}'_w \quad (32)$$

4.3. Context Alignment Projection

To maintain consistency between the global embedding $\mathbf{z}$ and the context feature encoding, an alignment projection module Ψ_{align} is defined.

This module takes the output of the context branch $\mathbf{h}^{(5)}$ as input and maps it into the same space as $\mathbf{z}$. The structure of Ψ_{align} is identical to that of Ψ_{proj} and consists of a single fully connected layer:

$$\mathbf{z}_{\text{ctx}} = \Psi_{\text{align}}(\mathbf{h}^{(5)}) \quad (33)$$

4.4. Prototype Classification Head and Linear Classification Head

After multi-branch feature fusion, the network obtains a global semantic representation $\mathbf{z} \in \mathbf{R}^d$ for each point. Conventional point cloud classification networks generally adopt a single linear classifier for category prediction, whose decision boundaries mainly rely on linear mapping parameters without explicit constraints on class semantic centers. When different categories share high geometric similarity, feature overlap easily emerges in the feature space and degrades classification accuracy. Especially for structural components such as beams, columns and walls in indoor ancient building scenes, merely relying on a linear classifier fails to deliver sufficient discriminative power due to their similar local geometric shapes.

To tackle this issue, this paper introduces an auxiliary classification branch based on class prototypes on top of the conventional linear classification head. By learning the semantic center of each category in the feature space, the network pulls samples of the same class close to their corresponding class centers while maintaining large feature margins between different classes, further boosting the model's discriminative capability and the interpretability of the feature space. The complete classification module consists of two components, a Linear Classification Head and a Prototype Head, which are jointly optimized during training and fused to produce the final prediction during inference.

For the fused global embedding feature $\mathbf{z}$, the linear classification head is first applied to obtain base classification scores. Essentially, the linear classification head is a fully connected mapping, whose calculation is formulated as:

$$\ell_{cls} = \mathbf{W}_c \mathbf{z} + \mathbf{b}_c \quad (34)$$

where $\mathbf{W}_c$ and $\mathbf{b}_c$ are learnable parameters; c denotes the total number of categories.

To further strengthen inter-class discriminability, this paper designs a prototype classification head. For each category c , the network learns a trainable class prototype vector $\mathbf{P}_c \in \mathbf{R}^d$, and all class prototypes collectively form the prototype matrix $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_c]^T$.

Different from traditional methods that compute class centers via sample mean values, this work adopts a learnable prototype mechanism. Upon network initialization, each class prototype is randomly initialized following a zero-mean Gaussian distribution. During training, prototype vectors are treated as trainable parameters of the network, which together with the parameters of encoders, gated fusion module and classifiers constitute a unified optimization objective. In each forward pass, the network first calculates the similarity between samples in the current mini-batch and all class prototypes. Gradients are then backpropagated according to the classification loss, and parameter updates are performed by the AdamW optimizer.

To perform classification based on cosine similarity, both the embedding $\mathbf{z}$ and the prototype $\mathbf{P}_c$ are first L2-normalized to eliminate the influence of vector magnitude:

$$\tilde{\mathbf{z}} = \frac{\mathbf{z}}{\|\mathbf{z}\|_2} \quad (35)$$

$$\tilde{\mathbf{P}}_c = \frac{\mathbf{P}_c}{\|\mathbf{P}_c\|_2} \quad (36)$$

$$\tilde{\mathbf{P}} = [\tilde{\mathbf{P}}_1, \tilde{\mathbf{P}}_2, \dots, \tilde{\mathbf{P}}_C] \quad (37)$$

where $\|\cdot\|_2$ denotes the Euclidean norm.

After normalization, the dot product between the embedding and the prototype becomes equivalent to cosine similarity.

The similarity between the normalized embedding $\tilde{\mathbf{z}}$ and each normalized prototype $\tilde{\mathbf{P}}$ is then computed and divided by a temperature coefficient τ to control the sharpness of the output distribution:

$$\ell_{proto} = \frac{\tilde{\mathbf{z}} \tilde{\mathbf{P}}^T}{\tau} \quad (38)$$

where τ is a positive scalar, set to 0.15 in this study. A smaller τ amplifies similarity differences and produces a sharper probability distribution, improving inter-class separability; a larger τ yields a smoother distribution and helps prevent overfitting.

The linear classification head exhibits strong capability for category mapping, while the prototype classification head provides more stable semantic constraints for each category. Accordingly, this paper performs weighted fusion of the two branches to jointly determine the final classification outcome, with the final prediction score denoted as ℓ .

$$\ell = \ell_{cls} + \lambda_p \ell_{proto} \quad (39)$$

where λ_p denotes the fusion weight of the prototype branch.

4.5. Joint Loss Function and Optimization Objective

To jointly optimize classification accuracy, prototype discriminability, context consistency, and feature-space compactness, a joint loss function is constructed:

$$L = L_{ce} + \lambda_1 L_{proto} + \lambda_2 L_{align} + \lambda_3 L_{center} \quad (40)$$

where the cross-entropy loss constrains the final classification results:

$$L_{ce} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C \frac{N}{C \cdot n_c} \cdot \prod(y_i = c) \cdot \log \frac{\exp(\ell_{i,c})}{\sum_{j=1}^C \exp(\ell_{i,j})} \quad (41)$$

The prototype supervision loss enhances the discriminability of class prototypes:

$$L_{proto} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C \frac{N}{C \cdot n_c} \cdot \prod(y_i = c) \cdot \log \frac{\exp(\ell_{proto,i,c})}{\sum_{j=1}^C \exp(\ell_{proto,i,j})} \quad (42)$$

The alignment loss enforces consistency between the context branch and the main representation:

$$L_{align} = \frac{1}{N} \sum_{i=1}^N \|z_i - z_{ctx,i}\|_2^2 \quad (43)$$

The intra-class compactness loss further reduces the dispersion of samples belonging to the same class in the embedding space:

$$L_{center} = \frac{1}{\sum_{c=1}^C |\mathcal{G}_c|} \sum_{c=1}^C \sum_{i \in \mathcal{G}_c} \|\mathbf{z}_i - \boldsymbol{\mu}_c\|_2^2 \quad (44)$$

where L denotes the overall joint training loss of the model, which is obtained by weighted summation of four sub-losses and adopted for backpropagation to update all network parameters; L_{ce} stands for the weighted cross-entropy loss of the linear classification head; L_{proto} refers to the weighted cross-entropy loss of the prototype classification branch; L_{align} means the context feature alignment loss, and L_{center} represents the intra-class compactness constraint loss; λ_{proto} , λ_{align} and λ_{center} are hyperparameters for balancing sub-losses, which separately control the proportion of prototype supervision constraint, context alignment constraint and intra-sample aggregation constraint in total loss; N is the total count of point cloud samples within a single training mini-batch; C indicates the total number of ancient building component categories, where $C = 5$ in this paper including wall, roof, beam, column and window; i is the index of a single point sample in the batch, c is the component category index, and j serves as an auxiliary category index for Softmax normalization. Shared variables of weighted cross-entropy: n_c represents the total point quantity of category c in the full dataset, which is used to calculate the class balance weight coefficient. This coefficient assigns larger weights to categories with scarce samples to amplify classification gradients of rare components; $\mathbb{I}(y_i = c)$ is a Boolean indicator function that outputs 1 if the ground-truth label of point i matches category c and outputs 0 otherwise, and only the ground-truth category participates in loss calculation; y_i is the ground-truth component label of point sample i ; $\ell_{i,c}$ is the raw prediction logit output by linear classification head, while $\ell_{proto,i,c}$ is the similarity logit calculated between sample embedding and category prototype; $\ell_{i,j}$ and $\ell_{proto,i,j}$ are prediction logits of linear branch and prototype branch corresponding to auxiliary category j respectively; $\exp(\cdot)$ is a natural exponential operation that converts raw scores into class probabilities via Softmax, and $\log(\cdot)$ is the natural logarithm operator for cross-entropy computation. Exclusive variables for context alignment loss: $\mathbf{z}_i$ is the trunk global embedding vector with a dimension of 128 output by multi-branch gated fusion module, and $\mathbf{z}_{ctx,i}$ is the context feature vector with identical dimension generated by neighborhood context projection layer; $\|\cdot\|_2$ denotes the L2 Euclidean norm. Exclusive variables for intra-class compactness loss: $\mathcal{G}_c$ is the set containing all embedding vectors of samples labeled category c in current mini-batch, $|\mathcal{G}_c|$ is the sample number inside this set, $\boldsymbol{\mu}_c$ is the mean embedding center of all category- c samples in the batch, and $i \in \mathcal{G}_c$ means traversing all samples belonging to category c in the set.

5. Experimental Results and Analysis

5.1. Sample Preparation and Experimental Setup

In this study, the constructed three-dimensional scene point cloud dataset was used as the training data. The dataset includes five categories of architectural components: windows, beams, walls, roofs, and columns. Detailed descriptions of each component category are provided in Table 1.

Table 1. Statistics of various object samples.

Category	Window	Beam	Wall	Roof	Column	Total
Sample Count	218,491	314,633	331,180	446,579	53,897	1,364,777
Proportion	16.01%	23.05%	24.27%	32.72%	3.95%	100.00%

This paper adopts an independent data partitioning scheme combining 3D spatial grid blocking and global component ownership constraints. First, the bounding box of the

complete indoor point cloud of Bawang Academy is evenly divided into non-overlapping 3D cubic spatial blocks. After spatial block segmentation, global binding constraints are applied to all annotated components: all point points belonging to one intact component are assigned exclusively to the single spatial block where the main body of the component resides. After finishing spatial block cutting and component binding, stratified sampling is conducted across all independent non-overlapping blocks to split the dataset into a test set, a validation set and a training set. The test set accounts for 15% of all blocks, and the validation set takes up 15% of the remaining blocks reserved for training.

The full implementation of the Geometric Feature Multi-scale Network (GFMN) with multi-branch geometric feature fusion is built on the PyTorch 2.0 deep learning framework (<https://pytorch.org/>, accessed 10 July 2026).. The hardware and software experimental configurations are listed as follows. For hardware, an Intel Xeon Gold CPU is equipped with 128 GB RAM and a single NVIDIA RTX 3060 graphics card. For software, Windows 10 serves as the underlying operating system, with CUDA 11.7 configured for parallel acceleration.

In terms of training strategy, the batch size was set to 8192, the maximum number of training epochs was 200, the initial learning rate was 1×10^{-3} , the weight decay coefficient was 1×10^{-4} , and the early stopping patience was set to 20. The optimizer used was AdamW, combined with ReduceLROnPlateau for adaptive learning rate adjustment, where the learning rate decay factor was set to 0.5 and the patience of the monitored metric was set to 2. To alleviate class imbalance, the training stage employed both a class-weighted cross-entropy loss constructed based on the inverse frequency of each class and WeightedRandomSampler for class-balanced sampling.

In terms of feature construction, local contextual features were extracted using k-nearest neighbor (k-NN) neighborhood statistics, with the number of neighbors set to 20. The input dimension of the context branch was 12. The internal branch encoding dimension of the model was set to 64, the context encoding dimension to 64, the latent embedding dimension after fusion to 128, the hidden dimension of the fusion projection layer to 128, and the dropout rate to 0.20.

For the loss function design, this study adopted a composite objective function. Specifically, the prototype branch loss weight λ_{proto} was set to 0.35, the context alignment loss weight λ_{align} was set to 0.20, and the intra-class compactness loss weight λ_{center} was set to 0.10. The temperature coefficient of the prototype classification head was set to 0.15.

To address the severe class imbalance of the dataset, where roofs, walls and beams dominate the sample distribution while columns form an extremely minority class (the total point count of majority classes is approximately 8 to 9 times that of columns), a dual strategy of weighted loss plus class-balanced sampling is adopted during training to mitigate the skewed class distribution.

Class weights are calculated as the reciprocal of the number of point samples for each category. Minor classes with fewer samples (columns and windows) are assigned larger loss weights, which amplify the gradient contribution of their classification errors during backpropagation and prevent the model from biasing toward majority classes including roofs and walls with abundant points.

Higher sampling probabilities are assigned to points of rare components in mini-batch sampling. The sampling quantity of each component category is forced to be balanced within every batch, alleviating the uneven sample distribution in a single batch.

The two strategies work synergistically to constrain the model from ignoring geometric features of minority classes due to imbalanced sample sizes, from the two dimensions of loss function design and data sampling mechanism.

5.2. Evaluation Metrics

The main metrics used to evaluate the classification performance of the model include accuracy, precision, recall, and F1 score. These four evaluation metrics can be calculated based on the confusion matrix shown in Table 2. In Table 2, *TP* represents the number of positive samples correctly predicted as positive, *FP* represents the number of negative samples incorrectly predicted as positive, *FN* represents the number of positive samples incorrectly predicted as negative, and *TN* represents the number of negative samples correctly predicted as negative.

Table 2. Confusion Matrix.

		Actual Class	
		0	1
Predicted Class	0	<i>TP</i>	<i>FP</i>
	1	<i>FN</i>	<i>TN</i>

Accuracy represents the proportion of all samples that are correctly classified among the total number of samples.

$$accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (45)$$

Precision (*p*) represents the proportion of correctly predicted positive samples among all samples predicted as positive.

$$p = TP / (TP + FN) \quad (46)$$

Recall represents the proportion of correctly predicted positive samples among all actual positive samples.

$$recall = TP / (TP + FN) \quad (47)$$

The F1 score (*f1*) is the harmonic mean of precision and recall and is primarily used to evaluate the robustness of the model.

$$f1 = 2 * p * recall / (p + recall) \quad (48)$$

where * denotes scalar multiplication.

5.3. Experimental Results

Figures 4 and 5 illustrate the variations in the loss function during training, as well as the changes in test accuracy and F1 score, respectively. As shown in the figures, the loss decreases rapidly during the early stages of training, indicating that the network is able to quickly learn the geometric and contextual features of ancient building point clouds. As the number of training epochs increases, both the training loss and validation loss exhibit an overall downward trend and gradually stabilize in the later stages, suggesting that the model has converged sufficiently.

Specifically, the training loss continuously decreases from an initial value of 0.6374 to 0.1260, and the minimum validation loss reaches 0.1793. This indicates that the model can effectively fit the training samples during optimization while maintaining strong generalization performance on the validation set. In terms of evaluation metrics, after hyperparameter tuning on the validation set, the model achieves an overall accuracy of 93.17% on the independent test set that does not participate in training or parameter adjustment, with the optimal macro-F1 score on the validation set reaching 0.8741. Overall, the validation loss and metric curves fluctuate slightly within the first 60 training epochs, and

gradually become smooth and stable after 60 epochs without persistent violent oscillations or severe overfitting throughout the training process. The minor fluctuations of validation metrics in the early training stage can be attributed to three factors.

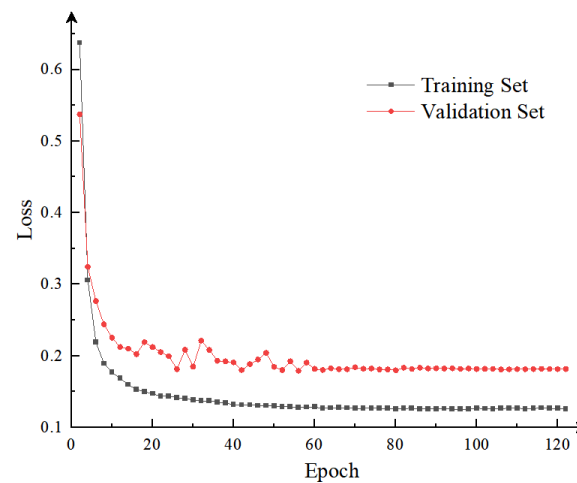


Figure 4. Variation Curve of Loss Function.

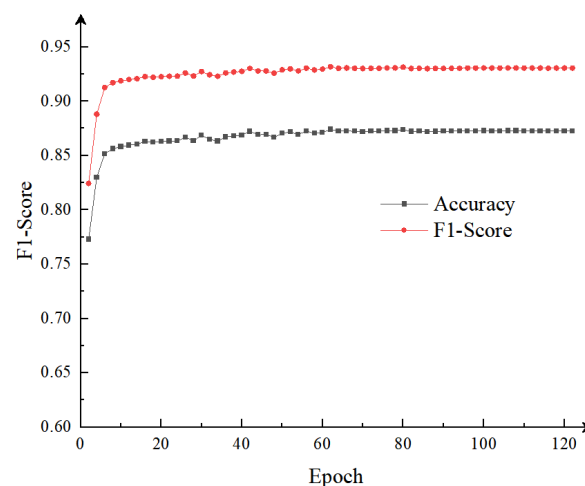


Figure 5. Variation Curves of Accuracy and F1-Score on the Test Set.

First, the total number of samples in the validation set is far smaller than that of the training set. A small sample base tends to amplify random fluctuations in single-epoch evaluation, and such fluctuations are naturally alleviated after the network representations stabilize in the later training phase.

Second, the initial learning rate is set to 1×10^{-3} . The randomly initialized weights and relatively large update step size in the early stage lead to fluctuations in metrics; the oscillations vanish after adaptive learning rate decay during training.

Third, the AdamW optimizer involves momentum accumulation. Disordered gradient directions at the beginning of training magnify the volatility of validation curves, and the curves gradually smooth out once the model sufficiently learns component features and gradients become stable.

Based on the confusion matrix shown in Figure 6, the segmentation results for the five categories of interior architectural components—windows, beams, walls, roofs, and columns—were statistically analyzed. Precision, recall, and F1 score for each category were calculated from the confusion matrix. Overall, the model achieved an overall accuracy of 93.18% on the validation set, with the F1 scores of all categories reaching approximately 0.93. These results demonstrate that the proposed multi-scale geometric feature

fusion network provides stable and highly accurate semantic segmentation performance for point clouds of ancient building interior components.

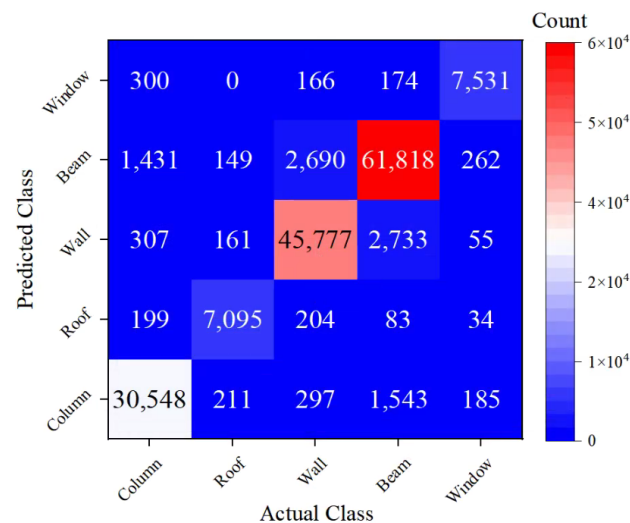


Figure 6. Confusion Matrix.

As shown by the sample statistics in Table 1, the dataset exhibits an extremely imbalanced class distribution: the point count of the most abundant roof class is approximately 8.3 times that of the rarest column class, making this a severely class-imbalanced dataset. Nevertheless, the precision and recall of each class calculated from the confusion matrix in Figure 6 steadily fall within the range of 92–93%, and the F1-score of every class approaches 0.93. No obvious performance gap exists between majority and minority classes, and the common issue of drastically degraded accuracy for small categories is avoided. This favorable result is jointly guaranteed by multiple complementary mechanisms:

1. A weighted cross-entropy loss and class-balanced sampler are adopted simultaneously during training. On one hand, higher loss weights are assigned to rare components such as columns, driving the model to prioritize learning the linear geometric features of minority classes. On the other hand, the sampling frequency of minority-class points is intentionally elevated during batch sampling, ensuring sufficient training samples for small components in every epoch and preventing the model from overfitting to majority planar classes including walls and roofs.
2. The multi-scale geometric descriptors extracted in this work can accurately capture the inherent morphological distinctions among various components. Columns and beams exhibit prominent linear features, walls and roofs are dominated by planar attributes, and windows contain abundant abrupt boundary features. Even with limited column samples, these highly discriminative geometric descriptors supply sufficient distinguishing information and mitigate recognition bias caused by class imbalance.
3. The prototype head learns a dedicated embedding prototype for each component category. The intra-class compactness loss reduces the feature dispersion of samples within the same class. Even with a limited number of column samples, their embeddings can converge to a stable class prototype, greatly alleviating inadequate feature learning for minority samples. Meanwhile, the prototype regularization loss enlarges the feature distance between prototypes of different components, further weakening the impact of sample quantity disparity on classification decision boundaries.
4. The gated fusion mechanism adaptively assigns weights to different feature branches according to local geometric morphology. For slender minority components like columns, the model automatically places greater emphasis on geometric branches such

as linearity and anisotropy, compensating for insufficient feature learning caused by scarce samples and preventing the collapse of recognition accuracy for small categories.

5.4. Comparative Experimental Analysis

To further verify the effectiveness of the proposed Geometric Feature Multi-scale Network (GFMN) for semantic segmentation of point clouds of indoor ancient building components, four state-of-the-art point cloud segmentation algorithms are selected for horizontal comparison, namely PointNet, PointNet++, Point Transformer and Point Cloud Transformer (PCT).

All experimental groups adopt the spatially independent blocking strategy proposed in this paper to split the training, validation and test sets, with identical software and hardware environments as well as training hyperparameters to guarantee fair comparison. To eliminate random fluctuations induced by single random weight initialization, each model undergoes five independent repeated training runs. All evaluation metrics (overall accuracy, total parameter count, inference speed and peak GPU memory consumption) take the average values of the five repeated experiments, and the corresponding standard deviations are calculated for statistical significance testing. The final comparative results are presented in Table 3.

Table 3. Accuracy Comparison of Different Segmentation Models.

Method	Accuracy	Parameters (M)	Inference Speed (Samples·s ⁻¹)	Memory Usage (GB)
PointNet	71.24%	3.6	196	35.5
PointNet++	87.35%	1.2	321	7.2
Point Transformer	82.29%	7.8	78	5.6
Point Cloud Transformer (PCT)	84.37%	0.97	165	18.4
Proposed Method	93.17%	7.05	144	15.2

As the first deep learning model capable of directly processing raw unordered point clouds, PointNet extracts features via point-wise MLPs combined with global max pooling. It boasts an extremely lightweight architecture with only 3.6 M total parameters and an inference speed of 196 samples per second. However, the model lacks the capacity to model local geometric information, leading to frequent misclassification of fine-grained ancient building components such as windows and beams. Its segmentation accuracy merely reaches 71.24%. Furthermore, global pooling induces excessive memory overhead, with peak GPU memory consumption hitting 35.5 GB, which raises high barriers to hardware deployment.

Built upon PointNet, PointNet++ constructs a hierarchical local feature extraction framework. It enhances perception of local details through multi-scale neighborhood grouping and layered feature aggregation, lifting overall accuracy to 87.35%. With merely 1.2 M parameters, it achieves the smallest model size among all comparative methods, delivering the fastest inference speed at 321 samples/s and a moderate memory footprint of only 7.2 GB, demonstrating prominent advantages in deployment efficiency. Nevertheless, its fixed-radius neighborhood structure is sensitive to fluctuations in point cloud density, resulting in unstable feature extraction for ancient building scenes with drastically uneven point density across different components.

Point Transformer and Point Cloud Transformer (PCT) both introduce self-attention mechanisms to model long-range dependencies within point clouds. Point Transformer implements dynamic neighborhood aggregation via vector attention and positional encoding. It has 7.8 M parameters and the lowest peak GPU memory usage of 5.6 GB, yet

the heavy computation cost of attention layers limits its inference speed to only 78 samples/s with poor real-time performance, yielding a segmentation accuracy of 82.29%. Drawing on the Transformer architecture from natural language processing, PCT adopts offset attention and lightweight downsampling strategies, achieving a parameter count of 0.97 M, an inference speed of 165 samples/s, and an accuracy of 84.37%. Although both models perform acceptably on general indoor scenes, they fail to extract sufficiently discriminative local features for the segmentation task of ancient building components, where inter-class similarity is high and subtle geometric discrepancies are hard to distinguish, resulting in obvious accuracy deficiencies.

Comprehensive multi-dimensional comparisons in Table 3 demonstrate that the proposed GFMN achieves the optimal trade-off among segmentation accuracy, parameter count, inference speed and GPU memory consumption. Equipped with 46-dimensional elaborate handcrafted geometric descriptors, as well as a complete pipeline including multi-branch encoding, gated adaptive fusion and a prototype classification head, the proposed model fully exploits distinctive geometric information of ancient building components. It attains an overall test set accuracy of 93.17%, approximately 22, 6, 11 and 9 percentage points higher than PointNet, PointNet++, Point Transformer and PCT respectively. The total parameter count of our model is 7.05 M, slightly lower than Point Transformer; its inference speed of 144 samples/s outperforms Point Transformer, while peak GPU memory usage of 15.2 GB is far less than PointNet.

Compared with the lightweight baseline PointNet++, our method sacrifices a modest amount of inference speed and memory overhead for a 6-percentage-point accuracy gain. When matched against the two Transformer-based models, our method achieves substantially superior segmentation accuracy while delivering significantly better real-time inference performance. Taking segmentation precision, hardware deployment cost and inference efficiency into comprehensive consideration, the proposed approach is better suited for practical engineering requirements of batch automatic segmentation of ancient building point clouds.

Figure 7 presents the classification results obtained by different methods in the study area. From the segmentation results, it can be observed that the proposed method significantly outperforms the other approaches. The experimental results fully demonstrate the superiority of the proposed method for semantic segmentation of ancient building point clouds, particularly in scenarios where architectural components exhibit distinctive geometric characteristics while labeled data remain limited.

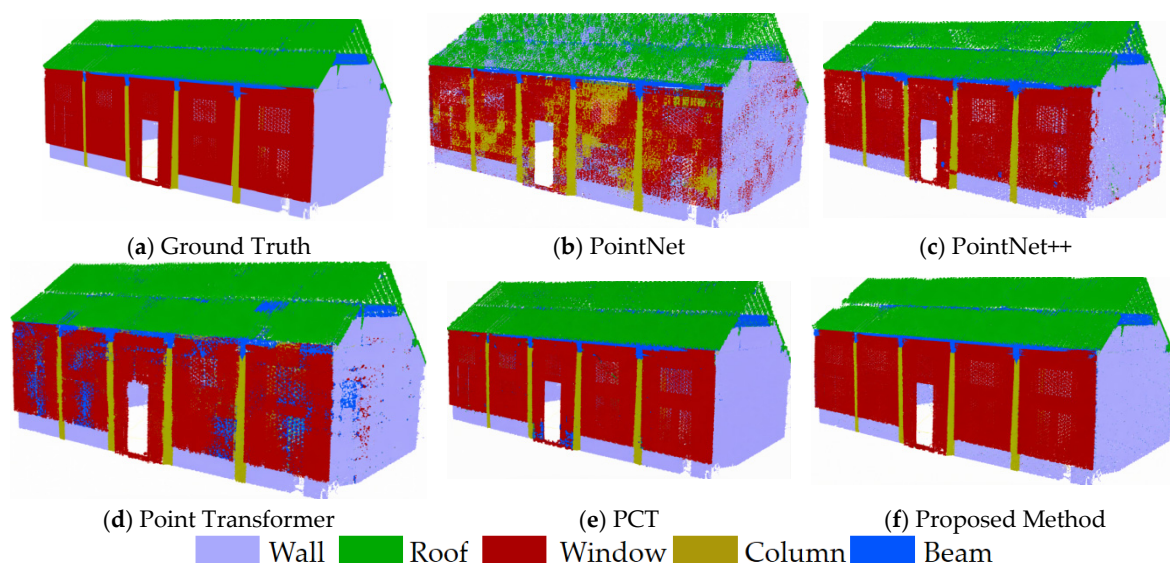


Figure 7. Comparison of Point Cloud Segmentation Results Using Different Methods.

6. Discussion

This study addresses the challenges of “complex local geometry, high inter-class similarity, and insufficient single-scale features” in semantic segmentation of point clouds for ancient building interior components by proposing a segmentation method based on a Geometric Feature Multi-scale Network (GFMN). Experimental results show that the proposed method achieved an overall accuracy of 93.17% on the validation set, with a validation F1 score of 0.8741. The F1 scores for all categories were approximately 0.93, indicating that the model can stably distinguish the five component types—windows, beams, walls, roofs, and columns—and possesses strong discriminative capability.

From a methodological perspective, the advantage of GFMN mainly comes from its explicit modeling of the geometric differences among ancient architectural components. Unlike methods that rely only on point coordinates or implicit feature learning, this study constructs a 46-dimensional point-level descriptor from four aspects: basic attributes, local geometric features, density and scale features, and multi-scale fusion. The features are then grouped according to semantics and fed into independent encoding branches. The benefit of this design is that different types of geometric information can be extracted separately, avoiding feature interference caused by simple concatenation, while better matching the characteristics of ancient architectural components, whose shape differences are subtle but structural regularities are clear.

The gated adaptive fusion mechanism is also an important reason for the performance improvement. In ancient building interior scenes, walls and roofs usually exhibit strong planar structures, beams and columns show obvious linear extension features, while windows are often characterized by boundaries, openings, and local structural discontinuities. In response to these differences, the gating module can dynamically adjust the contribution of each branch according to the local geometric context of the current point, enabling the model to automatically select more appropriate feature sources for different components and thereby improving its adaptability to complex structures. During training, the loss decreased from 0.6374 to 0.1260, and the minimum validation loss reached 0.1793, indicating that the fusion strategy not only improved classification performance but also provided good convergence and generalization.

The prototype classification head and joint loss function further enhanced class separability. The prototype head learns a semantic prototype for each category, causing samples of the same class to cluster around the corresponding prototype in the embedding space. The joint loss simultaneously constrains the final classification result, prototype discriminability, contextual consistency, and intra-class compactness. For ancient building point cloud tasks, where class boundaries are blurred and intra-class variation is large, this “pull same classes closer and push different classes apart” constraint is particularly effective. This is also why the model achieved its best overall performance at the 62nd epoch, with stable curves in the later stage and no obvious oscillation or severe overfitting.

Compared with baseline methods, the advantages of GFMN are more evident. PointNet lacks fine-grained modeling of local geometric structures, making it prone to misclassification in detailed regions such as windows and beams. Although PointNet++ improves performance through hierarchical local feature extraction, it is still affected by the sensitivity of fixed-radius neighborhoods to density variation. Point Transformer and PCT are better at modeling contextual dependencies, but in ancient architectural scenes dominated by subtle geometric differences, attention mechanisms alone are not sufficient to fully characterize component shapes. Under the same data split, the proposed method achieved 93.17% accuracy, significantly outperforming PointNet, PointNet++, Point Transformer, and PCT, demonstrating that explicit geometric features and multi-branch fusion strategies are more suitable for ancient building interior component segmentation.

Although the proposed method achieves favorable segmentation performance, the experiments still have certain limitations. First, the experimental data in this work are collected only from the single indoor scene of Bawang Academy. Although this building contains typical components such as beams, columns, walls, roofs and windows and can effectively verify the effectiveness of the proposed model, its architectural style, component forms, material characteristics and spatial layout remain relatively homogeneous. Accordingly, the current experimental results mainly reflect the segmentation performance of the model on this specific type of ancient building scene, and its generalization ability to ancient buildings of different regions, ages and structural forms requires further verification.

Future research will continuously expand the ancient building point cloud dataset by collecting indoor scene data covering diverse architectural types and styles. We will also design cross-scene training and testing experiments (Cross-scene Validation) to further evaluate the model's robustness, transfer adaptability and external validity under cross-scene conditions, so as to provide more sufficient experimental support for the popularization and application of the model in the digital preservation of ancient buildings.

Overall, the experimental results of GFMN indicate that, in semantic segmentation of point clouds for ancient building interiors, relying solely on end-to-end feature learning is not sufficient. Instead, combining domain knowledge-driven multi-scale geometric features, grouped encoding, and adaptive fusion mechanisms can more effectively improve the recognition accuracy of complex components. This method provides valuable technical support for digital heritage preservation, structural analysis, and subsequent restoration modeling of ancient buildings.

7. Conclusions

This paper addresses the semantic segmentation of point clouds for ancient building interior components by taking the Ba Wang Academy as the research object and proposing a Geometric Feature Multi-scale Network. The method constructs a 46-dimensional point-level descriptor from four aspects: basic attributes, local geometric features, density-scale features, and multi-scale fusion. Through grouped encoding, gated adaptive fusion, and a prototype classification head, it effectively improves the discriminative ability for components such as windows, beams, walls, roofs, and columns. Experimental results show that the proposed method achieved an overall accuracy of 93.17% on the validation set and a macro-averaged F1 score of 0.8741, significantly outperforming state-of-the-art methods such as PointNet, PointNet++, and Point Transformer, which verifies its superiority in ancient building point cloud segmentation.

Future work will focus on improving the generalization ability and computational efficiency of the model. We will further expand the point cloud dataset consisting of multiple ancient buildings, covering typical indoor scenes of ancient architectures with different regions, ages and architectural styles. Cross-scene validation experiments and cross-building transfer tests will be carried out to systematically evaluate the model's robustness, transfer adaptability and external validity in complex ancient building environments. Meanwhile, we will explore weakly supervised learning, self-supervised pre-training and lightweight network design to further reduce labeling costs and enhance the engineering practicability of the model.

Author Contributions: Conceptualization, J.M.; methodology, J.M.; software, D.W.; validation, J.M. and D.W.; formal analysis, J.M.; investigation, J.M.; resources, J.M.; data curation, D.W.; writing—original draft preparation, J.M.; writing—review and editing, D.W.; visualization, D.W.; supervision, D.W.; project administration, J.M.; funding acquisition, J.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Scientific Research Program of Higher Education Institutions of Liaoning Provincial Department of Education (2025), Grant No. LJ212510153048.

Data Availability Statement: The raw point cloud data collected on-site in this study involves the internal spatial information of ancient buildings, which cannot be publicly released due to cultural heritage protection restrictions. The processed analysis data, model evaluation indicators and statistical results that support the conclusions of this paper are fully presented within the manuscript, figures and tables. Any reasonable data access requests for academic research purposes can be sent to the corresponding author upon reasonable request.

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

References

1. Liang, S.S. *Chinese Architecture: A Pictorial History*; MIT Press: Cambridge, MA, USA, 1984.
2. Jiao, Z.; Zhang, J.; Ou, L. Study on the Dynamic Protection Strategy of Ancient Buildings in Historical and Cultural Neighbourhoods from the Perspective of Traditional Culture Based on Three-dimensional Modelling Technology. *Appl. Math. Nonlinear Sci.* **2025**, *10*. Available online: <https://www.semanticscholar.org/paper/7dd07c0e14dae9aaabc7460adef754b04787252a> (accessed 1 August 2026).
3. Yang, S.; Hou, M.; Li, S. Three-Dimensional Point Cloud Semantic Segmentation for Cultural Heritage: A Comprehensive Review. *Remote Sens.* **2023**, *15*, 548.
4. Yang, B.S.; Liang, F.X.; Huang, R.G. Progress, Challenges and Perspectives of 3D LiDAR Point Cloud Processing. *Acta Geod. Cartogr. Sin.* **2017**, *46*, 1509–1516.
5. Lovell, L.J.; Davies, R.J.; Hunt, D.V.L. The application of historic building information modelling (HBIM) to cultural heritage: A review. *Heritage* **2023**, *6*, 6691–6717.
6. Rashdi, R.; Martínez-Sánchez, J.; Arias, P.; Qiu, Z. Scanning Technologies to Building Information Modelling: A Review. *Infrastructures* **2022**, *7*, 49.
7. Zhao, J.; Hua, X.; Yang, J.; Yin, L.; Liu, Z.; Wang, X. A Review of Point Cloud Segmentation of Architectural Cultural Heritage. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *10*, 247–254.
8. Moyano, J.; León, J.; Nieto-Julián, J.E.; Bruno, S. Semantic Interpretation of Architectural and Archaeological Geometries: Point Cloud Segmentation for HBIM Parameterisation. *Autom. Constr.* **2021**, *130*, 103856.
9. Xie, Y.; Tian, J.; Zhu, X.X. Linking Points with Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 38–59.
10. Zhang, R.J.; Zhou, X.; Zhao, J.H. A Semantic Segmentation Algorithm of Ancient Building's Point Cloud Data. *Geomat. Inf. Sci. Wuhan Univ.* **2020**, *45*, 753–759.
11. Yan, L.S.; Bian, L.; Fan, Q.P. Ground-based LiDAR Point Cloud Classification Method for Ancient Building Scenes. *Geomat. Spat. Inf. Technol.* **2025**, *48*, 181–183.
12. Grilli, E.; Menna, F.; Remondino, F. A Review of Point Clouds Segmentation and Classification Algorithms. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 339–344.
13. Grilli, E.; Remondino, F. Classification of 3D Digital Heritage. *Remote Sens.* **2019**, *11*, 847.
14. Teruggi, S.; Grilli, E.; Russo, M.; Fassi, F.; Remondino, F. A Hierarchical Machine Learning Approach for Multi-Level and Multi-Resolution 3D Point Cloud Classification. *Remote Sens.* **2020**, *12*, 2598.
15. Croce, V.; Caroti, G.; De Luca, L.; Jacquot, K.; Piemonte, A.; Véron, P. From the Semantic Point Cloud to Heritage-Building Information Modeling: A Semiautomatic Approach Exploiting Machine Learning. *Remote Sens.* **2021**, *13*, 461.
16. Matrone, F.; Grilli, E.; Martini, M.; Paolanti, M.; Pierdicca, R.; Remondino, F. Comparing Machine and Deep Learning Methods for Large 3D Heritage Semantic Segmentation. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 535.
17. Fiorucci, M.; Khoroshiltseva, M.; Pontil, M.; Traviglia, A.; Del Bue, A.; James, S. Machine Learning for Cultural Heritage: A Survey. *Pattern Recognit. Lett.* **2020**, *133*, 102–108.
18. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-View Convolutional Neural Networks for 3D Shape Recognition. In *Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015*; IEEE: Piscataway, NJ, USA, 2015; pp. 945–953.
19. Boulch, A.; Guerry, J.; Le Saux, B.; Audebert, N. SnapNet: 3D Point Cloud Semantic Labeling with 2D Deep Segmentation Networks. *Comput. Graph.* **2018**, *71*, 189–198.

20. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Macau, China, 3–8 November 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 4213–4220.
21. Zhang, J.Y.; Zhao, X.L.; Chen, Z. Review of Semantic Segmentation of Point Cloud Based on Deep Learning. *Laser Optoelectron. Prog.* **2020**, *57*, 040002.
22. Maturana, D.; Scherer, S. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2015*; IEEE: Piscataway, NJ, USA, 2015; pp. 922–928.
23. Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. SEGCloud: Semantic Segmentation of 3D Point Clouds. In *Proceedings of the International Conference on 3D Vision, Qingdao, China, 10–12 October 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 537–547.
24. Riegler, G.; Ulusoy, A.O.; Geiger, A. OctNet: Learning Deep 3D Representations at High Resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2017; pp. 3577–3586.
25. 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 IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2019; pp. 8500–8508.
26. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4338–4364.
27. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2017; pp. 652–660.
28. 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*; Curran Associates: Long Beach, CA, USA, 2017; pp. 5099–5108.
29. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2020; pp. 11108–11117.
30. 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.
31. 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.
32. Zhou, C.; Dong, Y.; Hou, M.; Ji, Y.; Wen, C. MP-DGCNN for the Semantic Segmentation of Chinese Ancient Building Point Clouds. *Herit. Sci.* **2024**, *12*, 249.
33. Zhao, J.; Liu, R.; Hua, X.; Yu, H.; Zhao, J.; Wang, X.; Yang, J. DSC-Net: Learning Discriminative Spatial Contextual Features for Semantic Segmentation of Large-Scale Ancient Architecture Point Clouds. *Herit. Sci.* **2024**, *12*, 274.
34. Cai, Z.Y.; Yuan, Z.Y.; Pang, M.Y. Survey on Deep-Learning-Based Point Cloud Semantic Segmentation. *Comput. Eng. Appl.* **2025**, *61*, 22–30.
35. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2021; pp. 16259–16268.
36. 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.
37. Lai, X.; Liu, J.; Jiang, L.; Wang, L.; Zhao, H.; Liu, S.; Qi, X.; Jia, J. Stratified Transformer for 3D Point Cloud Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2022; pp. 8500–8509.
38. Wang, J.C.; Wang, H.Y.; Su, H. RandLA-CGNet for Large-scale Indoor Point Cloud Semantic Segmentation. *Comput. Syst. Appl.* **2026**, *35*, 175–186.

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.