International Journal of
Geo-Information

MDPI

Article

IAGC: Interactive Attention Graph Convolution Network for Semantic Segmentation of Point Clouds in Building Indoor Environment

Ruoming Zhai [1,2], Jingui Zou [1,*], Yifeng He [1,2] and Liyuan Meng [3]

[1] School of Geodesy and Geomatics, Wuhan University, Wuhan 430072, China; ruomingzhai@whu.edu.cn (R.Z.); yifeng.he@tum.de (Y.H.)

[2] Guangxi Key Laboratory of Spatial Information and Geomatics, Guilin 541004, China

[3] Beijing Key Laboratory of Urban Spatial Information Engineering, Beijing 100038, China; lymeng@whu.edu.cn

* Correspondence: jgzou@sgg.whu.edu.cn; Tel.: +86-1338-757-1063

Abstract: Point-based networks have been widely used in the semantic segmentation of point clouds owing to the powerful 3D convolution neural network (CNN) baseline. Most of the current methods resort to intermediate regular representations for reorganizing the structure of point clouds for 3D CNN networks, but they may neglect the inherent contextual information. In our work, we focus on capturing discriminative features with the interactive attention mechanism and propose a novel method consisting of the regional simplified dual attention network and global graph convolution network. Firstly, we cluster homogeneous points into superpoints and construct a superpoint graph to effectively reduce the computation complexity and greatly maintain spatial topological relations among superpoints. Secondly, we integrate cross-position attention and cross-channel attention into a single head attention module and design a novel interactive attention gating (IAG)-based multilayer perceptron (MLP) network (IAG–MLP), which is utilized for the expansion of the receptive field and augmentation of discriminative features in local embeddings. Afterwards, the combination of stacked IAG–MLP blocks and the global graph convolution network, called IAGC, is proposed to learn high-dimensional local features in superpoints and progressively update these local embeddings with the recurrent neural network (RNN) network. Our proposed framework is evaluated on three indoor open benchmarks, and the 6-fold cross-validation results of the S3DIS dataset show that the local IAG–MLP network brings about 1% and 6.1% improvement in overall accuracy (OA) and mean class intersection-over-union (mIoU), respectively, compared with the PointNet local network. Furthermore, our IAGC network outperforms other CNN-based approaches in the ScanNet V2 dataset by at least 7.9% in mIoU. The experimental results indicate that the proposed method can better capture contextual information and achieve competitive overall performance in the semantic segmentation task.

Keywords: deep learning; point cloud; semantic segmentation; self-attention mechanism; graph convolution

check for updates

Citation: Zhai, R.; Zou, J.; He, Y.; Meng, L. IAGC: Interactive Attention Graph Convolution Network for Semantic Segmentation of Point Clouds in Building Indoor Environment. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 181. https://doi.org/10.3390/ijgi11030181

Academic Editor: Wolfgang Kainz

Received: 11 January 2022
Accepted: 6 March 2022
Published: 8 March 2022

1. Introduction

In the reconstruction of indoor environments, laser scanning point clouds have been generally used, providing high-precision and rich spatial information for the next Building Information Modeling (BIM) [1]. Nonetheless, effective semantic segmentation should be conducted before retrieving geometric entities. It can boost better scene understanding and high-accuracy, entity-based modeling [2,3].

In the past few years, segmenting operations have mainly focused on designing handcrafted features [4–6] using empirical knowledge about spatial geometrics or symmetry. They are relatively limited to specific scenarios with particular geometric primitives. However, composite indoor entities, such as tables, chairs, bookshelves, etc., show irregular geometric structures or different physical information and are difficult to semantically segment by handcrafted features. In addition, some nonvisual latent information hidden in high-level features may be neglected, making it difficult to make a distinction between different categories of objects with marginal differences. Moreover, the complex layout of indoor environments leads to the occlusion and incompleteness of point clouds. It may impede artificial feature design and decrease the accuracy of segmentation. Meanwhile, with the remarkable achievements of deep learning techniques in the fields of natural language processing [7,8] and computer vision [9,10], much heuristic research attempted to directly apply the already matured 2D convolution neural network (CNN) to point clouds to automatically extract high-dimension features for shape classification, segmentation, and object detection and tracking [11,12]. Due to the irregular, uneven, and unstructured features of point clouds, most methods tended to reorganize the point cloud into regular data structures as the intermediate representation for the application of 2D CNN. This data projection, on the other hand, results in redundant, time-consuming conversion procedures and a considerable amount of storage space.

Recently, PointNet [13], a pioneering work, carried out pointwise feature extraction without transformation of input data. It comprises several shared permutation invariant operations, multilayer perceptron (MLP) layers, and a max-pooling layer. However, the efficient and concise representation of pointwise feature learning cannot capture local structures in more complicated scenes. An efficient and robust network, PointNet++ [14], is proposed to capture geometric structures from multiple scales by recursively applying PointNet to hierarchical structures. The hierarchical structure consists of a series of set abstraction levels. At each level, it implements the sampling layer and the grouping layer to randomly select points from input points and construct their regional neighborhoods, followed by a PointNet layer to learn the local geometric features. By stacking these set abstraction levels, the local geometric features are aggregated layer by layer, and finally the global features are progressively extracted to represent the whole complex scene. Then, the application of PointNet++ inspired many subsequent state-of-the-art networks. They perform pointwise operations on local neighborhoods while also hierarchically aggregating global features across the entire large-scale point cloud. Nonetheless, unlike the natural spatial connection between the regular adjacent pixels of images, the potential information of spatial topological relationships between different points cannot be fully learned among the unordered neighboring points. As a result, graph-based networks [15–18] have been proposed to treat each point as a vertex within a graph structure and update its feature based on contextual information between connected edges and vertexes. However, building a global graph structure among massive point clouds complicates the segmentation process.

Meanwhile, a great variety of Transformer networks [19] originally designed for machine translation have shown a greater capacity for modeling contextual dependencies than CNNs and recurrent neural networks (RNNs) by explicitly relating different positions of a sequence. Many Transformer variants have expanded their applications to the computer vision field. However, in the case of point clouds, the full attention mechanism demonstrates its inefficiency in indiscriminately calculating attention scores among massive points.

Overall, the current limitations of CNN-based methods lie in the following directions:

1. The high-dimensional features based on regional points, which can distinguish different objects with similar features but distinctive positions, are not fully utilized in designed convolution kernels.
2. The contextual information derived from the fully-connected graph structure for all points not only reduces the efficiency but adversely impacts the generalization of global interaction.

To balance the requirements for both rich feature information and high efficiency, we propose a novel interactive-attention-based graph convolution (IAGC) network to selectively pay attention to the notable features in each homogeneous clustering pointset, called superpoints [20]. And then, according to the global graph constructed by the superpoints, we progressively update the embedding by attaching features of connected superpoints to the input central superpoint. Concretely, inspired by the idea of gMLPs [21], a heuristic and lightweight interactive-attention gMLP architecture (IAG–MLP) is designed to dynamically assign proper attentional weights to parts of feature channels for local feature learning on superpoints, which can be more effective than Transformers. Furthermore, by globally integrating features from other superpoints, a simplified variation of the Long-Short-Term Memory (LSTM) architecture [22], known as the Gated Recurrent Unit (GRU) [23], can be implemented on superpoint-level semantic inference.

Thus, the main contributions of the proposed algorithm are summarized as follows:

1. A dual cross-attention Transformer variant, called IAG–MLP, is proposed to be directly oriented to superpoints that are reorganized from raw point clouds into geometry-based and color-based homogeneous segments, and it will enhance the capability of capturing high-dimensional contextual dependencies in local embeddings by learning both cross-position attention and cross-channel attention.
2. By propagating contextual messages via nearby superpoints and related super-edges, an end-to-end graph network is built to gradually update the feature embeddings of superpoints, finally translating superpoint-level semantic inference into point-level fine-grained inference.
3. We present theoretical and empirical analyses of the proposed IAGC architecture, as well as qualitative and quantitative experiments in three indoor benchmarks that indicate its effectiveness and remarkable performance.

The remainder of the paper is organized as follows. Section 2 gives a brief review of the related work. Section 3 describes the details of the proposed semantic segmentation method. Section 4 provides the qualitative and quantitative experimental results to validate the proposed segmentation approach, and Section 5 shows several concluding remarks.

2. Related Work

The classic approach to semantically label large-scale point clouds is to reorganize point clouds into a regular structure for a compact convolution function. Meanwhile, many aggregation functions based on graph structure are currently proposed for capturing the underlying connections between different points. This section mainly reviews semantic segmentation methods for point clouds, the data structure of point clouds fed to the deep learning-based networks, the prevailing Transformer variants, and graph convolutions for contextual understanding.

2.1. Semantic Segmentation for Point Clouds

In terms of semantic segmentation for better indoor scene understanding, they can be divided into model-driven, knowledge-driven, and data-driven approaches. Model-driven approaches first generate potential models of geometric primitives (e.g., lines, planes, cubes, and cylinders), and then find the largest cluster that best fits the geometric guesses [24]. They can iteratively implement the hypothesis and verification procedures to find multiple primitives, but are locally optimal in complex indoor environments due to the poor segmentation performance on unstructured geometries.

In order to fulfill global optimization solutions for different objects, the knowledge-driven, namely ontology-driven, approaches boost the optimal selection of algorithms for particular geometries by building an ontology integrating data characteristics, potential algorithms, and explicit prior knowledge. Specifically, the external information drives the ontology model to apply different algorithms according to different point characteristics, and the segmentation results reversely increase the feature gap among different categories in the initial acquisition process [25,26]. Therefore, the ontology serves as a meta diagram to share and reuse external knowledge throughout the whole workflow and finally contributes to global optimization.

On the other hand, the current state-of-the-art data-driven approaches focus on designing reasonable deep learning networks and improving the quality of training data. Theoretically, the characteristics of point clouds can be implicitly mapped to multilayer neural networks without external knowledge interference and be capable of segmenting different categories of objects, especially for composite entities and outliers.

2.2. Deep Learning Networks for Semantic Segmentation

Existing deep learning methods for semantic segmentation can be categorized into two aspects according to the granularity of point clouds on which the feature extraction is performed: projection-based networks and point-based networks. Most projected-based networks commonly apply convolution operations, which have achieved excellent performance on regular and compact 2D images or text sequences, to the unordered and unstructured point clouds by projecting them into intermediate regular representations, such as the voxel-based representation [27], the Multiview-based representation [28], or the higher dimensional lattice representation [29]. Normally, these methods not only result in unnecessary memory and computational consumption but also disrupt the natural spatial association among point clouds. In contrast, pointwise methods enable directly learning features on raw point clouds without introducing extra data transformation. Most of the later networks improved their capabilities in modeling local structures with the point-based bedrock network, PointNet. For instance, PointNet++ [14] exploited Furthest Point Sampling (FPS) to hierarchically downsample point clouds and iteratively extract features from PointNet in each sampling layer. In order to describe large-scale scenes from multi-resolutions, MSSCN [30] concatenated point features with different densities, PointSIFT [31] paid attention to encoding both multi-orientations and multi-scales for local details, and PointCNN [32] employed a fully convolutional point network with a series of abstraction layers, feature learners at different scales, and a merging layer. When the random sampling strategy in large-scale scenes is time-consuming since it works on the original points, the use of supervoxels [33], which were inspired by superpixels in 2D image processing, greatly cuts down the number of points and provides a more natural and compact representation for local operations. Nevertheless, the fixed resolution of supervoxels may lead to inaccurate segmentation in marginal areas of multiple objects since their local neighborhood characterizes different classes of points, and it is unnecessary with respect to objects with large areas, such as walls, ceilings or floors. At the same time, according to global energy optimization, the superpoint [20] was constructed by geometrically and even physically partitioning the point clouds without predefining the number of segments, which minimizes unnecessary segmentation of objects with large areas but maintains topological relationships among superpoints by building a global graph. Later, the construction of superpoints was improved by building a label consistency loss between true labels of points and pseudo labels of the superpoints in an end-to-end network [34]. Furthermore, the cascaded nonlocal network [35] adopted the superpoints as basic units and built a nonlocal operation with three granularity levels, including neighborhood-level, superpoint-level, and global-level. As a result, the contextualization among different supepoints was hierarchically aggregated through the stacking of a number of nonlocal modules.

2.3. Attentive-Based Transformer Networks

The Transformer architecture widely adopted in natural language processing focuses on building contextual relationships in the encoder–decoder sequential structure, which normally consists of several stacked multihead self-attention modules, a feed-forward network (FFN), and a residual connection in each encoder or decoder block. As a crucial element of Transformer, self-attention aims at generating dynamical weights from pair-wise relations of inputs and learning context-dependent representations for each token in a sequence.

Since the revolutionary Transformer network has made impressive progress in the computer vision field, such as the recent Vision Transformer (ViT) [36], it is inevitable to apply it to deep learning on point clouds. Nonetheless, when it comes to larger scene-level datasets, the Transformer structure for 3D point clouds is rather expensive as the computation cost grows quadratically with the input size. To address these limitations, several derivative models learn local contextual information utilizing the crucial self-attention mechanism and build long-range dependencies in the 3D point cloud space. For instance, instead of a simple MLP layer or max-pooling layer in PointNet, PointTransformer [37] employed vector-based self-attention coupled with relation subtraction and position encoding addition to hierarchically transit down in feature encoding and transit up in feature decoding. PointCloudTansformer (PCT) [38] used an offset-attention transformer with only encoded modules to improve feature learning for shape classification and part segmentation. The Pointformer [39] is proposed to model interactions among points in the local region with multiscale Local Transformers (LT), and the RandLA-Net [40] executed self-attention on random sampling points with high efficiency. The research in [41] exploited gated fusion in regional structures and spatial-wise and channel-wise attention in global structures. In general, performing the attention mechanism in large-scale scenes may lead to a considerable computational workload for pairwise attentional weights so integrating self-attention among regional structures and long-range dependencies in a global graph can be more effective.

2.4. Graph Convolutions

Due to its increasingly prominent capability in processing unstructured data, the Graph Convolution Network (GCN) [42–44] is undoubtedly employed in processing point clouds and is generally classified into two groups: the spectral-based method and the spatial-based method. Spectral-based approaches perform convolution by converting vertex representations into the spectral domain with the Fourier Transformation or its extensions [45,46]. In contrast, spatial-based approaches directly perform convolution based on graph topology. For example, PyramNet [47] formulated the covariance matrix within a directed acyclic graph to explore regional connections among points and proposed a Pyramid Attention Network to extract features with different semantic intensities. In addition, Graph Attention Convolution (GAC) [18] defined the shape of the convolution kernel by calculating attention weights among neighboring points in a connected graph to underpin the importance of relevant parts. In order to expand receptive fields, the MS-RRFSegNet [48] conducted supervoxel-level feature segmentation to obtain more descriptive contexts. However, these networks mentioned above constructed local graphs based on the regional distribution of neighboring points and inherently capture regional contextual information.

3. Methodology

In this paper, we propose a novel graph network for semantic segmentation for large-scale point clouds in indoor scenes. To be specific, we first introduce superpoints obtained from the oversegmentation of point clouds and then demonstrate our stacked IAG–MLP module for embedding learning of superpoint-level representation. Finally, with the recurrent convolution module, a global directed graph is presented to iteratively update previous feature embeddings.

3.1. Oversegmented Superpoint Generation

Considering the computation consumption in processing millions of points in a large-scale indoor scene, the point clouds are oversegmented into geometrically and physically homogeneous pointsets, i.e., superpoints can be viewed as basic operation units for deep learning networks. In this way, the number of superpoints in a scene and the number of points contained in a superpoint would not be defined in advance, which ensures the minimum amount and the maximum structural completion of simple segments for local feature learning. In this case, we assume points from the same superpoint should present similar features and consequently share the same class label. According to the Principal Component Analysis (PCA) [49] algorithm, shape features calculated on the optimal K-nearest neighborhood adjacency graph can be retrieved by constructing a covariance matrix and decomposing its eigenvalues, which are positive and ordered, i.e., $0 <\lambda_1 < \lambda_2 < \lambda_3$. Then, the PCA describes three principal directions, which reveal the volumetric, planar, and linear characteristics of the neighborhood.

$$\begin{cases} m_S = \frac{\lambda_1}{\lambda_3} \\ m_P = \frac{\lambda_2 - \lambda_1}{\lambda_3} \\ m_L = \frac{\lambda_3 - \lambda_2}{\lambda_3} \end{cases} \tag{1}$$

$$m_V = \frac{\sum\limits_{j=1}^{3} \lambda_j |[u_2]|}{\| \sum\limits_{j=1}^{3} \sum\limits_{i=1}^{3} \lambda_j \cdot |[u_2]| \|} \tag{2}$$

The scattering feature m_S refers to the isotropic elliptical shape of the neighborhood, the planarity m_P defines the average distance all around the center of gravity, and the linearity m_L describes how elongated the adjacency is. In addition, the verticality m_V can be also introduced for discriminating objects in different vertical distribution, where u_1, u_2, u_3 are the three eigenvectors associated with λ_1, λ_2, λ_3, respectively.

In order to cluster homogeneous points with similar features, the geometric features mentioned above and physical features such as color could all be taken into consideration for the generalized minimal partition problem, which is studied by a piecewise constant approximation of the global energy function [20]:

$$g^* = \arg \min_{g \in R^{7 \times V}} \sum\nolimits_{i \in V} \|g_i - f_i\|^2 + \mu \sum\nolimits_{(i,j) \in E} \delta(g_i - g_j) \tag{3}$$

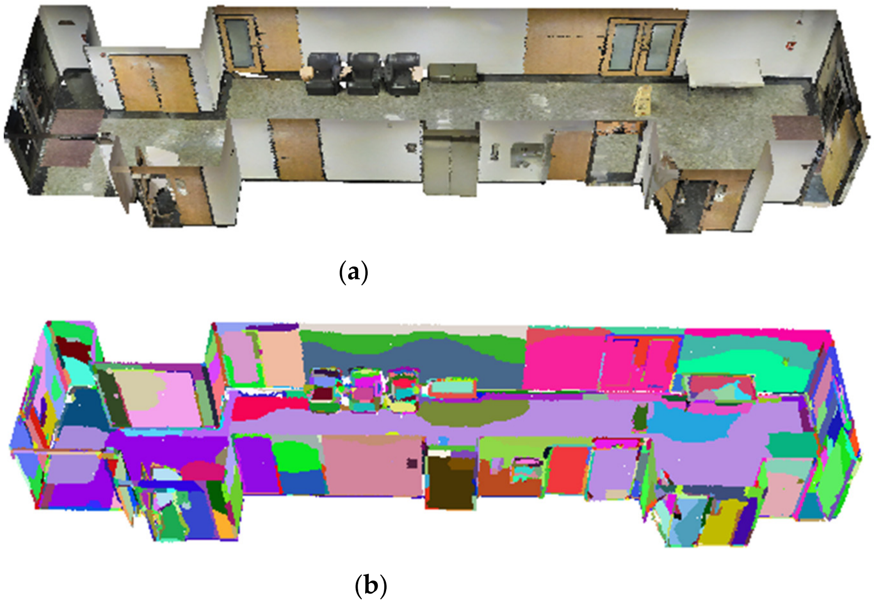
For each point $i \in V$, the shape of its local neighborhood is characterized by the combined features $f \in R^{7 \times V}$, which contains 4 geometrical features and 3 physical features. The first part of this energy function is the fidelity formulation, ensuring that the constant segments of g^* correspond to the homogeneous value of f. The second part is the regularized function that adds a constraint for each edge connecting two segments with different values. In addition, $\delta(\cdot \neq 0)$ refers to the Iverson bracket and the regularization strength characterizes the coarseness of the resulting partition as well as the granularity of the superpoint graph, i.e., the number of total segments. In fact, although the optimization problem is a non-convex and noncontinuous function, which cannot be straightforwardly resolved, the l_0-cut pursuit algorithm [50] can exploit graph cuts to recursively split the level-sets of a piecewise-constant candidate solution. In practice, as shown in Figure 1, the inferring segments corresponding to superpoints are partitioned into different sizes and shapes owing to their geometric and physical features.

Finally, as long as the point clouds are reconstructed by the superpoint graph with superpoints along with their connected edges, we concatenate the spatial position features (spatial position, normalized position, elevation), geometric features (scattering, planarity, linearity, verticality), and color features (RGB values) of superpoints as input features for our proposed local feature extraction network. In particular, the elevation is introduced for discriminating objects at different heights relative to the indoor floor, which is defined as follows:

$$E = \frac{\sum z}{z_{\max} - z_{\min}} - 0.5 \tag{4}$$

3.2. IAG–MLP

Local feature extraction is crucial for the later global graphical aggregation operation. However, automatically capturing local structural features based on the neighboring region is still challenging. In fact, most of the current state-of-the-art networks resort to the simple and concise PointNet for implementing convolution functions with permutation invariance of points in the local region. Nonetheless, the final max-pooling operation in PointNet serves as the "max attention" mechanism that only considers the most representative features such as the contours of pointsets in the feature space and ignores the structural correlations between the remaining interior points, which may be contributed differently to the feature learning.


(**a**)

(**b**)

Figure 1. Illustration of superpoint partition. (**a**) Raw point clouds; (**b**) Superpoint Geometric Partitioning Result.

To deal with the neglected important information in the local convolution operation induced by the PointNet, the Transformer with a self-attention mechanism that pays attention to the region with rich information can be adopted as a novel feature learning algorithm. We first review the self-attention mechanism applied by the most famous vanilla Transformer [19] (see Figure 2) in the machine translation field. Given an input feature, self-attention linearly projects the input feature into a query matrix Q, a key matrix K, and a value matrix V, which can be formulated as:

$$A(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{5}$$

where A is the attention matrix indicating the pairwise affinity among tokens of a sequence and as a scaling factor, d_k is the dimension of matrix K.

However, when it applies to point cloud processing, due to its high computational burden with dot-product attention calculation among points in each encoder or decoder layer, it is infeasible to directly implement the stacked encoder-decoder mode on massive point clouds. Furthermore, since the high dimension is indispensable to better high-level representation, separately using a linear combination of pairwise values as self-attentional weights to ameliorate input features in each feature channel may augment memory consumption in the high dimension. Therefore, inspired by both the vanilla Transformer and gMLPs [21], we propose a novel IAG–MLP network directly oriented to superpoints, which can automatically impact and encode the embedding representation of each superpoint with a lightweight interactive-attention gating architecture.

The overall scheme is shown in Figure 3. Given a superpoint together with per-point features (e.g., raw RGB, spatial coordinates, normalized location, elevation, and geometric features mentioned above), this local encoding unit first samples N internal points and embeds them from these handcrafted features into high-dimensional features, such that cross-attention can be calculated by two split high-dimensional features for complex local structure learning. Afterward, our proposed IAG block captures spatial interactions with the shared memory unit for cross-position attention and spreads channel communication in the gating operation for cross-channel attention. In practice, unlike the self-attention mechanism, which has a high computational complexity $O(n^2 d)$, the cross-attention mechanism allows learnable spatial transformation with a lower computational complexity $O(\frac{n^2 d}{2})$, where n denotes the number of regional points and d denotes the dimension of the input feature.

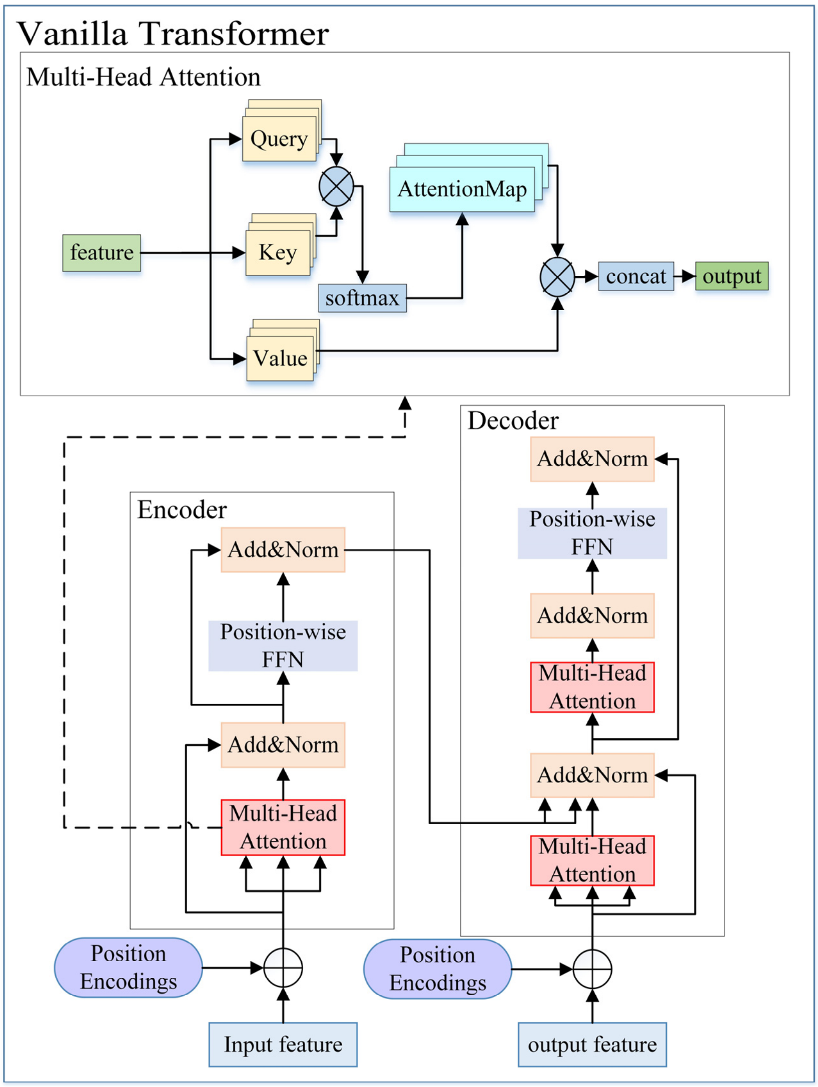


Figure 2. Overview of vanilla Transformer architecture.

To be specific, this IAG block includes the following steps:

(1) Contraction Operation. To enable cross-channel interaction, it is necessary to contain a contraction operation over the entire feature dimension. The succinct way is by applying a linear projection coupled with an anterior normalization function and a posterior activation function, which can be formulated as:

$$f_{W,b}(X) = \sigma(W(norm(X)) + b) \tag{6}$$

where the input feature X is normalized in a batchnorm way [51], which is necessary for the learning to converge. Then a linear projection is implemented by the matrix multiplication of $W \in R^{2d \times 2d}$ for which the size of $2d$ is the projected feature dimensions. In addition, b and σ refer to a bias, which can either be a matrix or a scalar and an activation function such as $RELU$ [52].

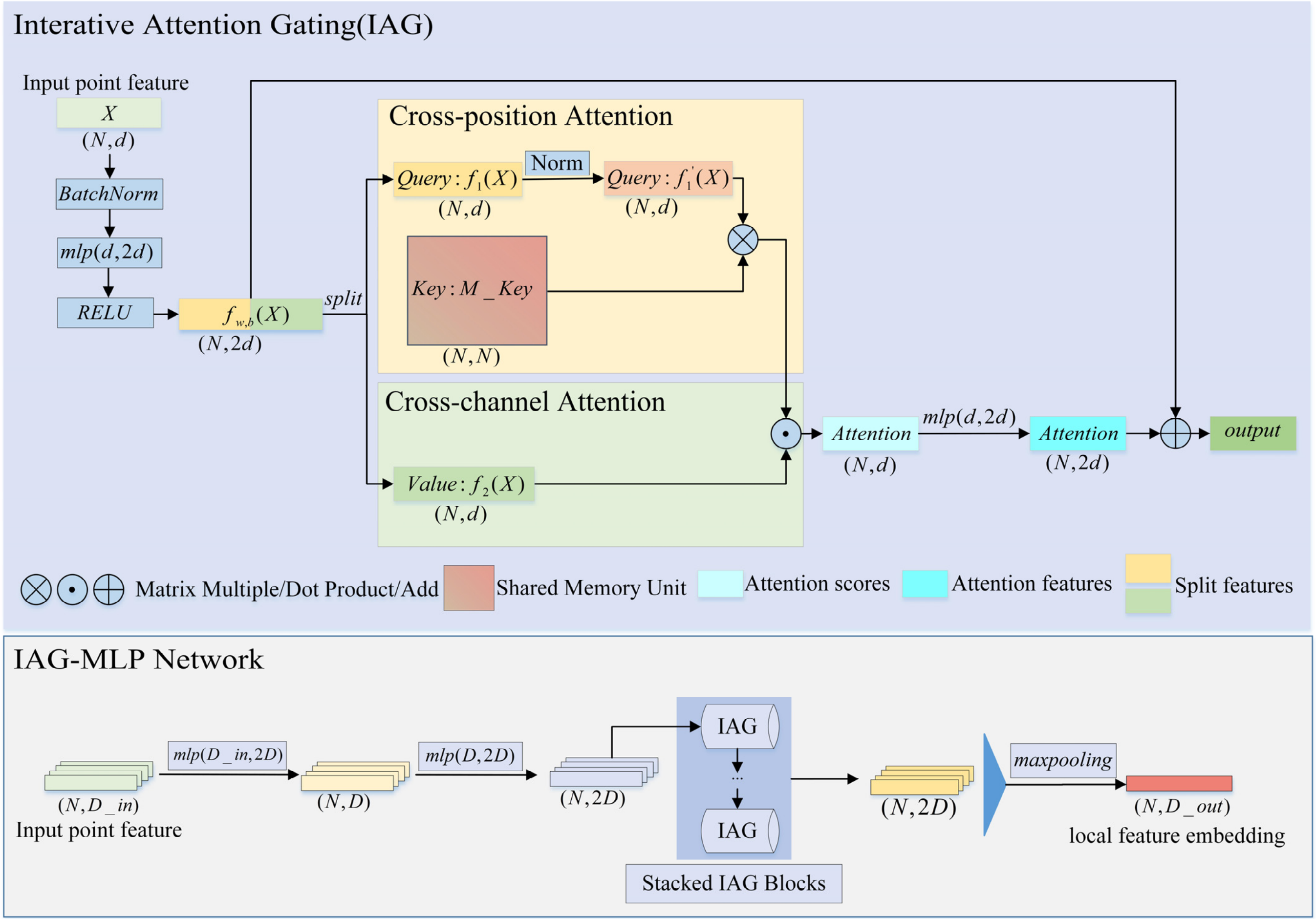


Figure 3. The proposed local feature aggregation module. The bottom panel illustrates the Interactive Attention Gating (IAG)–multilayer perceptron (MLP) Network for local feature encoding in preprocessed superpoints. The top panel illustrates the kernel IAG block, which exploits the dual cross-attention mechanism to compute the cross-position attention weights by a shared memory unit and operate the cross-channel attention on the feature channels, constructed as a variant of the self-attention module.

To compute cross-attention between channel dimensions effectively, we split $f_{W,b}(X)$ into two independent components, $f_1(X)$ and $f_2(X)$, along the feature channel and implemented element-wise spatial interaction, which is known as the gating function and has been demonstrated to be feasible in Gated Linear Units (GLUs) [53,54]. In fact, we implement cross-position attention and cross-channel attention in $f_1(X)$ and $f_2(X)$, respectively, and aggregate them in the gating function to alleviate the vanishing gradient problem. To be noticed, we do not employ the multihead structure for attention calculation as the dimension of point clouds is rather small compared with the dimension of text sequences, and it is unnecessary to split the features in the local embedding network for superpoint-level deep learning.

(2) Cross-position Attention. Self-attention is normally treated as a linear projection algorithm utilizing self-values of data samples to ameliorate their own features, but this $N \times N$ self-attention matrix can only explain the interrelationship among points in the same training dataset, and it is not clear whether there is a specific correlation among data samples in a scene. Additionally, despite the small magnitude of parameters involved in the self-attention module, pairwise attention calculation cannot be ignored. Thus, we designed a spatial interactive attention unit, inspired by the external attention network [55], to calculate cross-position attention between

high-dimensional features and an external memory unit, which is independent of the input feature and shares information across the entire training dataset. Specifically, we construct the spatial interaction unit with the paradigmatic structure of the self-attention layer of the vanilla Transformer and initially normalize it in a similar way to PCT [38] with the double normalization method, which empirically improves the stability of local embedding networks.

$$x''_{i,j} = softmax(x'_{i,j}) = \frac{\exp(x'_{i,j})}{\sum_n \exp(x'_{i,j})} \tag{7}$$

$$x_{i,j} = \frac{x''_{i,j}}{\sum_n x''_{i,j}} \tag{8}$$

Then, according to the vanilla Transformer structure, the external memory unit $M\_key \in R^{N\times N}$ serving on the key matrix K can gradually record the contextual information among regional points in a matrix multiplication operation $\otimes$ by multiplying with the double normalized $f_1(X_1) \in R^{N\times d}$ referring to the query matrix Q.

$$F = Norm(f_1(X)) \otimes M\_Key^T \tag{9}$$

(3) Cross-channel Attention. Unlike the self-attention weights derived from pairwise attention among points, the cross-channel attention unit can be viewed as a cross-attention mechanism to modulate individual point representation using spatial signal. Specifically, the cross-channel attention map is inferred from the dot production of $f_2(X)$ and the cross-position attentional weight matrix F:

$$A = F \odot f_2(X) \tag{10}$$

The $\odot$ denotes the element-wise multiplication, which rapidly tunes the magnitude of each element in X in a feature pairwise way. Actually, this is a gating mechanism, i.e., the dot product operation of the output of the convolutional layer without nonlinear transformation and the output of the convolutional layer with nonlinear transformation in our IAG–MLP unit, but both cross-position and cross-channel attention are computed and fused in this gating mechanism in our IAG–MLP unit.

Furthermore, in order to employ a residual connection between the input feature and the attention map, we project the attention map to a $2d$ dimensions and add it to the input feature.

(4) Residual Connection Block. Theoretically, a deep learning network with more variables should be better able to do challenging tasks, but it is proven that simply deepening the layers makes it harder to train the network, which is called the degeneration problem. Hence, considering the degeneration problem of a deep learning network with increasing layers as it stacks more IAG–MLP modules, the residual connection block is put forward to create a concise shortcut where the projected input is put into the IAG block and passed through several layers to be finally integrated with the projected attentional map.

Due to the requirement for local feature representation, the max-pooling operation is carried out to form a feature vector to aggregate the relatively global feature embedding of a superpoint from sampled points. Eventually, we update the input features based on the spatial dimension rather than the channel dimension, merely with several simple MLP layers for higher dimension feature expression and stacked IAG blocks for enhanced feature signal.

3.3. Interactive Attention Graph Convolution Network (IAGC)

For a large-scale indoor scene, the spatial topological relations between different types of objects can be utilized as object-level contextual information to globally ameliorate the embedding representation. In order to improve the performance of semantic segmentation with segment-level contextualization aggregation, the graph convolution derived from the Edge-Condition Convolution (ECC) [56] would fuse superpoint-level embedding features and contextual information in the global graph for progressively updating semantic segmentation inference. Hence, in this work, we propose an end-to-end superpoint-based semantic segmentation network that firstly clusters geometrically and physically homogeneous points as intermediate representations as well as superpoints for extracting local features within the IAG–MLP local network, and then constructs the global SuperPoint Graph (SPG) for hierarchically and globally updating embedding representations with connected edges.

Specifically, to further explain the whole architecture of our proposed IAGC, Figure 4 illustrates the semantic segmentation inference from a raw point cloud scene to the superpoint-based graph, $G = (V, E)$, where V denotes the superpoint set and E is the oriented attributed edge set. Once the local feature is embedded into superpoints, the ECC iteratively conducts convolution operations on each superpoint without processing on the entire graph structure.

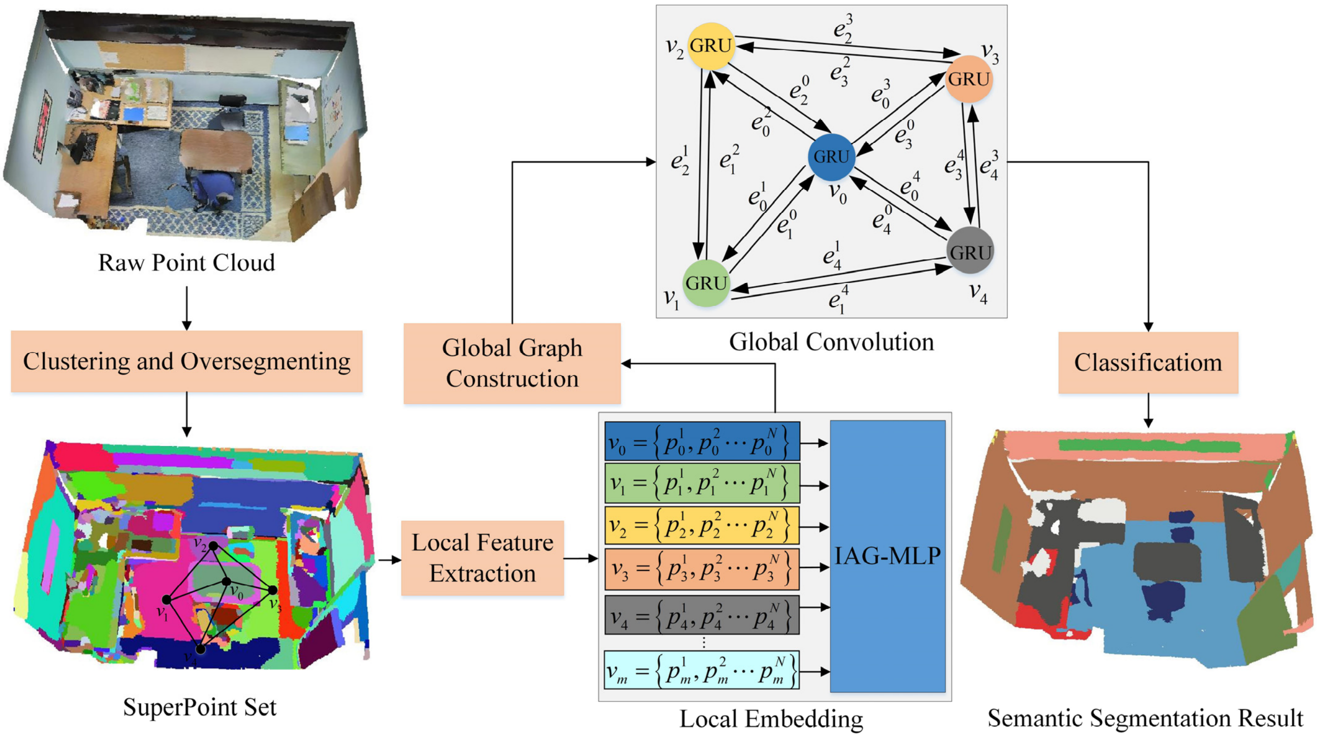

Raw Point Cloud
Clustering and Oversegmenting
SuperPoint Set
v_2
v_0
v_1
v_3
Local Feature Extraction
Global Graph Construction
Global Convolution
v_2 GRU
v_3 GRU
GRU v_0
v_1 GRU
GRU v_4
e_2^3
e_3^2
e_2^0
e_0^3
e_0^2
e_3^0
e_2^1
e_1^2
e_3^4
e_4^3
e_0^1
e_0^4
e_1^0
e_4^0
e_4^1
e_1^4
$v_0 = \{p_0^1, p_0^2 \cdots p_0^N\}$
$v_1 = \{p_1^1, p_1^2 \cdots p_1^N\}$
$v_2 = \{p_2^1, p_2^2 \cdots p_2^N\}$
$v_3 = \{p_3^1, p_3^2 \cdots p_3^N\}$
$v_4 = \{p_4^1, p_4^2 \cdots p_4^N\}$
$v_m = \{p_m^1, p_m^2 \cdots p_m^N\}$
IAG-MLP
Local Embedding
Classification
Semantic Segmentation Result

Figure 4. Architecture of the proposed interactive-attention-based graph convolution (IAGC) network. We perform the clustering and oversegmenting on raw point clouds for superpoint set for local feature extraction with IAG–MLP, which individually embeds each superpoint with N downsampling points. According to the connected vertexes and edges in the global graph, the embeddings are hierarchically finetuned in the Gated Recurrent Unit (GRU) and semantically labeled.

For instance, for a superpoint V_i and its connected superpoint V_j, assume that E_i^j is the connected edge between superpoints i and j with directional attributes regarding the ratio of geometric features, the ratio of point count, and the spatial relations of centroids.

Afterward, the global contextual information of each vertex in the SPG can be formulated by the following mean aggregation function as a global message:

$$m = \frac{1}{|N_j|} \sum_{j \in N_j} (W_i^j e_i^j) \cdot v_j \tag{11}$$

where W refers to dynamically generated parameters of the Dynamical Filter Network [57], which essentially is a MLP layer without bias, such that the feature dimension of the edge features is the same as the feature dimension of the superpoint embedding v, which facilitates the element-wise multiplication $(\cdot)$. In addition, $\frac{1}{N_j}$ normalizes and averages the contributions of other superpoints to the superpoint in the connected global graph.

Next, due to the capability of handling sequential input for contextual information processing, the GRU, which possesses fewer parameters but is as effective as the LSTM, is employed to construct and iteratively update a hidden state integrated by the embedding representation and the global contextual message propagating along the connected edges. In this case, we define the hidden state h_i as initialized with embedding v_i and conduct the global aggregation between the projected h and the global message m by an element-wise multiplication:

$$h_i^0 = v, x_i^t = \sigma\left(W_g h_i^t + b_g\right) \odot m_i^t \tag{12}$$

According to the gating mechanism in the GRU module, we first linearly project the current input x_i^t and the previous hidden state h_i^{t-1} into a higher dimensional embedding $\overline{x}_i^t$ and the hidden state $\overline{h}_i^{t-1}$ in $T = t$ iterations.

$$\overline{x}_i^t = W_x x_i^t + b_x, \overline{h}_i^{t-1} = W_h h_i^{t-1} + b_h \tag{13}$$

As shown in Figure 5, $\overline{x}_i^t$ and h_i^{t-1} are concatenated by vector addition and passed into a sigmoid function to squish their values between 0 and 1, where 0 means irrelevant features to throw away and 1 means useful features to keep. Thus, this filtering operation contributes to the update gate u_i and reset gate r_i.

$$u_i^t = \sigma(\overline{x}_i^t + \overline{h}_i^{t-1}), r_i^t = \sigma(\overline{x}_i^t + \overline{h}_i^{t-1}) \tag{14}$$

where u_i^t throws away irrelevant features and adds the new information, and r_i^t decided how much past features to forget in the later operation. Then, a new hidden state candidate $\overrightarrow{h}_i^t$ to emphasize strongly correlated dimensions and ignore weakly correlated dimensions is built by exploiting the tanh function to adjust the concatenated values of r_i^t, $\overline{h}_i^{t-1}$, and $\overline{x}_i^t$ between -1 and 1.

$$\overrightarrow{h}_i^t = \tanh\left(r_i^t \odot \overline{h}_i^{t-1} + \overline{x}_i^t\right) \tag{15}$$

Subsequently, the updating hidden state h_i^t for current iteration can be constructed by two components as followed:

$$h_i^t = \left(1 - u_i^t\right) \odot \overrightarrow{h}_i^t + u_i^t \odot \overline{h}_i^{t-1} \tag{16}$$

where $\left(1 - u_i^t\right) \odot \overrightarrow{h}_i^t$ decides the information flow of the current hidden state h_i^t, and $u_i^t \odot \overline{h}_i^{t-1}$ decides the previous one $\overline{h}_i^{t-1}$ through the update gate u_i^t. Eventually, the global contextualization is incorporated into the high-dimensional features, which are concatenated by the long sequences of hidden states.

$$y = W(h_i^0, h_i^1 \cdots h_i^t) \tag{17}$$

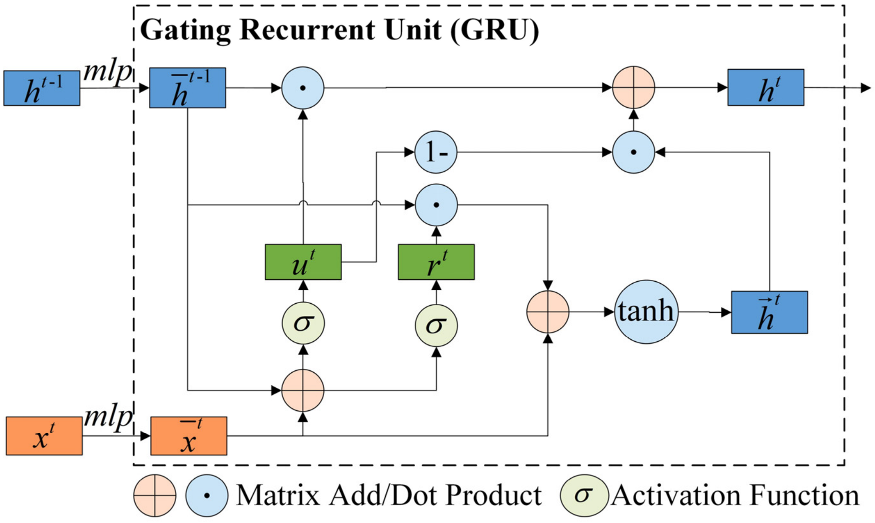


Figure 5. The structure of the GRU.

4. Experiments

In this section, we first introduce the details of the experimental setup. Secondly, we conduct ablation studies to validate the efficiency of individual components of our proposed IAGC on three 3D indoor datasets, including the SceneNN dataset, Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset, and ScanNet (V2) dataset. Lastly, we compare our network with several state-of-the-art networks, and end up with qualitative and quantitative evaluations of their performances.

4.1. Datasets

1. S3DIS [58]

The S3DIS dataset is collected in 6 large-scale indoor areas originating from 3 different office buildings, covering over 6000 m^2 with 271 rooms. Each point with geometric and physical attributes such as XYZ spatial coordinates and RGB features is categorized into 13 semantic classes. With respect to the dataset deployment and configuration, most state-of-the-art methods test their models on Area 5 since it comes from a different building but generally show poor performance in several categories, such as beam, column, and board, whose features are different from those of corresponding objects in other areas. Hence, in order to comprehensively verify our model, we provide the Area-4-fold results and the 6-fold cross-validation results.

2. ScanNet (V2) [59]

The ScanNet contains 1613 indoor scenes derived from RGB-D reconstruction, and its points are annotated into 20 classes, where 1513 scenes are generally divided into 1201 and 312 for training and validation, respectively, and the remaining 100 scenes without labeling are viewed as testing datasets submitted in its open benchmark challenge competition for verification. And yet, we split the validation dataset into two datasets for validating and testing in an ablation study to investigate the optimal voxelization of raw point clouds and the proper number of superpoints for graph aggregation.

3. SceneNN [60]

The SceneNN dataset is a scene mesh dataset consisting of 76 indoor rooms for semantic and instance segmentation. In particular, their semantic labeling complies with the NYU-D v2 [61] category standard with 40 semantic classes that range from building structural entities, such as walls, floors, and ceilings, to various furniture, but approximately 8 categories are rarely attached to the points, which inherently affect the overall performance of the whole categories. However, the diversity of semantic classes can verify the generalization of our model. Therefore, in our work, we apply them to our ablation experiments to explore both the effectiveness and generalization of our models by splitting them into three areas, which approximately follows the 51/15/10 room split for training, validating, and testing.

4.2. Implementation Details

All designed experiments are implemented with Pytorch on a high-performance server equipped with a 12-GB NVIDIA Tesla K80 GPU. Since our designed IAGC model aims at semantic segmentation in building interior scenarios with massive 3D points, in order to speed up the computation efficiency and improve segmenting performance during the training process, we preprocess the point clouds by 0.03 m interval voxelization for S3DIS and SceneNN, and especially 0.02 m for ScanNet due to its intricate environment and various objects. In addition, for each superpoint, 128 points are subsampled for spatial distribution learning in a local network, and at most 512 superpoints are randomly selected for globally contextualization iterations in graph convolutions. We use the ADAM optimizer during the training process, with an initial rate of 0.01 and a decay rate of 0.7. Also, since we train the whole superpoint graph of a scene at a time, the batch size is reduced to 2 for S3DIS, SceneNN, and ScanNet, respectively. Additionally, evaluation metrics such as mean class intersection-over-union (mIoU), mean class accuracy (mAcc), and overall accuracy (OA) are expressed in proportional terms and are employed to quantitatively evaluate segmentation results.

In this case, we employ the preprocessed superpoints as our basic unit for deep learning such that geometrically and physically similar points belonging to the same category can be clustered into one superpoint, which boosts the embedding learning with more homogeneous points. Specifically, we construct our IAGC method with the architecture illustrated in Figure 6.

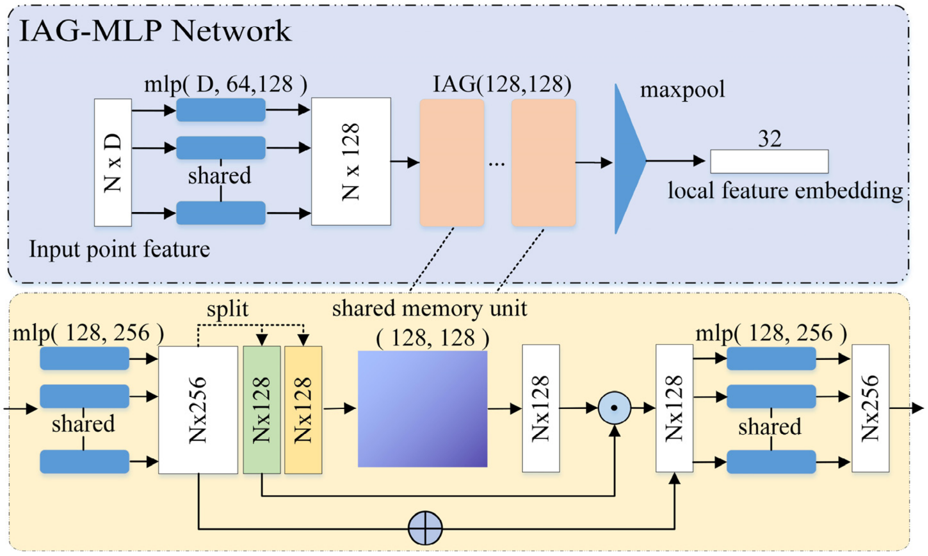


Figure 6. Implemented architecture of IAG–MLP.

4.3. Ablation Studies and Analyses

We conducted several ablation studies to validate the effectiveness of our proposed network by replacing the local embedding network and global aggregation work with the counterparts of current state-of-the-art networks and adjusting the amount of IAG blocks stacked in the IAG–MLP network. Moreover, we adjusted the granularity of the superpoint graph by preprocessing point clouds in different subsampling intervals and predefining the maximum number of superpoints for graph convolution to investigate the optimal graphical structure for deep learning.

4.3.1. Ablation Test of Local Embedding Function

To demonstrate the effectiveness of our local feature extraction network, we trained the models by stacking 1, 2, 3, 4, 5 IAG blocks in IAG–MLP, namely 1-IAG–MLP, 2-IAG–MLP, 3-IAG–MLP, 4-IAG–MLP, and 5-IAG–MLP. Then we compared their performance with that of two different networks, including the prevailing convolution network PointNet and the traditional self-attention network vanilla Transformer.

In addition, we adjusted the PointNet to a lightweight architecture adopted in SPG [16], which consists of a Transformer Network (i.e., T-Net, which is totally different from the vanilla Transformer Network), several sequential MLPs with a final 256-dimension feature, and a final max-pool layer with a 32-dimension feature vector. In the vanilla Transformer network, we used position encodings concatenated with other geometric features instead of adding them into input features.

Based on the corresponding numerical results in Table 1 (left panel), we can see that although PointNet achieved the best OA with 64.13%, it presents the lowest IoU compared with other networks in terms of local embedding blocks. Theoretically, large objects such as walls or floors commonly take up a great proportion of the building's interior, and the OA metric is generally dominated by large objects, containing a large quantity of points, while the mIoU is closely related to all categories. Consequently, we could conclude that dual attention-based networks are more sensitive to small targets than PointNet because most of the stacked IAG–MLP networks show better performance than the other two methods in mIoU, indicating better capability to distinguish multiple classes, especially suitable for indoor environments with complex architectural structures and distinctive equipment. Specifically, Figure 7a presents the semantic segmentation metrics curves in different stacked IAG–MLP networks, and they indicate that as the stacked IAG block increments, the IAG–MLP with fewer IAG blocks induced underfitting while the IAG–MLP with more IAG blocks shows overfitting, which contributes to the optimal 2-IAG–MLP outperforming the other networks in mIoU. On the other hand, the traditional vanilla Transformer underperformed stacked IAG–MLPs in both OA and mIoU in that the cross-attention utilized in IAG–MLP captures high-order relationships across channels other than self-attention.

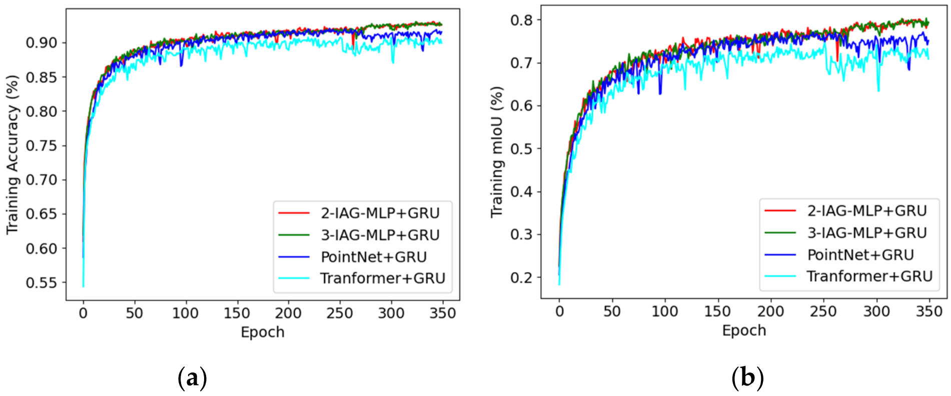


Figure 7. Training processes of the Transformer + GRU, PointNet + GRU, 2-IAG–MLP + GRU and 3-IAG–MLP + GRU models on S3DIS dataset in Area 4-fold. (**a**) Training Accuracy. (**b**) Training mean class intersection-over-union (mIoU).

Regarding computational complexity, we used Gflop to measure the number of floating-point operations 1 billion times per second during the training process. We can see that although the PointNet model has the largest number of parameters compared to Transformer and 2-IAG–MLP networks, it has the least computational complexity because the attention weight calculation in the attention-based network is more complex than the convolutional operation in PointNet.

Table 1. Ablation studies on the SceneNN dataset (testing on area 3 and training on the rest).

Method	Params	Gflops	OA	mIoU	mAcc	Method	Params	Gflops	OA	mIoU	mAcc
PoinNet [13]	189k	10.7	**64.13**	6.68	7.87	PointNet + GRU	289k	11.6	70.39	12.94	13.74
Transformer [19]	170k	16.6	61.88	6.71	7.29	Transformer + GRU	270k	23.1	69.38	14.16	**19.90**
1-IAG–MLP	87k	8.5	62.62	6.96	8.82	1-IAG–MLP + GRU	186k	10.2	71.44	12.77	16.73
2-IAG–MLP	153k	16.2	62.94	**7.57**	8.18	2-IAG–MLP + GRU	253k	17.5	**73.03**	**16.05**	17.73
3-IAG–MLP	219k	24.8	61.94	6.94	8.18	3-IAG–MLP + GRU	319k	27.0	71.00	14.14	18.35
4-IAG–MLP	286k	33.8	62.95	6.59	7.94	4-IAG–MLP + GRU	385K	34.1	70.83	14.46	18.60
5-IAG–MLP	352k	41.2	62.42	6.91	**8.95**	5-IAG–MLP + GRU	451K	43.4	71.55	14.04	16.60
						2-IAG–MLP + LSTM	255k	17.6	69.84	13.61	17.66
						2-IAG–MLP + GAT	190k	18.9	50.35	7.04	7.83

4.3.2. Ablation Test of Global Aggregation Function

In order to validate the effectiveness of the gating mechanism of RNN in the global aggregation block, we first integrated the three regional feature extraction networks mentioned above into GRU [23] for graph convolution. It can be seen from both the top right panel of Table 1 that all the local networks integrated with GRU for global aggregation obtained performance improvements, and in particular, the 2-IAG–MLP integrated with GRU, i.e., 2-IAG–MLP + GRU, achieved the best mIoU and OA performance, with 16.05% and 73.03%, respectively. Although Transformer + GRU presents a significant improvement in mAcc, its poor performance in mIoU makes it the worst in OA, indicating its great segmenting performance in particular categories but not all categories.

Secondly, we compared the simplified RNN module, namely GRU, with the more complicated RNN module, namely LSTM, to investigate the optimal graph network for global aggregation strategy. As we can see, although LSTM is exquisitely constructed with three gates to update the hidden state, 2-IAG–MLP + GRU with two gates achieved higher improvement than 2-IAG–MLP + LSTM by 3.19% and 2.44% in OA and mIoU. We also combined our IAG–MLP with other graphical strategies such as the Graph Attention Network (GAT) [62]. As it can be seen, in the original GAT, the concatenation of pairwise feature embeddings used in attentional weights calculation leads to a huge computational cost, and as a result, we replaced them with attributed edges in ECC for high efficiency. We can see that 2-IAG–MLP + GAT performed even worse than 2-IAG–MLP, mainly because indiscriminately assigning global attention weights of other superpoints to each superpoint in the superpoint graph may impede the feature representation, which further proves the necessity of the gating mechanism in LSTM and GRU for iteratively and selectively dropping irrelevant information and absorbing important information along the long sequence.

Furthermore, we selected the four best-performing networks in the global aggregation block to train on the S3DIS dataset to ulteriorly prove the generalization of our network. The specific training processes of Area 4-fold are depicted in Figure 7, from which we can indicate that the IAG–MLP-based methods best fit the data distribution, where they perform slightly better than PointNet + GRU in the first 250 epochs and continue to steadily grow while PointNet + GRU shows a great downtrend in the last 100 epochs. In addition, the testing results on Area 4 in Table 2 (left panel) show that as the training dataset increases, the gap between our IAGC and the other three networks is significantly wider, where the 2-IAG–MLP + GRU possesses the best performance in all metrics, especially more than 10.7% and 13.3% higher than the other three models in mIoU and mAcc, respectively. In addition, 6-fold cross validation results are also provided in Table 2 (bottom right panel) to prove that our proposed IAG–MLP-based method can achieve competitive accuracy with MLP-based methods and even outperform them.

Table 2. Ablation studies on the S3DIS dataset.

Method	Area 4-fold			Miro-Mean over 6-fold		
	OA	mIoU	mAcc	OA	mIoU	mAcc
Transformer + GRU	82.8	51.9	61.8	83.4	57.3	69.1
PointNet + GRU	82.5	52.4	63.3	84.6	58.6	70.2
IAGC(2-IAG–MLP + GRU)	**85.6**	**65.2**	**78.3**	**85.6**	**64.7**	**76.8**
IAGC(3-IAG–MLP + GRU)	83.5	54.5	65.0	84.7	58.8	70.8

4.3.3. Ablation Test of Granularity of Superpoint Graph

Due to the huge computation burden of the graph aggregation module with GRU for superpoints, it is necessary to explore the optimal number of superpoints and optimal graphical structure to implement the GRU module. Therefore, we analyzed the effect of the granularity of the superpoint graph on semantic segmentation results in the ScanNet dataset by implementing 2-IAG–MLP over the different subsampling sizes, partitioning granularities, and maximum numbers of superpoints for global aggregation. The Scannet dataset was split into 1201/156/156 scenes for training, validating, and testing. To be specific, we trained on the labeled point clouds for 50 epochs in different voxelization widths, regularization strengths, and max superpoints (denoted as v, μ, and max_sp), where v decides the subsampling interval and the number of points in each superpoint, and μ, quoted in the global energy function in Equation (3), dominates the number of superpoints in each superpoint graph, and max_sp describes the maximum number of superpoints for global aggregation operations. For instance, Figure 8 visualizes the preprocessing procedures of point clouds in geometric feature calculation, geometry-based and color-based partitioning, and global graph construction steps, which reorganize the data structure from the initial disordered point clouds to the final superpoint graphs. We should note that the geometric features such as scattering, planarity, and linearity are assigned to the red, green, and blue colors, respectively. The superpoints are randomly colored, and the grey lines illustrate the attributed edges in superpoint graphs.

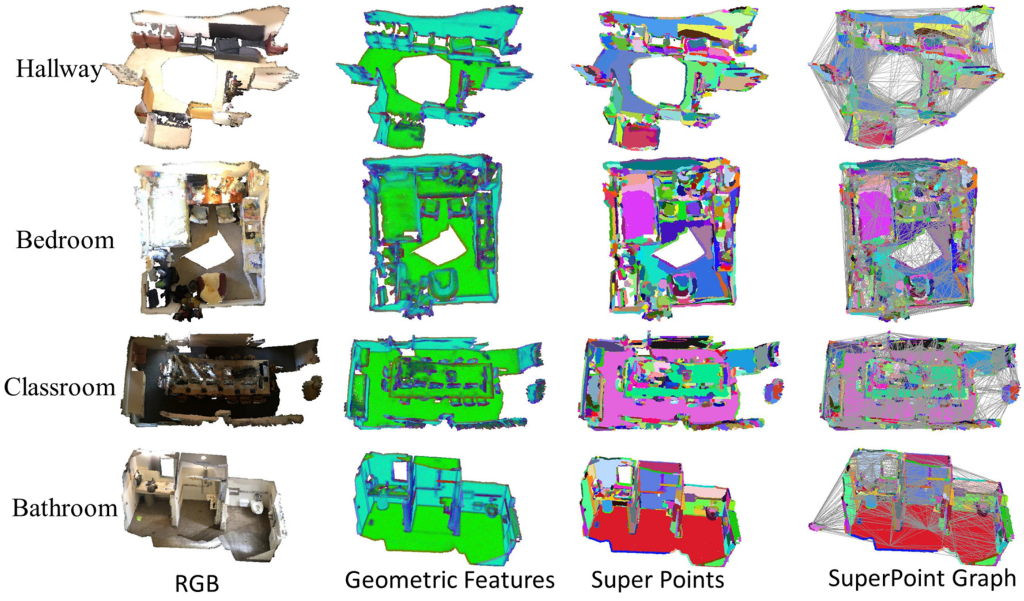


Figure 8. Visualization examples of preprocessing results on the ScanNet dataset with {v = 0.02, μ = 0.03} in different maximum number of superpoints for global aggregation.

On the other hand, as shown in Table 3, with the increase of the voxelization interval, the segmentation results with v = 0.03 decreased in comparison with their corresponding results with v = 0.02, which may underlie the fact that over subsampling may lead to insufficient points within superpoints for local embedding learning. In addition, partitioning with a larger μ increased the size of superpoints and mistakenly clustered points sharing similar characteristics but different labels into one superpoint, resulting in incorrect segmentation inference normally happening at the adjacent edge of two objects of different categories.

Table 3. Ablation studies on the ScanNet dataset (testing on 312 labeled scenes) for granularity of superpoint graphs.

v (m)	μ (m)	max_sp	OA	mIoU	mAcc
0.02	0.03	512	**79.2**	**51.0**	**75.4**
		1024	76.7	44.9	72.0
	0.05	512	74.8	39.3	71.7
		1024	74.8	37.8	70.1
0.03	0.03	512	77.1	46.9	74.1
		1024	69.2	30.0	62.5
	0.05	1024	73.8	35.7	70.8

Furthermore, in order to balance the cost of computation efficiency and segmentation accuracy, we investigated the performance of global aggregation with respect to the maximum number of superpoints. It should be noted that raw point clouds partitioned with $\mu > 0.05$ normally consist of less than 512 superpoints, and consequently, we only executed at most 1024 superpoints for global aggregation in $\{v = 0.02, \mu = 0.03\}$, $\{v = 0.02, \mu = 0.05\}$, and $\{v = 0.03, \mu = 0.03\}$, and at most 512 superpoints for other comparative experiments.

Notably, compared with experiments with W, those with max_sp = 512 achieved better segmentation results since excessive superpoints participating in global contextual information updating led to poor capability of validly retrieving available information during the limited iterations. As seen in Figure 9, we visualize the semantic segmentation results on the ScanNet dataset, which were trained at different maximum numbers of superpoints with max_sp = 512 and max_sp = 1024, respectively. As a result, the experiment with $\{v = 0.02, \mu = 0.03, \text{max\_}sp = 512\}$ achieved the best on all metrics, and we utilized its segmentation results to compare with several state-of-the-art methods on ScanNet in the following experiments.

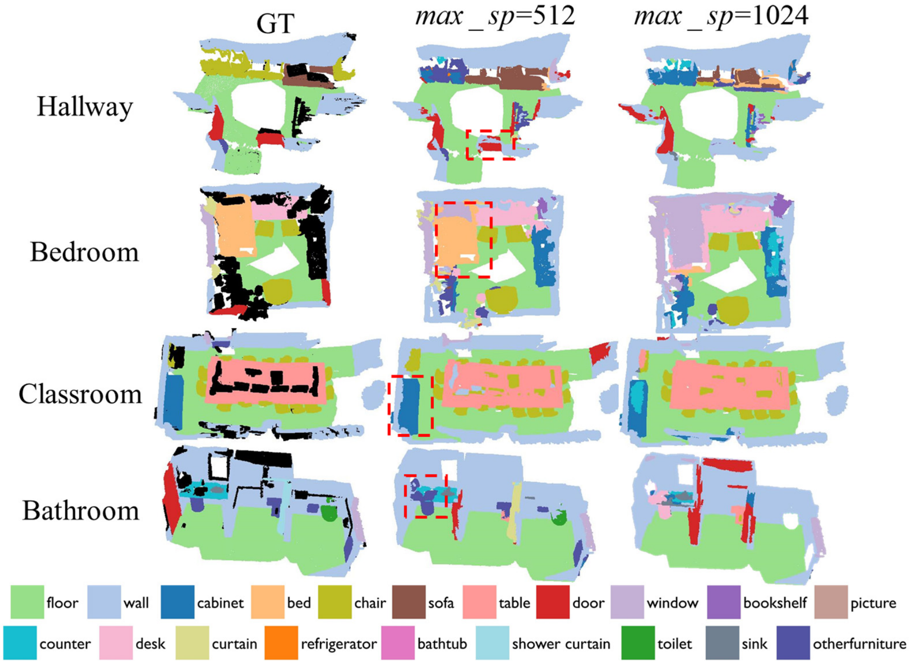


Figure 9. Visualization examples of segmenting results on the ScanNet dataset preprocessed with $\{v = 0.02, \mu = 0.03\}$ in different maximum number of superpoints for global aggregation.

4.4. Segmentation Results

In this section, we compared our proposed IAGC with several current state-of-the-art methods to further investigate and evaluate the segmenting performance of our network on two different open benchmarks, namely S3DIS and ScanNet.

4.4.1. Results on the S3DIS Dataset

With regard to the S3DIS dataset, we trained them with the same subsampled points (0.03 m) considering its dense and massive raw point clouds. We performed a 6-fold cross-validation across areas rather than buildings to prove the capability of our IAGC to distinguish multiple classes. The evaluation metrics of the miro-average results are average mIoU and average overall accuracy over 13 classes on raw point clouds.

We compared our IAGC with several existing state-of-the-art point cloud semantic segmentation methods, including PointNet [13], PointNet++ [14], SPG [16] and GAC [18]. Notably, PointNet and GAC only executed local networks with MLPs and graph attention convolution, respectively, while both PointNet++ and SPG implemented global aggregation with local embeddings derived from PointNet. As Table 4 shows, the mIoU of IAGC was higher by at least 17.9% compared with PointNet and GAC, which is attributed to the graph interaction among superpoints. Likewise, compared with the other two global networks, significantly increased mIoU, especially on objects of complex structure such as beams, bookcases, and sofas, are observed in IAGC with at least 12.3%, 7.2%, and 11.4% improvements. However, the mIoU of the board category is 20.9% and then 30.9% lower than PointNet's best performance. This huge performance gap in the board category probably derives from the initial data organization of point clouds as the PointNet is implemented in regular 3D voxel grids divided by spatial distribution.

Table 4. Semantic segmentation results of S3DIS (micro-mean over all 6-folds).

Method	OA	mIoU	Ceiling	Floor	Wall	Column	Bookcase	Beam	Wind	Door	Tab	Chair	Sofa	Board	Clutter
PointNet [13]	74.0	44.7	87.2	92.1	61.9	12.9	39.9	28.8	41.1	48.3	47.2	44.3	14.4	28.1	35.2
GAC [18]	79.4	46.8	90.1	86.5	67.4	12.7	34.1	17.2	44.5	51.2	58.5	54.8	16.1	25.9	49.5
PointNet++ [14]	81.1	56.9	91.7	92.4	71.3	15.6	51.5	29.6	56.0	57.5	62.2	65.3	45.1	51.8	50.0
SPG [16]	84.6	58.6	90.9	95.2	**74.3**	35.7	59.8	41.4	48.6	60.3	66.3	74.9	49.0	12.5	52.9
IAGC(Ours)	**85.6**	**64.7**	**93.2**	**95.4**	73.3	**39.8**	**67.0**	**53.7**	**64.6**	**61.0**	**74.2**	**79.4**	**60.4**	**20.9**	**58.3**

We further provided visualization examples of three scenes in different models. As seen in Figure 10, our proposed IAGC can predict more accurately in all building structures and most furniture, and shows distinct segmenting edges between different categories, while PointNet and PointNet++ present irregular shapes of segmented objects and mistakenly segmented points in discrete distributions, which results in ambiguous segmenting edges and low semantic segmentation performance.

4.4.2. Results on the ScanNet Dataset

As for the ScanNet dataset, we trained IAGC on the training dataset and submitted prediction results on an unlabeled test dataset to the testing server. Generally, those state-of-the-art methods submitted to the testing server are classified mainly by their convolution categories and input data types. As shown in Table 5, some of the networks such as 3DMV [63], PFCNN [64], and Tangent Convolution [65] retrieved semantic labels from 2D and 3D information, while the other networks listed below were trained with only 3D input data and can be divided into pointwise convolution and graph convolution. For convincing comparison, we adjusted the depth of the IAG–MLP network to ensure comparable capacity of the IAGC model with the point convolution baseline.

Obviously, there are more semantic categories in ScanNet than in S3DIS, which leads to the decline of the whole mIoU. However, our method merely leveraging 3D input data still attained a 53.4% mIoU score, achieving a significant performance gain of at least 5% compared to networks trained with both 2D and 3D information. Similarly, permanent building structures, which can also be seen in S3DIS, such as walls, floors, and doors, still maintained high performance. In addition, most furniture, including beds, cabinets, chairs, sofas, tables, and toilets, obtained an average IoU of 56.03%, outperforming the point convolution-based methods by a large margin (7.4%), and the other furniture, such as bathtubs, bookshelves, counters, and refrigerators, achieved competitive segmenting results compared with point convolution methods. In terms of graph-based methods, both our IAGC and SPG implemented global graph convolution with GRU but different local embedding strategies, and our IAGC was 8.3% higher than SPG in mIoU. In general, the gating mechanism coupled with channel attention interaction leads to an extra attention module to capture more spatial relationships than typical pointwise convolution with a channel-specific filter.

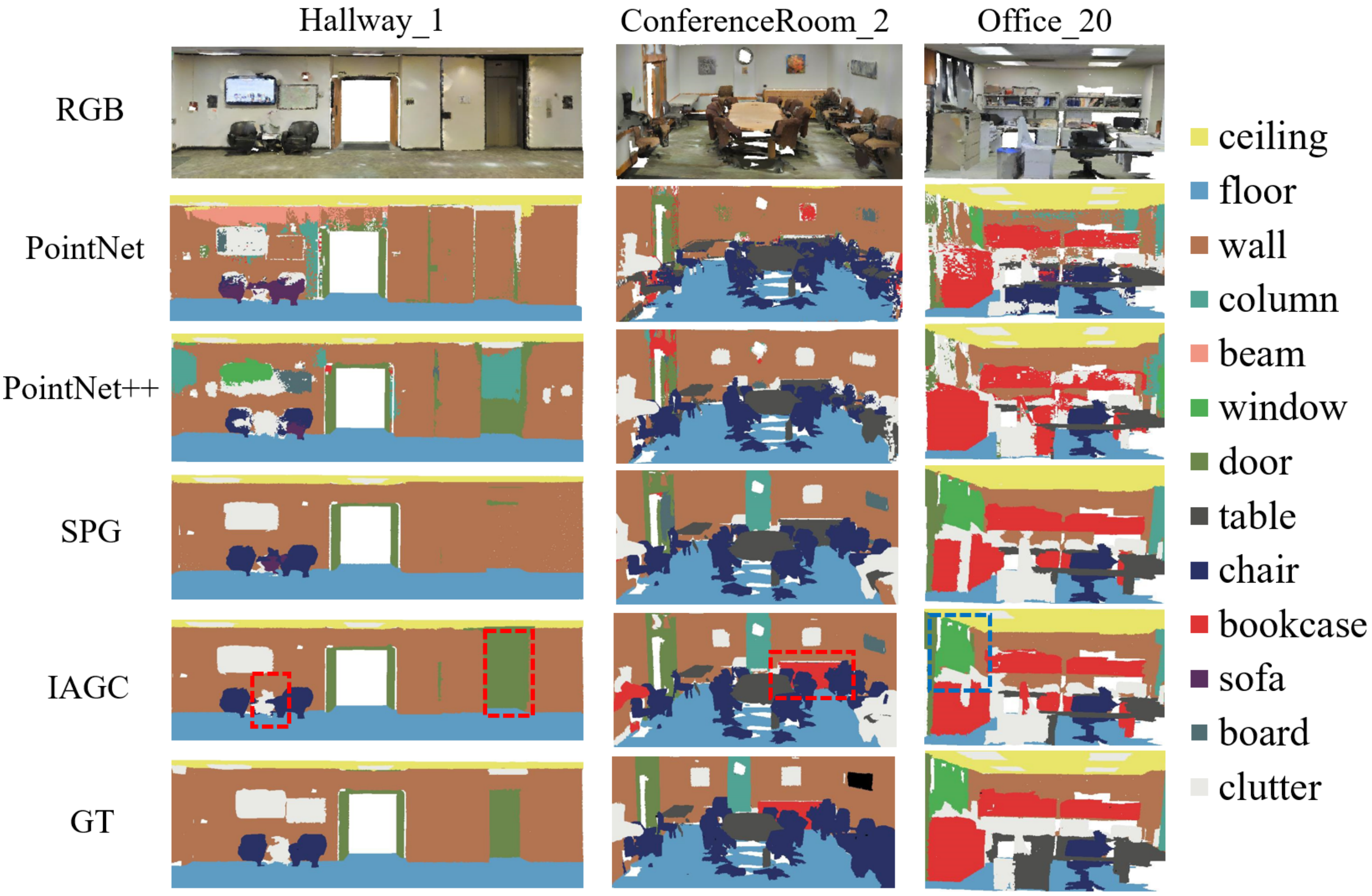


Figure 10. Visualization examples of three scenes (office, hallway and conference rooms) in Area 4 of the S3DIS dataset.

4.4.3. Results Analysis

Generally speaking, our proposed IAG–MLP presents competitive performance on the local representation embedding task compared with common pointwise convolution networks, and the combination of local IAG–MLP and global graph convolution networks even outperforms other MLP-based or Transformer-based networks on the semantic segmentation task. Accordingly, we attribute these improvements to two underlying factors. First of all, IAG–MLP executes an interactive-attention mechanism where the embeddings can be dominated by the augmented features from the combination of multiple feature channels in the dot-production enhanced procedure, which is beneficial to objects that show distinctive geometry-based and color-based characteristics (i.e., chairs, sofas, and tables). Because of the high-order features spread across the high-level feature channel, objects with similar geometry but different colors from the surrounding objects (i.e., board and window) and objects with similar color but different geometry (i.e., beam and column) can be clearly distinguished. In addition, the distinctive representations derived from the IAG–MLP network further facilitate interaction of contextual information among superpoints. In practice, it distinguishes the floor from the ceiling by considering the effect of the adjacent objects such as indoor furniture and surrounding walls. In contrast, the floor and ceiling may mutually enhance the segmentation of indoor furniture. Overall, our proposed IAGC generates more distinctive features for complicated objects and better handles permanent structures thanks to the interactive attention module.

Table 5. Semantic segmentation results of ScanNet (testing on unlabeled dataset).

Category	Method	mIoU	Bath	Bed	Shelf	Cab	Chair	Cntr	Curt	Desk	Door	Floor	Other	Pic	Fridg	Show	Sink	Sofa	Table	Toil	Wall	Wind
2D + 3D	3DMV [63]	48.4	48.4	53.8	64.3	42.4	60.6	**31.0**	**57.4**	43.3	**37.8**	79.6	30.1	**21.4**	**53.7**	20.8	47.2	50.7	41.3	69.3	60.2	53.9
	PFCNN [64]	44.2	50.5	62.2	38.0	34.2	65.4	22.7	39.7	36.7	27.6	92.4	24	19.8	35.9	26.2	36.6	58.1	43.5	64	66.8	39.8
	Tangent Conv [65]	43.8	43.7	64.6	47.4	36.9	64.5	35.3	25.8	28.2	27.9	91.8	29.8	14.7	28.3	29.4	48.7	56.2	42.7	61.9	63.3	35.2
Point Conv	PointNet++ [14]	33.9	58.4	47.8	45.8	25.6	36.0	25.0	24.7	27.8	26.1	67.7	18.3	11.7	21.2	14.5	36.4	34.6	23.2	54.8	52.3	25.2
	FCPN [66]	44.7	**67.9**	60.4	57.8	38.0	68.2	29.1	10.6	48.3	25.8	92.0	25.8	2.5	23.1	32.5	48.0	56	46.3	72.5	66.6	23.1
	PointCNN [32]	45.8	57.7	61.1	35.6	32.1	71.5	29.9	37.6	32.8	31.9	**94.4**	28.5	16.4	21.6	22.9	**48.4**	54.5	45.6	**75.5**	70.9	47.5
	ScanNet [59]	30.6	20.3	36.6	50.1	31.1	52.4	21.1	0.2	34.2	18.9	78.6	14.5	10.2	24.5	15.2	31.8	34.8	30	46.0	43.7	18.2
Graph Conv	SPG [16]	45.5	54.1	60.2	60.1	42.3	74.8	26.9	24.7	44.8	28.3	90.7	32.2	4.1	23.6	41.5	30.9	63.5	48.1	61.0	69.7	29.4
	2-IAG–MLP + GRU	**53.8**	49.5	**69.3**	**64.7**	47.1	**79.3**	30	47.7	**50.5**	35.8	90.3	**32.7**	8.1	47.2	**52.9**	44.8	**71.0**	**50.9**	74.6	**73.7**	**55.4**
	3-IAG–MLP + GRU	50.4	55.6	63.6	61.4	**47.7**	75.7	23.3	41.9	44	36.5	91.6	31.5	0.1	33.9	50.9	44.3	64.1	49.7	72.7	71.9	46.6

5. Conclusions and Discussion

In this paper, we present a novel 3D deep architecture for semantic segmentation in door scenes, named Interactive Attention-based Graph Convolution (IAGC). We first reorganized the raw point clouds into homogeneous superpoints based on geometry-based and color-based information to effectively reduce the computational complexity while retaining the characteristics of the objects in each superpoint to the greatest extent. At the same time, using superpoints as an input data unit may significantly extend the receptive field to obtain more rich information. Consequently, aiming to address the problem of insufficient local feature learning by PointNet, which is bedrock for most state-of-the-art networks, we proposed a dual attention module, Interactive Attention Gating MLP, namely IAG–MLP, that is oriented to fully capture high-level features in superpoints by both cross-position and cross-channel attention filters. Furthermore, we implemented another RNN architecture called GRU, which performs on the entire set of superpoints to extract global contextual information in order to update the local embedding of superpoints and boost the final semantically segmented inference. Lastly, extensive experiments on challenging open benchmarks show that our proposed method could be a potential local network with strong capability in stronger feature expressiveness for 3D point clouds. We hope that our work will inspire further investigation into the idea of augmenting the MLP architecture with an interactive attention mechanism, the design of superpoint-based networks, and instance or part segmentation. In addition, there are constraint rules about the entities in indoor environments, such as geometric characteristic intervals, topological relationships between entities, etc., which allow us to further investigate a smarter system to integrate deep learning networks with ontology-driven pipelines for adaptive segmentation of point clouds.

Author Contributions: All authors contributed to the manuscript and discussed the results; Ruoming Zhai developed the original idea and designed the deep learning network; Yifeng He conceived the experiments; Liyuan Meng finished the open data collection and download work; Jingui Zou performed data processing and contributed to the final revision of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by "National Natural Science Foundation of China" (Grant No. 41871373), "Guangxi Key Laboratory of Spatial Information and Geomatics" (Grant No. 19-185-10-13) and "Beijing Key Laboratory of Urban Spatial Information Engineering" (Grant No. 20210101).

Data Availability Statement: The data that support the findings of this study are available from the first and corresponding author, upon reasonable request.

Acknowledgments: The authors would like to thank the anonymous reviewers and editors for their valuable comments. We also thank the students who participated in the project.

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

References

1. Tang, P.; Huber, D.; Akinci, B.; Lipman, R.; Lytle, A. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Autom. Constr.* **2010**, *19*, 829–843. [CrossRef]
2. Pintore, G.; Mura, C.; Ganovelli, F.; Fuentes-Perez, L.; Pajarola, R.; Gobbetti, E. State-of-the-art in Automatic 3D Reconstruction of Structured Indoor Environments. *Comput. Graph. Forum* **2020**, *39*, 667–699. [CrossRef]
3. Xia, S.; Chen, D.; Wang, R.; Li, J.; Zhang, X. Geometric primitives in LiDAR point clouds: A review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 685–707. [CrossRef]
4. Lalonde, J.F.; Vandapel, N.; Huber, D.F.; Hebert, M. Natural terrain classification using three-dimensional ladar data for ground robot mobility. *J. Field Robot.* **2006**, *23*, 839–861. [CrossRef]
5. Golovinskiy, A.; Kim, V.G.; Funkhouser, T. Shape-based recognition of 3D point clouds in urban environments. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 2154–2161.
6. Guo, Y.; Sohel, F.; Bennamoun, M.; Lu, M.; Wan, J. Rotational projection statistics for 3D local surface description and object recognition. *Int. J. Comput. Vis.* **2013**, *105*, 63–86. [CrossRef]
7. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
8. Yin, W.; Kann, K.; Yu, M.; Schütze, H. Comparative study of CNN and RNN for natural language processing. *arXiv* **2017**, arXiv:1702.01923.
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
10. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
11. Bello, S.A.; Yu, S.; Wang, C. Review: Deep learning on 3D point clouds. *Remote Sens.* **2020**, *12*, 1729. [CrossRef]
12. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4338–4364. [CrossRef]
13. Qi, C.R.; Su, H.; Kaichun, M.; Juibas, 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; IEEE Computer Society: Los Alamitos, CA, USA, 2017.
14. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5099–5108.
15. Wang, C.; Samari, B.; Siddiqi, K. Local spectral graph convolution for point set feature learning. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 52–66.
16. Landrieu, L.; Simonovsky, M. Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
17. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3323–3332.
18. Wang, L.; Huang, Y.; Hou, Y.; Zhang, S.; Shan, J. Graph attention convolution for point cloud semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10296–10305.
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the NIPS'17: 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 6000–6010.
20. Guinard, S.; Landrieu, L. Weakly supervised segmentation-aided classification of urban scenes from 3D LiDAR point clouds. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *XLII-1/W1*, 151–157. [CrossRef]
21. Liu, H.; Dai, Z.; So, D.R.; Le, Q.V. Pay Attention to MLPs. *arXiv* **2021**, arXiv:2105.08050.
22. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2222–2232. [CrossRef] [PubMed]
23. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
24. Xiao, G.; Wang, H.; Lai, T.; Suter, D. Hypergraph modelling for geometric model fitting. *Pattern Recognit.* **2016**, *60*, 748–760. [CrossRef]
25. Truong, Q.H. Knowledge-Based 3D Point Clouds Processing. Ph.D. Thesis, Université de Bourgogne, Dijon, France, 2013.
26. Ponciano, J.J.; Roetner, M.; Reiterer, A.; Boochs, F. Object Semantic Segmentation in Point Clouds—Comparison of a Deep Learning and a Knowledge-Based Method. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 256. [CrossRef]
27. Qi, C.R.; Su, H.; Niebner, M.; Dai, A.; Yan, M.; Guibas, L.J. Volumetric and Multi-View CNNs for Object Classification on 3D Data. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.

28. Feng, Y.; Zhang, Z.; Zhao, X.; Ji, R.; Gao, Y. GVCNN: Group-View Convolutional Neural Networks for 3D Shape Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
29. Su, H.; Jampani, V.; Sun, D.; Maji, S.; Kalogerakis, E.; Yang, M.-H.; Kautz, J. SPLATNet: Sparse Lattice Networks for Point Cloud Processing. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
30. Du, J.; Jiang, Z.; Huang, S.; Wang, Z.; Su, J.; Su, S.; Wu, Y.; Ca, G. Point cloud semantic segmentation network based on multi-scale feature fusion. *Sensors* **2021**, *21*, 1625. [CrossRef]
31. Jiang, M.; Wu, Y.; Zhao, T.; Zhao, Z.; Lu, C. PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation. *arXiv* **2018**, arXiv:1807.00652.
32. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. *Adv. Neural Inf. Proc. Syst.* **2018**, *31*, 820–830.
33. Lin, Y.; Wang, C.; Zhai, D.; Li, W.; Li, J. Toward better boundary preserved supervoxel segmentation for 3D point clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 39–47. [CrossRef]
34. Hui, L.; Yuan, J.; Cheng, M.; Xie, J.; Zhang, X.; Yang, J. Superpoint network for point cloud oversegmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
35. Cheng, M.; Hui, L.; Xie, J.; Yang, J.; Kong, H. Cascaded Non-Local Neural Network for Point Cloud Semantic Segmentation. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 8447–8452.
36. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
37. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.; Koltun, V. Point transformer. *arXiv* **2020**, arXiv:2012.09164.
38. Guo, M.H.; Cai, J.X.; Liu, Z.N.; Mu, T.; Martin, R.; Hu, S. PCT: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [CrossRef]
39. Pan, X.; Xia, Z.; Song, S.; Li, L.; Huang, G. 3d object detection with pointformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7463–7472.
40. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11108–11117.
41. Wang, X.; He, J.; Ma, L. Exploiting Local and Global Structure for Point Cloud Semantic Segmentation with Contextual Point Representations. In Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–15 December 2019; pp. 4573–4583.
42. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph Neural Networks: A Review of Methods and Applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
43. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef]
44. Zhang, Z.; Cui, P.; Zhu, W. Deep Learning on Graphs: A Survey. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 249–270. [CrossRef]
45. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
46. Shuman, D.I.; Narang, S.K.; Frossard, P.; Ortega, A.; Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Proc. Mag.* **2013**, *30*, 83–98. [CrossRef]
47. Zhiheng, K.; Ning, L. PyramNet: Point cloud pyramid attention network and graph embedding module for classification and segmentation. *arXiv* **2019**, arXiv:1906.03299.
48. Luo, H.; Chen, C.; Fang, L.; Khoshelham, K.; Shen, G. Ms-rrfsegnet:f Multiscale regional relation feature segmentation network for semantic segmentation of urban scene point clouds. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8301–8315. [CrossRef]
49. Demantké, J.; Mallet, C.; David, N.; Vallet, B. Dimensionality based scale selection in 3D lidar point clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2011**, *38*, 97–102. [CrossRef]
50. Landrieu, L.; Obozinski, G. Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM J. Imaging Sci.* **2017**, *10*, 1724–1766. [CrossRef]
51. Santurkar, S.; Tsipras, D.; Ilyas, A.; Dry, A. How does batch normalization help optimization? In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; pp. 2488–2498.
52. Agarap, A.F. Deep learning using rectified linear units (relu). *arXiv* **2018**, arXiv:1803.08375.
53. Shazeer, N. Glu variants improve transformer. *arXiv* **2020**, arXiv:2002.05202.
54. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. *Int. Conf. Mach. Learn. PMLR* **2017**, *70*, 933–941.
55. Guo, M.H.; Liu, Z.N.; Mu, T.J.; Hu, S.M. Beyond Self-attention: External Attention using Two Linear Layers for Visual Tasks. In Proceedings of the 2021 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), A Virtual Event, 19 June 2021.
56. Simonovsky, M.; Komodakis, N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 3693–3702.

57. Jia, X.; De Brabandere, B.; Tuytelaars, T.; Gool, L. V Dynamic filter networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 667–675.
58. 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 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1534–1543.
59. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Niessner, M. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017; pp. 2432–2443.
60. Hua, B.S.; Pham, Q.H.; Nguyen, D.T.; Tran, M.K.; Yu, L.F.; Yeung, S.K. Scenenn: A Scene Meshes Dataset with annotations. In Proceedings of the International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016.
61. Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor Segmentation and Support Inference from RGBD Images. In *Computer Vision—ECCV 2012*; Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; pp. 746–760.
62. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
63. Dai, A.; Nießner, M. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 452–468.
64. Yang, Y.; Liu, S.; Pan, H.; Liu, Y.; Tong, X. PFCNN: Convolutional neural networks on 3d surfaces using parallel frames. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 13578–13587.
65. Tatarchenko, M.; Park, J.; Koltun, V.; Zhou, Q.Y. Tangent convolutions for dense prediction in 3d. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3887–3896.
66. Rethage, D.; Wald, J.; Sturm, J.; Navab, N.; Tombari, F. Fully-convolutional point networks for large-scale point clouds. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 596–611.