remote sensing

MDPI

Technical Note

SVASeg: Sparse Voxel-Based Attention for 3D LiDAR Point Cloud Semantic Segmentation

Lin Zhao [1], Siyuan Xu [1], Liman Liu [2,*], Delie Ming [1] and Wenbing Tao [1]

[1] National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
[2] School of Biomedical Engineering, South-Central University for Nationalities, Wuhan 430074, China
* Correspondence: limanliu@mail.scuec.edu.cn

Abstract: 3D LiDAR has become an indispensable sensor in autonomous driving vehicles. In LiDAR-based 3D point cloud semantic segmentation, most voxel-based 3D segmentors cannot efficiently capture large amounts of context information, resulting in limited receptive fields and limiting their performance. To address this problem, a sparse voxel-based attention network is introduced for 3D LiDAR point cloud semantic segmentation, termed SVASeg, which captures large amounts of context information between voxels through sparse voxel-based multi-head attention (SMHA). The traditional multi-head attention cannot directly be applied to the non-empty sparse voxels. To this end, a hash table is built according to the incrementation of voxel coordinates to lookup the non-empty neighboring voxels of each sparse voxel. Then, the sparse voxels are grouped into different groups, and each group corresponds to a local region. Afterwards, position embedding, multi-head attention and feature fusion are performed for each group to capture and aggregate the context information. Based on the SMHA module, the SVASeg can directly operate on the non-empty voxels, maintaining a comparable computational overhead to the convolutional method. Extensive experimental results on the SemanticKITTI and nuScenes datasets show the superiority of SVASeg.

Keywords: sparse voxel-based attention; LiDAR point cloud; semantic segmentation; SVASeg

Citation: Zhao, L.; Xu, S.; Liu, L.; Ming, D.; Tao, W. SVASeg: Sparse Voxel-Based Attention for 3D LiDAR Point Cloud Semantic Segmentation. *Remote Sens.* **2022**, *14*, 4471. https://doi.org/10.3390/rs14184471

Academic Editors: Martin Weinmann, Florent Poux and Eleonora Grilli

Received: 4 August 2022
Accepted: 6 September 2022
Published: 7 September 2022

1. Introduction

Scene perception is a very crucial task in computer vision which has a wide range of applications (e.g., autonomous driving and robotics). LiDAR is an indispensable device in modern autonomous driving vehicles. It captures precise and far distance measurements of the environments surrounding conventional visual cameras. The obtained measurements naturally form a 3D point cloud, which can be used to identify and locate dynamic objects and drivable areas. Therefore, LiDAR point cloud semantic segmentation is a crucial task for autonomous driving, which assigns a special semantic label for each point and provides point-wise perception information of the overall scene.

Previous LiDAR segmentation approaches can be roughly grouped into four main categories: point-based, projection-based, voxel-based and multi-view fusion-based methods. Point-based approaches [1–6] directly operate on point clouds and predict the semantic label of each point. Those methods generally apply point-based operators [5,7–9] (e.g., sampling, grouping and ordering) to extract semantic features from raw point clouds, but they are limited to adapting to the outdoor point cloud under the property of varying density and large range of scenes, and the large number of points also results in computational difficulties. Projection-based methods [10–13] project the LiDAR point clouds into a 2D space (e.g., range image and bird-eye-view images) so that 2D convolutions can be used to process them. However, those projection-based methods cannot completely model the geometric information because the original topology will inevitably be lost or altered during the 3D-to-2D projection process.

Voxel-based methods [14,15] rasterize LiDAR point clouds into voxels and then apply 3D convolutions to extract features. Those methods are computational expensive and entail high memory consumption. Recently, more efficient sparse convolutions [16–18] have been proposed to accelerate 3D convolution and can achieve state-of-the-art segmentation performance. Multi-view fusion-based methods [17,19,20] combine multiple different operations (i.e., voxel-based, projection-based and/or point-wise operations) to segment point clouds, and show promising results.

Sparse convolution is a crucial operation in most segmentation models [20,21] that include voxel-based operation. Although these models have the advantage of efficiency, they cannot efficiently capture large amounts of context information, resulting in limited receptive fields and unsatisfactory performance. The receptive field of sparse convolution is related to voxel size, kernel size, stride size and layers. When trading off performance and resource consumption, it is difficult to directly increase these parameters to obtain larger receptive fields. Compared with convolutional neural networks, the transformer has showed its superiority and achieved promising results in most 2D vision tasks [22,23] and 3D object detection [24,25], because it can model the long-range relationships between pixels by self-attention and multi-head attention.

Motivated by above findings and inspired by VoTr [24] for 3D object detection, a sparse voxel-based attention network is proposed for LiDAR semantic segmentation (SVASeg). The SVASeg is mainly composed of multiple submanifold sparse convolution layers, multiple sparse inverse convolution layers and a sparse, voxel-based multi-head attention module (SMHA). For the key component, SMHA, according to the increment of voxel coordinates, a hash table is built to lookup the non-empty neighbor voxels for each voxel. Then, all sparse voxels can be grouped into different local groups. For each group, we perform position embedding, multi-head attention and feature fusion to capture the context information and enlarge the receptive fields. The SMHA only focuses on the non-empty voxels in a local region and maintains a comparable computational overhead to the convolutional method. Experimental results on SemanticKITTI and nuScenes datasets showed the superiority of SVASeg.

2. Related Work

In this section, we briefly review existing works related to our approach: LiDAR semantic segmentation and the transformer for point clouds. We mainly focus on the LiDAR-only methods.

2.1. LiDAR Semantic Segmentation

As the public datasets [26,27] of outdoor scenes increase in size and number, LiDAR semantic segmentation research is developing. These methods are grouped into four categories, including point-based, projection-based, voxel-based and multi-view fusion-based methods.

Point-based methods directly learn the point features based on the raw point clouds through point-based operators [5,7–9] (e.g., sampling, grouping and ordering). KPConv [5] uses kernel point convolution which utilizes kernel-points to convolve local point sets. ASAP-Net [28] designs a flexible module as soon as possible to improve spatio-temporal point cloud feature learning by considering both attention and structure information across frames. PointASNL [29] proposes an adaptive sampling module to re-weight the neighbors around the initial sampled points via farthest point sampling. S-BKI [30] develops a Bayesian, continuous 3D semantic occupancy map of point clouds by generalizing the Bayesian kernel inference model. PointNL [31] aims to build the long-range dependencies of point clouds from the neighborhood-level, superpoint-level and global-level. To capture and represent implicit geometric structures of point cloud, STPC [32] introduces a spatial direction dictionary to learn those latent geometric components and designs a sparse deformer to transform unordered neighbor points into the canonical ordered dictionary space by using direction dictionary learning. RandLA-Net [1] introduces a lightweight

architecture for large-scale point clouds by using random sampling instead of complex sampling approaches. Based on RandLA-Net, MSAAN [33] proposes a multi-scale attentive aggregation network to achieve the global consistency of point cloud feature representation. However, these methods which mainly focus on indoor point cloud are limited to adapting to the outdoor point cloud under the conditions of varying density and a large range of scenes, and the large number of points also results in the computational difficulties for these methods when shifting from indoor to outdoor settings.

Projection-based methods project the input point clouds to a 2D pseudo-image. Then, a 2D convolutional neural network is used to process the pseudo-image. RangeNet++ [34], SqueezeSegV3 [11], TemporalLidarSeg [35], SalsaNext [10], KPRNet [12] and Lite-HDSeg [36] utilize the spherical projection mechanism to map the raw point clouds into a range image, and an encoder–decoder network is applied to the range image to obtain semantic information. For instance, to tackle the feature distribution of drastic LiDAR image changes at different image locations, SqueezeSegV3 [11] uses spatially-adaptive convolution to adopt different filters for different locations according to the input image. SalsaNext [10] introduces a new context module, consisting of a residual dilated convolution stack fusing receptive fields at various scales, for the uncertainty-aware semantic segmentation of a LiDAR point cloud. Lite-HDSeg [36] is a new encoder–decoder architecture with light-weight harmonic dense convolutions as its core. PolarNet [13] projects the raw point cloud into a polar bird's-eye view (BEV). However, the original topology of point clouds will inevitably be lost or altered during projection, resulting in projection-based methods failing to completely model the geometric information.

Voxel-based methods rasterize the raw point clouds into voxels, and then apply vanilla 2D or 3D convolutions to generate LiDAR semantic segmentation results. Recently, more efficient works [16,17] have been proposed to accelerate the 3D convolution and reduce the memory consumption. Following the previous works [16,17], MinkNet42 [21] and PCSCNet [37] achieved better semantic segmentation results on outdoor scenarios. Among them, PCSCNet [37] is a fast semantic segmentation model based on the voxel-based point convolution and 3D sparse convolution. Furthermore, Cylinder3D [18] groups the raw point cloud into the cylindrical partitions and designs an asymmetrical residual block to further reduce computation and improve the segmentation performance.

Multi-view fusion-based methods construct the LiDAR segmentation model by using the combination of voxel-based, projection-based and/or point-wise operations. To capture richer semantic information, some methods [15,19,38–42] fuse two or more different views together. For instance, [39,40] fused the point-wise semantic information from a bird's-eye view and a range image in the early stage, and then fed it into a 3D detector to obtain the detection results. AMVNet [38] fuses the outputs of different views in a late stage. PVCNN [15] and FusionNet [41] utilize point–voxel fusion strategies to achieve better LiDAR segmentation performance. However, the performances of these methods are also limited due to the lack of rich contextual information.

2.2. Transformer in Point Cloud

A transformer can model the long-range relationships between pixels by self-attention and multi-head attention. It has achieved promising results in most 2D vision tasks [22,23,43] and in 3D object detection [24,25]. As for 3D semantic segmentation, some works [44–47] applied the point-based transformer to point clouds for indoor scene semantic segmentation. However, these methods cannot be used for outdoor LiDAR segmentation due to the inherent properties of LiDAR points (e.g., sparsity and varying density). STPC [32] uses spatial transformer point convolution to tackle the semantic segmentation of both indoor and outdoor scenes, but its segmentation performance is unsatisfactory.

3. Proposed Method

3.1. Network Architecture

In this section, we describe the whole network architecture of SVASeg for LiDAR point cloud semantic segmentation. As illustrated in Figure 1, the whole network is an encoder–decoder architecture which mainly contains four encoding layers, four decoding layers and a sparse voxel-based multi-head attention module. For each encoding layer, four submanifold sparse convolution layers are used to encode and down-sample the input sparse features. For the decoding layer, we firstly use four sparse inverse convolution layers to recover the spatial resolutions of the sparse features. The decoded features and corresponding encoded features are fused together by concatenation to further refine the fused features and improve its discrimination. Specially, after two successive decoding layers, a sparse voxel-based multi-head attention module (described in Section 3.2) is applied to the sparse features to capture the contextual information and enlarge the receptive fields for better LiDAR semantic segmentation.

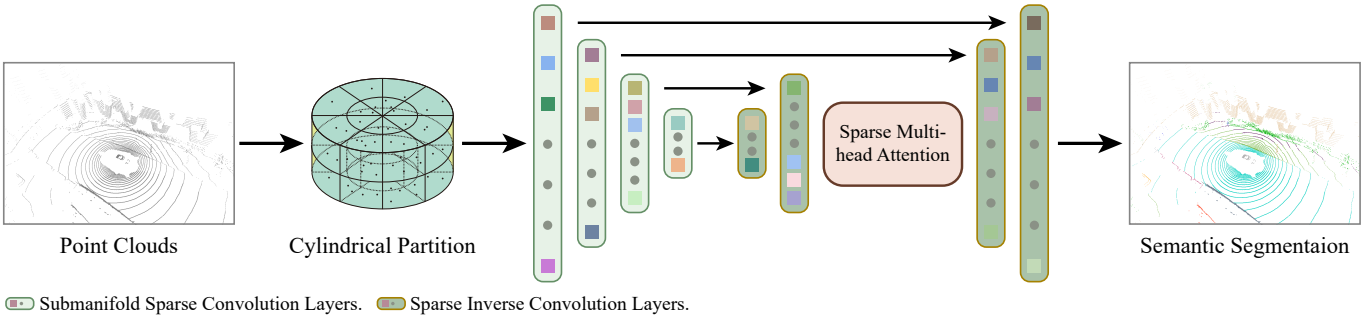

Figure 1. The network architecture overview of our SVASeg. LiDAR point clouds are firstly voxelized into cylindrical partitions as sparse features. Then, multiple submanifold sparse convoluation layers, sparse inverse convolution layers and a sparse multi-head attention module are used to process the sparse features and generate the point-wise semantic predictions. The voxel-based sparse multi-head attention is described in Section 3.2.

For the whole pipeline, our SVASeg takes a LiDAR point cloud as input. Then, following the studies [18,20], the raw point clouds are transformed into a cylinder coordinate system and further voxelized into the cylindrical partitions as the input sparse features. Subsequently, the feature encoder and decoder are used to process the sparse features and generate sparse semantic features. Then, those sparse semantic features are converted to a dense cylindrical representation $\mathcal{F}_{dense} \in \mathbb{R}^{B\times C\times H\times W\times L}$, where B is batch size; C denotes the number of feature dimensions; and H, W and L indicate the radius, angle and height, respectively. Finally, the point-wise semantic predictions can be obtained by applying a simple argmax operation and the voxelized inverse indexes to the dense semantic features.

During training, the classical cross-entropy loss function $\mathcal{L}_{ce}$ is used to supervise the learning of our network, SVASeg. The $\mathcal{L}_{ce}$ is a voxel-wise loss and used to maximize the point accuracy. Following previous works [18,20], the lovasz–softmax loss function [48] is also taken as an auxiliary loss $\mathcal{L}_{aux}$ to maximize the intersection-over-union score. Therefore, the total training loss of our network is

$$\mathcal{L}_{loss} = \mathcal{L}_{ce} + \mathcal{L}_{aux}. \tag{1}$$

3.2. Sparse Voxel-Based Multi-Head Attention

The transformer has been widely used in various 2D vision tasks and achieves promising results, because it can build the long-range relationships between pixels by self-attention and multi-head attention. However, it is difficult to directly apply a standard tranformer module to non-empty voxels due to its sparsity. Inspired by VoTr [24] for 3D object detection, a multi-head attention module (depicted in Figure 2) is adapted to sparse non-empty

voxels to capture the contextual information and enlarge the receptive fields for better LiDAR semantic segmentation.

Grouping. Given a voxel set $V = \{v_i \mid i = 1, 2, \ldots, N\}$ with N non-empty voxels and its indices I and spatial shape S, we firstly build a hash table for each querying voxel v_i according to the incrementation of voxel coordinate $(\triangle v_{i,x}, \triangle v_{i,y}, \triangle v_{i,z})$ and a specific hash size K. For example, given the incrementation of voxel coordinate $(\triangle v_{i,x}, \triangle v_{i,y}, \triangle v_{i,z}) = \{(0,0,0), (1,0,0), \ldots, (5,5,4), (5,5,5)\}$, we can search K non-empty neighbor voxel indices for v_i. The new indices can be obtained by $(v_{i,x} \pm \triangle v_{i,x}, v_{i,y} \pm \triangle v_{i,y}, v_{i,z} \pm \triangle v_{i,z})$, and the indices of K non-empty neighbor voxels are added into the hash table. In addition, the dimension K is also used for the position embedding. Afterwards, we can lookup the non-empty neighbor voxels $V_i = \{v_{ij} \mid j = 1, 2, \ldots, K\}$ from the hash table. Thus, the geometry coordinates $\mathcal{G}_{sparse} \in \mathbb{R}^{N \times 3}$ and the features $\mathcal{F}_{sparse} \in \mathbb{R}^{N \times C}$ of the sparse voxels can be divided into different groups and generate $\mathcal{G}^g_{sparse} \in \mathbb{R}^{N \times 3 \times K}$ and $\mathcal{F}^g_{sparse} \in \mathbb{R}^{N \times C \times K}$, respectively. Each group corresponds to a local region, and K is the number of non-empty voxels in the neighborhood of centroid voxels.

Position Embedding. In a transformer, position embedding can effectively capture the position information of each element. In this work, the relative position embedding is used because the multi-head attention will be performed in a local region. Specifically, the relative geometry coordiantes can be obtained as follows:

$$\mathcal{G}^r_{sparse} = \mathcal{G}^g_{sparse} - \phi(\mathcal{G}_{sparse}), \tag{2}$$

where $\phi(\cdot)$ extends an axis in the last of $\mathcal{G}_{sparse}$. Afterwards, a linear projection function $\varphi(\cdot)$ is applied to the relative coordiantes $\mathcal{G}^r_{sparse}$ to generate high-dimensional embedding features. The embedding features are further fused with the grouped sparse features:

$$\mathcal{F}^{g,e}_{sparse} = \mathcal{F}^g_{sparse} + \varphi\left(\mathcal{G}^r_{sparse}\right). \tag{3}$$

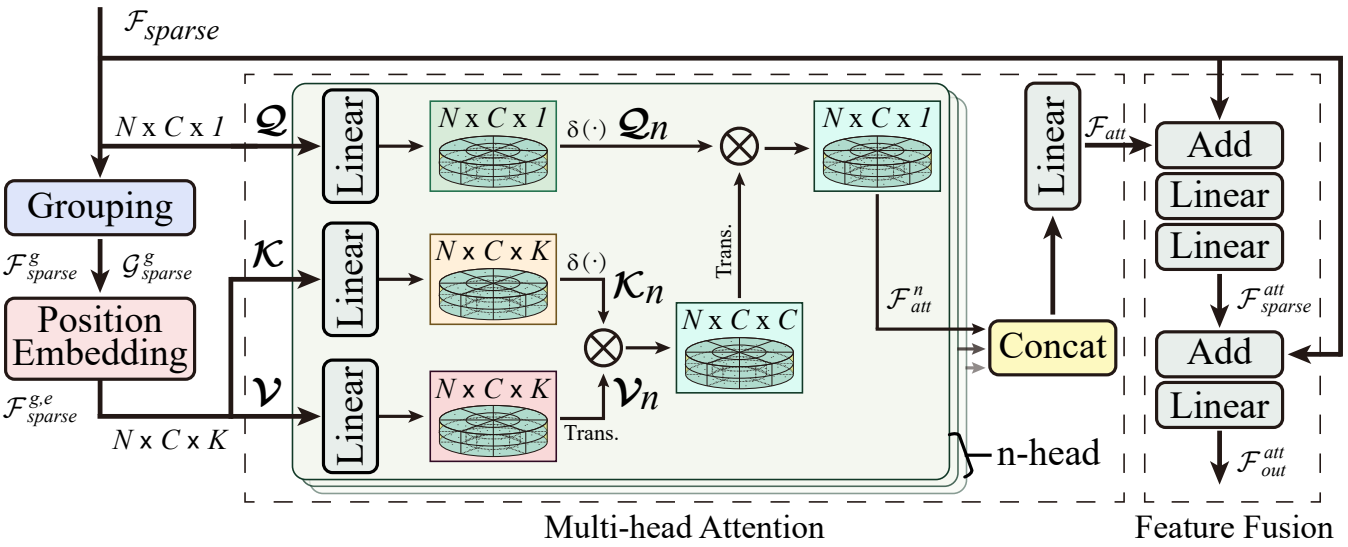

Figure 2. The sparse voxel-based multi-head attention. $\mathcal{Q}$, $\mathcal{K}$ and $\mathcal{V}$ indicate the query features, key features and value features, respectively. $N \times C \times K$ is the shape of $\mathcal{K}$ and $\mathcal{V}$, where N, C and K indicate the number of all non-empty voxels, the feature dimension and the number of local non-empty neighbor voxels, respectively. $\delta(\cdot)$ is the softmax normalization function. Trans. represents the transpose operation of a tensor in the last two dimensions. The n-head is a parameter of multi-head attention.

Multi-head Attention. The multi-head attention is responsible for modeling the long-range relationships between non-empty voxels and aggregating the context information in a local region for better segmentation, which is a key component of the sparse voxel-based multi-head attention module. After getting the query features $\mathcal{Q} = \varphi(\mathcal{F}_{sparse})$, key features $\mathcal{K} = \mathcal{F}^{g,e}_{sparse}$ and value features $\mathcal{V} = \mathcal{F}^{g,e}_{sparse}$, $\mathcal{Q}$, $\mathcal{K}$ and $\mathcal{V}$ are projected and generate corresponding multi-head features:

$$\mathcal{Q}_n = \varphi_q(\mathcal{Q}), \quad \mathcal{K}_n = \varphi_k(\mathcal{K}), \quad \mathcal{V}_n = \varphi_v(\mathcal{V}), \tag{4}$$

where φ_q, φ_k and φ_v are the linear projection functions. n is the number of multi-head. Following the work [49], the voxel-based multi-head attention can by formulated as:

$$\mathcal{F}^n_{att} = \frac{\delta(\mathcal{Q}_n)}{\sqrt{d}}\left(\frac{\delta(\mathcal{K}_n)^T}{\sqrt{d}}\mathcal{V}_n\right), \tag{5}$$

where $\delta(\cdot)$ is the softmax normalization function. d is the number of query feature channels. The features of all heads are fused together by using a concatenation operation and a linear fusion function φ_f:

$$\mathcal{F}_{att} = \varphi_f\left(\left[\mathcal{F}^1_{att}, \mathcal{F}^2_{att}, \ldots, \mathcal{F}^n_{att}\right]\right), \tag{6}$$

where $[\cdot]$ is the concatenation operation. The sparse voxel-based multi-head attention is directly performed on sparse non-empty voxels, which is an efficient attention mechanism due to its extension from an approximate linear 2D transformer [49].

Feature Fusion. After aggregating the contextual information of sparse voxels by using the voxel-based multi-head attention, two shortcut connections are used to speed up the convergence of our segmentation network. Specifically, the attention features $\mathcal{F}_{att}$ are firstly fused with the sparse features $\mathcal{F}_{sparse}$ by element-wise addition. Then, two linear fusion functions are applied to the fused features to refine it:

$$\mathcal{F}^{att}_{sparse} = \varphi_f\Big(\varphi_f\left(\mathcal{F}_{att} + \mathcal{F}_{sparse}\right)\Big). \tag{7}$$

Afterwards, the sparse features $\mathcal{F}_{sparse}$ are added with the $\mathcal{F}^{att}_{sparse}$, and a linear fusion function is applied to the fused features to further refine it once again and generate the final attention features $\mathcal{F}^{att}_{out}$:

$$\mathcal{F}^{att}_{out} = \varphi_f\Big(\mathcal{F}^{att}_{sparse} + \mathcal{F}_{sparse}\Big). \tag{8}$$

The obtained attention features $\mathcal{F}^{att}_{out}$ will be further processed by the following sparse inverse convolution layers in the decoder structure.

4. Experiments

Our proposed SVASeg was evaluated on the large-scale LiDAR semantic segmentation datasets SemanticKITTI [26] and nuScenes [27] to demonstrate its effectiveness. We firstly provide a brief introduction to the dataset and evaluation metrics in Section 4.1. Then, Section 4.2 presents the implementation details of our method. Subsequently, we exhibit the detailed experiments and the comparisons with other methods on the SemanticKITTI and nuScense datasets in Sections 4.3 and 4.4. Finally, we show ablation study experiments with various numbers of hash size in Section 4.5.

4.1. Datasets and Evaluation Metrics

The SemanticKITTI [26] dataset was collected from the KITTI Vision Benchmark and contains 22 sequences involving autonomous driving scenarios. According to the official settings, sequences from 00 to 10 should be used for training (19,130 frames) and validation (sequence 08, 4071 frames), and sequences from 11 to 21 are the test split (20,351 frames). All

semantic labels on the testing split are unavailable. Each scan in the dataset contains more than 100,000 points with pointwise semantic labels of 28 classes. After merging similar categories and ignoring rare classes, a total of 19 classes remained for the task of LiDAR point cloud semantic segmentation.

The nuScenes [27] dataset is another large-scale dataset for autonomous driving which contains more than 1000 scenes collected from different areas of Boston and Singapore. This dataset has 28,130 training frames and 6019 validation frames. nuScenes provides up to 32 classes of annotations. After merging similar classes, a total of 16 classes remained for the LiDAR semantic segmentation. Furthermore, this dataset has the property of class imbalance. Specifically, cars and pedestrians are the most frequent categories, whereas bicycles and construction vehicles have limited training data. Moreover, the challenge of the nuScenes dataset also comes from the fact that it was collected at different locations and with different diverse weather conditions. Compared to the SemanticKITTI dataset, the point clouds of nuScenes are also less dense, because its sensor (Velodyne HDL-32E) has fewer beams and lower horizontal angular resolution.

Mean intersection over union ($mIoU$) is a standard evaluation metric for semantic segmentation tasks. In this work, the mIoU over all classes was taken as the evaluation metric to evaluate the LiDAR segmentation performance of our proposed approach. It can be formulated as

$$mIoU = \frac{1}{C}\sum_{i=1}^{C} IoU_i, \tag{9}$$

$$IoU_i = \frac{p_{ii}}{p_{ii} + \sum_{j \neq i} p_{ij} + \sum_{k \neq i} p_{ki}}, \tag{10}$$

where p_{ij} is the number of points that belong to class i and are predicted to be class j, and C is the number of classes.

4.2. Implementation Details

For the SemanticKITTI [26] and nuScenes [27] datasets, following the works [18,20], the LiDAR point clouds were split into cylindrical partitions with the size $480 \times 360 \times 32$, where three dimensions are the radius, angle and height, respectively. Then, we followed the procedure of [18] to construct a UNet-like structure [50] with submanifold sparse convolution and sparse inverse convolution. Considering the balance between segmentation performance and computation and memory consumption, we only utilized a sparse voxel-based multi-head attention module after the second decoding layer. The number of specific hash size K was set to 32, the number of n-heads was set to 4, and 256 input channels were used for the sparse multi-head attention. During the training, the proposed SVASeg was trained along with the whole framework with the ADAM optimizer with initial learning rate set to 0.001, and batch size was set to 2 for 40 epochs on a single NVIDIA RTX3090 GPU. Our proposed SVASeg was implemented based on the deep learning framework PyTorch.

4.3. Evaluation on the SemanticKITTI Dataset

Following most previous works, we conducted experiments on SemanticKITTI [26] to evaluate the performance of our SVASeg. Table 1 reports the segmentation results of our SVASeg and other LiDAR segmentation methods on the validation set of the SemanticKITTI dataset. From Table 1, we can see that our proposed method outperforms the point-based methods (e.g., RandLANet [1]) and projection-based methods (e.g., SqueezeSegV3 [11] and SalsaNext [10]) with a large margin. Compared to the voxel-based method Cylinder3D [18], the multi-view fusion-based method AMVNet [38] and the multi-modality fusion-based method PMF [51], our approach achieved better segmentation results. Qualitative results of LiDAR segmentation are presented in Figure 3.

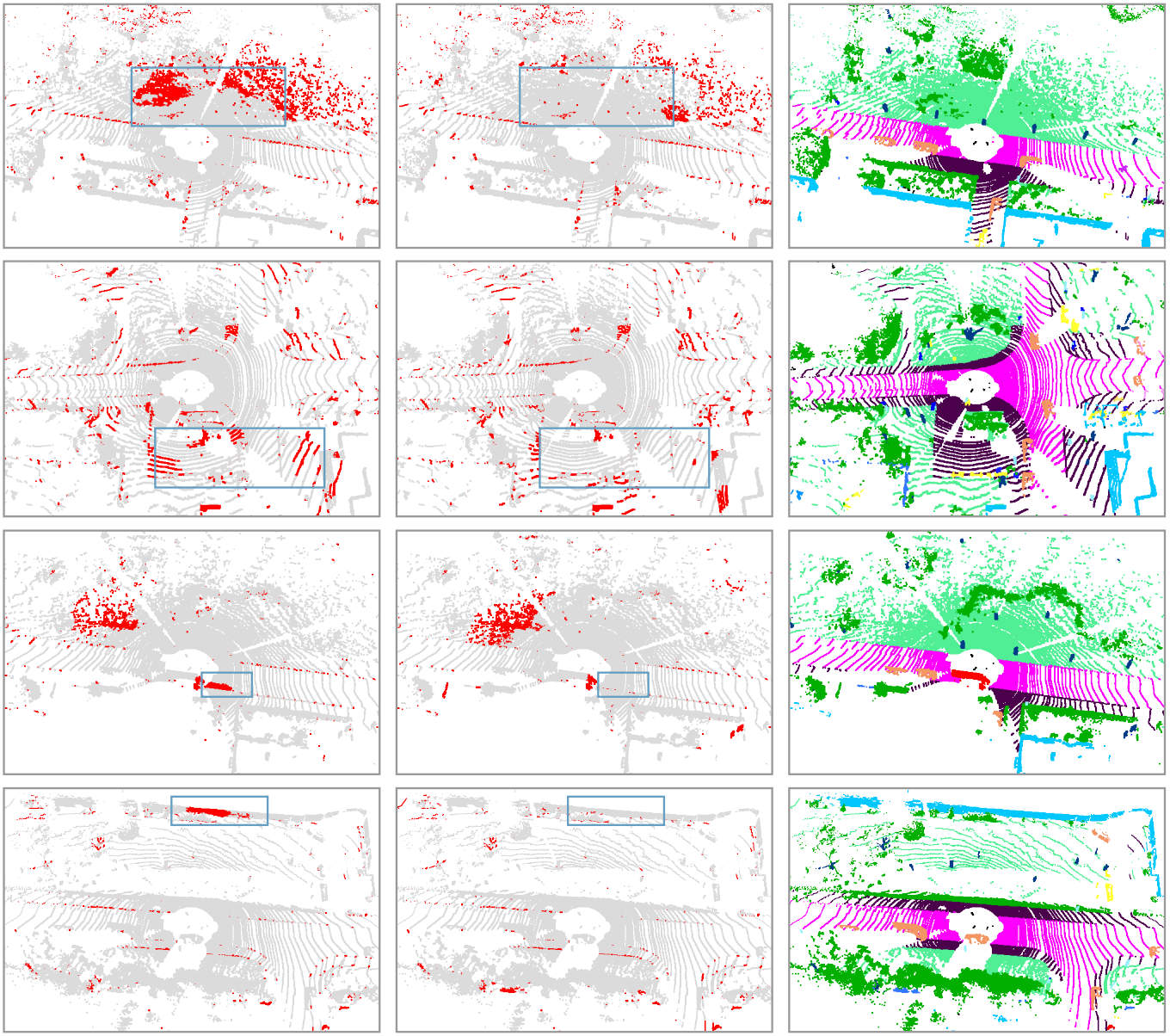

Figure 3. Compared to Cylinder3D, our method had less error (shown in red) when recognizing the surface region on the SemanticKITTI dataset's validation set, thanks to the voxel-based sparse multi-head attention module. Best viewed in color.

We also submit our segmentation results on the SemanticKITTI evaluation server. Table 2 provides the detailed class-wise quantitative results of our SVASeg and other state-of-the-art methods on the SemanticKITTI LiDAR semantic segmentation challenge. From Table 2, we can see that our proposed SVASeg achieved better segmentation performance than most state-of-the-art methods and dominated greatly in many categories. Specifically, compared to the point-based methods, including PointNL [31], STPC [32], RandLANet [1] and KPConv [5], our method outperformed the point-based methods and significantly improved the performances of LiDAR semantic segmentation. Compared to the projection-based methods, including RangeNet++ [34], SqueezeSegV3 [11], KPRNet [12], SalsaNext [10] and Lite-HDSeg [36], our proposed SVASeg achieved an about 1.4∼13.0% performance gain in terms of mIoU due to the 3D geometric information lost in projection-based methods. Compared to some voxel-based methods (e.g., PolarNet [13],

MinkNet42 [21] and PCSCNet [37]) and multi-view fusion-based methods (e.g., FusionNet [41], TORANDONet [19] and SPVCNN [17]), the proposed method also performs better than these LiDAR semantic segmentation methods. The sparse voxel-based multi-head attention module is a plug-and-play module, which could be applied to other voxel-based methods to achieve further performance improvements. These results demonstrate the effectiveness and superiority of our proposed SVASeg.

Table 1. Experimental results of our proposed SVASeg and other LiDAR segmentation methods on the SemanticKITTI dataset validation set. All results were obtained from the literature. Best and second best results are **bolded** and underlined.

Methods	*mIoU*	Car	Bicycle	Motorcycle	Truck	Other-Vehicle	Person	Bicyclist	Motorcyclist	Road	Parking	Sidewalk	Other-Ground	Building	Fence	Vegetation	Trunk	Terrain	Pole	Traffic-Sign
#Points (k)	-	6384	44	52	101	471	127	129	5	21434	974	8149	67	6304	1691	20391	882	8125	317	64
RandLANet [1]	50.0	92.0	8.0	12.8	74.8	46.7	52.3	46.0	0.0	93.4	32.7	73.4	0.1	84.0	43.5	83.7	57.3	73.1	48.0	27.3
RangeNet++ [34]	51.2	89.4	26.5	48.4	33.9	26.7	54.8	69.4	0.0	92.9	37.0	69.9	0.0	83.4	51.0	83.3	54.0	68.1	49.8	34.0
SequeezeSegV3 [11]	53.3	87.1	34.3	48.6	47.5	47.1	58.1	53.8	0.0	95.3	43.1	78.2	0.3	78.9	53.2	82.3	55.5	70.4	46.3	33.2
MinkowskiNet [21]	58.5	95.0	23.9	50.4	55.3	45.9	65.6	82.2	0.0	94.3	43.7	76.4	0.0	87.9	57.6	87.4	67.7	71.5	63.5	43.6
SalsaNext [10]	59.4	90.5	44.6	49.6	86.3	54.6	74.0	81.4	0.0	93.4	40.6	69.1	0.0	84.6	53.0	83.6	64.3	64.2	54.4	39.8
SPVNAS [17]	62.3	96.5	44.8	63.1	59.9	64.3	72.0	86.0	0.0	93.9	42.4	75.9	0.0	88.8	59.1	88.0	67.5	73.0	63.5	44.3
PMF [51]	63.9	95.4	47.8	62.9	68.4	**75.2**	78.9	71.6	0.0	**96.4**	43.5	80.5	0.1	88.7	60.1	88.6	**72.7**	75.3	**65.5**	43.0
Cylinder3D [18]	64.9	96.4	**61.5**	78.2	66.3	69.8	**80.8**	**93.3**	0.0	94.9	41.5	78.0	**1.4**	87.5	50.0	86.7	72.2	68.8	63.0	42.1
AMVNet [38]	65.2	95.6	48.8	65.4	88.7	54.8	70.8	86.2	0.0	95.5	**53.9**	**83.2**	0.15	**90.9**	**62.1**	87.9	66.8	74.2	64.7	**49.3**
SVASeg (Ours)	**66.1**	**96.8**	53.0	**80.2**	**88.9**	62.8	78.1	91.4	**1.1**	93.7	41.0	78.7	0.1	89.7	55.1	**89.2**	65.8	**76.7**	65.1	49.0

Table 2. Experimental results of our proposed SVASeg and state-of-the-art LiDAR segmentation methods on the SemanticKITTI dataset's official leaderboard. All results were obtained from the literature or leaderboard.

Methods	*mIoU*	Car	Bicycle	Motorcycle	Truck	Other-Vehicle	Person	Bicyclist	Motorcyclist	Road	Parking	Sidewalk	Other-Ground	Building	Fence	Vegetation	Trunk	Terrain	Pole	Traffic-Sign
S-BKI [30]	51.3	83.8	30.6	43.0	26.0	19.6	8.5	3.4	0.0	92.6	65.3	77.4	30.1	89.7	63.7	83.4	64.3	67.4	58.6	67.1
PointNL [31]	52.2	92.1	42.6	37.4	9.8	20.0	49.2	57.8	28.3	90.5	48.3	72.5	19.0	81.6	50.2	78.5	54.5	62.7	41.7	55.8
RangeNet++ [34]	52.2	91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	91.8	65.0	75.2	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9
LatticeNet [52]	52.9	92.9	16.6	22.2	26.6	21.4	35.6	43.0	46.0	90.0	59.4	74.1	22.0	88.2	58.8	81.7	63.6	63.1	51.9	48.4
RandLANet [1]	53.9	94.2	26.0	25.8	40.1	38.9	49.2	48.2	7.2	90.7	60.3	73.7	20.4	86.9	56.3	81.4	61.3	66.8	49.2	47.7
PolarNet [13]	54.3	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5
MinkNet42 [21]	54.3	94.3	23.1	26.2	26.1	36.7	43.1	36.4	7.9	91.1	63.8	69.7	29.3	92.7	57.1	83.7	68.4	64.7	57.3	60.1
STPC [32]	54.6	94.7	31.1	39.7	34.4	24.5	51.1	48.9	15.3	90.8	63.6	74.1	5.3	90.7	61.5	82.7	62.1	67.5	51.4	47.9
MINet [53]	55.2	90.1	41.8	34.0	29.9	23.6	51.4	52.4	25.0	90.5	59.0	72.6	25.8	85.6	52.3	81.1	58.1	66.1	49.0	59.9
3D-MiniNet [54]	55.8	90.5	42.3	42.1	28.5	29.4	47.8	44.1	14.5	91.6	64.2	74.5	25.4	89.4	60.8	82.8	60.8	66.7	48.0	56.6
SqueezeSegV3 [11]	55.9	92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9
TemporalLidarSeg [35]	58.2	94.1	50.0	45.7	28.1	37.1	56.8	47.3	9.2	91.7	60.1	75.9	27.0	89.4	63.3	83.9	64.6	66.8	53.6	60.5
KPConv [5]	58.8	96.0	32.0	42.5	33.4	44.3	61.5	61.6	11.8	88.8	61.3	72.7	**31.6**	**95.0**	64.2	84.8	69.2	69.1	56.4	47.4
SalsaNext [10]	59.5	91.9	48.3	38.6	38.9	31.9	60.2	59.0	19.4	91.7	63.7	75.8	29.1	90.2	64.2	81.8	63.6	66.5	54.3	62.1
FusionNet [41]	61.3	95.3	47.5	37.7	41.8	34.5	59.5	56.8	11.9	91.8	68.8	77.1	30.8	92.5	69.4	84.5	69.8	68.5	60.4	66.5
PCSCNet [37]	62.7	95.7	48.8	46.2	36.4	40.6	55.5	68.4	**55.9**	89.1	60.2	72.4	23.7	89.3	64.3	84.2	68.2	68.1	60.5	63.9
KPRNet [12]	63.1	95.5	54.1	47.9	23.6	42.6	65.9	65.0	16.5	**93.2**	**73.9**	**80.6**	30.2	91.7	**68.4**	**85.7**	69.8	71.2	58.7	64.1
TORANDONet [19]	63.1	94.2	55.7	48.1	40.0	38.2	63.6	60.1	34.9	89.7	66.3	74.5	28.7	91.3	65.6	85.6	67.0	**71.5**	58.0	65.9
SPVCNN [17]	63.8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lite-HDSeg [36]	63.8	92.3	40.0	55.4	37.7	39.6	59.2	**71.6**	54.1	93.0	68.2	78.3	29.3	91.5	65.0	78.2	65.8	65.1	59.5	**67.7**
SVASeg (Ours)	**65.2**	**96.7**	**56.4**	**57.0**	**49.1**	**56.3**	**70.6**	67.0	15.4	92.3	65.9	76.5	23.6	91.4	66.1	85.2	**72.9**	67.8	**63.9**	65.2

4.4. Evaluation on the nuScenes Dataset

Besides evaluation on the large scale outdoor dataset SemanticKITTI [26], we also conducted experiments on another large-scale autonomous driving dataset, nuScenes [27], to further evaluate the performance of our method. Table 3 reports the LiDAR semantic segmentation results on the validation set of nuScenes. RangeNet++ [34] and Salsanext [10]

use KNN as post-processing to further improve the LiDAR segmentation performance. From Table 3, we can see that our SVASeg achieves better performance than other LiDAR segmentation methods. Specifically, the proposed method outperforms the state-of-the-art projection-based methods (e.g., RangeNet++ and Salsanext) by about 6∼12% mIoU. Compared to $(AF)^2$-S3Net [55], PolarNet [13] and Cylinder3D [18], SVASeg achieved 11.5 mIoU, 2.7 mIoU and 1.1 mIoU performance gains, respectively. Compared to the SemanticKITTI dataset, the points in the nuScenes dataset are very sparse (35k points/frame), especially bicycles, traffic-cones, etc. Therefore, the LiDAR segmentation task is more challenging. From Table 3, we can see that our proposed SVASeg also shows its effectiveness in those sparse categories.

Table 3. Experimental results of our method and other methods on the nuScenes validation set. * is our reproduced Cylinder3D.

Methods	*mIoU*	Barrier	Bicycle	Bus	Car	Construction	Motorcycle	Pedestrian	Traffic-Cone	Trailer	Truck	Driveable	Other	Sidewalk	Terrain	Manmade	Vegetation
#Points (k)	-	1629	21	851	6130	194	81	417	112	370	2560	56048	1972	12631	13620	31667	21948
$(AF)^2$-S3Net [55]	62.2	60.3	12.6	82.3	80.0	20.1	62.0	59.0	49.0	42.2	67.4	94.2	68.0	64.1	68.6	82.9	82.4
RangeNet++ [34]	65.5	66.0	21.3	77.2	80.9	30.2	66.8	69.6	52.1	54.2	72.3	94.1	66.6	63.5	70.1	83.1	79.8
PolarNet [13]	71.0	**74.7**	28.2	85.3	**90.9**	35.1	77.5	71.3	58.8	57.4	76.1	**96.5**	**71.1**	**74.7**	74.0	87.3	85.7
PCSCNet [37]	72.0	73.3	42.2	87.8	86.1	44.9	**82.2**	76.1	62.9	49.3	77.3	95.2	66.9	69.5	72.3	83.7	82.5
Salsanext [10]	72.2	74.8	34.1	85.9	88.4	42.2	72.4	72.2	63.1	**61.3**	76.5	96.0	70.8	71.2	71.5	86.7	84.4
Cylinder3D * [18]	74.0	74.5	36.6	**89.5**	88.0	47.9	76.5	**78.1**	63.0	59.7	80.3	96.3	70.8	74.5	**75.0**	**87.5**	86.7
SVASeg (Ours)	**74.7**	73.1	**44.5**	88.4	86.6	**48.2**	80.5	77.7	**65.6**	57.5	**82.1**	**96.5**	70.5	**74.7**	74.6	87.3	**86.9**

4.5. Ablation Studies

In this sub-section, we show ablation experiments on the validation set of SemanticKITTI [26] to investigate the effect of different hash size K for grouping in the sparse voxel-based multi-head attention (SMHA) module. For a fairer and clearer comparison, we used the same configuration as in Section 4.2 for all models. Detailed experiment results are presented in Table 4. We first removed the SMHA module from SVASeg, which was taken as our baseline method. It achieved 65.2 mIoU on the validation set of SemanticKITTI. From Table 4, we can see that increasing the specific hash size K can improve the LiDAR segmentation performance, which indicates the SMHA can capture richer context information and enlarge the receptive fields for better segmentation. We also incorporated the proposed SMHA with Cylinder3D for LiDAR semantic segmentation. From Table 4, it can be observed that SMHA can effectively improve the segmentation performance, demonstrating the effectiveness of our SMHA.

Table 4. Ablation results on the validation set of SemanticKITTI.

	Baseline	**Hash Size *K***			**Cylinder3D**	
		16	**24**	**32**	**Original**	**+SHMA**
mIoU	65.2	65.7	66.0	66.1	64.9	65.6

Table 5 illustrates a comparison of the memory consumption, model size, time performance and performance on the validation set of the SemanticKITTI dataset. All experiments were conducted on a single GTX 3090Ti GPU with the same environment. Note that the time unit is milliseconds, and the memory and model size units are MB. Compared with the baseline model, the SHMA module only added 108 M memory consumption, 2 M parameters and 8 ms running time, which shows the SMHA maintains a comparable computational overhead to the convolutional method.

Table 5. Results of model complexity. The units for memory, model size and time are MB, MB and milliseconds.

Method	Memory	Model Size	Time	*mIoU*
Baseline	3041	214	102	65.2
Baseline + SHMA	3149	216	110	66.1

5. Conclusions

In this paper, a sparse voxel-based attention network was proposed for 3D LiDAR point cloud semantic segmentation (SVASeg). SVASeg mainly consists of four encoding layers, four decoding layers and a sparse voxel-based multi-head attention module. The encoding and decoding layers are implemented by using the submanifold sparse convolution and sparse inverse convolution, respectively, which are used to learn high-level semantic features from the input sparse voxels. The sparse voxel-based multi-head attention module is used to enlarge the receptive fields and capture rich contextual information for better segmentation, which only focuses on the non-empty voxel positions in a given local region. Extensive experimental results on the SemanticKITTI and nuScenes datasets showed the effectiveness and superiority of our SVASeg. In the future, the shift window-based transformer can be applied to SVASeg to further reduce the computational consumption and improve the performance of LiDAR semantic segmentation.

Author Contributions: Conceptualization, L.Z., L.L. and W.T.; methodology, L.Z. and S.X.; software, L.Z. and D.M.; validation, L.Z., S.X., W.T. and D.M.; writing—original draft preparation, L.Z. and S.X.; writing—review and editing, L.Z., S.X., L.L. and W.T.; supervision, W.T. and L.L.; project administration, W.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant 61976227 and 62176096).

Data Availability Statement: Not applicable.

Acknowledgments: The authors are grateful to the editor and reviewers for their constructive comments, which significantly improved this work.

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

References

1. 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 Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11108–11117.
2. Liu, L.; Yu, J.; Tan, L.; Su, W.; Zhao, L.; Tao, W. Semantic Segmentation of 3D Point Cloud Based on Spatial Eight-Quadrant Kernel Convolution. *Remote Sens.* **2021**, *13*, 3140. [CrossRef]
3. Xu, T.; Gao, X.; Yang, Y.; Xu, L.; Xu, J.; Wang, Y. Construction of a Semantic Segmentation Network for the Overhead Catenary System Point Cloud Based on Multi-Scale Feature Fusion. *Remote Sens.* **2022**, *14*, 2768. [CrossRef]
4. Zhao, L.; Tao, W. JSNet: Joint Instance and Semantic Segmentation of 3D Point Clouds. *Proc. Aaai Conf. Artif. Intell.* **2020**, *34*, 12951–12958. [CrossRef]
5. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October 2019–2 November 2019.
6. Ballouch, Z.; Hajji, R.; Poux, F.; Kharroubi, A.; Billen, R. A Prior Level Fusion Approach for the Semantic Segmentation of 3D Point Clouds Using Deep Learning. *Remote Sens.* **2022**, *14*, 3415. [CrossRef]
7. 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.
8. Wu, W.; Qi, Z.; Fuxin, L. PointConv: Deep Convolutional Networks on 3D Point Clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
9. Gao, F.; Yan, Y.; Lin, H.; Shi, R. PIIE-DSA-Net for 3D Semantic Segmentation of Urban Indoor and Outdoor Datasets. *Remote Sens.* **2022**, *14*, 3583. [CrossRef]

10. Cortinhal, T.; Tzelepis, G.; Aksoy, E.E. SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds. In Proceedings of the International Symposium on Visual Computing, San Diego, CA, USA, 5–7 October 2020; pp. 207–222.
11. Xu, C.; Wu, B.; Wang, Z.; Zhan, W.; Vajda, P.; Keutzer, K.; Tomizuka, M. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 1–19.
12. Kochanov, D.; Nejadasl, F.K.; Booij, O. KPRNet: Improving projection-based LiDAR semantic segmentation. *arXiv* **2020**, arXiv:2007.12668.
13. Zhang, Y.; Zhou, Z.; David, P.; Yue, X.; Xi, Z.; Gong, B.; Foroosh, H. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9601–9610.
14. Riegler, G.; Osman Ulusoy, A.; Geiger, A. Octnet: Learning deep 3d representations at high resolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3577–3586.
15. Liu, Z.; Tang, H.; Lin, Y.; Han, S. Point-voxel cnn for efficient 3d deep learning. *arXiv* **2019**, arXiv:1907.03739.
16. Graham, B.; Engelcke, M.; van der Maaten, L. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
17. Tang, H.; Liu, Z.; Zhao, S.; Lin, Y.; Lin, J.; Wang, H.; Han, S. Searching efficient 3d architectures with sparse point-voxel convolution. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 685–702.
18. Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Li, W.; Ma, Y.; Li, H.; Yang, R.; Lin, D. Cylindrical and asymmetrical 3d convolution networks for lidar-based perception. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef]
19. Gerdzhev, M.; Razani, R.; Taghavi, E.; Bingbing, L. Tornado-net: Multiview total variation semantic segmentation with diamond inception module. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation, Xi'an, China, 30 May–5 June 2021; pp. 9543–9549.
20. Zhao, L.; Zhou, H.; Zhu, X.; Song, X.; Li, H.; Tao, W. LIF-Seg: LiDAR and Camera Image Fusion for 3D LiDAR Semantic Segmentation. *arXiv* **2021**, arXiv:2108.07511.
21. Choy, C.; Gwak, J.; Savarese, S. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019, pp. 3075–3084.
22. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv* **2021**, arXiv:2103.14030.
23. Li, Z.; Wang, W.; Xie, E.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Lu, T.; Luo, P. Panoptic SegFormer. *arXiv* **2021**, arXiv:2109.03814.
24. Mao, J.; Xue, Y.; Niu, M.; Bai, H.; Feng, J.; Liang, X.; Xu, H.; Xu, C. Voxel transformer for 3d object detection. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 3164–3173.
25. Fan, L.; Pang, Z.; Zhang, T.; Wang, Y.X.; Zhao, H.; Wang, F.; Wang, N.; Zhang, Z. Embracing Single Stride 3D Object Detector with Sparse Transformer. *arXiv* **2021**, arXiv:2112.06375.
26. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9297–9307.
27. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.
28. Cao, H.; Lu, Y.; Lu, C.; Pang, B.; Liu, G.; Yuille, A. Asap-net: Attention and structure aware point cloud sequence segmentation. *arXiv* **2020**, arXiv:2008.05149.
29. Yan, X.; Zheng, C.; Li, Z.; Wang, S.; Cui, S. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5589–5598.
30. Gan, L.; Zhang, R.; Grizzle, J.W.; Eustice, R.M.; Ghaffari, M. Bayesian spatial kernel smoothing for scalable dense semantic mapping. *IEEE Robot. Autom. Lett.* **2020**, *5*, 790–797. [CrossRef]
31. 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 International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 24 October–24 January 2020; pp. 8447–8452.
32. Fang, Y.; Xu, C.; Cui, Z.; Zong, Y.; Yang, J. Spatial transformer point convolution. *arXiv* **2020**, arXiv:2009.01427.
33. Geng, X.; Ji, S.; Lu, M.; Zhao, L. Multi-scale attentive aggregation for LiDAR point cloud segmentation. *Remote Sens.* **2021**, *13*, 691. [CrossRef]
34. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. Rangenet++: Fast and accurate lidar semantic segmentation. In Proceedings of the 2019 IEEE International Conference on Intelligent Robots and Systems, Macau, China, 3–8 November 2019; pp. 4213–4220.
35. Duerr, F.; Pfaller, M.; Weigel, H.; Beyerer, J. LiDAR-based recurrent 3D semantic segmentation with temporal memory alignment. In Proceedings of the 2020 International Conference on 3D Vision, Fukuoka, Japan, 25–28 November 2020, pp. 781–790.
36. Razani, R.; Cheng, R.; Taghavi, E.; Bingbing, L. Lite-hdseg: Lidar semantic segmentation using lite harmonic dense convolutions. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation, Xi'an, China, 30 May 2021–5 June 2021; pp. 9550–9556.

37. Park, J.; Kim, C.; Jo, K. PCSCNet: Fast 3D Semantic Segmentation of LiDAR Point Cloud for Autonomous Car using Point Convolution and Sparse Convolution Network. *arXiv* **2022**, arXiv:2202.10047.
38. Liong, V.E.; Nguyen, T.N.T.; Widjaja, S.; Sharma, D.; Chong, Z.J. AMVNet: Assertion-based Multi-View Fusion Network for LiDAR Semantic Segmentation. *arXiv* **2020**, arXiv:2012.04934.
39. Wang, Y.; Fathi, A.; Kundu, A.; Ross, D.; Pantofaru, C.; Funkhouser, T.; Solomon, J. Pillar-based object detection for autonomous driving. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
40. Zhou, Y.; Sun, P.; Zhang, Y.; Anguelov, D.; Gao, J.; Ouyang, T.; Guo, J.; Ngiam, J.; Vasudevan, V. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In Proceedings of the Conference on Robot Learning, PMLR, Virtual, 16–18 November 2020; pp. 923–932.
41. Zhang, F.; Fang, J.; Wah, B.; Torr, P. Deep fusionnet for point cloud semantic segmentation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 644–663.
42. Chen, K.; Oldja, R.; Smolyanskiy, N.; Birchfield, S.; Popov, A.; Wehr, D.; Eden, I.; Pehserl, J. MVLidarNet: Real-Time Multi-Class Scene Understanding for Autonomous Driving Using Multiple Views. *arXiv* **2020**, arXiv:2006.05518.
43. 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 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
44. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 16259–16268.
45. Mazur, K.; Lempitsky, V. Cloud transformers: A universal approach to point cloud processing tasks. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10715–10724.
46. Wang, J.; Chakraborty, R.; Stella, X.Y. Spatial transformer for 3D point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef] [PubMed]
47. Guo, M.H.; Cai, J.X.; Liu, Z.N.; Mu, T.J.; Martin, R.R.; Hu, S.M. Pct: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [CrossRef]
48. Berman, M.; Triki, A.R.; Blaschko, M.B. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4413–4421.
49. Shen, Z.; Zhang, M.; Zhao, H.; Yi, S.; Li, H. Efficient attention: Attention with linear complexities. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2021; pp. 3531–3539.
50. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
51. Zhuang, Z.; Li, R.; Jia, K.; Wang, Q.; Li, Y.; Tan, M. Perception-aware Multi-sensor Fusion for 3D LiDAR Semantic Segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 16280–16290.
52. Rosu, R.A.; Schütt, P.; Quenzel, J.; Behnke, S. Latticenet: Fast point cloud segmentation using permutohedral lattices. *arXiv* **2019**, arXiv:1912.05905.
53. Li, S.; Chen, X.; Liu, Y.; Dai, D.; Stachniss, C.; Gall, J. Multi-scale interaction for real-time lidar data segmentation on an embedded platform. *IEEE Robot. Autom. Lett.* **2021**, *7*, 738–745. [CrossRef]
54. Alonso, I.; Riazuelo, L.; Montesano, L.; Murillo, A.C. 3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5432–5439. [CrossRef]
55. Cheng, R.; Razani, R.; Taghavi, E.; Li, E.; Liu, B. AF2-S3Net: Attentive Feature Fusion with Adaptive Feature Selection for Sparse Semantic Segmentation Network. *arXiv* **2021**, arXiv:2102.04530.