Point HorNet: Higher-Order Spatial Interaction Network for Point Clouds

Hao Yuan
College of Information Engineering, Northwest A&F University
Shaanxi, China.
1980450353@ qq.com

Linqing Liu
College of Information Engineering, Northwest A&F University
Shaanxi, China.
13474008839@nwafu.edu.cn

Tingting Yan
College of Information Engineering, Northwest A&F University
Shaanxi, China.
15339112376@163.com

Wenjing Zhang
College of Information Engineering, Northwest A&F University
Shaanxi, China.
3261208385@qq.com

Qinghe Liu
College of Information Engineering, Northwest A&F University
Shaanxi, China.
liuqinghe@nwafu.edu.cn

Juanjie Wei
College of Information Engineering, Northwest A&F University
Shaanxi, China.
weijj@nwafu.edu.cn

Ziang Wu
School of Software，Northwestern Polytechnical University
Shaanxi, China.
ziangwu@mail.nwpu.edu.cn

Huijun Yang*
College of Information Engineering, Northwest A&F University
Shaanxi, China.
yhj740225@163.com

***Abstract*—The progression of 3D scanning technology has amplified the importance of point cloud data in computer vision, robotics navigation, and virtual reality. Point clouds, consisting of discrete points in space, harbor rich geometric and structural information. The rise of deep learning has ushered in innovative methods for point cloud processing, improving the efficiency and precision of tasks such as feature extraction, classification, segmentation, and reconstruction. The introduction of HorNet and Recursive Gated Convolution has facilitated spatial interactions of any order, addressing issues of feature loss inherent in low-order interactions. This has resulted in significant advancements in image analysis tasks, including image classification, object detection, and semantic segmentation. Drawing inspiration from this success, we have investigated the application of HorNet and Recursive Gated Convolution to point clouds, constructing a network specifically designed for semantic segmentation tasks. Point HorNet has achieved commendable results in semantic segmentation, attaining a mIoU of 68.1% in Area 5 and 73.6% in a six-fold cross-validation on the S3DIS dataset.**

Keywords—point cloud, high-order spatial interaction, deep learning, indoor scenes

I. INTRODUCTION

With the rapid advancement of 3D data acquisition technology, point cloud data has become increasingly important in multiple domains. Applications such as autonomous driving [1], artificial intelligence [2], virtual reality [3], and 3D reconstruction heavily rely on accurate and efficient point cloud processing techniques to gain in-depth and practical spatial cognition capabilities. Point cloud data, due to its rich geometric and topological information, has become an indispensable part of these technological developments.

Despite this, processing point cloud data still faces numerous challenges. Firstly, point cloud data is often large-scale, unordered, and noisy, which poses significant technical difficulties for effective data processing. Secondly, traditional point cloud processing methods often fail to fully utilize the high-order spatial features within point cloud data, limiting the depth and efficiency of data analysis. For example, the point by point MLP method limits the ability to establish complex high-order interactions between points; The point convolution method may increase computational complexity and time due to the inherent characteristics of point clouds; If too many local fusion features are used in RNN methods, the rich geometric features of the original point cloud will be lost, and problems such as vanishing gradients and exploding are also prone to occur. Meanwhile, existing models for processing point cloud data rarely propose the method of high-order point cloud feature space interaction.

Given the above context, this study is dedicated to exploring and developing high-order point cloud feature interaction technology, aiming to enhance the effectiveness of information extraction and the breadth of applications by establishing more complex models of relationships between points. Therefore, we conducted a thorough study of existing point cloud processing methods and proposed a network based on high-order spatial feature interactions in point clouds, Point HorNet, to address some of the issues present in current technology. Our main contributions are:

We extended recursive gated convolution (g^nConv) [4] to three-dimensional space, enabling high-order spatial feature interactions and addressing feature loss due to low-order spatial interactions.

On this basis, we proposed a deep learning-based feature extraction network, PointHorNet, allowing the model to more effectively process 3D point cloud data and enhance its ability to capture spatial information.

II. Related work

The processing and application of point cloud data has long been a hot topic in the fields of computer vision and machine learning. In recent years, point cloud semantic segmentation networks have rapidly evolved in areas such as autonomous driving, artificial intelligence, virtual reality, and 3D reconstruction. The primary objective of point cloud semantic segmentation is to divide a set of point clouds into several subsets with distinct semantic meanings and unique properties [5]. Current methods for point cloud semantic segmentation can be mainly categorized into two types based on the processing approach of point cloud data: indirect semantic segmentation and direct semantic segmentation. Indirect semantic segmentation includes two main methods: multi-view-based [6][7][8][9][10] and voxel-based approaches [11][12]; direct semantic segmentation comprises six methods: neighborhood feature learning, optimized CNN, graph convolution, attention mechanism, instance segmentation combination, and RNN-based methods [13].

Indirect semantic segmentation methods transform raw point cloud data using multi-view or voxelization techniques to extract feature information indirectly from the point cloud data. The feature information is then projected back onto the original point cloud to achieve semantic segmentation. Direct semantic segmentation methods extract feature information directly from the point cloud data, retaining inherent information in the original point cloud due to the absence of data transformation. Consequently, direct semantic segmentation methods have become the primary research direction in recent years due to their simplicity, effectiveness, and adaptability to large data volumes.

1) ***Neighborhood Feature Learning-based Methods****:* To capture local features, numerous models rely on neighborhood feature learning to obtain contextual information from point clouds, combining global and local features effectively to enhance segmentation performance. PointNet++ [14] employs a hierarchical structure, with each layer consisting of sampling, grouping, and feature extraction (PointNet [15]). The sampling layer selects the center points of local regions, the grouping layer constructs subsets of point clouds around these center points, and the feature extraction layer obtains the feature representation of these subsets. RandLA-Net [16] adopts random point sampling instead of the farthest point sampling method used in PointNet++ [14], capturing and preserving local geometric features through a local feature aggregation module. This method achieves significant improvements in storage and computation but is highly dependent on accurate local structure information and sensitive to the density and uniformity of point cloud data. If the data contains irregular or missing parts, the model's performance will significantly degrade, and the computational cost is high, especially when dealing with large-scale point clouds, as it requires calculating the relationship between each point and its neighboring points. This type of method usually focuses on learning local neighborhood feature representations, while ignoring high-order relationships between objects. They often use fixed size neighborhoods or simple statistical methods to extract features, and it is difficult to capture the interactions between objects in complex scenes, so their performance may be poor when dealing with multi-scale objects and complex scenes.

2) ***Graph Convolution-based Methods****:* Although PointNet++ effectively addresses the issue of local feature extraction compared to PointNet, it still extracts point features in isolation, ignoring the relationships between points. To overcome the limitations of PointNet++'s isolated point feature learning, Wang et al. proposed the Dynamic Graph Convolutional Neural Network (DGCNN [17]), constructing a local neighborhood graph and edge convolution operation to extract the features of the center point and the edge vectors between the center point and its K nearest neighbors to capture local features of the point cloud. To address data structure and computational issues, Xu et al. proposed Grid-GCN [18], a fast query method that introduces a graph convolution module with coverage-aware network queries and grid context aggregation for efficient data structure and computation. However, this method is highly dependent on the graph structure and has a high computational complexity for large-scale point clouds, requiring significant computational resources. It uses the connection relationships of nodes in the graph structure to learn the feature representation of nodes. Although it can model the relationships between objects through graph structures, it usually does not involve higher-order interactions. Therefore, for data with complex structures, it may not be possible to fully model high-order relationships between objects.

3) ***Optimized CNN-based Methods****:* Convolutional Neural Networks (CNNs) consist of one or more convolutional layers and fully connected layers, along with associated weights and pooling layers, enabling CNNs to extract high-level features from 3D point cloud data. However, the unordered nature of point cloud data makes it difficult to apply convolution operations directly, prompting researchers to optimize CNN models to address this issue and leverage their advantages. Thomas et al. [19] proposed kernel point convolutions with deformable convolution operators, applying weights from the nearest kernel points within each local neighborhood for convolution. Wu et al. [20] introduced PointConv, a continuous convolution operation method, training a multilayer perceptron on local point coordinates to approximate the continuous weight function and density function in the convolution filter, providing permutation and translation invariance. Additionally, PointConv was extended to a deconvolution operation (PointDeconv) to propagate features from subsampled point clouds back to the original resolution. Although CNNs perform well in image processing, they are not directly applicable to irregular point cloud data and require preprocessing such as voxelization or projection to regular structures, which may result in information loss and may not fully capture the local complexity in 3D space.

*4) **RNN-Based Methods**:* Recurrent neural networks (RNNs) not only learn from current input but also from sequences of previous data, enhancing the utilization of contextual information. Based on this, Engelmann et al. [21] expanded upon the PointNet framework, introducing input-level and output-level contextual information. Input-level context involves converting point clouds into multiscale and networked blocks, while output-level context involves merging extracted input-level features through a Concatenation Unit (CU) and a Recurrent Concatenation Unit (RCU), ultimately providing features with rich context at the output level. Ye et al. [22] scanned the 3D space sequentially along the x and y directions to extract information and constructed a per-point pyramid pooling module to capture local features of varying point cloud densities. They also utilized a hierarchical bidirectional RNN to learn spatial context information, allowing for the fusion of multilevel semantic features, however, the ability to model space is limited and the computational efficiency is low.

*5) **Attention Mechanism-Based Methods**:* The primary role of attention mechanisms is to enable the system to selectively ignore less important and irrelevant information, focusing instead on important details. Attention mechanisms compute gradients for neural networks and learn the weights of attention through forward and backward propagation. To improve semantic segmentation accuracy, researchers have integrated attention mechanisms into segmentation algorithms. Yang et al. [23] developed a Point Attention Transformer (PAT) for point cloud reasoning, proposing a Group Shuffle Attention (GSA) to model relationships between points. Tsinghua scholars introduced the concept of Transformers into point cloud processing, proposing the efficient and accurate PCT network [24]. This network encodes input point cloud features into a higher-dimensional feature space and then connects local geometric information that has been processed through four attention layers to capture semantic similarity at different scales, finally aggregating local and global features to accomplish classification and segmentation tasks. Although attention mechanisms have advantages in capturing key features in point clouds, they typically involve less high-order feature space interaction.

*6) **Instance Segmentation-Based Methods**:* Combining semantic and instance segmentation not only reduces repetitive operations and complexity of calculations but also increases segmentation accuracy. Building on this, Wang et al. [25] proposed an Association and Segmentation framework (ASIS), learning semantic-aware point-level instance embeddings that benefit instance segmentation from semantic segmentation, and concurrently, merging the semantic features of points within the same instance for more accurate semantic segmentation. Pham et al. [26] developed a multitask point network based on PointNet, performing two tasks simultaneously: predicting the semantic information of 3D points and embedding these points in high-dimensional vectors to create similar embeddings for points from the same object instance. Then, using a multi-value conditional random field model, they combined semantic and instance labels, formulating the segmentation problem as a joint optimization of labels within the field model. These methods require extensive labeled data for training, with the labeling of point cloud data being costly and time-consuming. The task of instance segmentation itself is challenging, especially in complex scenes and occluded conditions, with the accuracy and robustness of model segmentation still needing improvement. The instance segmentation method mainly focuses on separating different instances in the point cloud, rather than explicitly modeling high-order feature space interactions.

III. METHOD

A. Transformer

Transformer, a framework built upon the self-attention mechanism, is designed for processing sequence data. The essence of the self-attention [27] mechanism lies in calculating the relationships between each element of a sequence and all other elements within that sequence, thereby capturing the global dependencies within the sequence. Additionally, to preserve positional information in the sequence, Transformer incorporates positional encoding into the input sequence.

*1) **Self-attention**:* Add The self-attention mechanism empowers the model to calculate attention scores at each position within a sequence with respect to various positions. This process is executed through the following three steps:

a) Computation of Q, K, and V: For a given input, the first step involves computing the query vector (Q), key vector (K), and value vector (V).These vectors are derived by multiplying the input vector with three separate weight matrices.

b) Calculation of Attention Scores: The dot product between the query and all keys is computed, followed by scaling (dividing by the square root of d_k, where d_k is the dimension of the key vector), and applying the softmax function to obtain the attention weights. This procedure can be represented by the following formulma:

$$Attention(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

c) Output Computation: Finally, the obtained attention weights are multiplied with the value vectors to yield the final output, which serves as the input for the next layer or as the ultimate output.

*2) **Position Encoding**:* Since Transformers process sequences in parallel rather than sequentially like RNNs, they require a method to account for the order of elements in a sequence. Transformers achieve this by adding position encodings to the input embeddings, which share the same dimensionality, allowing for their summation. Position encodings are represented by a combination of sine and cosine functions.

B. Point Transformer

Due to the unordered nature, irregularity, and varying point density of point cloud data, specialized handling challenges are posed to deep learning models. To effectively address these challenges, the Point Transformer [28] is specifically designed

with several key components: Transition Up, Transition Down, and the Transformer Layer.

Transition Up and Transition Down correspond to addressing the multi-scale representation issues of point cloud data. Transition Up aims to increase the resolution of the point cloud by increasing the number of points, achieved through interpolation of point positions and appropriate propagation of features, ensuring that while more points are added, features of new points are reasonably allocated and inferred to maintain data coherence. Conversely, Transition Down reduces data complexity by decreasing the number of points, selectively retaining representative points to reduce processing demands, and aggregating features of nearby points to retain as much information as possible. This design of Transition Up and Transition Down directly confronts the challenges of irregularity and changing point density in point cloud data, allowing the model to work effectively at different resolutions. The computation process can be represented by the following formula:

$$\boldsymbol{y_i} = \sum_{\boldsymbol{x_j} \in X(i)} \omega\left(\sigma\left(\gamma(\boldsymbol{x_i}) - \rho(\boldsymbol{x_j}) + \theta\right)\right) \odot \left(\mu(\boldsymbol{x_j}) + \theta\right) \quad (2)$$

The Transfromer Layer employs a local self-attention mechanism to effectively utilize local neighborhood information, applying self-attention locally within the neighborhood around each data point. Let $X = \{\boldsymbol{x}\}_{\boldsymbol{i}}$ be a set of feature vectors, then: the subset $X(i) \subseteq X$ is the set of points in the local neighborhood of $\boldsymbol{x_i}$ (specifically, the k-nearest neighbors), $\boldsymbol{y_i}$ is the output feature, ω is a normalization function, such as softmax, σ is a mapping function (e.g., MLP) to generate attention vectors for feature aggregation (an MLP with two linear layers and one ReLU nonlinearity). γ, ρ, and μ are pointwise feature transformations, such as linear projections or MLPs, the relational function employs subtraction, and θ is a positional encoding function. The point transformer layer is illustrated in Fig. 1.

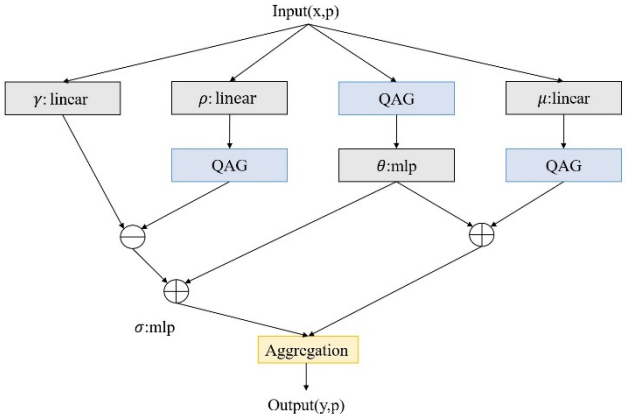

Fig. 1. Point transformer layer.

C. Hornet

Traditional standard convolutions do not account for spatial interactions, leading to issues such as limited receptive fields, context loss, and feature dropout. The success of self-attention and other dynamic networks demonstrates that complex and high-order interactions often exist between spatial positions in nonlinear, deep models. The introduction of explicit and high-order spatial interactions in the model is beneficial for enhancing the modeling capabilities of visual models. To address the feature dropout and other issues caused by low-order spatial interactions, Yongming Rao et al. [4] proposed the HorNet framework, which utilizes g^nConv to recursively perform high-order spatial interactions, extending spatial interactions to arbitrary orders and achieving the goal of input-adaptive spatial mixing. Below, we will demonstrate the implementation process of g^nConv.

1) ***gconv****:* Before introducing gnconv, we first introduce gconv, which is the core component of gnconv. Let $x \in R^{N*C}$ be the input features, the output y of gconv can be represented as:

$$f_{in}(x) = [a_0^{N*C}, b_0^{N*C}] \in R^{N*2C}$$
$$a_1 = f(b_0) \odot a_0, \quad y = f_{out}(a_1) \quad (3)$$

Where f_{in}, f_{out} are linear projection layers that perform channel mixing, f represents a depthwise convolution and element-wise multiplication is used to introduce mutual feature information between a_0 and b_0. This constitutes one interaction process. Inspired by the large receptive field in Vision Transformers, which facilitates the capture of long-range dependencies, the depthwise convolution f in gconv adopts two implementations: 7×7 Convolution or Global Filter(GF).

2) $\boldsymbol{g^n Conv}$ *:* Multiple applications of the first-order interaction gconv are employed to introduce higher-order interactions, further extracting high-dimensional features and enhancing model capacity and feature representation.

Initially, gconv is used with f_{in} to obtain a set of projected features a_0 and $b_{0\ldots n-1}$:

$$f_{in}(x) = \left[a_0^{N*C_0}, b_0^{N*C_0}, \ldots, b_{n-1}^{N*C_{n-1}}\right] \in R^{N*2C} \quad (4)$$

Subsequently, gated convolution is executed recursively:

$$a_{i+1} = f_i(b_i) \odot g_i(a_i)/\alpha \quad i = 0,1,\ldots,n-1 \quad (5)$$

Finally, the output of the last recursive step is passed to the projection layer f_{out}, yielding the result of $g^n Conv$.

$$y = f_{out}(a_n) \quad (6)$$

where we scale the output by 1/α to stabilize the training. $\{f_i\}$ are a set of depth-wise convolution layers and $\{g_i\}$ are a linear layer used to match different order dimensions. When $i = 0$, no operation is performed; when $1 \leq i < n-1$, $g_i = Linear(C_i, C_{i+1})$.

From the recursive formula above, it can be seen that the interaction order of a_i increases by 1 after each step. Multi-order feature interactions are achieved through multiple element-wise multiplications to introduce mutual feature information. Therefore, $g^n Conv$ realizes n-order spatial interactions.

3) ***Formula proof****:* According to Equation (3.8) in Hor-Net [4], the spatial mixing result for the cth channel at position i in g^nConv (before f_{out}) is given by:

$$x_{g^n\,\mathrm{Conv}}^{(i,c)} = p_n^{(i,c)} = \sum_{j \in \Omega_i} \sum_{c'=1}^{C} \underline{w_{n-1,\,i \to j}^{c}\, \mathbf{g_{n-1}^{(i,c)}}}\, w_{f_{\mathrm{in}}}^{(c',c)}\, x^{(j,c')} \equiv \sum_{j \in \Omega_i} \sum_{c'=1}^{C} \underline{h_{ij}^{c}}\, w_{f_{\mathrm{in}}}^{(c',c)}\, x^{(j,c')} \quad (7)$$

Where Ω_i is the local window centered at i , w_{n-1} represents the convolution weights of f_{n-1}, w_{fin} denotes the linear weights of f_{in} , and $\mathbf{g_{n-1}} = g_{n-1}(a_{n-1})$ is the projection of a_{n-1} . From this formula, we can observe that g^nConv implements inputadaptive spatial mixing, with weights $\{h_{ij}^c\}$. Since h_{ij} is computed from a_{n-1}, which include n-1 order interactions, g^nConv incorporates higher-order influences of spatial mixing weights. Consequently, g^nConv is capable of better modeling more complex spatial interactions.

D. Point HorNet

Point cloud data often encompasses a wealth of information, including position, color, depth, and more. In such scenarios, low-order spatial interactions, due to their simplicity in mathematical representation, struggle to accurately capture these complex features, leading to information loss during the feature extraction process [15][33].

Consequently, we endeavor to extend the g^nConv within HorNet and apply it to point clouds, leveraging its high-order spatial interactions to mitigate the issue of feature loss. And with this, we introduce our proposed network framework, the Point HorNet.

The proposed point-based network framework(Fig. 2) takes as input data formatted with xyz coordinates (positional information) and RGB values (feature information). Employing an encoder-decoder architecture [45], the encoder's stages progressively downsample the point sets with rates [1, 4, 4, 4, 4], followed by feature extraction through one or multiple HorBlock3D layers(Fig. 3). Conversely, the decoder's stages consist of an upsampling layer paired with a HorBlock3D layer, enabling the decoding of features through upscaling.

1) ***Downingsampling Layer****:* Downsampling refers to the process of reducing the number of data points or decreasing the dimensionality to cut down computational requirements, enhance processing speed, and prevent overfitting [29]. For a downsampling rate of 1, a linear layer is directly employed for mapping. When the rate exceeds 1, the farthest point sampling [14] algorithm is utilized to select point sets, followed by regrouping the sampled points around each new point using the query and group method (employing k-NN [14] with k values of [16,32,32,32,32]). The features of these points are extracted and processed through a linear layer, normalization function, and ReLU activation function. Finally, a max pooling layer pools the features to obtain the downsampled point features.

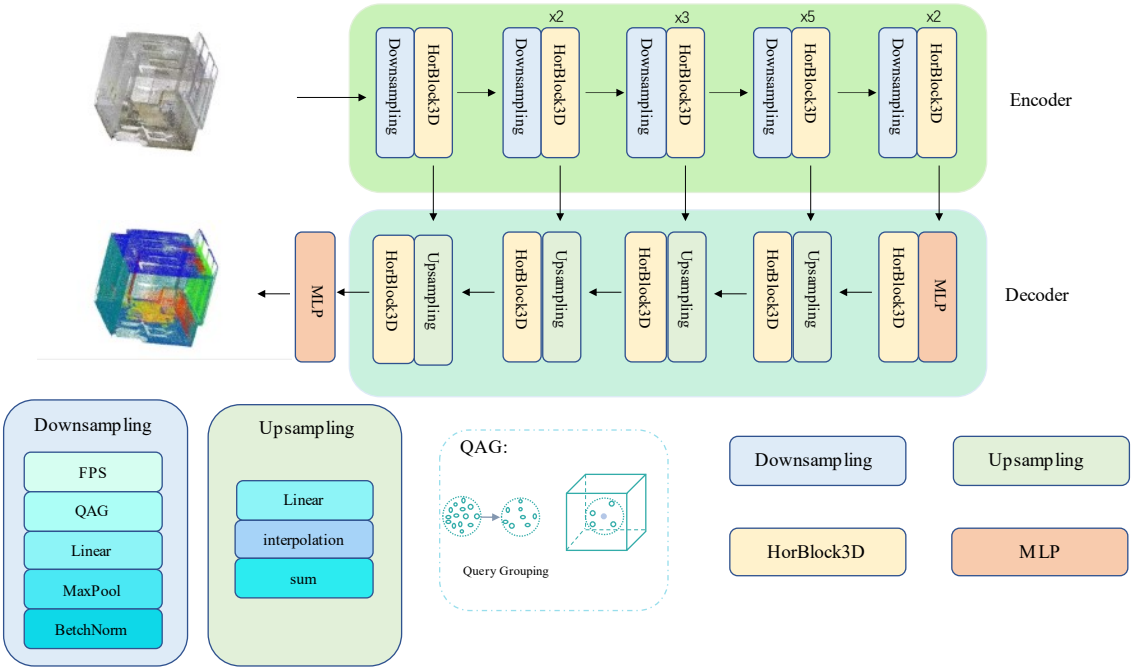

Fig. 2. Point HorNet network structure.

2) ***Usampling Layer****:* Upsampling is used to increase the dimensionality of data. Our network's upsampling includes a linear transformation and an optional interpolation operation, which not only preserves the original information but also introduces additional dimensions, aiding in the performance of more complex tasks within the decoder module.

For each input point feature, initial processing is done through a linear layer, followed by batch normalization [30] and ReLU activation. The processed features are then mapped to a higher resolution point set using trilinear interpolation [11], a technique that considers the eight closest neighboring points in space to estimate the values at new positions. The interpolated features are combined with features from the corresponding encoding stage. These encoder features are provided to the

decoder through skip connections [31], maintaining and utilizing the detail information captured during encoding for the decoding process.

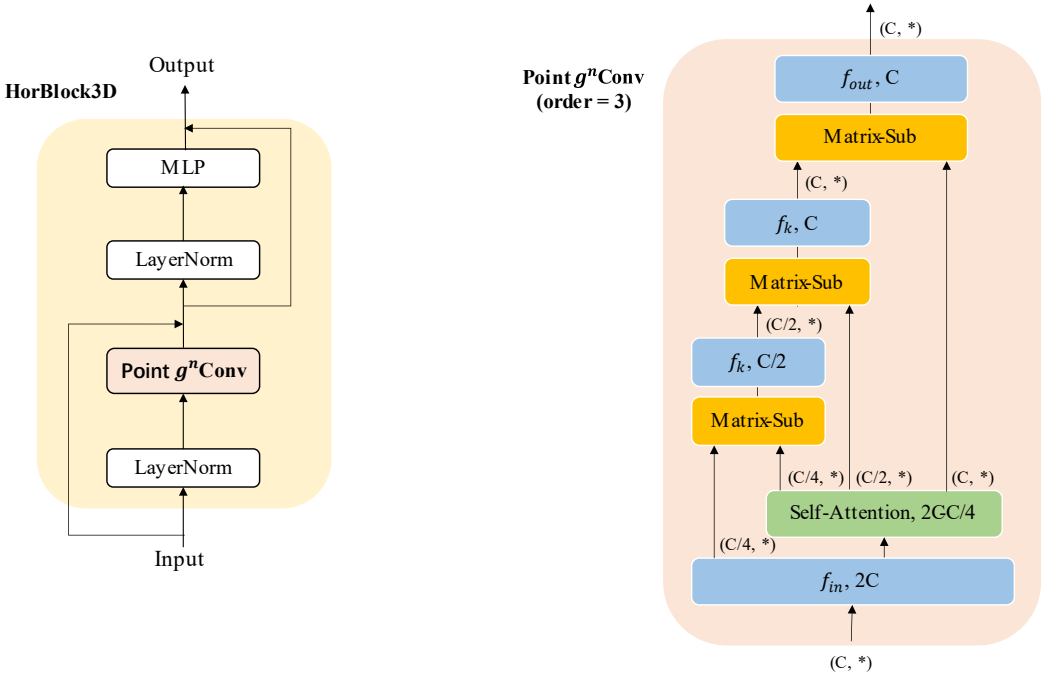

Fig. 3. Detailed structure design for HorBlock3D and Point g^nConv.

*3) **Depthwise convolutions**:* The two types of depthwise convolutions used in HorNet are not suitable for point cloud data due to the unique characteristics of point clouds.

The 7×7 convolution, being a local operation, is constrained by its fixed receptive field size, which limits its flexibility in extracting features at different scales. In point cloud data, variations in object scale may prevent certain features from being effectively captured within a 7×7 window, impacting the model's generalization and accuracy. Additionally, the 7×7 convolution lacks rotational invariance, meaning that after a rotation of the point cloud data, the convolution kernel cannot maintain its original feature extraction capabilities, leading to degraded performance.

Global filters (GFs) [32], while providing a global perspective, also face issues with scale invariance and rotational invariance. The design of global filters is typically based on fixed global parameters, making it difficult for them to adaptively adjust their filtering behavior when dealing with point cloud data of varying scales. Similarly, the filtering effect of global filters is also compromised when faced with rotation operations, as they lack an inherent mechanism to handle rotational invariance.

Therefore, we opt to implement the depth function using a self-attention mechanism [28]:

Feature Scale Invariance [15]: The self-attention mechanism can adaptively adjust the weights between different points, meaning it can capture features at various scales. In point cloud data, the scale of objects can vary significantly, and the self-attention mechanism can effectively handle this scale variation without relying on a fixed convolution kernel size.

Rotational Invariance [15]: When computing attention weights, the self-attention mechanism focuses on the relative positions and features of points rather than absolute positions. This allows the model to maintain its feature extraction capabilities when faced with point cloud data rotation, as the relative positional relationships remain unchanged after rotation.

Flexible Receptive Field: Compared to the 7×7 convolution or global filters (GFs [32]), the self-attention mechanism offers a more flexible receptive field. It is not confined to a fixed window size but can dynamically adjust the receptive field based on the characteristics of the input data, better capturing long-range dependencies.

Global Information Interaction: The self-attention mechanism allows each point to interact with all other points, enabling the model to globally understand the spatial structure of point cloud data. This global information interaction is crucial for comprehending complex point cloud scenes.

*4) **Incorporating Adjacent Features**:* In PointHorNet, subtraction is used as the operator to incorporate adjacent features, as opposed to multiplication in HorNet [4]:

Direct Expression of Feature Differences: Subtraction directly reflects the differences between adjacent features, which is particularly important in point cloud data as it helps the model capture local shape variations such as edges and corners. Multiplication, on the other hand, emphasizes interactions between features, which may blur the boundaries between them in some cases.

Computational Efficiency: Subtraction is generally simpler and more efficient computationally than multiplication. When dealing with large-scale point cloud data, computational efficiency is a factor that cannot be overlooked. Subtraction can be executed more quickly, thus reducing the consumption of computing resources, which is especially important for real-time processing and large datasets.

Avoid amplification of eigenvalues: Multiplication may lead to amplification of eigenvalues, especially when there are significant numerical differences in point cloud data. This kind of amplification may lead to a decrease in the sensitivity of the model to small features, while subtraction can maintain the relative stability of feature values and help maintain the model's generalization ability.

Maintain linear relationships of features: The subtraction operation can maintain the linear relationship between features, which is very important for models that need to maintain the original linear structure of features. In point cloud processing, maintaining the linear relationship of features helps the model better understand the geometric structure in three-dimensional space.

Numerical Stability: In numerical computations, multiplication may lead to a rapid increase in numerical range, which can cause instability and overflow issues. In contrast, the numerical range for subtraction is relatively smaller, contributing to stability in computations.

Feature Scale Invariance [15]: Subtraction operation has a degree of scale invariance since it is based on the relative differences between features. This means that even if the scale of the point cloud data changes, the outcome of the subtraction operation remains consistent, which is beneficial for processing point clouds of different scales.

Rotational Invariance [15]: Under certain conditions, subtraction may more easily achieve rotational invariance. For example, in a local coordinate system, the position differences between adjacent points remain unchanged after rotation, whereas multiplication could be affected by rotation.

IV. Experiments

We evaluated our proposed Point HorNet network design through experiments. For 3D semantic segmentation, we tested using the Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset [34] and the ScannetV2 dataset. [35]

A. Implementation details

We implemented Point HorNet with PyTorch. We used the AdamW optimizer, setting the momentum and weight decay at 0.9 and 0.01 respectively. Before training, the point cloud data was subject to data augmentation such as random rotations, scaling, jittering, random dropout, or color alteration to enhance the model's generalization capability and robustness. The initial learning rate and epochs were set at 0.003 and 100 respectively.

B. S3DIS

1) ***Data and metric****:* The S3DIS dataset [34] contains over 200 million points. These point cloud data involve about 271 indoor spaces, covering more than 6000 square meters in total. Each point in the dataset contains positional information (x, y, z) and color information (r, g, b), and is assigned a specific semantic label. These labels fall into 13 categories, including walls, floors, ceilings, windows, doors, tables, chairs, etc. During training and testing, the point cloud data from the S3DIS dataset were divided into 272 separate .npy files, the smallest of which contains 85,855 points and the largest contains 9,273,742 points. Following standard protocols, we evaluated our proposed method in two modes: (a) the Area 5 training-test results, and (b) 6-fold cross-validation. For evaluation metrics, we used mean Intersection over Union (mIoU), mean class accuracy (mAcc), and overall point-wise accuracy (OA).

2) ***Results****:* The results are shown in Tables I. and II. Point HorNet performed excellently under both evaluation modes. On Area 5, Point HorNet's mIoU/mAcc/OA reached 68.1%/74.2%/90.2%, surpassing neighborhood feature learning-based frameworks (such as PointNet), voxel-based architectures (like SegCloud), graph-based methods (like SPGraph), attention-based methods (like PAT), and sparse convolutional networks (like MinkowskiNet). In 6-fold cross-validation, the model performed better than in Area 5, improving by 0.2 absolute percentage points over the Point Transformer. It can be seen that in S3DIS area 5, the performance of the model on the column and board is not very outstanding. It is because the local features of these two regions are not obvious, and the PointHornet network relies on the interaction of higher-order feature spaces. If the local features of these regions are not significant enough, the network may find it difficult to learn effective feature representations.

TABLE I. Semantic segmentation results on the S3DIS dataset, evaluated on Area 5.

Method	OA	mAcc	mIoU	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet [15]	-	49.0	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	59.0	52.6	5.9	40.3	26.4	33.2
SegCloud [36]	-	57.4	48.9	90.1	96.1	69.9	0.0	18.4	38.4	23.1	70.4	75.9	40.9	58.4	13.0	41.6
TangentConv [37]	-	62.2	52.6	90.5	97.7	74.0	0.0	20.7	39.0	31.3	77.5	69.4	57.3	38.5	48.8	39.8
PointCNN [38]	85.9	63.9	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
SPGraph [47]	86.4	66.5	58.0	89.4	96.9	78.1	0.0	42.8	48.9	61.6	84.7	75.4	69.8	52.6	2.1	52.2
PCCN [40]	-	67.0	58.3	92.3	96.2	75.9	0.3	6.0	69.5	63.5	66.9	65.6	47.3	68.9	59.1	46.2
PAT [23]	-	70.8	60.1	93.0	98.5	72.3	1.0	41.5	85.1	38.2	57.7	83.6	48.1	67.0	61.3	33.6
PointWeb [41]	87.0	66.6	60.3	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
HPEIN [42]	87.2	68.3	61.9	91.5	98.2	81.4	0.0	23.3	65.3	40.0	75.5	87.7	58.5	67.8	65.5	49.4
MinkowskiNet [43]	-	71.7	65.4	91.8	98.7	86.2	0.0	34.1	48.9	62.4	81.6	89.8	47.2	74.9	74.4	58.6
KPConv [43]	-	72.8	67.1	92.8	97.3	82.4	0.0	23.9	58.0	69.0	81.5	91.0	75.4	75.3	66.7	58.9
PointTransformer[28]	90.8	76.5	70.4	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
OneFormer3D [47]	-	-	72.4	92.0	96.5	81.5	0.0	40.9	66.2	81.4	43.9	87.0	48.5	46.0	81.3	43.9
Ours	**90.2**	**74.2**	**68.1**	95.1	98.2	82.8	0.0	26.1	59.9	72.8	81.7	91.2	80.3	75.8	60.4	58.9

TABLE II. SEMANTIC SEGMENTATION RESULTS ON THE S3DIS DATASET, EVALUATED WITH 6-FOLD CROSS-VALIDATION.

Method	**OA**	**mAcc**	**mIoU**
PointNet [15]	78.5	66.2	47.6
SPGraph [39]	85.5	73.0	62.1
PointWeb [41]	87.3	76.2	66.7
PointCNN [38]	88.1	75.6	64.3
PAT [23]	-	76.5	64.3
KPConv [43]	-	79.1	70.6
CBL [48]	89.6	79.4	73.1
Point Transformer [28]	90.2	81.9	73.5
PointMetaBase-XXL [49]	91.3	-	77.0
Ours	**90.6**	**81.9**	**73.7**

C. ScannetV2

1) ***Data and metric****:* ScanNetV2 is a large-scale 3D indoor scene understanding dataset, containing over 1500 scanned indoor scenes, with each 3D point annotated with a semantic category, such as walls, furniture, chairs, etc., across 21 categories. The training and testing process includes 1201 training sets, 312 validation sets, and 100 test sets. The smallest dataset contains 27,711 points, while the largest contains 1,612,218 points. For evaluation metrics, we use mean Intersection over Union (mIoU).

2) ***Results****:* The results are shown in Table III. On the ScannetV2 dataset, Point HorNet achieved val mIoU/test mIoU of 70.3%/71.5%. It outperformed neighborhood feature learning-based frameworks(such as PointNet++ [14]), and powerful point-based models (like PointConv [20], KPConv [43]).

TABLE III. SEMANTIC SEGMENTATION RESULTS ON THE SCANNETV2.

Method	val mIoU	test mIoU
PointNet++ [14]	53.5	33.9
PointConv [20]	61.0	55.6
KPConv [43]	69.2	68.0
SparseConvNet[44]	69.3	72.5
EMSANet [50]	71.0	60.0
Ours	**70.3**	**71.5**

V. CONCLUSION

The introduction of HorNet and recursive gated convolution has enabled arbitrary-order spatial interactions, addressing issues of feature loss due to low-order spatial interactions, and has achieved impressive progress in image analysis tasks such as image classification, object detection, and semantic segmentation. Inspired by this success, we have researched and implemented a recursive gated convolution for 3D point cloud data, enabling high-order spatial interactions on point clouds, compensating for the low-order spatial interactions in traditional point cloud deep learning models, and on this basis, proposed a new network, Point HorNet. Experiments have demonstrated the effectiveness and accuracy of recursive gated convolution on point clouds and Point HorNet. We hope our efforts will inspire future work to further explore high-order spatial interactions in point cloud data models.

In future work, we will attempt to prune the Point HorNet model by removing unimportant parameters or layers, thereby reducing the model size and computational load. We plan to divide large-scale point cloud data into smaller blocks or hierarchies and process each part separately, which will decrease the number of points processed at once and enhance scalability. Leveraging GPU parallel computing capabilities, we aim to process point cloud data in parallel to improve the scalability of handling large-scale point cloud data. Additionally, we will optimize the spherical query process to reduce redundant computational overhead.

REFERENCES

[1] Zermas D, Izzat I, Papanikolopoulos N. Fast segmentation of 3D point clouds:aparadigm on LiDAR data for autonomous vehicle applications[C] // 2017 IEEE International Conference on Robotics and Automation (ICRA),May 29-June 3,2017,Singapore.New York:IEEE Press,2017:5067-5073.

[2] Cotella V A. From 3D point clouds to HBIM: application of artificial intelligence in cultural heritage[J]. Automation in Construction, 2023, 152: 104936.

[3] Wirth F, Quehl J, Ota J, et al. Pointatme: efficient 3d point cloud labeling in virtual reality[C]//2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019: 1693-1698.

[4] Chen C, Asoni D E, Barrera D, et al. HORNET: High-speed onion routing at the network layer[C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015: 1441-1454.

[5] JING Zhuangwei, GUAN Haiyan, ZANG Yufu, NI Huan, LI Dilong, YU Yongtao. Survey of Point Cloud Semantic Segmentation Based on Deep Learning[J]. Journal of Frontiers of Computer Science and Technology, 2021, 15(1): 1-26.

[6] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. In RSS, 2016.

[7] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In CVPR, 2018.

[8] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In CVPR, 2018.

[9] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In CVPR, 2019.

[10] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In ICCV, 2015.

[11] Maturana D, Scherer S. Voxnet: A 3d convolutional neural network for real-time object recognition[C]//2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2015: 922-928.

[12] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In CVPR, 2017.

[13] HIMMELSBACH M, HUNDELSHAUSEN F V, WÜNSCHEH J. Fast segmentation of 3D point clouds for ground vehicles[C]//Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, La Jolla, Jun 21-24, 2010. Piscataway:IEEE, 2010: 560-565.

[14] Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[J]. Advances in neural information processing systems, 2017, 30.

[15] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.

[16] Hu, Q., Yang, B., **e, L., Rosa, S., Guo, Y., Wang, Z., ... & Markham, A. (2020). Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11108-11117).

[17] Phan A V, Le Nguyen M, Nguyen Y L H, et al. Dgcnn: A convolutional neural network over large-scale labeled graphs[J]. Neural Networks, 2018, 108: 533-543.

[18] Xu, Q., Sun, X., Wu, C. Y., Wang, P., & Neumann, U. (2020). Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5661-5670).

[19] THOMAS H, QI C R, DESCHAUD J E, et al. Kpconv: flexible and deformable convolution for point clouds[C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Oct 27-Nov 2, 2019. Piscataway: IEEE, 2019: 6410-6419.

[20] Wu, W., Qi, Z., & Fuxin, L. (2019). Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition* (pp. 9621-9630).

[21] ENGELMANN F, KONTOGIANNI T, HERMANS A, et al. Exploring spatial context for 3D semantic segmentation of point clouds[C]//Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops, Venice, Oct 22-29, 2017. Washington: IEEE Computer Society, 2017:716-724.

[22] YE X Q, LI J M, HUANG H X, et al. 3D recurrent neural networks with context fusion for point cloud semantic segmentation[C]//LNCS 11211: Proceedings of the 15th European Conference on Computer Vision, Munich, Sep 8-14, 2018. Berlin, Heidelberg: Springer, 2018: 415-430.

[23] YANG J C, ZHANG Q, NI B B, et al. Modeling point clouds with self-attention and gumbel subset sampling[C]//Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, Jun 16- 20, 2019. Piscataway: IEEE, 2019: 3323-3332.

[24] GUO M H, CAI J X, LIU Z N, et al. PCT:Point cloud transformer[J]. arXiv:2012.09688, 2020.

[25] WANG X L, LIU S, SHEN X Y, et al. Associatively segmenting instances and semantics in point clouds[C]//Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, Jun 16-20, 2019. Piscataway: IEEE, 2019: 4096-4105.

[26] PHAM Q H, NGUYEN T, HUA B S, et al. JSIS3D: joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields[C]//Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, Jun 16- 20, 2019. Piscataway: IEEE, 2019: 8827-8836.

[27] Humphreys G W, Sui J. Attentional control and the self: the Self-Attention Network (SAN)[J]. Cognitive neuroscience, 2016, 7(1-4): 5-17.

[28] Zhao H, Jiang L, Jia J, et al. Point transformer[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 16259-16268.

[29] Nezhadarya, E., Taghavi, E., Razani, R., Liu, B., & Luo, J. (2020). Adaptive hierarchical down-sampling for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12956-12964).

[30] Bjorck N, Gomes C P, Selman B, et al. Understanding batch normalization[J]. Advances in neural information processing systems, 2018, 31.

[31] K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. NIPS, pp. 2377‑2385, 2015.

[32] Rao Y, Zhao W, Zhu Z, et al. Global filter networks for image classification[J]. Advances in neural information processing systems, 2021, 34: 980-993.

[33] H. Su et al. "SPLATNet: Sparse Lattice Networks for Point Cloud Processing." In CVPR, 2018.

[34] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis,Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In CVPR, 2016.

[35] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser and M. Nießner, "ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 2432-2443, doi: 10.1109/CVPR.2017.261.

[36] Tchapmi L, Choy C, Armeni I, et al. Segcloud: Semantic segmentation of 3d point clouds[C]//2017 international conference on 3D vision (3DV). IEEE, 2017: 537-547.

[37] Tatarchenko M, Park J, Koltun V, et al. Tangent convolutions for dense prediction in 3d[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 3887-3896.

[38] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, XinhanDi, and Baoquan Chen. Pointcnn: Convolution on X - transformed points. In NIPS, 2018.

[39] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In CVPR, 2018.

[40] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In CVPR, 2018.

[41] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia.PointWeb: Enhancing local neighborhood features for point cloud processing. In CVPR, 2019.

[42] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi Wing Fu, and Jiaya Jia. Hierarchical point-edge interaction network for point cloud semantic segmentation. In ICCV, 2019.

[43] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Franc ¸ois Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In ICCV, 2019.

[44] Wang J, Li W, Zhang M, et al. Large kernel sparse ConvNet weighted by multi-frequency attention for remote sensing scene understanding[J]. IEEE Transactions on Geoscience and Remote Sensing, 2023, 61: 1-12.

[45] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer International Publishing, 2015: 234-241.

[46] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In CVPR, 2017.

[47] Kolodiazhnyi M, Vorontsova A, Konushin A, et al. Oneformer3d: One transformer for unified point cloud segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 20943-20953.

[48] Tang L, Zhan Y, Chen Z, et al. Contrastive boundary learning for point cloud segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 8489-8499.

[49] Lin H, Zheng X, Li L, et al. Meta architecture for point cloud analysis[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 17682-17691.

[50] Seichter D, Stephan B, Fischedick S B, et al. PanopticNDT: Efficient and Robust Panoptic Map**[C]//2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2023: 7233-7240.