remote sensing

MDPI

Article

AIFormer: Adaptive Interaction Transformer for 3D Point Cloud Understanding

Xutao Chu [1], Shengjie Zhao [1,*] and Hongwei Dai [2]

[1] School of Computer Science and Technology, Tongji University, Shanghai 201804, China; chuxutao@tongji.edu.cn
[2] School of Computer Engineering, Jiangsu Ocean University, Lianyungang 222005, China; hwdai@jou.edu.cn
* Correspondence: shengjiezhao@tongji.edu.cn

Abstract: Recently, significant advancements have been made in 3D point cloud analysis by leveraging transformer architecture in 3D space. However, it remains challenging to effectively implement local and global learning within irregular and sparse structures of 3D point clouds. This paper presents the Adaptive Interaction Transformer (AIFormer), a novel hierarchical transformer architecture designed to enhance 3D point cloud analysis by fusing local and global features through the adaptive interaction of features. Specifically, AIFormer mainly consists of several stacked AIFormer Blocks. Each AIFormer module employs the Local Relation Aggregation Module and the Global Context Aggregation Module, respectively, to extract local details of relationships within the reference point and long-range dependencies between reference points. Then, the local and global features are fused using the Adaptive Interaction Module for adaptive interaction to optimize the point representation. Additionally, the AIFormer Block further designs geometric relation functions and contextual relative semantic encoding to enhance local and global feature extraction capabilities, respectively. Extensive experiments on three popular 3D point cloud datasets verify that AIFormer achieves state-of-the-art or comparable performances. Our comprehensive ablation study further validates the effectiveness and soundness of the AIFormer design.

Keywords: 3D point cloud; semantic segmentation; 3D point cloud analysis; deep learning; feature fusion

check for updates

Citation: Chu, X.; Zhao, S.; Dai, H. AIFormer: Adaptive Interaction Transformer for 3D Point Cloud Understanding. *Remote Sens.* **2024**, *16*, 4103. https://doi.org/10.3390/rs16214103

Academic Editors: Jiju Poovvancheri, Dong Chen, Jiaming Na and Norbert Pfeifer

Received: 13 September 2024
Revised: 26 October 2024
Accepted: 31 October 2024
Published: 2 November 2024

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

1. Introduction

In recent years, with the development of deep learning and the widespread availability of 3D acquisition devices, 3D point cloud data has played an increasingly important role in various 3D applications, such as autonomous driving, robotics, and augmented reality. Unlike 2D images, the 3D point cloud is a discrete set of points with irregular, sparse, and non-uniform distribution, which hampers the direct application of mature convolutional methods to process the 3D point cloud. Therefore, exploring advanced methods for 3D point cloud analysis is a significant and challenging task.

As a fundamental yet challenging task, various 3D point cloud analysis methods have been proposed to address this problem, including projection-based [1–4] and voxel-based [5–8] methods. They transform irregular 3D point clouds into regular grids or voxels to be processed with mature convolution methods. However, such transformations inevitably lead to information loss. Therefore, point-based methods [9–13] operating directly on the raw point cloud have become popular. These methods retain the detailed spatial information of the 3D point cloud, but the Multi-Layer Perceptrons (MLPs) used to capture point features are limited in establishing long-range dependencies, resulting in insignificant performance in large-scale 3D scenes.

With ViT [14] successfully dominating the field of 2D visual understanding, extending the Transformer-like architecture to 3D is a natural idea. PCT [15] and PTv1 [16] are the first concurrent methods to utilize the Transformer architecture for 3D point cloud analysis. PCT

employs global attention on the entire point set, leading to unacceptable $O(N^2)$ memory consumption. PTv1, on the other hand, adopts a hierarchical aggregation paradigm similar to PointNet++ [17], which utilizes a local self-attention mechanism to capture the local region features and progressively expands the size of the local region along the network hierarchy, achieving an effective balance between memory usage and computational efficiency. Several follow-up approaches, such as PVT [18] and FastPointTransformer [19], explore enhancement to local region representation to improve the capability of point modeling.

However, Transformer-based architecture extensions of 3D do not yield the same superior performance as 2D tasks. We analyze and identify the key issues: point features only rely on the interactions of points in a local region and lack actual long-range dependencies. Some existing methods [20–25] expand the receptive field of a point to enhance the long-range dependence of the point. For example, ST [20] densely samples nearby points and sparsely samples distant points to increase the receptive field of the points. However, this approach necessitates a sophisticated and complex design to manage the varying number of points across different windows. LRP [22] optimizes the pooling operator with dilated maximal pooling to provide networks with large adaptive receptive fields with few parameters. This method is lightweight but has limited ability to capture features. CDFormer [23] introduces a long-range contextual feature capture mechanism for local features and then utilizes cross-attention to achieve short- and long-range contextual communication of point clouds, which improves the global awareness of point cloud features but incurs higher computational and storage costs.

In this paper, we propose an Adaptive Interaction Transformer (AIFormer), which achieves local region feature extraction and modeling long-range dependencies via local relation aggregation and global semantic aggregation, respectively, and then fuses the local and global features with adaptive interaction to improve the representation of points. Specifically, we first use a grid sample to obtain the reference point set that contains a variable number of nearest-neighbor points. Subsequently, we use the Local Relation Aggregation Module and Global Context Aggregation Module, respectively, to extract local details of relationships within the reference point and long-range dependencies between reference points. Finally, the Adaptive Interaction Module achieves adaptive interaction fusion between local and global features. To further enhance feature learning, we designed the geometric relation function for the Local Relation Aggregation Module, which is more suitable for extracting high-frequency geometric features in the local region of the reference point. Additionally, contextual relative semantic encoding serves as position encoding in the Global Context Aggregation Module to capture richer position and semantic information. We refer to the transformer block comprising the above three main modules as Adaptive Interaction Transformer Block (AIFormer Block), as shown in Figure 1, and we build the hierarchical AIFormer architecture by stacking the AIFormer Block. Extensive experiments on three popular 3D point cloud datasets verify that AIFormer achieves state-of-the-art or comparable performances. Comprehensive ablation studies further validate the soundness and effectiveness of the AIFormer design. Our main contributions can be summarized as follows:

- We propose the Adaptive Interaction Transformer Block by extracting local relational features within the local region of a reference point and modeling long-range dependencies between reference points, respectively. Then the local and global features are fused with Adaptive Interaction. AIFormer achieves effective local and global semantic capture of points.
- We propose enhancement approaches for local and global features that facilitate the information communication of points. The former extracts high-frequency geometric features from the local region of a reference point using the geometric relation function, while the latter captures richer positional and semantic information through contextual relative semantic encoding as position encoding.
- We propose a hierarchical transformer architecture based on the AIFormer Block, called AIFormer, which is demonstrated effectiveness via extensive experiments and

analysis, and achieves state-of-the-art or comparable performance in several 3D point cloud segmentation task.

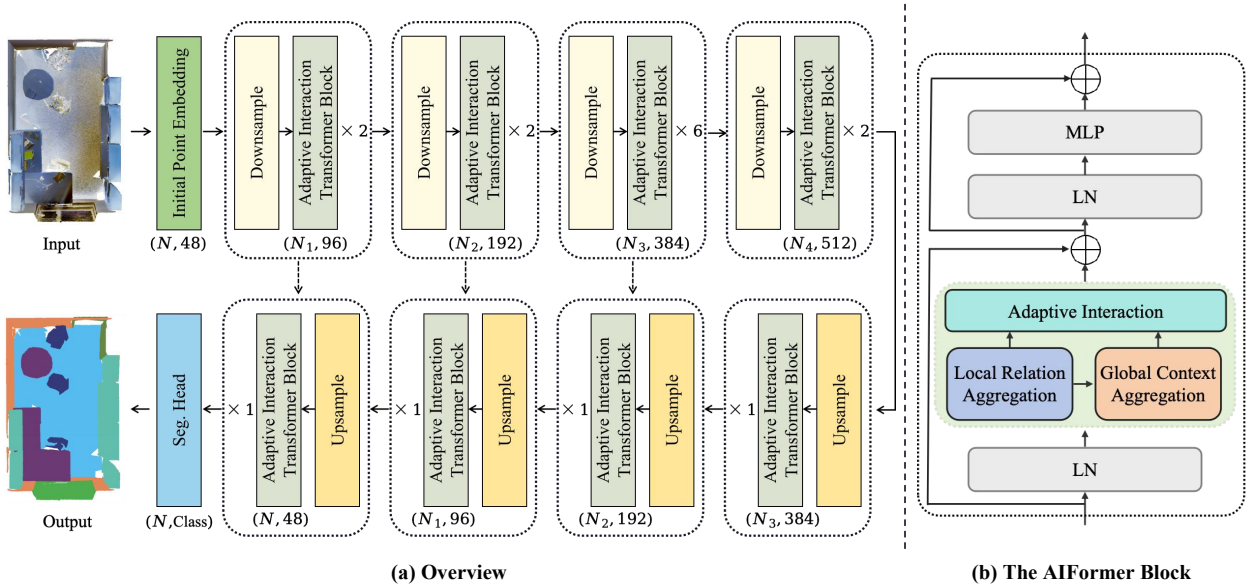


Figure 1. (**a**) Framework Overview. (**b**) Structure of Adaptive Interaction Transformer Block (AIFormer Block). Best viewed in color.

2. Related Work

2.1. 3D Point Cloud Analysis

With the popularity of 3D point cloud sensors, the deep learning-based 3D point cloud analysis has been widely studied [26], and the existing 3D point cloud analysis methods can be roughly categorized into three: voxel-based methods, projection-based methods, and point-based methods. The voxel-based methods [5–8,27–29] voxelize the 3D space and then use 3D convolution on the voxels. The earlier method [8] uses dense 3D convolution, which produces cubic memory consumption. To improve computational efficiency, methods such as MinkowskiNet [6] and SSCNs [7] proposed sparse tensors and developed open-source libraries to provide memory-efficient generalized processing operators for 3D voxels. The projection-based methods [1–4,30–37] project the 3D point cloud to the 2D mesh in order to use standard 2D convolutions such as PointCNN [2] learning the $\mathcal{X}$-transformation from the raw data and then applying the typical element-wise product and sum operations of the typical convolution operator. DFAMNet [31] maps the 3D point cloud to the image and extracts the features using a 2D grid, and then the features are fused by a dual fusion attention mechanism. Voxel-based and projection-based methods suffer from spatial information loss due to quantization or projection operations. In contrast, point-based methods operate directly on the raw data without information loss. PointNet [9] and PointNet++ [17] were pioneering work in point-based methods, proposing Multi-Layer Perceptron (MLP) and hierarchical local feature aggregation structures to capture local features, and this became the paradigm for follow-up point-based methods [10–13,38–41]. For example, RandLA-Net [10] utilizes a random sampling strategy to improve computational and memory significantly and then proposes a local feature aggregation module to keep geometric details efficiently. PointASNL [13] adaptively adjusts the sampling points in the local range by reweighting the neighboring points and fuses the information of local points and the entire point cloud in multi-scale with the Local-Nonlocal module.

2.2. Transformer Architectures

In recent years, as the Transformer architecture [42] has dominated the natural language processing tasks, it has also been used to explore the realm of vision tasks [43] with equal success. ViT [14] divides the image into non-overlapping local patches, treats each

patch as a token, and extracts the image features directly using a Transformer encoder. Building upon the success of ViT, various architectures with multi-scale resolution or incorporate local-global information [44–50] have been proposed to address the issue of low-resolution output in dense prediction and the high computational burden of global self-attention. For instance, PVT [48] devises a pyramid hierarchical shrinking architecture to explore multi-resolution features and utilize spatial-reduction attention to save memory. At the same time, Swin Transformer [46] employs window-based attention to introduce a shifted window strategy in successive transform blocks to enable cross-window communication. MixFormer [45] considers the feature fusion between channels and spatial dimensions and designs their bidirectional interaction to provide complementary clues. Further, some approaches proposed local-global feature fusion to improve the performance, such as Twins-SVT [51] leverages interleaved local and global attention for higher throughput and significant performance. SepViT [52] proposed a depthwise separable self-attention to facilitate local-global information interaction within and among the windows.

Our work is inspired by FAT [53], which uses the human visual system to inspire a bidirectional adaptation process to local and global information in a context-aware manner. Both it and our goal is to explore the bidirectional interaction of local and global information. However, FAT is based on 2D image design, and employs classical self-attention for local information processing, while our approach utilizes relational functions for extracting local information.

2.3. Point Cloud Transformers

As the Transformer architecture has shown powerful capabilities in modeling long-range dependencies, it is a natural idea to explore [54,55] the application to non-regular 3D point cloud data. PTv1 [16] and PCT [15] explore the Transformer architecture for 3D point cloud data at the same period. PCT performs global attention on the entire 3D point cloud and achieves decent performance in object classification and shape segmentation, but it suffers from high memory. PTv1 introduces local attention, called vector attention, for each point, which reduces memory consumption and extends the network to large-scale outdoor scene segmentation. Follow-up variants like FastPointTransformer [19] and PTv2 [56] have introduced voxel hashing or grouped vector attention to improve network efficiency and performance. Additionally, several methods [18,57,58] have projected point features into regular features to use more mature 2D transformer technologies. Other approaches [20,59–61] have extended the 2D transformer architecture to 3D. For instance, ST [20] adopted the Swin Transformer, facilitating cross-window communication for successive windows and proposing a stratified strategy to extend the receptive field.

Our work explores the modeling and interaction of local and global information of points, alongside many recent studies [19,21–25,62–66]. For example, PointCAT [65] introduces two separate cross-attention transformer branches combined with multi-scale features for point representation. APPT [25] introduces global pivot attention to extract global features and expand the effective receptive field. An asymmetric parallel structure is also designed to integrate local and global information effectively.

3. Method

We first review typical point-based transformer architectures and overview the architecture of AIFormer in Section 3.1. Then, we propose the Adaptive Interaction Transformer Block in Section 3.2, which mainly consists of the Local Relation Aggregation Module, Global Context Aggregation Module, and Adaptive Interaction Module. Finally, we introduce the relevant components of AIFormer in Section 3.3.

3.1. Review Point-Based Transformer

Formally, given an unordered 3D point cloud set $\mathcal{X} = (\mathcal{P}, \mathcal{F})$ with N points, where the coordinate set $\mathcal{P} = \{p_i | i = 1, 2, \ldots, N\} \in \mathbb{R}^{N \times 3}$ denotes the (x, y, z) position feature of each point p_i, and the feature set $\mathcal{F} = \{f_i | i = 1, 2, \ldots, N\} \in \mathbb{R}^{N \times c)}$ denotes the extra

features, which may contain in the corresponding point p_i, such as (r, g, b) colors, normal vectors, strengths, etc., and c denote the channels of f_i.

Then, the nearest-neighbor points set $\mathcal{M}(x_i) = \{(p_j, f_j) | p_j \in \text{Neighborhood}(p_i)\}$ is constructed for the point x_i in 3D space. Neighborhood() denotes the nearest neighbor algorithm, such as KNN, Ball Query, etc. We treat the nearest-neighbor set $\mathcal{M}(x_i)$ as a local region of the point x_i. Then, we feed $\mathcal{M}(x_i)$ into individual Linear layers or MLPs to generate the query, key, and value, respectively. The formulation is represented as follows:

$$\mathbf{Q} = \text{Linear}_q(\mathcal{M}(x_i)), \mathbf{K} = \text{Linear}_k(\mathcal{M}(x_i)), \mathbf{V} = \text{Linear}_v(\mathcal{M}(x_i)), \tag{1}$$

here, $\mathbf{Q} \in \mathbb{R}^{K \times d}, \mathbf{K} \in \mathbb{R}^{K \times d}, \mathbf{V} \in \mathbb{R}^{K \times d}$, K denotes the number of nearest-neighbor points of $\mathcal{M}(x_i)$ and d represents the channels of feature. Then, we can represent the self-attention (SA) operation on a 3D point cloud as:

$$\mathbf{W}^{attn} = \text{Softmax}(\mathbf{Q}\mathbf{K}^\top / \sqrt{d}), \tag{2}$$

$$\mathbf{f}_{x_i} = \mathcal{A}(\mathbf{W}^{attn}\mathbf{V}), \tag{3}$$

where $\mathbf{W}^{attn} \in \mathbb{R}^{K \times K}$ denotes the attention map, $\mathcal{A}$ denotes the SUM operation to aggregate the features of the local region. The $\mathbf{f}_{x_i} \in \mathbb{R}^d$ is the output feature vector of $\mathcal{M}(x_i)$, aggregated to the point x_i. For the sake of clarity, we omit **bias** in Softmax() and describe details in Section 3.2.2. Note that the above equations only show the calculation process for a single local region, and all local regions work in the same manner independently.

Review that the self-attention operation in point-based Transformer, which can only capture features within the nearest-neighbor points set $\mathcal{M}(x_i)$. This operation makes establishing a favorable long-distance dependency on the reference point difficult, even if the number of nearest-neighbor points is increased.

To address this problem, we propose the Adaptive Interaction Transformer. As shown in Figure 1a, our network architecture utilizes a U-Net structure with skip connections, which contains four levels of encoders and decoders. Specifically, before the Encoder, the Initial Point Embedding layer is used to embed the local information of the raw points. Then, the backbone is stacked by a Downsample or Upsample block and several AIFormer Blocks. The Downsample or Upsample block provides a reference point set, while the AIFormer Block processes and fuses local and global semantic features of the points in an adaptive interaction manner. Following the backbone, the Segmentation Head (Seg. Head) block predicts the category label of each point. Detailed parameters of the model architecture are described in Section 4.2. In the next subsections, we introduce the modules of the AIFormer Block in detail.

3.2. Adaptive Interaction Transformer Block

As shown in Figure 1b, the AIFormer Block consists of three modules: the Local Relation Aggregation Module (LRA) to capture the local relational features of the point, the Global Context Aggregation Module (GCA) to capture the long-range semantic information of the point, and the Adaptive Interaction Module (AI) that provides adaptive interaction for local and global features.

3.2.1. Local Relation Aggregation Module

Considering the balance between computation and memory, we sample a subset of point cloud set $\mathcal{X}$ as the reference point set $\mathcal{R}$. we take the reference point set $\mathcal{R}$ as the center point and use the entire point cloud set $\mathcal{X}$ in the current stage as the query set to construct the nearest-neighbor point set $\mathcal{M}^{local}(\mathcal{R}) \in \mathbb{R}^{L \times K^{\text{local}} \times d}$, where L is the number of reference points, and K^{local} is the number of nearest-neighbor points in each local region in the LRA.

By observing the local regions, we find two simple facts: First, local regions contain a large amount of high-frequency geometric information in 3D, i.e., low-level geometric relation features. Second, local regions naturally have a large amount of overlap between them.

Based on the above two simple facts, we take the convolution instead of complex self-attention operation as the feature extraction unit for local regions. The convolution has an inductive bias that benefits the extraction of high-frequency local information. The natural overlap between local regions facilitates cross-region communication. We formulate the convolution operation for the local region as follows:

$$\mathbf{f}_{r_i}^{local} = \mathcal{A}(\phi(r_j)), r_j \in \mathcal{M}^{local}(r_i), \tag{4}$$

where ϕ denotes the convolution operation and $\mathbf{f}_{r_i} \in \mathbb{R}^d$ is the output vector after aggregation of the local region features.

However, we realized that the naive convolution operation ignores the rich geometric relation expression in the local region [17]. Therefore, we design local relational convolutions to obtain inductive local representations for more robust representation.

Figure 2a illustrates the proposed LRA. Specifically, for each local region $\mathcal{M}^{local}(r_i)$, we explicitly encode the low-level geometric relation $\mathbf{g}_{ij}$ between the reference point r_i and its nearest-neighbor points with a position vector. We obtain an explicit representation of the low-level geometric details of the local region in 3D space. Then, instead of naive convolution, we use geometric relation function θ to map $\mathbf{g}_{ij}$ to high dimensions to explore high-level semantic relations. This process can be described as:

$$\hat{\mathbf{g}}_{ij} = \theta(\mathbf{g}_{ij}), \tag{5}$$

$$\mathbf{f}_{r_i}^{local} = \mathcal{A}(\hat{\mathbf{g}}_{ij} \odot x_j), \tag{6}$$

where θ is the geometric relation function, achieved through PointNet-like MLPs, and $\odot$ is the Hadamard product. The details of the low-level geometric relation $\mathbf{g}_{ij}$ are discussed in Section 4.4.1.

To conclude, the Local Relation Aggregation Module is capable of summarizing the high-level semantic relations within the reference point set $\mathcal{R} = (\mathcal{P}^{\mathcal{R}}, \mathcal{F}^{\mathcal{R}_{local}})$. It is important to note that the semantic representation at this stage remains localized and does not encompass long-range information out of the local region. To address this, we employ the Global Context Aggregation Module to facilitate long-range semantic awareness of the points.

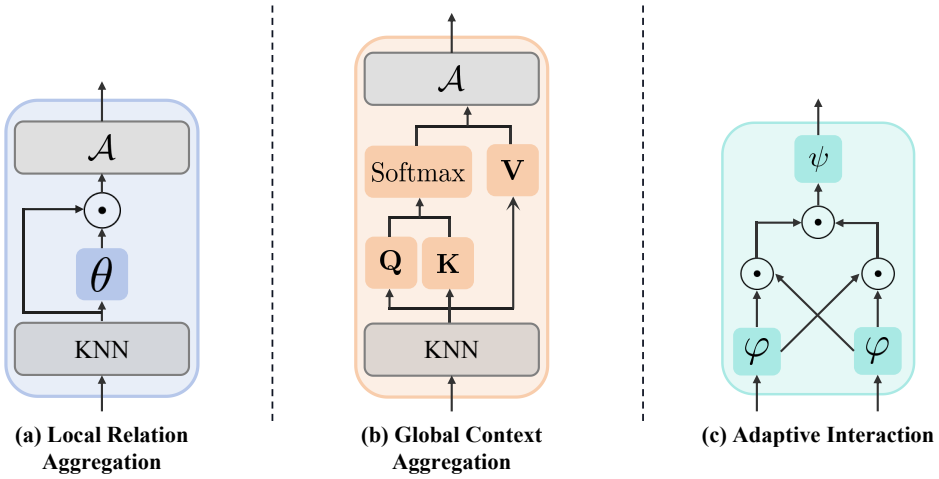


Figure 2. Illustration of the Local Relation Aggregation Module, the Global Context Aggregation Module and the Adaptive Interaction Module, respectively. Best viewed in color.

3.2.2. Global Context Aggregation Module

Like the LRA module, modeling long-range semantic dependencies also requires the construction of local regions. In the Global Context Aggregation Module. We take the reference point set $\mathcal{R}$ as the center point, and use the reference point set as the query set instead of the entire point cloud set $\mathcal{X}$ to construct the nearest-neighbor point set $\mathcal{M}^{global}(\mathcal{R}) \in \mathbb{R}^{L \times K^{\text{global}} \times d}$. L represents the number of reference points, and K^{global} denotes the number of nearest-neighbor points in GCA.

Each nearest-neighbor point represents a local region in the Local Relation Aggregation Module. So, the reference point at Global Context Aggregation Module obtains the long-range semantic information whose perception range is far beyond the Local Relation Aggregation Module. In this way, we achieve long-distance dependencies of points within the local region of the reference point and also express the entire point set in terms of a small subset, i.e., the reference point set. Then, as in Equation (1), we project the local region to query, key and value, which can be expressed as:

$$\mathbf{Q} = \text{Linear}_q(\mathcal{M}^{global}(r_i)), \mathbf{K} = \text{Linear}_k(\mathcal{M}^{global}(r_i)), \mathbf{V} = \text{Linear}_v(\mathcal{M}^{global}(r_i)). \quad (7)$$

Typically, the **bias** in Equation (2) is usually to add relative position information. Such as, Swin Transformer [46] utilizes relative position encoding to improve the model performance, ST [20] captures fine-grained position information in 3D by learning contextual relative position encoding, and CDFormer [23] proposes context-aware position encoding, which computes the relative position differences with the current feature interaction, which further enhances the positional cues.

Inspired by these works, we propose contextual relative semantic encoding (CRSE). Specifically, CRSE is calculated by the difference between relative positions and relative features as **bias**. For given any local region $\mathcal{M}^{global}(r_i)$, we denote the $\mathbf{p}_{r_i} \in \mathbb{R}^{1 \times 3}$ is a coordinates vector of r_i, and $\mathbf{p}_{r_j} \in \mathbb{R}^{\text{K}^{\text{global}} \times 3}$ is the coordinate vector of the nearest-neighbor points. Thus, we give the relative position difference $\Delta\mathbf{p}_{r_i} = \mathbf{p}_{r_i} - \mathbf{p}_{r_j}$, where $\Delta\mathbf{p}_{r_i} \in \mathbb{R}^{\text{K}^{\text{global}} \times 3}$ denotes the relative distance difference between the center point of the the local region and its nearest-neighbor points. Similarly, the relative feature difference defined as $\Delta\mathbf{f}_{r_i} = \mathbf{f}_{r_i} - \mathbf{f}_{r_j}$, where $\Delta\mathbf{f}_{r_i} \in \mathbb{R}^{\text{K}^{\text{global}} \times 3}$ denotes the relative feature difference between the center point of the the local region and its nearest-neighbor points. Then, concatenated along the channel dimensions to obtain the relative semantic difference $\Delta\mathbf{r}_i = \Delta\mathbf{p}_{r_i} \oplus \Delta\mathbf{f}_{r_i}$. CRSE considers both relative position information and feature information of the point r_i with its nearest-neighbor points compared to previous work, which adaptively enhances the position and semantic information of the local region. Then the self-attention operation with CRSE can be represented as:

$$\mathbf{bias} = (\mathbf{Q})(\alpha(\Delta\mathbf{r}_i))^\top + (\mathbf{K})(\beta(\Delta\mathbf{r}_i))^\top, \quad (8)$$

$$\mathbf{W}^{attn} = \text{Softmax}((\mathbf{Q}\mathbf{K}^\top + \mathbf{bias})/\sqrt{d}), \quad (9)$$

$$\mathbf{f}_{r_i}^{global} = \mathcal{A}((\mathbf{W}^{attn})(\mathbf{V} + \gamma(\Delta\mathbf{r}_i))), \quad (10)$$

where α, β, γ is a trainable function that maps relative semantic differences $\Delta\mathbf{r}_i$ to the same dimension as $\mathbf{Q}, \mathbf{K}, \mathbf{V}$. $\mathbf{W}^{attn} \in \mathbb{R}^{K \times K}$ denotes the attention map, $\mathcal{A}$ denotes the SUM function, and $\mathbf{f}_{r_i}^{global} \in \mathbb{R}^d$ denotes the output vector summarized to r_i after aggregation of the nearest-neighbor point set $\mathcal{M}^{local}(r_i)$. Figure 2 illustrates the proposed Global Context Aggregation Module.

The above equation is the output for a reference point, and the final output is obtained as $\mathcal{R} = (\mathcal{P}^{\mathcal{R}}, \mathcal{F}^{\mathcal{R}_{global}})$ after summarization. Using the Global Context Aggregation Module, the reference point set $\mathcal{R}$ achieves the semantic representation of long-range dependencies.

3.2.3. Adaptive Interaction Module

For each reference point r_i, after the LRA and the GCA, it ideally captures both the detailed relational features of the local region within the reference point and the long-range semantic dependencies between the reference points are included. However, the features of $\mathcal{F}^{\mathcal{R}_{local}}$ and $\mathcal{F}^{\mathcal{R}_{global}}$ still suffer from insufficient interaction.

As shown in Figure 2c, we designed the Adaptive Interaction Module to implement global-to-local and local-to-global cross-interaction to achieve full fusion of the two, which can be represented as follows:

$$\hat{\mathbf{f}}_{r_i}^{local} = \mathbf{f}_{r_i}^{local} \odot \varphi(\mathbf{f}_{r_i}^{global}), \tag{11}$$

$$\hat{\mathbf{f}}_{r_i}^{global} = \mathbf{f}_{r_i}^{global} \odot \varphi(\mathbf{f}_{r_i}^{local}), \tag{12}$$

$$\hat{\mathbf{f}}_{r_i} = \psi(\hat{\mathbf{f}}_{r_i}^{local} \odot \hat{\mathbf{f}}_{r_i}^{global}), \tag{13}$$

where φ is the adaptive function to normalize the feature, here we use Sigmoid(), $\odot$ is the Hadamard product, and ψ is the interaction function, we use Linear layer to achieve interaction fusion of the two features. Equation (11) represents the global-to-local feature interaction, $\hat{\mathbf{f}}_{r_i}^{local} \in \mathbb{R}^d$ is the output of the local relation feature after it gets interacted with the global semantic feature. Similarly, Equation (12) represents local-to-global feature interactions, yielding $\hat{\mathbf{f}}_{r_i}^{global} \in \mathbb{R}^d$. Equation (13) represents the process of adaptive interaction, and $\hat{\mathbf{f}}_{r_i}$ indicates the fusion of local and global features to obtain the final output. Summarizing the output features of all reference points yields $\hat{\mathcal{R}} = (\mathcal{P}^{\mathcal{R}}, \hat{\mathcal{F}}^{\mathcal{R}})$.

3.3. Relevant Components

3.3.1. Initial Point Embedding

Typically, using linear to project the original features prior to input into the transformer block to a uniform high dimensional facilitates the model to efficiently process and learning the sophisticated patterns of the input data. However, we found poor performance after embedding the original features by linear or Position Encoding [16] in the Initial Point Embedding layer. Inspired by the ST [20] employ KPConv [11] in the Initial Point Embedding layer to summarize the features of the local region of each point at the finest level. It yields decent improvements while suffering only minimal additional computation. For different datasets, we set the initial receptive field of KPConv as the corresponding grid size in Table 1 and set the maximum number of nearest neighbor points as 16. Meanwhile, to enhance the position information and avoid the interference of other features, we only use (x, y, z) coordinates as the input. The number of channels is also set to a quarter of the first AIFormer Block following the setting of ST. With this setup, we achieve a performance improvement far beyond Linear, while the architectural impact of this change is minimal. We demonstrate the effectiveness of this modification in our experiments (Sections 4.4.6 and 4.4.5). It also demonstrates the core idea in our paper that a larger receptive field has a more powerful characterization.

Table 1. Data augmentation settings.

Dataset	Drop Points	Rotate	Flip	Scale	Jitter	Distort	Chromatic	Grid Size
ScanNetv2	✓	✓	✓	✓	✓	✓	✓	0.02 m
S3DIS	✓		✓	✓	✓		✓	0.02 m
SemanticKITTI		✓	✓	✓	✓			0.04 m

3.3.2. Downsample and Upsample

Traditional point-based Transformer architectures [16,20,56] typically utilize downsampling blocks to reduce the scale of the point cloud. For instance, PTv2 [56] introduced Partition-based Pooling, aggregating points within the same grid into a point. This type of downsampling block integrates reference point sampling, nearest-neighbor point querying,

and feature aggregation, thereby improving pooling efficiency but resulting in losing point features. In contrast, AIFormer provides a simpler sampling strategy and avoids any loss of detail in the 3D point cloud. We provide only the reference point set in the Downsample Block, and integrate the nearest-neighbor point query and feature aggregation operations in the AIFormer Block.

Specifically, in the Downsample Block, as shown in Figure 3, inspired by Partition-based Pooling, the 3D point cloud space is partitioned into non-overlapping grids based on the (x, y, z) coordinate of the 3D point cloud set $\mathcal{X}$. Then record the indexes of all points in each non-empty grid and construct the mapping table. For each grid, we take the point that is closest to the center is referred to as the reference point. Collecting all the reference point is called a reference point set $\mathcal{R}$. In our implementation, the grid size for each stage is determined by the sampling multipliers and the grid size of the previous stage. In this way, both the precise coordinates of the points are preserved and the overlapping area of the receptive field is effectively controlled.

In the Upsample Block, as shown in Figure 3, unlike common practices such as interpolation, we have implemented a more straightforward and precise upsampling operation by index mapping. We use the mapping table saved in the Downsample Block to map the features from the reference point to other points in the same grid via index lookups.

It is worth mentioning that index lookup is the method with the lowest time complexity of $O(1)$. The space complexity is $O(N)$. Therefore, it has negligible time consumption and RAM space consumption.

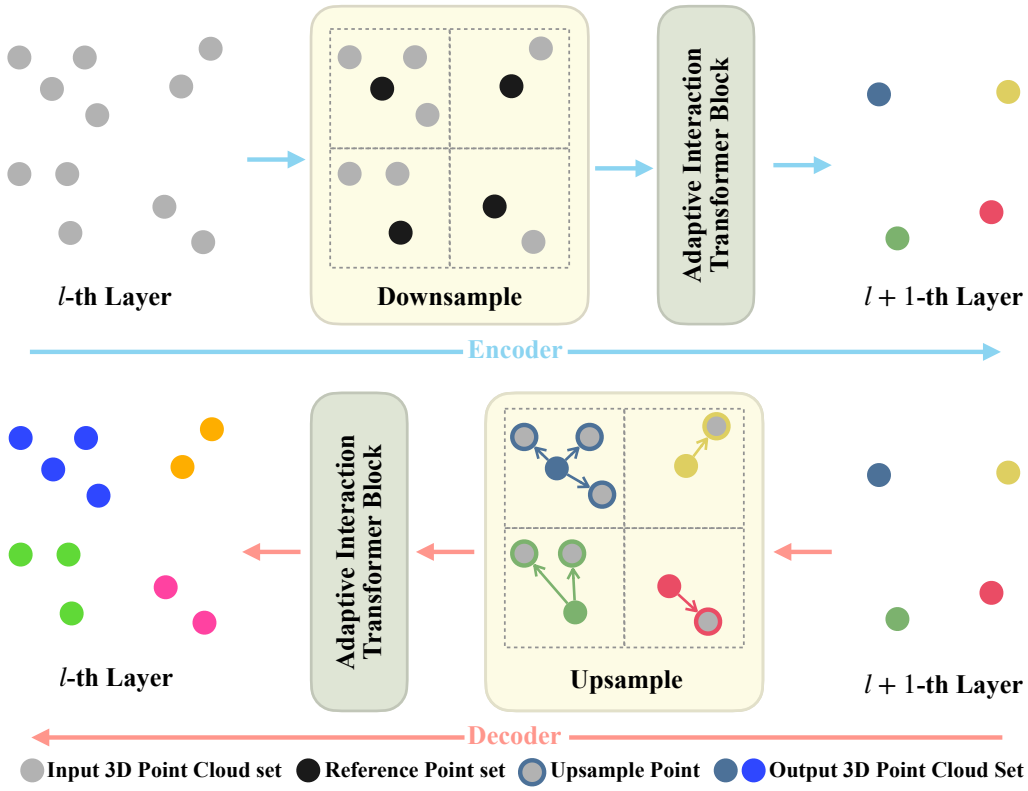


Figure 3. Illustration of the Downsample and Upsample, best viewed in color.

4. Experiments

4.1. Data and Metric

Dataset Description. We conducted 3D point cloud semantic segmentation experiments on two indoor scene datasets, S3DIS [67] and ScanNetv2 [68], and an outdoor scene dataset, SemanticKITTI [69].

- The Stanford Large-Scale 3D Indoor Spaces (S3DIS) is one of the notable indoor datasets. The dataset focuses on indoor environments, scanning six large indoor areas in three different buildings. These data are primarily structured as RGB-D imagery, and converted into point clouds. S3DIS offers detailed semantic labels, consisting mainly of thirteen common object categories (e.g., walls, floors, chairs, tables, etc.). Typically, Areas 1–4 and 6 are used as the training set, while Area 5 serves as the test set. We follow this convention to obtain results that can be fairly evaluated with existing methods.
- ScanNetv2 is a richly annotated dataset of 3D scans of indoor environments. It covers a wider variety of indoor environments, extending from educational and office spaces to residential rooms and public spaces. ScanNetv2 includes semantic annotations for over 2.5 million segments in more than 1500 scans, across hundreds of different spaces, containing more than 17.5 million annotation points. Each point is assigned a semantic label from 20 categories (e.g., show curtain, refrigerators, picture, etc.). The dataset is divided into three parts: 1201 scenes for the training, 312 scenes for the validation, and 100 scenes for online testing. Due to its high-quality point cloud annotations and challenging scenes, it is widely used for tasks such as 3D semantic segmentation, 3D object recognition, and other forms of scene understanding. Owing to its high-quality point cloud annotations and the variety of challenging indoor scenes, ScanNetv2 is widely used for tasks such as 3D semantic segmentation, 3D object recognition, and other forms of scene understanding.
- SemanticKITTI is a dataset specifically focused on outdoor large-scale scenes. The dataset is an extension of the KITTI Vision Benchmarking Suite, using LiDAR point clouds collected from a vehicle moving through urban and rural areas, but with dense semantic annotation of the dataset's 3D point clouds. Different in scale and classes from the indoor scenes dataset, the data provided consists mainly of nineteen common outdoor environments classes. Not only traffic participants are included, but also functional classes for ground such as parking lots and sidewalks. Typically, sequences 0–7 and 9–10 are used for training, sequence 8 for validation, and sequences 11–21 for online testing. The dataset is primarily used to evaluate semantic segmentation and other tasks in autonomous driving scenes.

Evaluation Metrics. For our experimental dataset, we employed three evaluation metrics: overall accuracy (OA), mean Accuracy (mAcc), and mean intersection over union (mIoU):

$$\mathrm{OA} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}, \tag{14}$$

$$\mathrm{mAcc} = \frac{1}{C} \sum_{i=1}^{C} \frac{\mathrm{TP_i} + \mathrm{TN_i}}{\mathrm{TP_i} + \mathrm{FP_i} + \mathrm{FN_i} + \mathrm{TN_i}}, \tag{15}$$

$$\mathrm{mIoU} = \frac{1}{C} \sum_{i=1}^{C} \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i}, \tag{16}$$

where C is the number of classes, TP is the true positive sample, TN is the true negative sample, FP is the false positive sample, and FN is the false negative sample.

4.2. Experiment Settings

Data Preprocessing and Augmentation. For the S3DIS dataset, we adopted the data preprocessing and data augmentation from PTv1 [16], using (x, y, z) coordinates and (r, g, b) colors as inputs fed into the network. For ScanNetv2, we followed the data preprocessing and data augmentation of ST [20] and used (x, y, z) coordinates and normal vectors as inputs. Meanwhile, SemanticKITTI uses (x, y, z) coordinates and strength as inputs, and we followed the data augmentation of SPVNAS [5]. During training, the maximum number of points fed into the network is 80,000 for S3DIS, 100,000 for ScanNetv2, and 120,000 for SemanticKITTI, respectively. Table 1 provides the data augmentation implementation details.

Implementation Details. The main architecture is shown in Figure 1a, and the main hyper-parameter is available in Table 2. There are four stages of encoder and decoder with block depths of [2, 2, 6, 2] and [1, 1, 1, 1], respectively. Specifically, for ScanNetv2, the point number is much larger. Therefore, we equip the four stages of the encoder with additional block depths of [3, 9, 3, 3].

The sampling multipliers for the downsample stage are [×3.5, ×3.0, ×2.5, ×2.0], respectively. The sampling multiplier is the grid size scaling ratio over the previous stage. Take the grid size of ScanNetv2 in Table 1 as an example, the basic grid size is 0.02 m. According to the sampling multiplier, the downsample stage sizes are [0.07, 0.21, 0.525, 1.05] m.

Table 2. Main hyper-parameter settings.

Dataset	Epoch	Leraning Rate	Weight Decay	Scheduler	Optimizer	Batch Size
ScanNetV2	1200	0.001	0.02	Cosine	AdamW	16
S3DIS	1200	0.006	0.01	OneCycleLR	AdamW	12
SemanticKITTI	80	0.002	0.005	Cosine	AdamW	8

Experimental Environment. All experiments in this paper are performed on a single node machine with four RTX 3090 GPUs, 345G RAM, and Intel(R) Xeon(R) Silver 4210R CPU. Pytorch version is 1.12.1 and CUDA version is 11.3.

4.3. Experiment Result and Analysis

4.3.1. Evaluation on S3DIS

We spent about 18 h training S3DIS, with an average training latency of 314 ms for a single sample and an average inference latency of 259 ms. However, we must emphasize that the time required may vary from device to device or load to load. Table 3 demonstrates the results of AIFormer compared with previous methods. Our method outperforms the point-based transformer methods [15,16,19], achieving improvements of 0.3%, 9.4%, and 0.4%, respectively, and reaches 70.7% on mIoU. AIFormer also outperforms other representative methods, including voxel-based method [6], projection-based method [2], point-based method [10,13], exhibiting significant performance advantages.

Table 3. Semantic segmentation results evaluated on S3DIS [67] (Area 5). The best results are presented in **bold**, the second-best results are underlined, and "-" means unknown.

Method	mIoU (%)	mAcc (%)	OA (%)	Ceiling	Floor	Wall	Beam	Colum	Window	Door	Table	Chair	Sofa	Bookcase	Board	Clutter
PointNet [9]	41.1	49.0	-	88.8	97.3	69.8	0.1	3.9	46.3	10.8	52.6	58.9	40.3	5.9	26.4	33.2
PointNet++ [17]	57.3	63.5	-	91.3	96.9	78.7	0.0	16.0	54.9	31.9	83.5	74.6	67.2	49.3	54.2	45.9
PointCNN [2]	57.3	63.9	85.9	92.3	98.2	79.4	0.0	17.6	22.8	62.1	80.6	74.4	66.7	31.7	62.1	56.7
PointWeb [3]	60.3	66.6	87.0	92.0	98.5	79.4	0.0	21.1	59.7	34.8	76.3	88.3	46.9	69.3	64.9	52.5
MinkwskiNet [6]	65.4	71.7	-	91.8	**98.7**	86.2	0.0	34.1	48.9	62.4	89.9	81.6	74.9	47.2	74.4	58.6
PointConv [12]	58.3	64.7	85.4	92.8	96.3	77.0	0.0	18.2	47.7	54.3	87.9	72.8	61.6	65.9	33.9	49.3
KPConv [11]	65.4	70.9	-	92.6	97.3	81.4	0.0	16.5	54.5	69.5	90.1	80.2	74.6	66.4	63.7	58.1
PointWeb [3]	61.9	68.3	87.2	91.5	98.2	81.4	0.0	23.3	**65.3**	40.0	75.5	87.7	58.5	67.8	65.6	49.7
PointASNL [13]	62.6	68.5	87.7	**94.3**	98.4	79.1	0.0	26.7	55.2	66.2	83.3	86.8	47.6	68.3	56.4	52.1
PCT [15]	61.3	67.7	-	92.5	98.4	80.6	0.0	19.3	61.6	48.0	76.5	85.2	46.2	67.7	67.9	52.2
PTv1 [16]	70.4	76.5	-	94.0	98.5	**86.3**	0.0	38.0	63.4	74.3	89.1	82.4	74.3	**80.2**	76.0	59.3
PointNeXt [39]	70.5	76.8	**90.6**	94.2	98.5	84.4	0.0	37.7	59.3	74.0	83.1	91.6	77.4	77.2	78.8	60.6
CBL [70]	69.4	75.2	**90.6**	93.9	98.4	84.2	0.0	37.0	57.7	71.9	**91.7**	81.8	77.8	75.6	69.1	**62.9**
FastPointTrans. [19]	70.3	**77.9**	-	94.2	98.0	86.0	**0.2**	**53.8**	61.2	**77.3**	81.3	89.4	60.1	72.8	80.4	58.9
Our	**70.7**	75.9	90.5	91.7	98.3	84.1	0.0	28.0	63.5	75.5	81.7	**92.1**	**80.3**	80.1	**82.5**	61.2

The qualitative results of indoor semantic segmentation are visualized in Figure 4. AIFormer leverages both local and global features of points, utilizing adaptive interactive fusion to enhance feature representation. This approach significantly improves the model's ability to predict small objects in challenging 3D scenes accurately. For example, in the first and second rows of Figure 4, AIFormer is relatively complete in predicting the category of windows and doors, while the comparison methods are relatively poor in semantic segmentation. In the third row, the AIFormer achieves more complete predict the categories of bookcase in this sample compared to the PCT and PTv1.

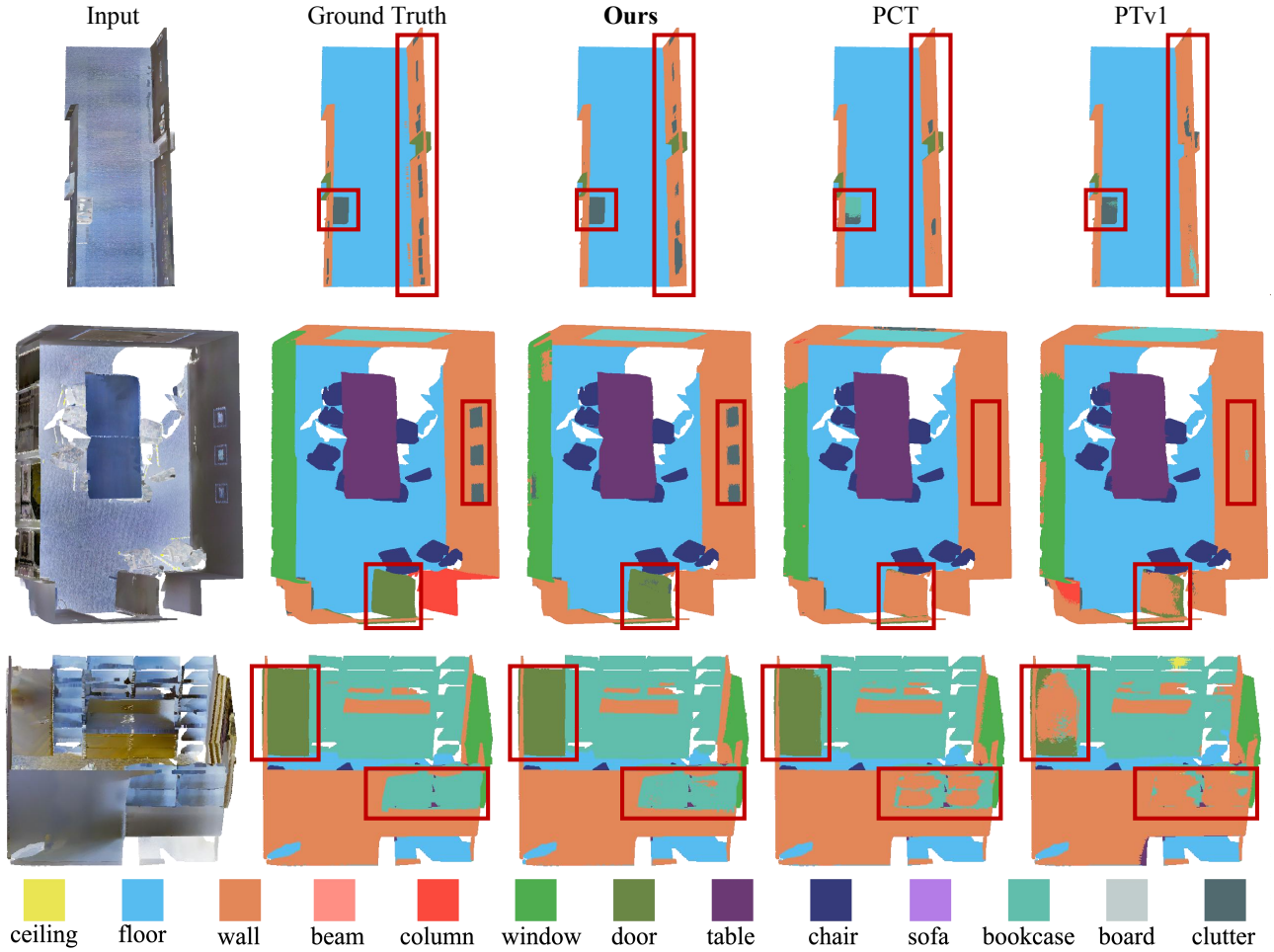


Figure 4. Visual comparison of our model with other methods [15,16] on S3DIS. Differences in semantic segmentation results are highlighted using red box for clarity.

4.3.2. Evaluation on ScanNetv2

ScanNetv2 contains more categories and scenarios than S3DIS, presenting more difficult challenges. We spent about 14 h training ScanNetv2, with an average training latency of 613 ms for a single sample and an average inference latency of 189 ms. And we summarize the results of comparing AIFormer with existing methods in Table 4. Our method maintains a competitive advantage over the majority of approaches and achieves up to 74.9% on mIoU. AIFormer achieves the second highest accuracy in 8 out of 20 categories, while PTv3 [71], the newest model with significant current performance, dominates the highest accuracy in almost all categories. However, it is worth noting that the performance is significantly improved by greatly expanding the receptive field from 16 to 1024 points, a useful but more brute force approach. In contrast, we indirectly, capture the local region first and then use the local feature to establish the global dependency, which is a more elegant way. On the other hand, PTv3 implements many new techniques and is difficult

to deploy compared to other approaches. In contrast, our approach has a simple overall architecture and is easy to deploy.

Figure 5 visualizes the qualitative semantic segmentation results on ScanNetv2. In the first and second rows, AIFormer accurately segments irregularly overlapping objects by capturing short- and long-range features and an adaptive interactive fusion mechanism. It is worth mentioning that the third row, where the objects are chaotically placed and spatially very close, poses a challenge for existing methods with limited receptive fields to distinguish their semantic information. However, AIFormer still predicts the cabinet category more accurately. At the same time, we also note that points with the ground truth labeled "table" were incorrectly predicted as "cabinet". We attribute the misprediction to an overly large receptive field that allows the point to capture a large number of features from the surrounding space, while the subdivision category fails to achieve sufficient convergence in training and is an issue we need to improve upon further.

By analyzing both quantitative and qualitative results and integrating findings from two indoor datasets, ScanNetv2 and S3DIS, AIFormer effectively achieves the capture of local and global information, models detailed structural information through adaptive interaction, and achieves superior segmentation performance for small objects such as chair, sofa, window, etc., which validates the effectiveness of AIFormer.

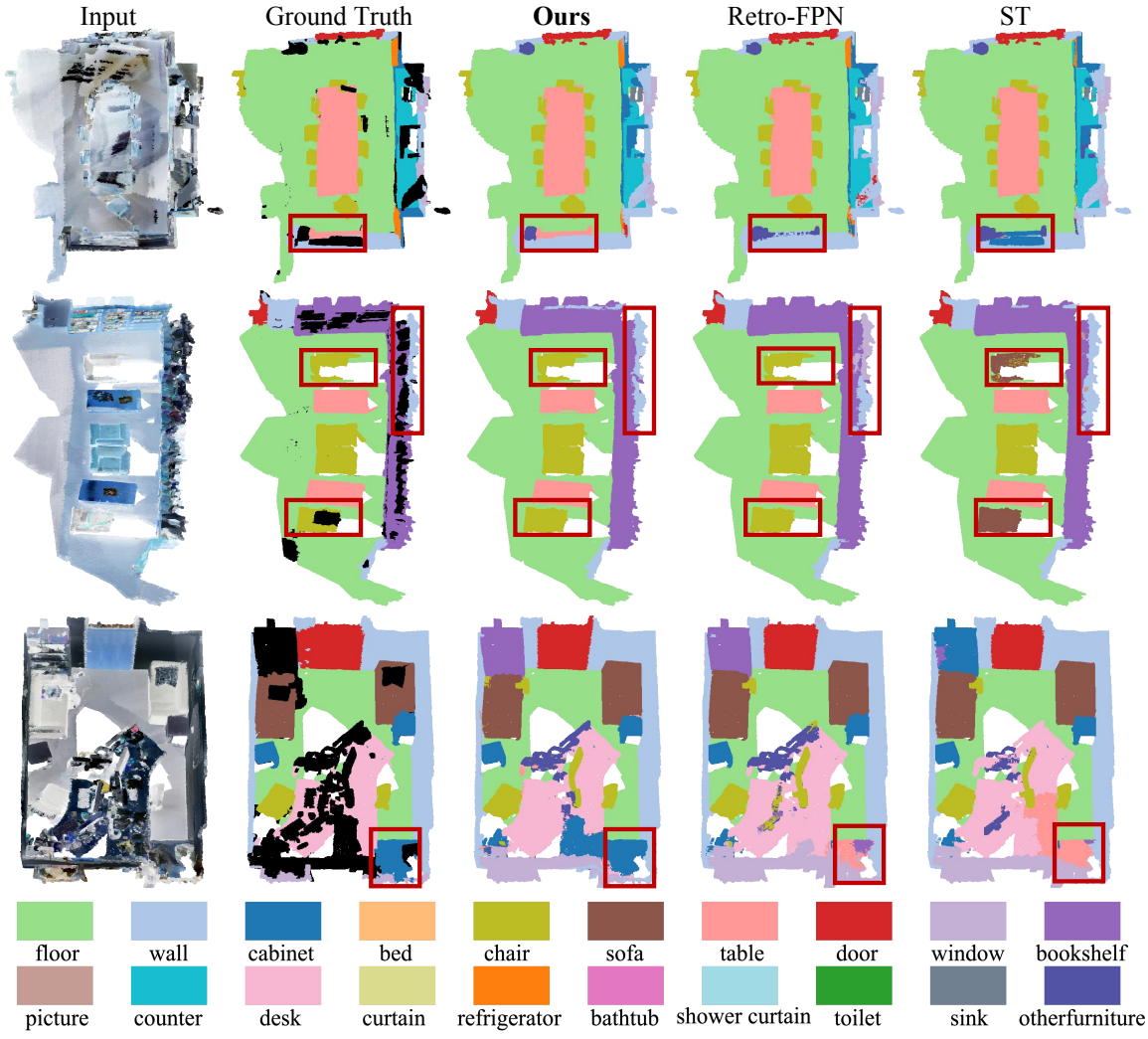


Figure 5. Visual comparison of our model with other methods [20,59] on ScanNetv2. Note that black indicates ignored labels, and differences in semantic segmentation results are highlighted using red box for clarity.

Table 4. Quantitative results of our proposed method and state-of-the-art LiDAR semantic segmentation methods on the *test* set of ScanNetv2 [68]. The best results are presented in **bold**, and the second-best results are underlined. The scores are obtained from the official leaderboard of ScanNetv2 when available, otherwise from the respective paper.

Method	mIoU (%)	Bathtub	Bed	Bookshelf	Cabinet	Chair	Counter	Curtain	Desk	Door	Floor	Other Furniture	Picture	Refrigerator	Shower Curtain	Sink	Sofa	Table	Toilet	Wall	Window
PointNet++ [17]	33.9	58.4	47.8	45.8	25.6	36.0	25.0	24.7	27.8	26.1	67.6	18.3	11.7	21.2	14.5	36.4	34.6	23.2	54.8	52.3	25.2
PointConv [12]	66.6	70.3	78.1	75.1	65.5	83.0	47.1	76.9	47.4	53.7	95.1	47.5	27.9	63.5	69.8	67.5	75.1	55.3	81.6	80.6	70.3
KPConv [11]	68.4	84.7	75.8	78.4	64.7	81.4	47.3	77.2	60.5	59.4	93.5	45.0	18.1	58.7	80.5	69.0	78.5	61.4	88.2	81.9	63.2
PointWeb [3]	61.8	72.9	66.8	64.7	59.7	76.6	41.4	68.0	52.0	52.5	94.6	43.2	21.5	49.3	59.9	63.8	61.7	57.0	89.7	80.6	60.5
MinkowskiNet [6]	73.6	85.9	81.8	83.2	70.9	84.0	52.1	85.3	66.0	64.3	95.1	54.4	28.6	73.1	89.3	67.5	77.2	68.3	87.4	85.2	72.7
FPConv [34]	63.9	78.5	76.0	71.3	60.3	79.8	39.2	53.4	60.3	52.4	94.8	45.7	25.0	53.8	72.3	59.8	69.6	61.4	87.2	79.9	56.7
PointASNL [13]	66.6	78.1	75.9	69.9	64.4	82.2	47.5	77.9	56.4	50.4	95.3	42.8	20.3	58.6	75.4	66.1	75.3	58.8	90.2	81.3	64.2
BPNet [35]	74.9	90.9	81.8	81.1	75.2	83.9	48.5	84.2	67.3	64.4	95.7	52.8	30.5	77.3	85.9	78.8	**81.8**	69.3	91.6	85.6	72.3
RFCR [38]	70.2	88.9	74.5	81.3	67.2	81.8	49.3	81.5	62.3	61.0	94.7	47.0	24.9	59.4	84.8	70.5	77.9	64.6	89.2	82.3	61.1
CBL [70]	70.5	76.9	77.5	80.9	68.7	82.0	43.9	81.2	66.1	59.1	94.5	51.5	17.1	63.3	85.6	72.0	79.6	66.8	88.9	84.7	68.9
SegGroup [72]	62.7	81.8	74.7	70.1	60.2	76.4	38.5	62.9	49.0	50.8	93.1	40.9	20.1	56.4	72.5	61.8	69.2	53.9	87.3	79.4	54.8
ST [20]	74.7	90.1	80.3	84.5	75.7	84.6	51.2	82.5	**69.6**	64.5	95.6	57.6	26.2	74.4	86.1	74.2	77.0	**70.5**	89.9	86.0	73.4
LargeKernel [29]	74.0	91.0	82.0	80.6	74.0	85.2	54.5	82.6	59.4	64.3	95.5	54.1	26.3	72.3	85.8	77.5	76.7	67.8	93.3	84.8	69.4
REFL-Net [40]	72.9	73.7	**82.3**	76.6	72.6	85.2	46.8	86.5	68.4	63.4	95.3	56.5	29.7	77.3	77.4	77.7	74.9	66.6	90.7	85.0	70.8
Retro-FPN [59]	74.4	84.2	80.0	76.7	74.0	83.6	54.1	91.4	67.2	62.6	95.8	55.2	27.2	77.7	88.6	69.6	80.1	67.4	94.1	85.8	71.7
PTv3 [71]	**79.4**	**94.1**	81.3	**85.1**	**78.2**	**89.0**	**59.7**	**91.6**	**69.6**	**71.3**	**97.9**	**63.5**	**38.4**	**79.3**	**90.7**	**82.1**	79.0	69.6	**96.7**	**90.3**	**80.5**
Ours	74.9	73.8	80.9	84.1	77.0	83.0	53.8	90.9	67.8	67.9	96.0	54.5	33.2	76.7	79.4	74.2	78.6	69.6	91.9	88.0	77.0

4.3.3. Evaluation on SemanticKITTI

S3DIS and ScanNetv2 offer uniformly distributed indoor 3D point clouds, yet we have also evaluated AIFormer's versatility in outdoor environments with non-uniform distributions. We spent about 20 h training SemanticKITTI, with an average training latency of 369 ms for a single sample and an average inference latency of 143 ms. And we summarize in Table 5 the results of comparing AIFormer with relevant methods, and our method has the same competitive advantage. For instance, compared to the classical Cylinder3D [27] and RangeViT [73], AIFormer improves mIoU metrics by 0.2% and 4%, respectively. Although DFAMNet [31] performs better, it relies on a more sophisticated architecture and requires additional image data to form a pseudo-point cloud. In contrast, AIFormer operates solely on 3D point cloud data and is designed to be simple and lightweight. Our method also achieves surpassing or approaching performance compared to other methods. The qualitative results of the outdoor scene segmentation are visualized in Figure 6. As can be seen within the red rectangle, AIFormer takes full advantage of capturing long-range dependencies at small-sized points, reducing a large number of false predictions compared to other methods.

Modeling effective segmentation models in non-uniformly distributed 3D spaces presents a considerable challenge. Through quantitative and qualitative analysis, it is demonstrated that by expanding the receptive field of a point in AIFormer and then performing local and global feature fusion in an adaptive interactive manner, the points can extract more effective semantic information. The superior performance demonstrates the effectiveness of the AIFormer under different environmental conditions.

Table 5. Quantitative results of our proposed method and state-of-the-art LiDAR semantic segmentation methods on the *test* set of SemanticKITTI [69]. The best results are presented in **bold**, and the second-best results are underlined. The scores are obtained from the official leaderboard of SemanticKITTI when available, otherwise from the respective paper.

Method	mIoU (%)	Car	Bicycle	Motorcycle	Truck	Other-Vehicle	Person	Bicyclist	Motorcyclist	Road	Parking	Sidewalk	Other-Ground	Building	Fence	Vegetation	Trunk	Terrain	Pole	Traffic-Sign
PointNet [9]	14.6	46.3	1.3	0.3	0.1	0.8	0.2	0.2	0.0	61.6	15.8	35.7	1.4	41.4	12.9	31.0	4.6	17.6	2.4	3.7
PointNet++ [17]	20.1	53.7	1.9	0.2	0.9	0.2	0.9	1.0	0.0	72.0	18.7	41.8	5.6	62.3	16.9	46.5	13.8	30.0	6.0	8.9
KPConv [11]	58.8	96.0	32.0	42.5	33.4	44.3	61.5	61.6	11.8	88.8	61.3	72.7	31.6	**95.0**	64.2	84.8	69.2	69.1	56.4	47.4
RangeNet++ [1]	52.2	91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	91.8	65.0	75.2	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9
RandLANet [10]	50.3	94.0	19.8	21.4	42.7	38.7	47.5	48.8	4.6	90.4	56.9	67.9	15.5	81.1	49.7	78.3	60.3	59.0	44.2	38.1
SqueezeSegV3 [30]	55.9	92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9
SPVNAS [5]	66.4	**97.3**	51.5	50.8	59.8	58.8	65.7	65.2	**43.7**	90.2	67.6	75.2	16.9	91.3	65.9	86.1	73.4	71.0	64.2	66.9
PointASNL [13]	46.8	87.9	57.6	25.1	39.0	29.2	34.2	57.6	0.0	87.4	24.3	74.3	1.8	83.1	43.9	84.1	52.2	70.6	57.8	36.9
PolarNet [33]	54.3	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5
Cylinder3D [27]	67.8	97.1	**67.6**	64.0	50.8	58.6	73.9	67.9	36.0	91.4	65.1	75.5	32.3	91.0	66.5	85.4	71.8	68.5	62.6	65.6
JS3C-Net [28]	66.0	95.8	59.3	52.9	54.3	46.0	69.5	65.4	39.9	88.9	61.9	72.1	31.9	92.5	**70.8**	84.5	69.8	67.9	60.7	**68.7**
WaffleIron [32]	67.3	96.5	62.3	64.1	55.2	48.7	70.4	77.8	29.6	90.5	69.5	75.9	24.6	91.8	68.1	85.4	70.8	69.6	62.0	65.2
RangeViT [73]	64.0	95.4	55.8	43.5	29.8	42.1	63.9	58.2	38.1	93.1	**70.2**	80.0	**32.5**	92.0	69.0	85.3	70.6	71.2	60.8	64.7
DFAMNet [31]	**69.0**	96.7	54.5	**80.8**	**95.5**	**68.5**	**79.7**	**91.7**	0.2	**94.3**	50.6	**81.8**	4.1	91.8	65.8	**89.4**	73.4	**76.8**	63.1	51.9
Ours	68.0	96.6	65.4	59.8	56.2	53.6	74.6	69.3	37.3	89.6	67.7	74.2	23.4	91.8	68.9	85.9	**73.9**	71.4	**65.5**	67.4

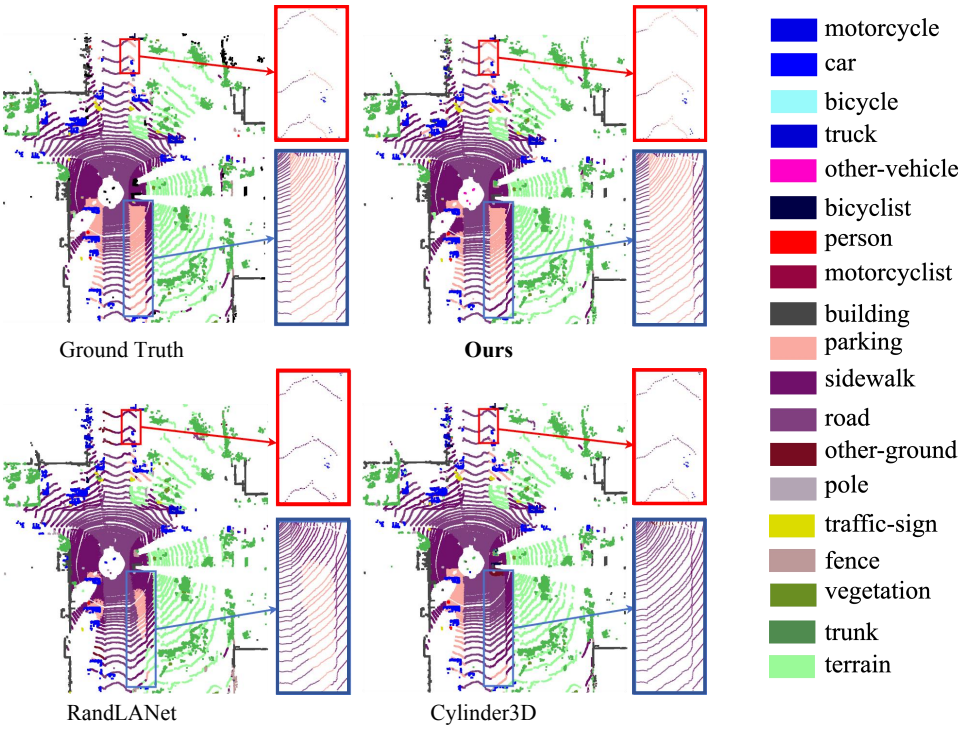


Figure 6. Visual comparison of our model with other methods [10,27] on SemanticKITTI. Note that black indicates ignored labels.

4.4. Ablation Study

Since *test* set of ScanNetv2 requires online submission, we selected *validation* set of ScanNetv2 as the baseline for conducting the ablation experiments. This setting is followed throughout in this subsection, if not otherwise specified.

4.4.1. Low-Level Geometric Relation $\mathbf{g}_{ij}$

The key to the Local Relation Aggregation Module is learning from low-level geometric relations, thereby how to define $\mathbf{g}_{ij}$ is a critical issue. We explore four intuitive definitions to verify that $\mathbf{g}_{ij}$ can reflect the geometric relations through flexible definitions, whose results are summarized in Table 6. The data indicates that using only the 3D Euclidean distance (Exp. 1), or the coordinates p_i of the reference point p_j and its nearest-neighbor point p_j (Exp. 2) as $\mathbf{g}_{ij}$, the accuracy can reach 74.3% and 74.5%, respectively. This underscores the efficacy of our LRA, which surpasses self-attention in learning high-level semantic relations from low-level geometric relations within local regions. Notably, as we incorporate additional relations such as coordinate differences, there is a corresponding increase in performance, reaching up to 75.5% (Exp. 4). These findings demonstrate that explicit definitions of geometric relations in local regions with small receptive fields can significantly enhance the exploration and understanding of geometric relations, leading to improved model performance.

Table 6. The results (%) of four intuitive low-level geometric relations $\mathbf{g}_{ij}$ on the ScanNetv2. The best results are presented in **bold**.

ID	Low-Level Relation $\mathbf{g}_{ij}$	Channels	mIoU (%)
(1)	(Ed)	1	74.3
(2)	(p_i, p_j)	6	74.5
(3)	$(\text{Ed}, p_i - p_j)$	4	75.0
(4)	$(\text{Ed}, p_i - p_j, p_i, p_j)$	10	**75.5**

4.4.2. Geometric Relation Function θ

The performance and parameters of the geometric relation function θ deployed with different layers are summarised in Table 7. It is found that the best performance of 75.52% is achieved with three layers of MLP. On the contrary, the performance decreases slightly when the number of layers is increased further. There might be too many layers that make it difficult for the network to converge. It is worth noting that a decent performance of 75.21% is achieved with only two layers of θ. This validates the power of the geometric relation function in capturing low-level set relationships in the local region.

Furthermore, combining ID 2 in Table 6 and 1 layer of MLP in Table 7, we can roughly regard it as a simple implementation of Equation (4), i.e., without additional processing of points in the local region, only using the coordinates of the center point and the nearest-neighbor points, and then extracting the features by using the convolution operator. Observation of the data reveals that the performance results are not satisfactory. By improving the low-level geometric relations, we observe that the performance of the model is further improved, which proves our view.

Table 7. The performance and parameters of different layers of the geometric relation function θ on the ScanNetv2. The best results are presented in **bold**.

MLP Layer	1	2	3	4	5
Layer Params (M)	2.06	4.11	6.17	8.22	10.27
mIoU (%)	74.92	75.21	**75.52**	75.51	75.31

4.4.3. Contextual Relative Semantic Encoding

We evaluate the impact of using only relative position difference $\Delta\mathbf{p}_{r_i}$, only relative feature difference $\Delta\mathbf{f}_{r_i}$ and using relative semantic difference $\Delta\mathbf{r}_i$ as **bias** on the Global Context Aggregation Module, respectively. Furthermore, we also evaluate the performance of the global aggregation module without using position encoding. As shown in Table 8, without position encoding comes with unsatisfactory performance, which is consistent with the observation of ST [20]. Next, using $\Delta\mathbf{p}_{r_i}$ or $\Delta\mathbf{f}_{r_i}$ as **bias** increases the performance by 0.2% and 0.8%, respectively, when compared to the model without position encoding.

Further, using $\Delta \mathbf{r}_i$ performance reaches 75.5%, achieving superior performance over ST and PTv2 [56]. This experiment shows that Contextual Relative Semantic Encoding (CRSE) effectively enhances the position and feature information. CRSE-equipped Global Context Aggregation Module significantly improves the ability to capture global semantic relations.

Table 8. Ablation study of Contextual Relative Semantic Encoding on the ScanNetv2. The best results are presented in **bold**.

Method	mIoU (%)
without **bias**	74.1
$\Delta \mathbf{p}_{r_i}$ only	74.3
$\Delta \mathbf{f}_{r_i}$ only	74.9
$\Delta \mathbf{r}_i$	**75.5**
ST [20]	74.3
PTv2 [23]	75.4

4.4.4. Efficacy of Adaptive Interaction Processing

The core operation of the Adaptive Interaction Module is the adaptive interaction process between global and local features. First, we validate the effectiveness of the adaptive function φ by comparing two typical normalization methods. As demonstrated in Table 9, results obtained without normalization of local and global features were unsatisfactory, which shows the importance of the adaptive function. Further, considering the performance and computational efficiency of the Sigmoid() and Softmax() methods, we choose Sigmoid() as the adaptive function φ more appropriately.

Note that Adaptive Interaction goes through the local and global features and returns their adaptive weight vector. This vector, which is non-trainable and generated directly by the features themselves, avoids the introduction of additional parameters. Instead of employing linear fusion, the Hadamard product of the features is computed, aligning more closely with the mechanisms of attention. In Table 9, we further show the performance impact of different feature interaction methods, showing that our proposed Adaptive Interaction method outperforms traditional linear approaches to feature fusion.

Table 9. The results(%) of different designs on adaptive interaction processing on the ScanNetv2 and S3DIS. The best results are presented in **bold**.

Function	Method	ScanNetv2	S3DIS
φ	without normalization	73.2	68.4
	Softmax()	75.3	70.3
	Sigmoid()	**75.5**	**70.7**
ψ	add + Linear	74.2	68.6
	cat + Linear	74.7	69.9
	Adaptive Interaction	**75.5**	**70.7**

4.4.5. Efficacy of Initial Point Embedding

To intuitively explore the impact of Initial Point Embedding, we show the results of different point embedding methods on training loss and validation mIoU in Figure 7. It is clear that using KPConv [11] as a point embedding method increases 0.5% and 1.1% compared to using PTv1 [16] and linear projection, respectively. Observing the training loss curve, it is found that KPConv converges faster and remains relatively stable without significant fluctuations during training. The impact of Initial Point Embedding on overall performance is further explored in Table 10, and this minor change in architecture resulted in a 1.6% improvement.

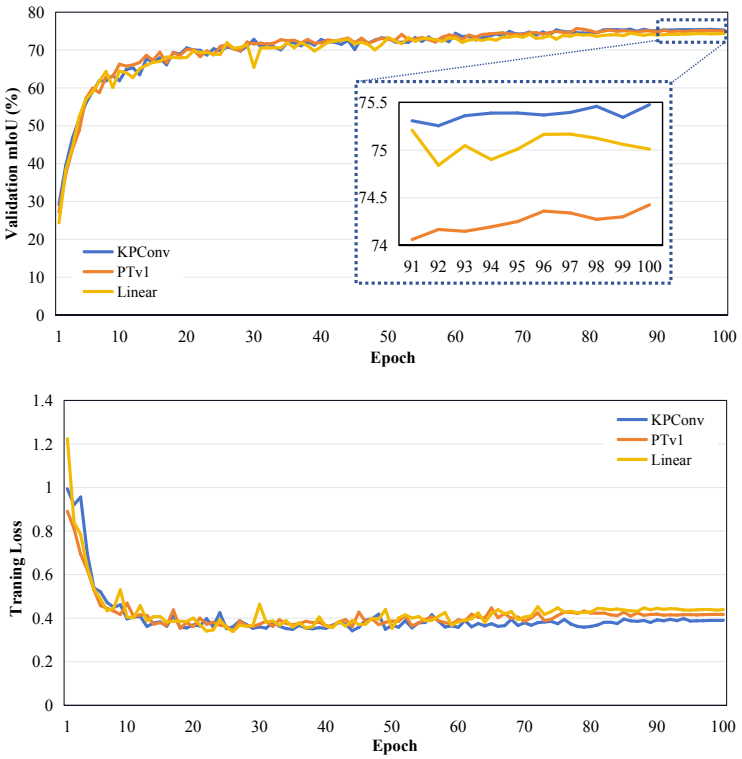


Figure 7. Validation mIoU and training loss curves of different point embedding methods on the ScanNetv2.

4.4.6. Module Design

We conduct extensive ablation studies of the modules introduced in AIFormer to verify the effectiveness of the modules. As shown in Table 10, we ablate different modules as follow: PointEmb. (Initial Point Embedding), DataAug. (Data Augmentation), LRA (Local Relation Aggregation Module), GCA (Global Context Aggregation Module), and AI (Adaptive Interaction Module).

Table 10. Ablation study of AIFormer architecture design on the ScanNetv2. The best results are presented in **bold**.

ID	PointEmb.	DataAug.	LRA	GCA	AI	mIoU (%)
(1)			✓			67.8
(2)				✓		71.8
(3)			✓	✓		72.1
(4)			✓	✓	✓	73.8
(5)	✓		✓	✓	✓	74.1
(6)		✓	✓	✓	✓	73.9
(7)	✓	✓	✓	✓		74.5
(8)	✓	✓	✓	✓	✓	**75.5**

Experiment (1) verifies the effectiveness of the LRA module, which is similar to a direct extension of PointNet++ [17] to the Transformer architecture. Meanwhile, Experiment (2) evaluates the effectiveness of the GCA module, similar to those used in PTv1 [16] and PCT [15]. Together, these experiments establish the benchmark for our model. In Experiment (4), benefiting from the AI, our benchmark results are improved by 5.8% and 3.7%, respectively. Compared to Experiment (3), the mIoU increased from 72.1% to 73.8%, demonstrating the significant role of the adaptive interaction module in enhancing long-range awareness of points. Experiments (5) and (6) validate the effects of the Initial Point Embedding and Data Augmentation strategy, respectively. The observed improvements in mIoU from Experiment (4) demonstrate the nontrivial role of proper data processing

in enhancing model performance. The comparison of Experiments (7) and (8) further highlights the effectiveness and reasonableness of our AIFormer architecture.

5. Conclusions

This paper introduces the AIFormer for 3D point cloud analysis, which mainly consists of stacked AIFormer Blocks. The AIFormer Blocks achieve effective capturing of local and global features through the included Local Relation Aggregation Module and Global Context Aggregation Module, and adaptive fusion of the local and global features by using the Adaptive Interaction Module to optimize the point representation in a feature interaction manner. Furthermore, the AIFormer Block further designs geometric relation functions and contextual relative semantic encoding to enhance local and global feature extraction capabilities. We conducted 3D semantic segmentation and extensive ablation experiments on S3DIS, ScanNetv2, and SemanticKITTI to demonstrate the superiority of AIFormer and the soundness of each design. Further, we will explore the potential of AIFormer for other 3D point cloud processing tasks.

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

Funding: This research was funded by the National Key R&D Program of China 2023YFC3806000 and 2023YFC3806002, in part by the National Natural Science Foundation of China under Grant 61936014, in part by Shanghai Municipal Science and Technology Major Project No. 2021SHZDZX0100, in part by the Shanghai Science and Technology Innovation Action Plan Project 22511105300 and in part by Fundamental Research Funds for the Central Universities.

Data Availability Statement: Publicly available datasets were analyzed in this study. The SemanticKITTI can be found here (https://semantic-kitti.org/ (accessed on 5 May 2024)). The S3DIS dataset was obtained based on the Stanford Large-Scale 3D Indoor Spaces Dataset by Matterport Camera (https://cvgl.stanford.edu/resources.html (accessed on 5 May 2024)). The ScanNetv2 can be found here (https://kaldir.vc.in.tum.de/scannet_benchmark/) (accessed on 5 May 2024).

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

References

1. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), The Venetian Macao, Macau, China, 4–8 November 2019; pp. 4213–4220. [CrossRef]
2. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution On $\mathcal{X}$-Transformed Points. In Proceedings of the 31th Advances in Neural Information Processing Systems (NeurIPS), Montréal, QC, Canada, 2–8 December 2018.
3. Jiang, L.; Zhao, H.; Liu, S.; Shen, X.; Fu, C.-W.; Jia, J. Hierarchical Point-Edge Interaction Network for Point Cloud Semantic Segmentation. In Proceedings of the 17th IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 10432–10440. [CrossRef]
4. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* **2019**, *38*, 1–12. [CrossRef]
5. Tang, H.; Liu, Z.; Zhao, S.; Lin, Y.; Lin, J.; Wang, H.; Han, S. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In Proceedings of the European Conference on Computer Vision (ECCV), Virtual, 23–28 August 2020; pp. 685–702 [CrossRef]
6. Choy, C.; Gwak, J.; Savarese, S. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3070–3079. [CorssRef]
7. Graham, B.; Engelcke, M.; van der Maaten, L. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 9224–9232. [CrossRef]

8. 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. [CrossRef]
9. Qi, C.R.; 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. 652–660. [CrossRef]
10. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. Learning Semantic Segmentation of Large-Scale Point Clouds with Random Sampling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 8338–8354. [CrossRef] [PubMed]
11. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6410–6419. [CrossRef]
12. Wu, W.; Qi, Z.; Fuxin, L. PointConv: Deep Convolutional Networks on 3D Point Clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9613–9622. [CrossRef]
13. Yan, X.; Zheng, C.; Li, Z.; Wang, S.; Cui, S. PointASNL: Robust Point Clouds Processing Using Nonlocal Neural Networks with Adaptive Sampling. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5588–5597. [CrossRef]
14. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.
15. Guo, M.; Cai, J.; Liu, Z.; Mu, T.; Martin, R. R.; Hu, S. PCT: Point Cloud Transformer. *Comp. Vis. Media* **2021**, *7*, 187–199. [CrossRef]
16. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.; Koltun, V. Point Transformer. In Proceedings of the 19th IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 16239–16248. [CrossRef]
17. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the 30th Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5105–5114.
18. Zhang, C.; Wan, H.; Shen, X.; Wu, Z. PVT: Point-Voxel Transformer for Point Cloud Learning. *arXiv* **2021**, arXiv:2108.06076. [CrossRef]
19. Park, C.; Jeong, Y.; Cho, M.; Park, J. Fast Point Transformer. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 16928–16937. [CrossRef]
20. Lai, X.; Liu, J.; Jiang, L.; Wang, L.; Zhao, H.; Liu, S.; Qi, X.; Jia, J. Stratified Transformer for 3D Point Cloud Segmentation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 8490–8499. [CrossRef]
21. Duan, L.; Zhao, S.; Xue, N.; Gong, M.; Xia, G.-S.; Tao, D. ConDaFormer: Disassembled Transformer with Local Structure Enhancement for 3D Point Cloud Understanding. In Proceedings of the 36th Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 10–16 December 2023.
22. Li, X.-L.; Guo, M.-H.; Mu, T.-J.; Martin, R.R.; Hu, S.-M. Long Range Pooling for 3D Large-Scale Scene Understanding. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 20–22 June 2023; pp. 10300–10311. [CrossRef]
23. Qiu, H.; Yu, B.; Tao, D. Collect-and-Distribute Transformer for 3D Point Cloud Analysis. *arXiv* **2023**, arXiv:2306.01257.
24. He, Y.; Yu, H.; Yang, Z.; Liu, X.; Sun, W.; Mian, A. Full Point Encoding for Local Feature Aggregation in 3-D Point Clouds. *IEEE Trans. Neural Netw. Learn. Syst.* 2024, *early access*. [CrossRef]
25. Li, H.; Zheng, T.; Chi, Z.; Yang, Z.; Wang, W.; Wu, B.; Lin, B.; Cai, D. APPT: Asymmetric Parallel Point Transformer for 3D Point Cloud Understanding. *arXiv* **2023**, arXiv:2303.17815.
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. [CrossRef]
27. Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Ma, Y.; Li, W.; Li, H.; Lin, D. Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 9934–9943. [CorssRef]
28. Yan, X.; Gao, J.; Li, J.; Zhang, R.; Li, Z.; Huang, R.; Cui, S. Sparse Single Sweep LiDAR Point Cloud Segmentation via Learning Contextual Shape Priors from Scene Completion. In Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI), Virtually, 2–9 February 2021; pp. 3101–3109. [CorssRef]
29. Chen, Y.; Liu, J.; Zhang, X.; Qi, X.; Jia, J. LargeKernel3D: Scaling up Kernels in 3D Sparse CNNs. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 20–22 June 2023; pp. 13488–13498. [CrossRef]
30. Xu, C.; Wu, B.; Wang, Z.; Zhan, W.; Vajda, P.; Keutzer, K.; Tomizuka, M. SqueezeSegV3: Spatially-Adaptive Convolution for Efficient Point-Cloud Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV); Virtual, 23–28 August 2020; pp. 1–19. [CrossRef]
31. Li, M.; Wang, G.; Zhu, M.; Li, C.; Liu, H.; Pan, X.; Long, Q. DFAMNet: Dual Fusion Attention Multi-Modal Network for Semantic Segmentation on LiDAR Point Clouds. *Appl. Intell.* **2024**, *54*, 3169–3180. [CrossRef]

32. Puy, G.; Boulch, A.; Marlet, R. Using a Waffle Iron for Automotive Point Cloud Semantic Segmentation. In Proceedings of the 20th IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 3356–3366. [CorssRef]
33. Zhang, Y.; Zhou, Z.; David, P.; Yue, X.; Xi, Z.; Gong, B.; Foroosh, H. PolarNet: An Improved Grid Representation for Online LiDAR Point Clouds Semantic Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 9598–9607. [CorssRef]
34. Lin, Y.; Yan, Z.; Huang, H.; Du, D.; Liu, L.; Cui, S.; Han, X. FPConv: Learning Local Flattening for Point Convolution. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4292–4301. [CrossRef]
35. Hu, W.; Zhao, H.; Jiang, L.; Jia, J.; Wong, T.-T. Bidirectional Projection Network for Cross Dimension Scene Understanding. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 14368–14377. [CrossRef]
36. Peng, B.; Wu, X.; Jiang, L.; Chen, Y.; Zhao, H.; Tian, Z.; Jia, J. OA-CNNs: Omni-Adaptive Sparse CNNs for 3D Semantic Segmentation. *arXiv* **2024**, arXiv:2403.14418.
37. Fan, Y.-C.; Liao, K.-Y.; Xiao, Y.-S.; Lu, M.-H.; Yan, W.-Z. 3D Point Cloud Semantic Segmentation System Based on Lightweight FPConv. *IEEE Access* **2023**, *11*, 31767–31777. [CrossRef]
38. Gong, J.; Xu, J.; Tan, X.; Song, H.; Qu, Y.; Xie, Y.; Ma, L. Omni-Supervised Point Cloud Segmentation via Gradual Receptive Field Component Reasoning. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 11668–11677. [CrossRef]
39. Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.; Elhoseiny, M.; Ghanem, B. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. In Proceedings of the 35th Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022; pp. 23192–23204.
40. Kang, X.; Wang, C.; Chen, X. Region-Enhanced Feature Learning for Scene Semantic Segmentation. *IEEE Trans. Multimed.* 2023, *early access*. [CrossRef]
41. Wei, M.; Wei, Z.; Zhou, H.; Hu, F.; Si, H.; Chen, Z.; Zhu, Z.; Qiu, J.; Yan, X.; Guo, Y.; et al. AGConv: Adaptive Graph Convolution on 3D Point Clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 9374–9392. [CrossRef] [PubMed]
42. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Polosukhin, I. Attention is all you need. In Proceedings of the 30th Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017.
43. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *ACM Comput. Surv.* **2022**, *54*, 1–41. [CrossRef]
44. Han, Q.; Fan, Z.; Dai, Q.; Sun, L.; Cheng, M.-M.; Liu, J.; Wang, J. On the Connection between Local Attention and Dynamic Depth-Wise Convolution. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
45. Chen, Q.; Wu, Q.; Wang, J.; Hu, Q.; Hu, T.; Ding, E.; Cheng, J.; Wang, J. MixFormer: Mixing Features across Windows and Dimensions. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5239–5249. [CrossRef]
46. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the 19th IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022. [CrossRef]
47. Gu, J.; Kwon, H.; Wang, D.; Ye, W.; Li, M.; Chen, Y.-H.; Lai, L.; Chandra, V.; Pan, D.Z. Multi-Scale High-Resolution Vision Transformer for Semantic Segmentation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 12084–12093. [CrossRef]
48. Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. In Proceedings of the 19th IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 548–558. [CrossRef]
49. Tu, Z.; Talebi, H.; Zhang, H.; Yang, F.; Milanfar, P.; Bovik, A.; Li, Y. MaxViT: Multi-Axis Vision Transformer. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 459–479. [CrossRef]
50. Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; Guo, B. CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June; pp. 12124–12134. [CrossRef]
51. Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; Shen, C. Twins: Revisiting the Design of Spatial Attention in Vision Transformers. In Proceedings of the 34th Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–14 December 2021; pp. 9355–9366.
52. Li, W.; Wang, X.; Xia, X.; Wu, J.; Li, J.; Xiao, X.; Zheng, M.; Wen, S. SepViT: Separable Vision Transformer. *arXiv* **2022**, arXiv:2203.15380.
53. Fan, Q.; Huang, H.; Zhou, X.; He, R. Lightweight Vision Transformer with Bidirectional Interaction. In Proceedings of the 37th Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 9–15 December 2024.
54. Lahoud, J.; Cao, J.; Khan, F.S.; Cholakkal, H.; Anwer, R.M.; Khan, S.; Yang, M.-H. 3D Vision with Transformers: A Survey. *arXiv* **2024**, arXiv:2208.04309.
55. Lu, D.; Xie, Q.; Wei, M.; Xu, L.; Li, J. Transformers in 3D Point Clouds: A Survey. *arXiv* **2022**, arXiv:2205.07417.

56. Wu, X.; Lao, Y.; Jiang, L.; Liu, X.; Zhao, H. Point Transformer V2: Grouped Vector Attention and Partition-Based Pooling. In Proceedings of the 35th Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022; pp. 33330–33342.
57. Liu, Z.; Yang, X.; Tang, H.; Yang, S.; Han, S. FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 20–22 June 2023; pp. 1200–1211. [CrossRef]
58. Fan, L.; Pang, Z.; Zhang, T.; Wang, Y.-X.; Zhao, H.; Wang, F.; Wang, N.; Zhang, Z. Embracing Single Stride 3D Object Detector with Sparse Transformer. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 8458–8468. [CrossRef]
59. Xiang, P.; Wen, X.; Liu, Y.-S.; Zhang, H.; Fang, Y.; Han, Z. Retro-FPN: Retrospective Feature Pyramid Network for Point Cloud Semantic Segmentation. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 20–22 June 2023; pp. 17780–17792. [CrossRef]
60. Yang, Y.; Guo, Y.; Xiong, J.; Liu, Y.; Pan, H.; Wang, P.; Tong, X.; Guo, B. Swin3D: A Pretrained Transformer Backbone for 3D Indoor Scene Understanding. *arXiv* **2023**, arXiv:2304.06906.
61. Hui, L.; Yang, H.; Cheng, M.; Xie, J.; Yang, J. Pyramid Point Cloud Transformer for Large-Scale Place Recognition. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 6078–6087. [CrossRef]
62. Yang, J.; Zhang, Q.; Ni, B.; Li, L.; Liu, J.; Zhou, M.; Tian, Q. Modeling Point Clouds with Self-Attention and Gumbel Subset Sampling. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3318–3327. [CrossRef]
63. Ai, D.; Xu, C.; Zhang, X.; Ai, Y.; Bai, Y.; Liu, Y. ASSA-Net: Semantic Segmentation Network for Point Clouds Based on Adaptive Sampling and Self-Attention. In Proceedings of the 2023 5th International Conference on Natural Language Processing (ICNLP), Guangzhou, China, 24–26 March 2023; pp. 60–64. [CrossRef]
64. Zhang, C.; Wan, H.; Shen, X.; Wu, Z. PatchFormer: An Efficient Point Transformer with Patch Attention. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 11789–11798. [CrossRef]
65. Yang, X.; Jin, M.; He, W.; Chen, Q. PointCAT: Cross-Attention Transformer for Point Cloud. *arXiv* **2023**, arXiv:2304.03012.
66. Huang, Z.; Zhao, Z.; Li, B.; Han, J. LCPFormer: Towards Effective 3D Point Cloud Analysis via Local Context Propagation in Transformers. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 4985–4996. [CrossRef]
67. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D Semantic Parsing of Large-Scale Indoor Spaces. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1534–1543. [CrossRef]
68. Rozenberszki, D.; Litany, O.; Dai, A. Language-Grounded Indoor 3D Semantic Segmentation in the Wild. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 125–141. [CrossRef]
69. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In Proceedings of the 17th IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9296–9306. [CrossRef]
70. Tang, L.; Zhan, Y.; Chen, Z.; Yu, B.; Tao, D. Contrastive Boundary Learning for Point Cloud Segmentation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 8479–8489. [CrossRef]
71. Wu, X.; Jiang, L.; Wang, P.-S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; Zhao, H. Point Transformer V3: Simpler, Faster, Stronger. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 4840–4851. [CrossRef]
72. Tao, A.; Duan, Y.; Wei, Y.; Lu, J.; Zhou, J. SegGroup: Seg-Level Supervision for 3D Instance and Semantic Segmentation. *IEEE Trans. Image Process.* **2022**, *31*, 4952–4965. [CrossRef] [PubMed]
73. Kong, L.; Liu, Y.; Chen, R.; Ma, Y.; Zhu, X.; Li, Y.; Hou, Y.; Qiao, Y.; Liu, Z. Rethinking Range View Representation for LiDAR Segmentation. In Proceedings of the 120th IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 228–240. [CorssRef]

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.