SpSequenceNet: Semantic Segmentation Network on 4D Point Clouds

Hanyu Shi[1], Guosheng Lin[1]*, Hao Wang[1], Tzu-Yi HUNG[2], and Zhenhua Wang[3]

[1]Nanyang Technological University, [2]Delta Research Center,
[3]Zhejiang University of Technology
E-mail: hanyu001@ntu.edu.sg, gslin@ntu.edu.sg

Abstract

Point clouds are useful in many applications like autonomous driving and robotics as they provide natural 3D information of the surrounding environments. While there are extensive research on 3D point clouds, scene understanding on 4D point clouds, a series of consecutive 3D point clouds frames, is an emerging topic and yet under-investigated. With 4D point clouds (3D point cloud videos), robotic systems could enhance their robustness by leveraging the temporal information from previous frames. However, the existing semantic segmentation methods on 4D point clouds suffer from low precision due to the spatial and temporal information loss in their network structures. In this paper, we propose SpSequenceNet to address this problem. The network is designed based on 3D sparse convolution, and it includes two novel modules, a cross-frame global attention module and a cross-frame local interpolation module, to capture spatial and temporal information in 4D point clouds. We conduct extensive experiments on SemanticKITTI, and achieve the state-of-the-art result of 43.1% on mIoU, which is 1.5% higher than the previous best approach.

1. Introduction

Scene understanding is a basic problem in computer vision. For autonomous driving cars and robotic systems that work in the real world, the performance and robustness of scene understanding is extremely crucial, since wrong decisions may result in fatal accidents. Researchers are trying to use more information to improve the performance and robustness. 3D point clouds, collected by Lidar or Depth cameras, provide more natural geometry information than 2D images. Further more, auto-driving cars and robots always work continuously within a period of time, thus the environments change continuously. Under this constraint, the systems could utilize temporal information from previous timestamps as hints and restrictions.

(a) Frame $t = 0$.

(b) Frame $t = 1$.

Figure 1: **Two-frame samples of normal camera video and point cloud sequence.** In each frame, the first row is collected with the normal front camera, and the second row is a projection of the annotated LiDAR point cloud. The point cloud is 360° around the car captured by the LiDAR sensors, which has broader perception fields than the normal camera video.

Semantic segmentation is a fundamental task in scene understanding. On 2D images, the task is a per pixel classification problem that assign corresponding categories to every pixel in the image. Inspired by the FCN [13], great acchievements have been made in this area, such as Deeplab

*Corresponding author: G. Lin (e-mail: gslin@ntu.edu.sg)

V3+[3], RefineNet [12] and PSPNet [27]. Also, many tasks are developed based on image semantic segmentation, such as point cloud segmentation, video segmentation and etc. Our work combines point cloud semantic segmentation and video semantic segmentation to improve the performance of scene understanding. 4D semantic segmentation is a more challenging task since both spatial and temporal information are involved.

The 4D datasets have rich real-world information. SemanticKITTI [2] (Figure 1) is one of the biggest 4D point cloud datasets, containing about 44,000 point cloud frames in total. The SemanticKITTI baseline method simplify the 4D semantic segmentation setting into a 3D one, where they combine multi point cloud frames into one point cloud, and apply the 3D segmentation method on the transitioned 3D point cloud. It causes temporal and spatial information loss during the combination of multiple point clouds frames. To resolve this problem, we propose SpSequenceNet to manipulate the 4D point cloud data in the 3D cube style, which reduces the spatial information loss. Meanwhile, we design a cross-frame global attention module and a novel cross-frame local interpolation module to extract the temporal features from different frames. We evaluate our network on SemanticKITTI [2]. The main contributions are:

- We design a network SpSequenceNet to directly capture spatial and temporal information from 4D point clouds (3D point cloud video) for semantic segmentation.
- We introduce the Cross-frame Global Attention (CGA) module to generate a global mask from previous point cloud frame and use the generated mask for the current point cloud frame segmentation.
- We propose the Cross-frame Local Interpolation (CLI) to fuse the information between two point cloud frames. It combines the temporal and spatial information together and improves the semantic segmentation quality.
- We achieve a new state-of-the-art result on SemanticKITTI [2], which is 1.5% higher than the existing methods.

2. Related work

Currently, there are few research works on 4D semantic segmentation. 4D semantic segmentation requires the network to extract both spatial information and temporal information. Thus, we separate the 4D semantic segmentation task into two sub tasks, i.e. spatial perception in 3D semantic segmentation and temporal perception, which is a novel area to explore. We will cover these two related parts in the following sections.

2.1. 3D Semantic Segmentation

A point cloud is collected by the depth sensors to reflect the objects' shape in the real world. The predicament in mining the semantics from point clouds is the sparsity and disorder of point cloud data. In previous researches, traditional 3D convolution [20] use a dense calculation and the complexity reaches $O(n^3)$. The sparsity of point cloud leads to high computation consumption and high resource waste for 3D convolution.Therefore, many works are done on point cloud processing and there are still many divergences on the utilization of point cloud data. Generally, there are three major ways for processing point cloud, namely projection-based method, PointNet-like method, and 3D convolution.

First, projection-based methods are the extension of the 2D semantic segmentation [24, 25, 23]. These methods perform projections, usually spherical projections, to transform the 3D points onto a surface. Then, they apply an image semantic segmentation network on the projected surface. The projection-based methods reach the real-time requirement (SqueezeSeg[24] reaches 13.5ms/per frame) while the final performance of projection-based methods is typically lower than other methods.

PointNet-like methods are developed from the novel structure PointNet [15]. This series of methods manipulate raw point cloud data directly, and treat the coordinate and RGB feature of the points as the input features. Then, the network applies a shared MLP on each point individually to generate the predictions. The performance is limited as it drops the local spatial relationship. PointNet++ [16] restricts a small region to extract the the local spatial relationship. PointCNN [11] redefines a convolution operation with MLP and neighbor weights to get a flexible local spatial information. KPConv [21] apply a more flexible neighbour mechanism and get the state-of-the-art performance in PointNet-like methods. Pointwise CNN [9] uses the kernel weights with voxel bins to combine the local information. KPConv [21] is followed by the PointCNN and PCNN and acchieve the state-of-the-art performance in PointNet-like methods.

The last method is the 3D convolution network. As stated in the beginning of this section, the computation consumption of 3D convolution is high. The major researches in this area focus on effectiveness. In OctNet [17], an octree structure is enrolled to represent 3D space, and guide the network on convolutions. Many works [18, 6] are developed based on this method. They arrange point cloud data into cubes and index them with Octree, Kdtree and etc so the convolution can be easily performed using this index. Furthermore, sparse 3D convolution based methods [8, 4] execute the 3D convolutions only along the active voxels in the inputs. Sparse 3D convolution can accelerate the convolution operations and share the knowledge base with dense

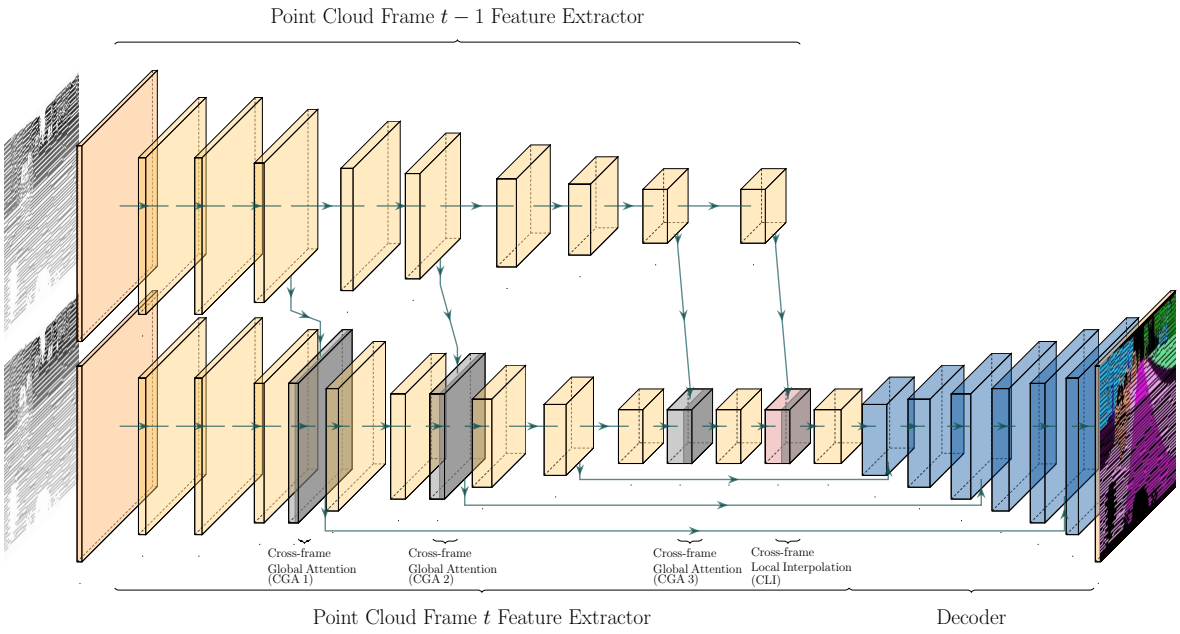


Figure 2: **Sparse sequence network structure.** The input data is the point cloud frames P_{t-1} and P_t. The output is the semantic label for P_t. In addition, we use colors to represent the different functions. The yellow blocks are the basic neural network blocks, which is a 3D residual network. Grey blocks are the Cross-frame Global Attention (CGA) modules, which is designed to fuse the comprehensive information from the last frame. The red blocks is the Cross-frame Local Interpolation (CLI) module, which is proposed to combine the local information from previous frames and current frames. The Blue blocks are the decoder modules for segmentation outputs respectively.

convolutions.

2.2. 4D Temporal Feature Extraction

4D temporal feature extraction focuses on mining the information in a time series. One recent research is Minkowski Convolutional Neural Networks(MinkowskiNet) [4]. It generalizes convolution function from 2D to 4D so that the theory of deep neural network is shared no matter the number of dimensions. The 4D MinkowskiNet lacks scalability since the computation consumption increase rapidly with the increases of points and frames.

There are some other researches on the 4D temporal feature extraction aside from semantic segmentation. In ST-CNN [28], a 3D U-Net and a 1-D encoder for time information are enrolled to auto-encode brain fMRI images. ST-CNN locates sight on the auto-encoder with a 4D temporal feature, which cannot be generalized to semantic segmentation tasks. OpenPose [10] focuses on a task to track human pose with the 4D point clouds. It uses 4D volumetric data to detect human hands' position in real-time with human detection and 2-D regression. PointFlowNet [1] is based on the pointNet-like method and fuses two features from frame t and $t-1$ to infer the motion of each point. Then, different losses are designed to extract the ego motion.

Overall, there are few methods which directly manipulate 4D point clouds on segmentation tasks. Therefore, we also explore some ideas from video semantic segmentation methods. MaskTrack and the network modulation [14, 26] use the the information and prediction from last frame to guide the current prediction.

3. Sparse Sequence Network

We show our proposed model structure in Figure 2. Generally, the problem setting of the 4D point cloud segmentation is similar to the normal 3D semantic segmentation. We built up the dataset based on the sensors, which are two sources, i.e. RGB-D camera (r, g, b) and LiDAR (r) Note that we take the coordinates (x, y, z) of each point and the point features $f_{i,t}$ as the model inputs, whose dimension is shaped as $(X, Y, Z, 3)$ (RGB-D) or $(X, Y, Z, 1)$ (LiDAR). The group of point clouds with n frames $P_t, t \in n$ is composed of $p_{i,t} = \{x_{i,t}, y_{i,t}, z_{i,t}\}, i \in m_t$. In our setting, we use a voxel method, and all the points are projected into a 3D tensor. As a result, (x, y, z) will be projected to (x', y', z'), which represents the point position in the cube. We set the $f_{i,t}$ as the value of each voxel. Our goal is to

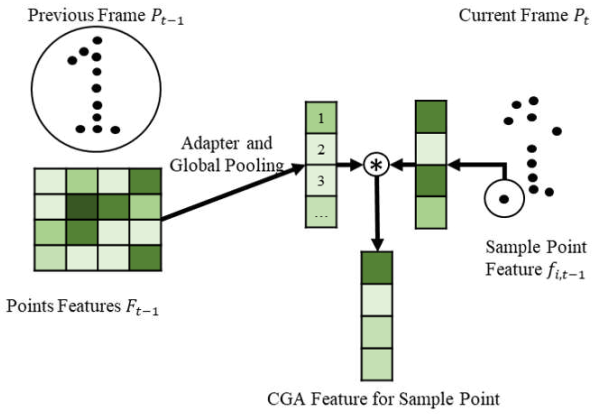


Figure 3: **A simple example for Cross-frame Global Attention (CGA).** There is a sample point in the current frame to show the process of the CGA.

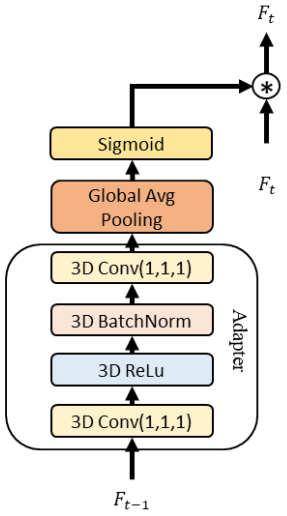


Figure 4: **The structure of Cross-frame Global Attention (CGA) in our network.**

predict the label $l_{i,t}$ of each $p_{i,t}$ when t is given. In our proposed framework, we use two frames, P_{t-1} and P_t, to do the predictions.

3.1. Network Architecture Overview

Our network is based on 3D convolution, which utilizes the voxel method. We predict the label $p_{i,t}$ with the inputs P_t and P_{t-1}, which are two 3D tensors.

The design of the proposed network follows the style of U-net, implemented by Submanifold Sparse Convolution Network (SSCN) [7]. To balance the speed and performance of training and inference, we made some modifications to the backbone network. Specifically, in the original version of SSCN, there are seven encoder blocks with skip paths to the deconvolution blocks, which forms a symmetrical structure.However, there are some drawbacks in the symmetrical desing, such as the limited representation abilities and the massive wastes of computation. Therefore, we reduce the number of skip paths. Besides, we add some blocks into the encoder, which is aimed to increase the expression ability and adjust the network. The decoder is streamlined, which contains the reduction of skip paths.

After the construction of our model, the next step is to build up our blocks to fuse the information from different frames. In the encoder phase, our network receives P_t and P_{t-1} with two different branches. It is described in the Figure 2. To construct better fused features, we define the information with two parts, global information and local information. Firstly, the cross-frame global attention module is designed for global information. In general, there are several cross-frame global attention modules in different phases. The cross-frame global attention module selects the features so that the backbone network can pay more attention to the key features. Secondly, cross-frame local interpolation focuses on local information, which is applied to fuse the information from both P_{t-1} and P_t at the end of encoder.

3.2. Cross-frame Global Attention

As stated above, we extract the temporal global semantics with our Cross-frame Global Attention (CGA) module. We show a simple explanation of the cross-frame global attention module in Figure 3. Inspired by the self-attention mechanism, we design the cross-frame global attention module to generate a mask for current frame P_t. The mask concludes the appearance information on the features of P_{t-1}. To highlight the crucial part of features F_t and inhibit irrelevant features, cross-frame global attention module uses the appearance information from $t-1$ to guide the model.

The global semantics are distributed to each level of the features. We select layers which are involved in the skip path and apply the cross-frame global attention. It reduces the computation complexity and brings precision improvement. Firstly, an adapter turns all feature vectors $f_{i,t-1}$ into $f'_{i,t-1}$ and applies a global average pooling on $f'_{i,t-1}$:

$$\mathbf{v}_j = \frac{\sum_i^{m_{t-1}} (g_j(f_{i,j,t-1}))}{m_{t-1}}. \quad (1)$$

Here, m_{t-1} is the total number of points from the previous frame P_{t-1}. g_j is a specific adapter function in the network and it is required to turn the features into a suitable one for attention. In our network, the adapter consists of two (1,1,1) 3D convolution layer, when a 3D ReLU layer

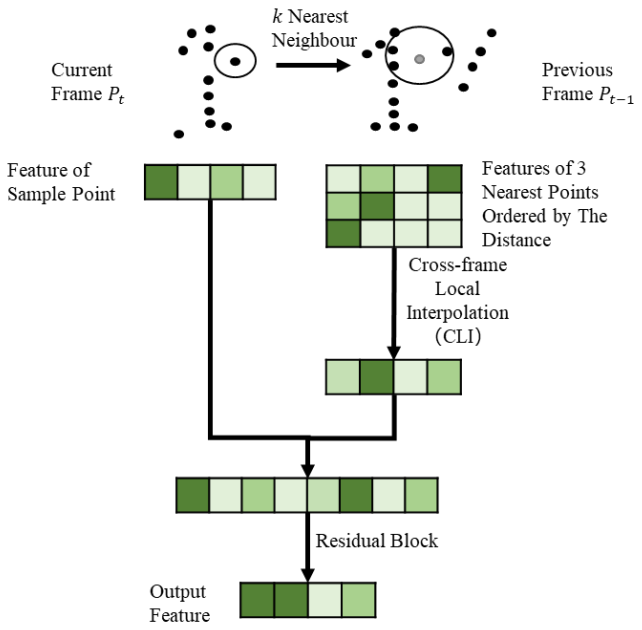


Figure 5: **The structure of Cross-frame Local Interpolation (CLI) in our network.** The process happens at every points in the current point cloud frame.

and a 3D batch normalization are in the middle of them. The global information is obtained by the average pooling. Then, we generate channel-wise attention maps a_j, which can be formulated as:

$$a_j = h_\theta(\mathbf{v}_j) = \frac{1}{1 + e^{-\theta^T \mathbf{v}_j}}. \tag{2}$$

When a_j is determined, the output features F'_t can be obtained by $F'_t = a_j * F_t$, where F_t is the input features of current point cloud frame. With cross-frame global attention, some channels in the features is set to zero. Therefore, it reduces the value in f_t, and keep the value of parts with high values in f'_t. P_{t-1} plays a role as a tutor. It teaches the network to focus on the true important part in P_t. A brief structure of this function is available in Figure 4.

3.3. Cross-frame Local Interpolation

At the end of the encoder phase, we design a cross-frame local interpolation (CLI) module to combine the information locally and capture the temporal information between two point cloud frames. Optic flow methods [22, 29] use the nearest pixel from two different frames to generate local optic flow and achieve significant performance. Inspired by these methods, cross-frame local interpolation is designed to extract partial difference between point clouds P_{t-1} and P_t. The basic idea of cross-frame local interpolation is shown in Figure 5, which is to seek the k nearest neighbors $p_{i',t-1}$ of $p_{i,t}$, and generate a new local feature to help the model fuse the temporal information. At the same time, cross-frame local interpolation summarizes the area of nearest points and fuses the spatial information with the feature of selected points.

Firstly, distance metrics $D_{t-1,t}$ is calculated as following:

$$D_{t-1,t} = \frac{C_t \cdot C_t^T + C_{t-1} \cdot C_{t-1}^T - 2C_t \cdot}{\gamma}, \tag{3}$$

where C is the metric which consists of the points coordinates. γ is a hyper-parameter for re-scaling the distance to a approximating scale $[0, 1]$. It is based on the shape of input data. We set γ as 32 when the shape of input is $32 \times 32 \times 32$. $D_{t-1,t}$ is an approximate Euclid distance matrix, which subsides the square operation to speed up the calculation. Based on $D_{t-1,t}$, the top k nearest $f_{j,t-1}$ is obtained, representing the area features. The weight $w_{i,t-1}$ for each point is

$$w_{i,t-1} = (\alpha - \min(d_{i,j,t,t-1}, \alpha)) * \beta, \tag{4}$$

where α and β are handcrafted parameters to adjust $w_{i,t-1}$. Note that α has an influence on the weights of distance. A low value of α makes network only considers the adjoining $p_{i,t-1}$ as the valid features. β modifies the range of final features to avoid gradient vanishing. In the experiment, we define α and β as 0.5 and 2. $d_{i,j,t,t-1}$ is the distance of position i,j in $D_{t-1,t}$. The $\min$ operation confirms no negative weight. $w_{i,t-1}$ is a weight for neighbour point $p_{i,t-1}$. Because of the point cloud's sparsity, the possibility of k nearest neighbours containing the points from another object remains high, while $w_{i,t-1}$ reduces this effect of features.The CLI features $L_{i,t-1}$ are calculated by:

$$L_{i,t-1} = \sum_{i}^{k} f_{i,t-1} * w_{i,t-1}. \tag{5}$$

Based on $L_{i,t-1}$, we concatenate $L_{i,t-1}$ and feature $f_{i,t}$ from current frames, and use a residual block to extract output features as Figure 5. We believe the network is capable to learn the relation between $L_{i,t-1}$ and $f_{i,t}$ and improve the segmentation quality.

4. Experiments

This section is divided into several parts. We first introduce the SemanticKITTI [2] dataset, the method and the experiment results. Then, we compare the results from different versions of the system. At last, we give some further discussions.

	mIoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	moving-car	moving-bicyclist	moving-person	moving-motorcyclist	moving-other-vehicle	moving-truck
TangentConv [19]	34.1	84.9	2.0	18.2	21.1	18.5	1.6	0.0	0.0	83.9	38.3	64.0	15.3	85.8	49.1	79.5	43.2	**56.7**	36.4	31.2	40.3	1.1	6.4	1.9	**30.1**	**42.2**
DarkNet53Seg	41.6	84.1	**30.4**	**32.9**	20.0	20.7	**7.5**	0.0	0.0	**91.6**	**64.9**	**75.3**	27.5	85.2	56.5	78.4	50.7	64.8	38.1	**53.3**	61.5	14.1	15.2	0.2	28.9	37.8
Backbone	41.9	**89.9**	20.6	23.3	23.4	**24.6**	3.5	0.0	0.0	89.8	59.9	73.5	**29.6**	90.2	65.0	82.3	63.6	64.1	**50.9**	49.6	**66.1**	40.7	21.6	7.5	7.5	1.0
Backbone+CGA	42.6	89.6	27.5	23.8	26.5	23.3	**7.5**	0.0	0.0	89.5	58.2	73.2	28.0	91.0	66.2	83.0	63.8	65.3	43.6	47.5	61.7	35.7	25.8	31.0	3.2	0.4
Backbone+CGA+CLI	**43.1**	88.5	24.0	26.2	**29.2**	22.7	6.3	0.0	0.0	90.1	57.6	73.9	27.1	**91.2**	**66.8**	**84.0**	**66.0**	65.7	50.8	48.7	53.2	**41.2**	**26.2**	**36.2**	2.3	0.1

Table 1: **Our results on the SemanticKITTI.** All the models were trained on the training set of SemanticKITTI, and evaluated on the testing set of SemanticKITTI. The performances of two state-of-the-art methods, TangentConv and DarkNet53Seg are from [2]. The evaluation metric for each column is mIoU. In the table, we list our three proposed methods. Our backbone network reaches 41.9% mIoU. The model in the fourth row is the backbone network with cross-frame global attention module. Cross-frame global attention module achieves a 0.7% improvement for the vanilla backbone network. The last row is the result from our proposed network SpSequenceNet, which apply the cross-frame local interpolation on the model in the fourth row. The network achieves +1.5% improvement with the DarkNet53Seg.

4.1. Dataset

We use the **SemanticKITTI** dataset, which is based on the data from the odometry task of KITTI [5]. In the SemanticKITTI paper [2], they built up a tool to manually annotate the semantic data on each frame. There are 22 3D point cloud videos, which contain 43,551 frames in total. In the experiment, the dataset is split into train (19,130 frames), validation (4,071), and test (20,351). In each scan, data is a series of points collected by LiDAR. The coordinates of points are related to the LiDAR's position. The test set is used for the final evaluations on their website[1]. The challenge in SemanticKITTI contains two parts, i.e. single-frame semantic segmentation and multi-frame semantic segmentation. Single-frame semantic segmentation is for the single-frame task, which contains 19 classes. Multi-frame semantic segmentation contains 6 more target categories than the single-frame task to distinguish between moving objects and stationary ones for several categories, including car, trunk, other-vehicle, person, bicyclist, motorcyclist. As mentioned before, our job is to predict the label at time t with the additional information from $t-1$, $t-2$.... We evaluate our model on 25 classes for the multi-frame semantic segmentation task.

4.2. Implementation Details

In the pre-possessing phase, we turn the coordinate system of previous frame P_{t-1} into that of the current frame P_t. Then, we apply a random rotation and scale on both P_t and P_{t-1} with the same random seeds, so that P_t and P_{t-1} are confirmed to be in the same coordinate system. Next, We use $0.05m$ as a unit to turn the coordinate of points P_{t-1} and P_t into the voxel format. The maximum scale of coordinate in the dataset is around $150m$, and the input cube in our network consists of $2048 \times 2048 \times 2048$ voxels. When the unit is set as $0.05m$, the input cube is capable of containing enough points. As a result, setting the unit to be $0.05m$ can achieve the best trade-off between computation and performance. Note that when $t = 0$, it is a special case for current point cloud frame P_t, which means it does not have a previous frame P_{t-1}. We simply build a cube with one point at $(0, 0, 0)$ and F_{t-1} is filled with 0. When the input is ready, we train SpSequenceNet with Adam optimizer and set the batch size as 14, which requires about 10GB GPU memory. The maximum number of the epoch is 40. We train the model with one Nvidia RTX 2080Ti. Each model takes about five days for training. In the inference phase, we apply the same process except data augmentation on the test data. In some cases, it is impossible to put all the points in the cube. The labels of these points are set as ignored label because the percentage of these points is below 1%, and the cost of covering these points is high.

[1] https://competitions.codalab.org/competitions/20331

4.3. Main Results

Baselines. The results are listed in Table 1. Baselines in SemanticKITTI are TangentConv [19] and DarkNet53Seg. They adjusted the coordinate system from P_{t-4} to P_{t-1} and combined all the frames into one point cloud as the input. TangentConv is a PointNet-like method, and DarkNet53Seg is a projection-based method.

Backbone Network. Backbone Network removes all the additional functions, and the input is just current point cloud frame P_t. The result is close to the best baseline DarkNet53Seg in SemanticKITTIs and is 7.9% higher than TangentConv [19].

Backbone + CGA. Here we adopt the backbone network and the cross-frame global attention. The input is based on two point cloud frames P_t and P_{t-1}. Compared to the backbone network, the performance has a 0.7% improvement on

	mIoU
backbone	41.9
backbone+CGA	42.6
backbone+CGA+CLI-1	42.0
backbone+CGA+CLI-3	**43.1**

Table 2: **Comparison between different top k for Cross-frame Local Interpolation (CLI).** The performance for top 3 CLI achieves a 0.5% improvement for the backbone network with cross-frame global attention, but top 1 CLI causes a performance decrease.

	Single mIoU	Move mIoU
Backbone	54.4	-
Backbone+CGA+CLI-3+Multi-head	56.0	**39.9**
Backbone+CGA+CLI-3+Reorganized	**57.1**	37.9

Table 3: **Single-Frame Task and Motion Status Segmentation.** The second column is single mIoU, which is the performance of the single-frame semantic segmentation task. The third column is the performance of the motion status segmentation.

mIoU.

Backbone + CGA + CLI. The structure is shown in Figure 2. The network contains the backbone network, the cross-frame global attention and the cross-frame local interpolation, which uses top 3 nearest neighbour to generate the area features. Our network achieves +1.5% mIoU with the DarkNet53Seg and achieves +1.2% mIoU with the backbone network.

In summary, compared with other advanced methods in the Table 1, our proposed methods are more sensitive with the movements of small objects and large static objects, while insensitive about the moving large objects. This phenomenon is caused by the characteristics of our proposed method. Specifically, in the proposed network, it detects the shifts of the features in the same voxel system between $t-1$ and t. When the object is moving, there are significant changes in the area of the small objects, while the large object areas do not change much.

4.4. Result Comparisons

Discussion of the methods in SemanticKITTI. The processing method for point cloud combination consume more resources than expected. Since the computation costs are highly related to the scope of the points, in our experiments, the batch size is forced to be set lower than 10. At the same time, the training time reaches over 6 hours per epoch when the reasonable minimum number of training epochs is 30, which spends about 8 days for the training process. It makes the training duration unacceptable. Therefore, we do not use this method on our backbone.

The effectiveness of the SpSequenceNet. We show a visualization for comparisons in Figure 6. For the backbone network, we can compare Figure 6b with Figure 6c and observe that the area in the red box of Figure 6b is untidiness. To be specific, Figure 6b represents vanilla backbone network. In Figure 6c, the previously mentioned area is more unitary. Therefore, cross-frame global attention and cross-frame local interpolation improve the smoothness of the results.

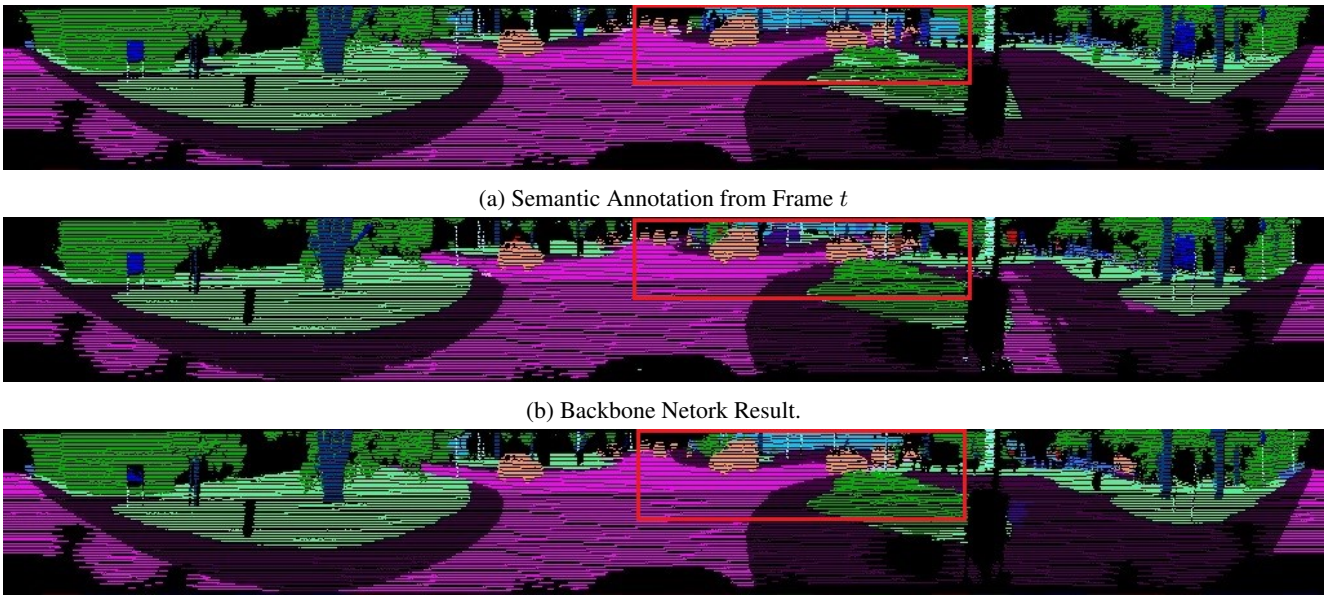
Cross-frame global attention. As shown in Table 1, the improvement of cross-frame global attention is of great significance. Specifically, cross-frame global attention enhances the performance of the vanilla backbone in some classes, because it helps the backbone track better on the small objects.

Top k cross-frame local interpolation. We choose K nearest neighbours from last frame P_{t-1} for point $p_{i,t}$ of current frame to generate the features of the cross-frame local interpolation. We train the model with the top 1, 3, and 5 nearest neighbours for the cross-frame local interpolation, which is named as top k CLI in the following part. For top 1 CLI and top 3 CLI, we submit the results to the SemanticKITTI for testing. The result shows that top 1 CLI causes the decrease in mIoU, which is in line with expectations. The precision of top 1 CLI in Table 1 is even poorer than backbone+CGA. For the points on the boundaries, the possibility of the nearest point with the same correct label is low, resulting in a 6% drop. At the same time, the result of the top 3 CLI reaches state-of-the-art. Finally, the result of the top 5 CLI is not shown here, since the performance on validation is similar to the top 3 CLI in every epoch. The performance of top 5 CLI is similar to top 3 CLI. Consequently, it is unnecessary to submit for the test results. According to the increase of computation consumption, 3 nearest neighbor is suitable for cross-frame local interpolation.

4.5. Single-Frame and Motion Status Experiment

We design an experiment to verify the effectiveness of our methods on the 4D point cloud semantic segmentation. The task of SemanticKITTI is to predict the semantics and the motion status for several specific objects. For objects within the same class, the gradients from moving and static objects may affect each other and degrade the training results. Therefore, the performance on the single-frame task can better reflect the overall performance of the networks. For better illustration, we compare the segmentation performance of motion status in different settings.

Accordingly, we train a backbone network for the single-frame task as a baseline. Then, Backbone+CGA+CLI-3 model is modified with a multi-head prediction in the end of the decoding phase, which is called multi-head method. One prediction head is for the single-frame task, and the



(a) Semantic Annotation from Frame t

(b) Backbone Netork Result.

(c) Backbone + CGA + CLI Result.

Figure 6: **Visual examples for different version networks.** Figure 6a is the ground truth for P_t and P_{t-1}. Figure 6b is the result from the backbone network when Figure 6c is the result from our proposed network. The result in the third row is better that that in the second row after comparing the top-left blue area of ground truth and our results.

other is for the object motion status. The input ground truth is also modified as single-frame and motion status ground truth, which can enhance the gradients from motion status. Finally, the multi-frame prediction of original SpSequenceNet, which is mentioned in Section 4.4 is reorganized to two outputs, the single-frame prediction and the motion status. We combine the moving objects and the static objects to generate a single-frame prediction, and extract the motion status from the moving objects and the static objects. The outputs of reorganized prediction is called reorganized prediction in Table 3.

The results are listed in Table 3. First of all, our network has the ability to improve the semantic segmentation. The mIoU improvement reaches 1.6% for multi-head network and 2.7% for reorganized prediction, compared to the performance of the backbone network, 54.4% on mIoU. Afterwards, compared to the reorganized prediction, multi-head network has a 2% improvement in the motion status, but there is a 1.1% decrease for single-frame task, which indicates it is harmful to the object representation ability if the model directly incorporates the motion status into the training objects.

5. Conclusion

In this paper, we propose a novel structure, SpSequenceNet, to fuse the spatial and temporal information from 4D point clouds. In the SpSequenceNet, we design two modules, cross-frame global attention and cross-frame local interpolation to improve the performance. Cross-frame global attention is an attention layer generated from the global features of the last frames, and highlights the key features of each point from current frames. Cross-frame local interpolation uses the features from the nearest last frames. With the experiments, we have shown the effectiveness of the whole model SpSequenceNet and its building components, cross-frame global attention and cross-frame local interpolation. Overall, our proposed method has significantly outperformed the state-of-the-art methods for 4D point cloud segmentation, and we believe our method can be effectively applied in other general 4D point cloud semantic segmentation tasks.

6. Acknowledgments

This work is supported by the Delta-NTU Corporate Lab with funding support from Delta Electronics Inc. and the National Research Foundation (NRF) Singapore. This work is also partly supported by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: AISG-RP-2018-003), the MOE Tier-1 research grant: RG22/19 (S), and the National Natural Science Foundation of China (61802348).

References

[1] Aseem Behl, Despoina Paschalidou, Simon Donné, and Andreas Geiger. Pointflownet: Learning representations for rigid motion estimation from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7962–7971, 2019.

[2] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[4] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[5] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[6] Benjamin Graham. Spatially-sparse convolutional neural networks. *arXiv preprint arXiv:1409.6070*, 2014.

[7] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *CVPR*, 2018.

[8] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017.

[9] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993, 2018.

[10] Hao Jiang and Quanzeng You. Real-time multiple people hand localization in 4d point clouds. *arXiv preprint arXiv:1903.01695*, 2019.

[11] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, pages 820–830, 2018.

[12] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, July 2017.

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[14] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2663–2672, 2017.

[15] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016.

[16] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[17] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3577–3586, 2017.

[18] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018.

[19] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2018.

[20] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *2017 International Conference on 3D Vision (3DV)*, pages 537–547. IEEE, 2017.

[21] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. *arXiv preprint arXiv:1904.08889*, 2019.

[22] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019.

[23] Yuan Wang, Tianyue Shi, Peng Yun, Lei Tai, and Ming Liu. Pointseg: Real-time semantic segmentation based on 3d lidar point cloud. *arXiv preprint arXiv:1807.06288*, 2018.

[24] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018.

[25] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019.

[26] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6499–6507, 2018.

[27] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[28] Yu Zhao, Xiang Li, Wei Zhang, Shijie Zhao, Milad Makkie, Mo Zhang, Quanzheng Li, and Tianming Liu. Modeling 4d fmri data via spatio-temporal convolutional neural net-

works (st-cnn). In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 181–189. Springer, 2018.

[29] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017.