Temporal Feature Matching and Propagation for Semantic Segmentation on 3D Point Cloud Sequences

Hanyu Shi, Ruibo Li, Fayao Liu, and Guosheng Lin

***Abstract*—In real-world LiDAR-based applications, data is generated in the form of 3D point cloud sequences or 4D point clouds. However, the topic of semantic segmentation on 4D point clouds is under-investigated and existing methods are still not able to achieve satisfactory performance to meet the requirement for real-world applications. The temporal information across different point clouds plays an important role in dynamic scene understanding, which is not well explored in existing work. In this paper, we focus on exploring effective temporal information across two consecutive point clouds for semantic segmentation on point cloud sequences. To this end, we design three novel modules to enhance the features of target frames by extracting different temporal information in the local regions and global regions. Experimental results on SemanticKITTI and SemanticPOSS demonstrate that our method achieves superior performance in 4D semantic segmentation by utilizing temporal information.**

***Index Terms*—4D Point Clouds, Semantic Segmentation, Scene Understanding.**

I. Introduction

With the development of 3D sensor technologies, 3D scanners such as LiDAR and RGB-D cameras become very affordable and they have been widely equipped into autonomous agents, personal and industrial equipment. In dynamic scenes, 4D point clouds, the temporal sequence of 3D point clouds captured by moving 3D sensors, are the most common data format. Therefore, in this paper, we target the task of semantic segmentation on 4D point clouds. An example of a 3D point cloud sequence is shown in Figure 1. Although significant progress has been made in semantic segmentation on 3D point clouds, semantic segmentation on 4D point clouds is under-investigated. In our view, fully utilizing the temporal information across different point clouds is the crucial point for semantic segmentation in dynamic environments.

As an essential step towards 3D scene understanding, 3D semantic segmentation predicts the label of each point in the whole scene. Existing methods [1]–[3] focus on the utilization of spatial information from local geometric features between the points. Two consecutive frames contain dynamic context information, i.e., the temporal information in the 4D point cloud. We argue that sufficiently leveraging temporal information is important to improve the segmentation performance. The temporal information can significantly improve the robustness and accuracy of the semantic segmentation method. Furthermore, the recognition of dynamic objects depends on their relative movement in different timestamps. Extracting the temporal information from different point cloud frames helps to recognize dynamic objects in 4D point clouds.

To fuse the temporal information, DarkNet53Seg [4] merges multiple point clouds into a huge point cloud and makes predictions on the merged point cloud. While simple frame aggregation is used to formulate the temporal information of consecutive frames, this approach does not consider potential temporal relationships between different frames of the point cloud, resulting in unsatisfactory performance of this method. 4D MinkowskiUNet [5] extends the 3D convolution to 4D convolution with a heavy computation cost. The inefficiency of temporal information extraction in this method brings challenges in large network design, network training and computation resources. SpSequenceNet [6] and its extension TVSN [7] design an efficient structure to extract temporal information. However, they only utilize local features from previous frames and simple pooled global features of the previous frames to enhance the features of target frames, which limits the final performance.

In this paper, we capture temporal information from three perspectives: local feature propagation, local semantic propagation, and global feature matching. These perspectives enhance the features in the current frame with different temporal information from previous frames. Firstly, it is often observed that the static or low-speed object within a specific region is located in a nearby region in the previous point cloud frame. To capture this local consistency information, we propose a local feature propagation module to enhance the features of the current frame. Secondly, the semantic information of each region shows temporal coherence between neighbouring frames. We propose a local semantic propagation module to project the predicted labels from the previous frame to the current frame. Thirdly, finding corresponding objects in neighbouring frames globally and propagating the corresponding features can be used to enhance the features of the current frame. We propose a global feature matching module to map object features from the previous frame into the current frame by finding corresponding objects in all spatial positions between neighbouring frames.

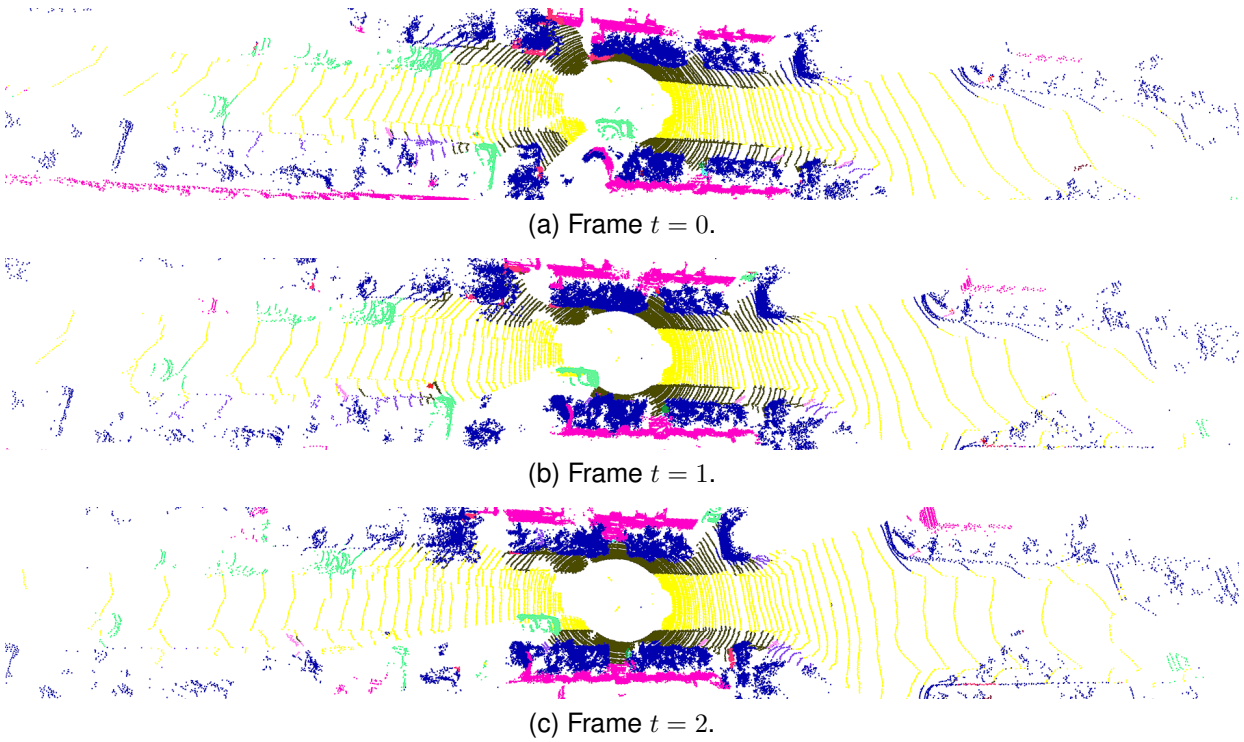
In summary, our contributions are:

Hanyu Shi, Ruibo Li and Guosheng Lin are with Nanyang Technological University. e-mail: {hanyu001, ruibo001}@e.ntu.edu.sg, gslin@ntu.edu.sg.

Fayao Liu is with Institute for Infocomm Research A*STAR, Singapore. e-mail: fayaoliu@gmail.com

Corresponding authors: Guosheng Lin and Fayao Liu

Manuscript received Nov 28, 2022.



(a) Frame $t = 0$.

(b) Frame $t = 1$.

(c) Frame $t = 2$.

Fig. 1. **A 3D point cloud video with 3 frames.** The point cloud covers a 360° area around the sensor. A sequence of point cloud frames constitutes a 4D point cloud. Different colours represent different classes of points: yellow for roads, blue for plants, green for cars and deep green for sidewalks.

- We design a Feature Propagation Network (FeatProp) for the 4D point cloud semantic segmentation by exploring the temporal information between two consecutive point clouds locally and globally. Our experimental results on SemanticKITTI [4] and SemanticPOSS [8] demonstrate the superior performance of our method.
- We propose a Local Feature Propagation module (LFP) to propagate the regional context information from the previous point cloud to the current point cloud as a clue to improve the prediction.
- We introduce a Local Semantic Propagation module (LSP) to propagate the predicted labels from the previous frame to the current frame, which improves the feature representation of the current features.
- We propose a Global Feature Matching module (GFM) which builds long-range connections among all points in the two neighbouring point clouds to find object correspondences, and map the object features from previous frames into the current frame to augment the current features.

II. Related Work

A. 3D Point Cloud Segmentation

In recent years, there has been an increasing amount of work on the 3D point cloud topic, especially 3D point cloud semantic segmentation. Point clouds contain rich spatial information from the environment. Spatial information benefits the performance of robotic systems. One of the greatest challenges of deep learning for 3D point clouds is that with the increased dimension, the efficiency and precision of a dense 3D convolution decrease significantly due to the sparsity and a large number of points in 3D point clouds. Therefore, existing methods for 3D point cloud semantic segmentation focus on developing feature extraction approaches on the sparse point cloud space. A recent survey [9] provides an overview of the state-of-the-art methods for 3D point cloud semantic segmentation and categorizes them into three approaches: projection-based, volumetric-based and point-based methods.

Projection-based Method. Projection-based methods extend the core of 2D image semantic segmentation. A projection-based method rebuilds the 3D point cloud data as one or more 2D planes. For example, multi-view approaches [10], [11] project the point cloud on multiple 2D images with different camera views. Spherical approaches [12]–[14] project the point cloud onto a spherical plane. Then, projection-based methods design a 2D dense network to process the projected images and re-project the 2D projections onto the original point cloud. Due to the 2D structure of projection-based methods, they can meet the real-time inference requirement. However, the projection drops the information of geometric structures partially. Furthermore, 2D dense convolution blurs the features of each projected point, which further drops the local geometric information. Hence, the drawbacks of projection limit the precision of projection-based methods, and the best usage scenario of projection-based methods is real-time tasks.

Volumetric-based Methods. To accelerate the process of 3D convolution, volumetric-based methods build an index on the 3D grids and apply convolution on the valid points except for the invalid space. Typically, the index is designed as a tree-

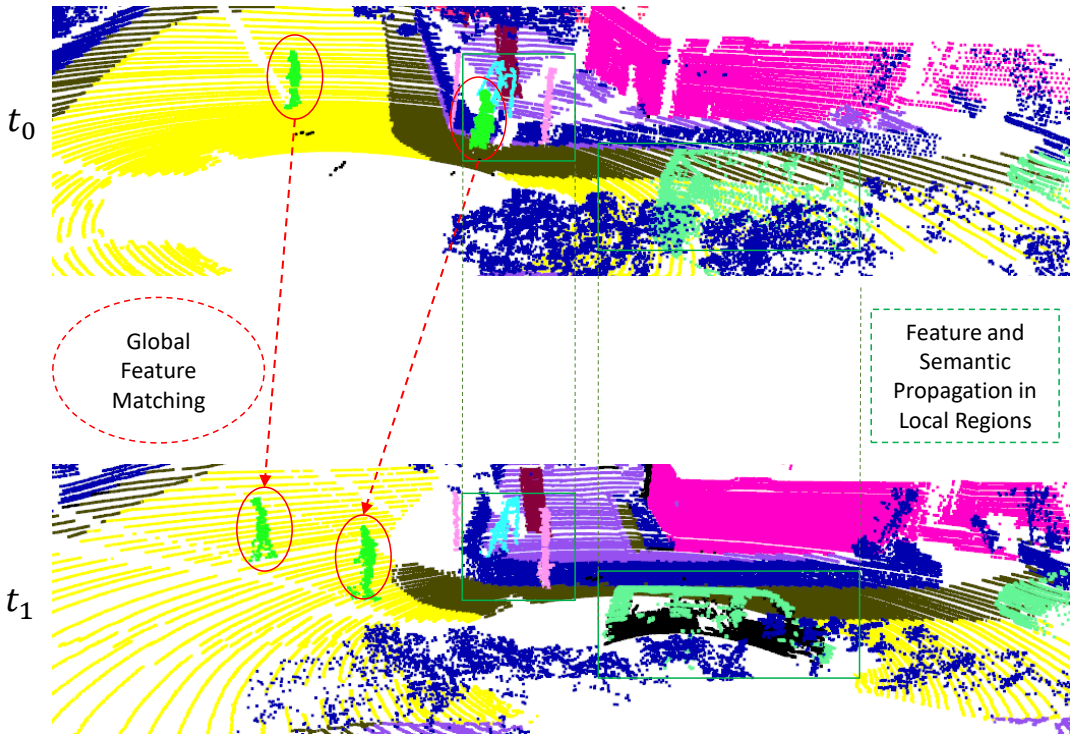


Fig. 2. **An illustration of the temporal information captured in our method.** As indicated by the red circles, the global temporal matching module builds the connections between the moving people in different locations and times. As indicated by the green rectangles, the feature and semantic propagation module propagates the information of the local regions in the previous frame to the current frame.

based index [15]–[17] or a hash table [5], [18]. The speed of volumetric-based methods depends on the index algorithms. Recently, some volumetric-based methods, like SPConv [18] or Minkowski Engine [5], update a GPU-based hash for high-performance computation. Afterwards, the convolution algorithm searches for the valid points and their neighbours along with the index and applies channel-wise operations on these points. As an extension to dense convolution, volumetric-based methods share some common knowledge with dense convolution and are easy to extend to other tasks. Nonetheless, the performance of volumetric-based methods is affected by the hyper-parameters in the volumetric algorithm. Similar to projection-based methods, the volumetric algorithm drops some local geometric information based on a subsample resolution. When the subsample resolution is unreasonable, the performance of sparse convolution is below the expectation.

Point-based Methods. Point-based method directly manipulates the points' features and coordinates to generate high-quality features. The first type of Point-based method is the Pointnet-like method [1], [3], [19], [20], which is inspired by Pointnet [21]. They design several novel MLP structures to build features with not only input RGB or remission features but also coordinates. Another type of Point-based method [2], [22], [23] is to build a special convolution operation to compute convolution on the continuous coordinate space instead of the grid space in typical dense convolutions. JSNet++ [24] combines PointNet++ [1] and PointConv [25] to predict the semantic-level and instance-level predictions together. Then, graph-based methods [26]–[28] build neighbourhood graphs on point clouds and use a graph neural network to capture the local spatial information.

B. Temporal Feature Extraction in 4D Point Cloud Segmentation

Temporal feature extraction is a special task not only in 2D image-based tasks but also in 3D point cloud-based tasks. It involves extracting features related to time in a video or sequence. A basic method for temporal feature extraction is to extend d dimensional convolution in the task to $d+1$ dimensional convolution by additional considering the temporal dimension. For example, 3D convolution network [29], [30] is applied to extract the information of human actions in 2D videos. However, the extension method consumes massive computation resources with limited improvement. As a result, there are a lot of novel structures proposed in both 3D point cloud sequence task and the 2D video task.

There are a few temporal feature extraction methods for 4D point cloud segmentation. MinkowskiUNet [5] extends the voxel-based 3D sparse convolution to 4D or higher dimension convolution. However, 4D MinkowskiUNet requires high computation resources, and their overall improvements are also not satisfied. MeteorNet [31] deigns a chain-based point grouping to extract the scene flow of points in different frames. P4Transformer [32] designs a point-based 4D convolution and a transformer to fuse the spatial and temporal information. P4Transformer [32] and MeteorNet [31] require significant downsampling of the point cloud, making it difficult to apply to real-world outdoor point clouds. PSTNet [33]

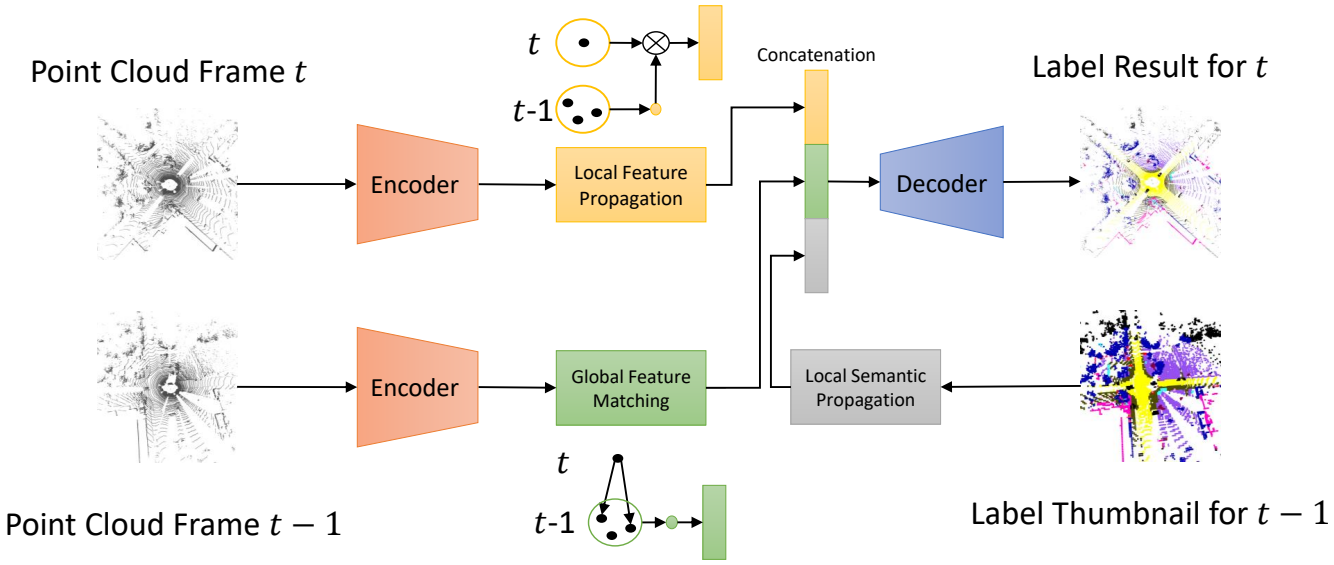


Fig. 3. **The structure of our proposed method.** The input data for both encoders of t and $t-1$ is a voxelized point cloud frame (3D sparse tensor). Label Thumbnail is a distribution statistic for regions in the point cloud.

proposes a point spatio-temporal convolution to process 4D information. PointMotionNet [34] further designs a pyramid-style network based on point spatio-temporal convolution and achieves leading performance. SpSequenceNet [6] designs a two-branch network and several modules to extract the temporal information. SpSequenceNet only considers local features and simple pooled global features from previous frames. Therefore, the performance of SpSequenceNet is limited. TVSN [7] improves the backbone network and some modules in SpSequenceNet [6]. TVSN [7] also designs a temporal voxel-point refiner to fuse the voxel-level predictions and low-level features in corresponding frames for high quality point-level predictions. SpSequenceNet [6] and TVSN [7] have a similar strategy for utilizing high-level temporal information. However, the utilization of temporal information in TVSN is still under-investigated.

We also investigate the usage of temporal information in other 4D tasks. SPU [35] design a gated feature fusion module to combine the features from multiple frames and upsample the target point cloud frame. Xiong et al. [36] utilize the temporal correlations to generate a guidance map and accelerate the video-based point cloud compression.

C. Temporal Feature Extraction in Video Object Segmentation

For a deeper exploration of temporal feature extraction, we discuss the 2D video object segmentation methods here, which have some common grounds. However, except for the data format, there are two differences between the 2D video object task and the 4D point cloud sequence task. Firstly, for a given street scene, the number of objects and pixels captured in a 2D video is typically lower than that in a 4D point cloud scenario. Secondly, the difference between objects in 2D videos and 3D point cloud sequences is that videos have RGB information while LiDAR-based point clouds only have the coordinates of points and their corresponding remission. One recent method in the 2D video object segmentation area is space-time memory [37]–[39]. Space-time memory designs a special memory mechanism to save the feature of previous image frames and uses attention-based or graph-based [40], [41] methods to generate a temporal feature from features of previous image frames. The memory mechanism retrieves the appearance features from the same object in the previous frame to support the prediction of the current frame.

III. Feature Propagation Network (FeatProp)

The task of 3D point cloud segmentation aims to predict the label $\mathbf{L}_t$ of each point $\mathbf{p}_{t,i}$ in $\mathbf{P}_t$. Note that t is the timestamp of the point cloud frame, and i is the index of point in the point cloud $\mathbf{P}_t$. The information of i-th point $\mathbf{p}_{t,i}$ consists of two components, the coordinate $\mathbf{c}_{t,i} = (x_{t,i}, y_{t,i}, z_{t,i})$ and the feature $f_{t,i}$. In the sparse convolution setting, the coordinate $\mathbf{c}_{t,i}$ of $\mathbf{P}_t$ is transformed into voxel grids and the points $\mathbf{p}_{t,i}$ in the same voxel grid are combined to one point $\mathbf{p}'_{t,i}$ with a certain combination rule. The voxelization process can be regarded as a voxelization sampling. Then, the transformed point cloud $\mathbf{P}'_t$ can be represented as a 3D tensor in the sparse convolution network and the output $\mathbf{L}'_t$ is also based on $\mathbf{P}'_t$. As our task here is 4D point cloud segmentation, we can use the information from $\mathbf{P}_{t'}, t' \leq t$ to support the prediction of $\mathbf{L}_t$.

Compared to 3D point cloud segmentation, designing an efficient algorithm to extract the temporal information is the key issue in 4D point cloud. We design three approaches to leverage the temporal information, which is shown in Figure 2. These approaches enhance the features in the current frame based on three different types of temporal information. Firstly,

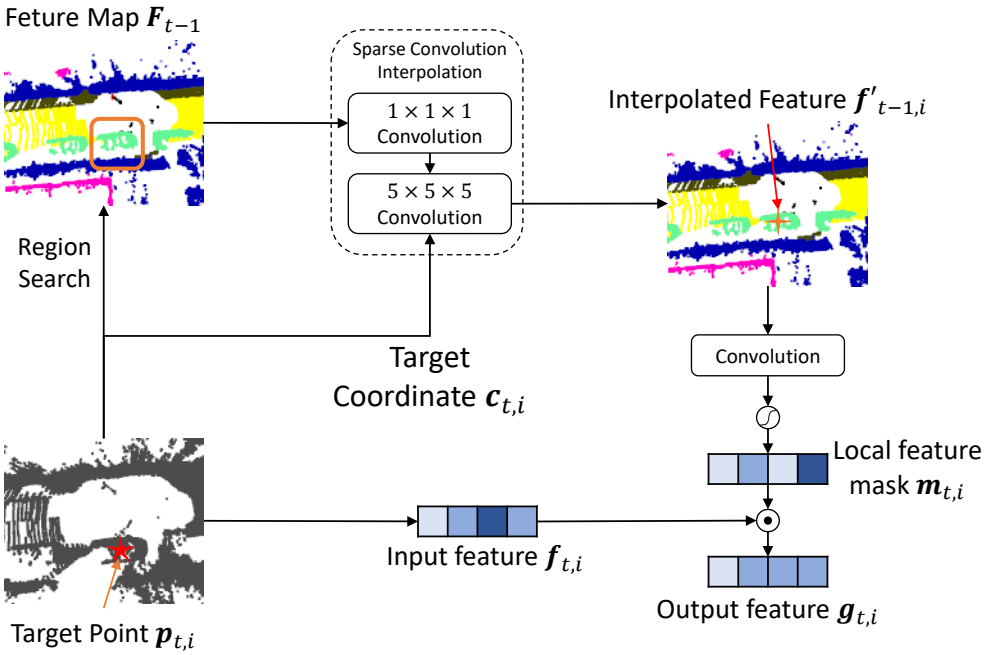


Fig. 4. **Local Feature Propagation.** Here is a simple example for local feature propagation on point cloud $\mathbf{P}_{t-1}$ and one point $\mathbf{p}_{t,i}$. The target point $\mathbf{p}_{t,i}$ is denoted as the the red star in the Point Cloud $\mathbf{P}_t$. The orange box represents the interpolation area in $\mathbf{P}_{t-1}$.

we propose a local feature propagation module to enhance the values of crucial features in the current frame with the local features from the previous frames. Secondly, the labels of points in previous frames are also an important clue for the predictions of the corresponding points in the current frame. We design local semantic propagation to propagate the semantic information from the previous frame to the current frame in local regions. In our experiment, the local semantic information from previous frames significantly improves the predictions of current frames. Thirdly, the global feature matching module selectively maps the same potential objects in different point cloud frames to build long-range connections and improve feature quality with matching results. Composing all the modules together, our method is illustrated in Figure 3.

FeatProp follows the structure of sparse 3D U-Net. In our implementation, the backbone network is MinkNet [5]. FeatProp uses the same encoder to encode the features of point cloud frame $\mathbf{P}_t$ and point cloud frame $\mathbf{P}_{t-1}$. Note that in the inference stage, we directly adapt the features from the inference step of $\mathbf{P}_{t-1}$ as the features of $\mathbf{P}_{t-1}$ in the inference step of $\mathbf{P}_t$. The encoder outputs of $\mathbf{P}_t$ and $\mathbf{P}_{t-1}$ are the input data for local feature propagation and global feature matching. Furthermore, the input of local semantic propagation is the label $\mathbf{L}_{t-1}$ from the inference step of $\mathbf{P}_{t-1}$. The decoder for $\mathbf{P}_t$ concatenates the outputs from local feature propagation, local semantic propagation, and global feature matching for the final prediction.

A. Local Feature Propagation

To extract and fuse the local context information, we design a module for Local Feature Propagation (LFP). The local feature propagation module generates a regional feature from the previous point cloud frame. Then, it generates an instructive feature mask $\mathbf{M}_{t-1}$ with this regional feature to select the crucial feature channel of $\mathbf{f}_{t,i}$ for each point $\mathbf{p}_{t,i}$. Figure 4 provides a simple explanation for local feature propagation. There are two steps, sparse convolution based interpolation and mask generation.

Sparse Convolution Based Interpolation. To generate regional features, we search the neighbours at the coordinate $\mathbf{c}_{t,i}$ of $\mathbf{p}_{t,i}$ in the point cloud P_{t-1}. With the neighbours in P_{t-1}, we interpolate a new regional feature $\mathbf{f}'_{t-1,i}$ as a local context feature with a highly efficient sparse convolution based interpolation. We show a comparison between the point based interpolation and sparse convolution based interpolation in Figure 5.

In the sparse convolution setting [5], [18], a sparse pooling or a sparse convolution on those specific points approximates the neighbour interpolation on these specific coordinates. The convolution based interpolation is

$$\mathbf{f}'_{t-1,i} = \sum_{\mathbf{j} \in \mathcal{N}^{\mathcal{D}}(\mathbf{c}_{t,i}, \mathbf{C}_{t-1})} \mathbf{W}_{\mathbf{j}} \cdot \mathbf{f}_{t-1,\mathbf{c}_{t,i}+\mathbf{j}} \; for \; \mathbf{c}_{t,i} \in \mathbf{C}_t, \quad (1)$$

where $\mathbf{C}_t$ and $\mathbf{C}_{t-1}$ are the coordinates sets of t and $t-1$, and $\mathcal{D}$ represents the $\mathcal{D}$-dimensional space. $\mathbf{f}_{t-1,\mathbf{c}_{t,i}+\mathbf{j}}$ is the features at the coordinate $\mathbf{c}_{t,i} + \mathbf{j}$ in P_{t-1}. $\mathcal{N}^{\mathcal{D}}(\mathbf{c}_{t,i}, \mathbf{C}_{t-1})$ is the coordinate set of the neighbour points at the coordinate $\mathbf{c}_{t,i}$ in the previous point cloud, which is $\{\mathbf{j} | \mathbf{c}_{t,i} + \mathbf{j} \in \mathbf{C}_{t-1}\}$. $\mathbf{W}_{\mathbf{j}}$ is the learnable weights of the grid at $\mathbf{j}$ in the convolution. Note that as the pose information is available in our task, $\mathbf{C}_t$ and $\mathbf{C}_{t-1}$ are aligned to be in the same coordinate system by default. Therefore, we combine a $1 \times 1 \times 1$ convolution and

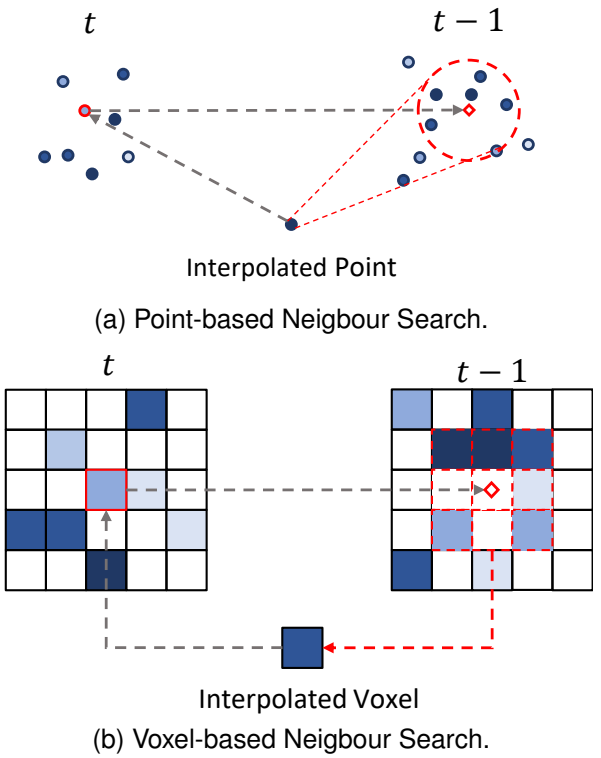


Fig. 5. **The comparison of sparse convolution based interpolation and normal point based interpolation.** Red dotted line represents the interpolation area.

a $5 \times 5 \times 5$ convolution to generate the interpolated features $\mathbf{f}'_{t-1,i}$.

As shown in Figure 5, a typical circuit of point interpolation is to search the neighbours of the target point in the local region and generate a new point to represent the point in the target area. The complexity of the neighbour searching algorithm is higher than $O(n^2)$ with matrix multiplication or $O(n \times n^{1-\frac{1}{k}})$ with predefined KDtree. On the other hand, the sparse convolution based interpolation is suitable for the voxel scenario. Compared to point interpolation, sparse convolution based interpolation checks neighbour voxels in the sparse 3D tensor as the neighbours. As we use a hash-based sparse convolution, the complexity of the voxel-based searching is $O(n)$, which is efficient in the voxel scenario.

Mask Generation. With the interpolated feature $\mathbf{f}'_{t-1,i}$, local feature propagation generates the mask $\mathbf{m}_{t,i}$ with the context feature $\mathbf{f}'_{t-1,i}$, which is formulated as:

$$\mathbf{m}_{t-1,i} = r \cdot \sigma(\mathbf{f}'_{t-1,i}), \tag{2}$$

where $\sigma(\cdot)$ is a sigmoid function and r is a hyperparameter. Note that we set r as 2 to rescale the sigmoid function to a range of 0 to 2. r avoids the feature modification when there is no point in the interpolation area. In our experiment, r slightly stabilized the training process. Based on $\mathbf{m}_{t,i}$, the output of local feature propagation is

$$\mathbf{g}_{t,i} = \mathbf{m}_{t-1,i} \odot \mathbf{f}_{t,i}, \tag{3}$$

where $\odot$ represents an element-wise multiplication.

Local feature propagation inherits the key sight of local cross-frame global attention in SpSequenceNet [6]. Cross-frame global attention guides the local feature generation with global summary information. However, outdoor point clouds contain complex environmental information and noise. A 256-dimensional channel-wise attention in cross-frame global attention barely summarizes the environment in outdoor point cloud scenes. Compared to global information, the local feature from the previous frame is more valuable for the feature generation of the current frame. Hence, our local feature propagation combines the guidance channel-wise attention and the local interpolation mechanism to improve the feature quality of the current frame.

B. Local Semantic Propagation

The semantic information from previous frames is the most valuable information in the time-related segmentation task, which has been widely used in 2D video object segmentation. We propose an algorithm that enhances the predictions of current frames by incorporating predicted semantics from previous frames. Initially, using the semantic predictions, we build the local label thumbnails of previous frames. Then, using the thumbnails, we generate local semantic features through local semantic propagation and fuse them with features of the current frames. With local semantic features from previous frames, the performance of our network is significantly improved. We also show an explanation in Fig 6.

Firstly, local semantic propagation transforms the label $l_{t-1,i}$ into the one-hot encoding $\mathbf{l}'_{t-1,i}$. The elements in $\mathbf{l}'_{t-1,i}$ is

$$l'_{t-1,i,k} = \begin{cases} 1 & k = l_{t-1,i} \\ 0 & k \neq l_{t-1,i} \end{cases}, k \leq n, \tag{4}$$

where n is the number of classes, and k is the index of label. Then, local semantic propagation transforms the voxelized coordinate $\mathbf{c}_{t-1,i}$ of point $\mathbf{p}_{t-1,i}$ with

$$\mathbf{c}'_{t-1,i} = (\lfloor \frac{x_{t-1,i}}{p} \rfloor, \lfloor \frac{y_{t-1,i}}{p} \rfloor, \lfloor \frac{z_{t-1,i}}{p} \rfloor), \tag{5}$$

where p is a down-sample scale parameter, which is 16 in our work. The local label thumbnail $\mathbf{d}_{t-1,o}$ is the mean value of all the one-hot encoding $\mathbf{l}'_{t-1,i}$ at the same coordinate $\mathbf{c}'_{t-1,i}$. We group all $\mathbf{d}_{t-1,o}$ as a new point cloud $\mathbf{D}_{t-1}$, which is the thumbnail for the prediction of $\mathbf{P}_{t-1}$. The thumbnail smooths the predictions of previous point cloud frames to reduce the effects of wrong predictions.

Ultimately, similar to local feature propagation, sparse convolution based interpolation is applied to get local semantic features with $\mathbf{D}_{t-1}$. Here, we use a $1 \times 1 \times 1$ convolution and a $3 \times 3 \times 3$ convolution to generate local semantic features of the coordinates in P_t. In the end, we concatenate local semantic features and other features together as the input for the next stage.

C. Global Feature Matching

Inspired by the memory network [37]–[39], [42], we design a soft matching to build long-range connections for objects

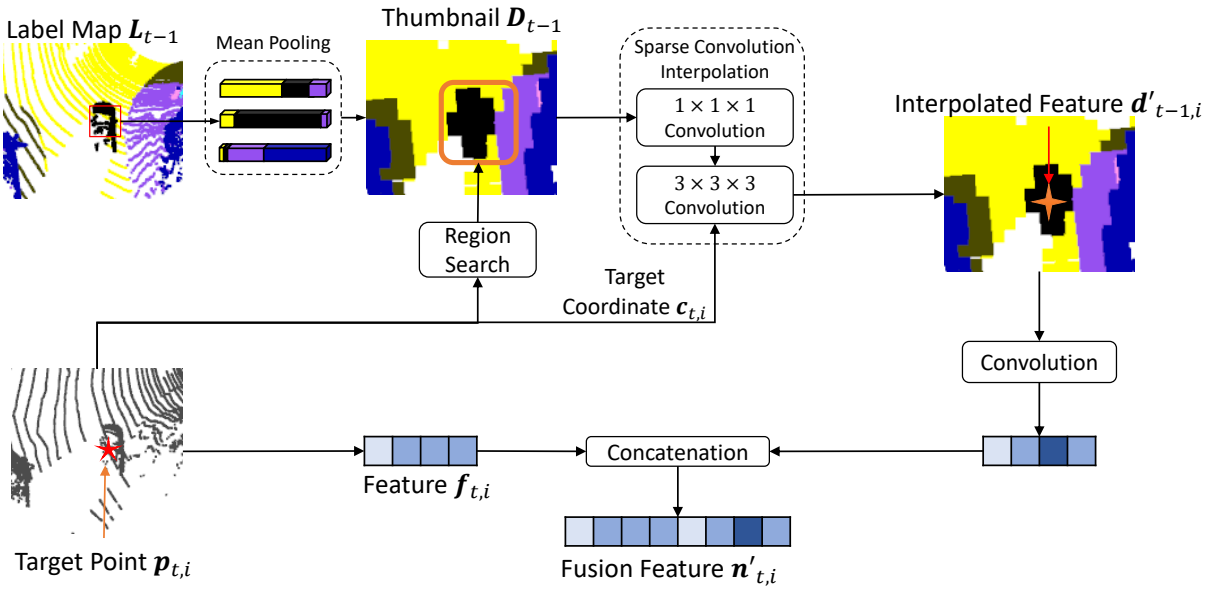


Fig. 6. **An explanation for local semantic propagation.** Similar to Figure 4, the red star is the target point and the area in the orange region is the interpolation area. Distribution map $\mathbf{D}_{t-1}$ is a statistics for $\mathbf{L}_{t-1}$.

in different point cloud frames. However, different from 2D video tasks, the massive number of points in a point cloud frame causes high noise, which affects the performance of the matching mechanism. Therefore, we also design an adjustment module to make global feature matching generate a sparse result to address the noise problem. We show the structure of global feature matching in Fig 7.

Firstly, the global feature matching module generates the key features $\mathbf{K}_t$ and $\mathbf{K}_{t-1}$ for both point cloud frames $\mathbf{P}_t$ and $\mathbf{P}_{t-1}$. The key generators are two residual blocks with shared parameters. The similarity function is:

$$s_{t,i,j} = \frac{exp(\alpha(\mathbf{k}_{t,i} \cdot \mathbf{k}_{t-1,j}))}{\sum_u exp(\alpha(\mathbf{k}_{t,i} \cdot \mathbf{k}_{t-1,u}))}. \tag{6}$$

Here α is a handcrafted parameter in the comparison, which is 2 in our network. With α, the difference of score map is enhanced to match a similar feature. Then, an adjustment function is applied on $\mathbf{S}_t$, as:

$$s'_{t,i,j} = \begin{cases} 0 & s_{t,i,j} \leq \frac{1}{m} \\ s_{t,i,j} & s_{t,i,j} > \frac{1}{m} \end{cases}, \tag{7}$$

where m is the number of points in $\mathbf{P}_{t-1}$. The adjustment function only considers the features whose probability is higher than the random guess. With the new sparse similarity map $\mathbf{S}_t$, the output feature for $\mathbf{P}_{t-1}$ is

$$\mathbf{v}_{t,i} = \sum_j s'_{t,i,j} \mathbf{f}_{t-1,j}. \tag{8}$$

$\mathbf{v}_{t,i}$ represents the matching result from $t-1$. Then, we concatenate $\mathbf{f}_{t,i}$ and $\mathbf{v}_{t,i}$ and use a residual block to fuse all the information together.

Our formulation can be seen as an extension of memory approach [42] in video object segmentation. The key idea of memory is to search the similar potential objects in the previous history. A video object segmentation task typically contains a few objects. In our task, an outdoor scene contains about 100k points and dozens of objects. The massive number of objects is beyond the capability of the naive memory approach. Furthermore, the differences between each object are significant with the RGB information, while a LiDAR-based point cloud contains the remission for the signal strength. Accordingly, we design an α and an adjustment function to generate a sparse matching result. The α in similarity function enhances the difference between the points. Nevertheless, as the formulation sums all the points in the previous frame, the long tail of negative matching results accumulates together and affects the feature quality of the target point. Accordingly, we design an adjustment function to eliminate the long tail effect.

IV. Evaluation

We evaluate our model on the multiple scan segmentation task on SemanticKITTI [4] and SemanticPOSS [8]. In SemanticKITTI, there are 43,551 LiDAR point cloud frames in 22 sequences. We use 9 sequences (19,130 frames) as the training set and validate our network with 1 sequence (4,071 frames). The remaining 12 sequences (20,351 frames) form the test dataset. In the multiple scan segmentation task, the model can infer with multiple scans, which is 1 scan in the single scan segmentation task on SemanticKITTI. The 25 labels contain 19 classes for static objects and 6 classes for some moving objects. SemanticPOSS is also a LiDAR outdoor scene point cloud dataset. SemanticPOSS contains 6 scenes with 2988 frames. Specifically, there are 2488 frames (5 scenes) in the training dataset and 500 frames (1 scene) in the validation dataset. SemanticPOSS contains a higher percentage of small objects than that in SemanticKITTI. The amount of small objects makes SemanticPOSS a challenging task.

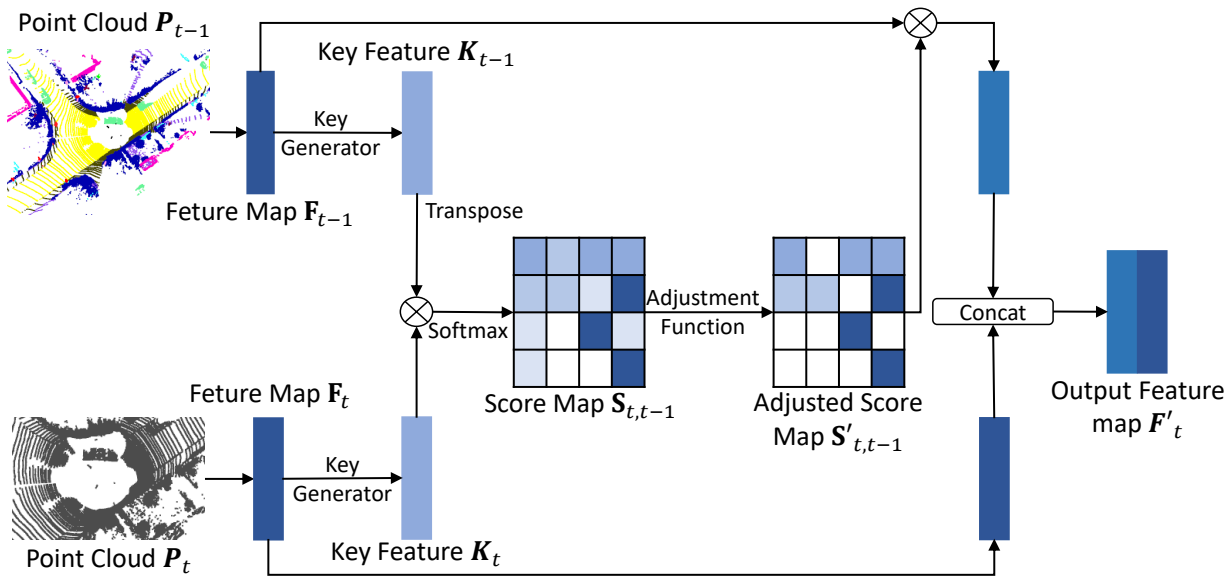


Fig. 7. **The structure of global feature matching.** Here, global feature matching generates two key features for F_{t-1} and F_t with two key generators, which share the parameters. $\otimes$ is the dot product operation.

A. Training Pipeline

In the training step, one problem is the choice of predictions L_{t-1} for the input of local semantic propagation. The first choice is to directly use the ground truth of $t-1$ as L_{t-1}. However, the network overfits the ground truth of $t-1$ and the performance in the inference stage is poor without the ground truth. Furthermore, in the video object segmentation [38], [39], [42], the choice is the predictions from the previous frame and they use a sequential training pipeline to maintain the predictions in the memory. The cost of a sequential training pipeline is high, especially in 4D point cloud segmentation setting. Consequently, we design a special non-sequential training pipeline based on the approximation mid prediction L_{t-1}.

In the first training epoch, we randomly set the previous frame P_{t-1} as an empty point cloud without any point or prediction with an 80% possibility, which approximates the P_t as the first frame P_0. Otherwise, we use the previous frame P_{t-1} and set the ground truth as the labels L_{t-1} to train our proposed model. Therefore, the first training epoch warms up all the modules and limits the usage of ground truth in case of the overfitting issue. Then, before starting the next training epoch, we sequentially predict all the point cloud frames in the training dataset and store the predictions as mid predictions.

Then, we apply a data augmentation mechanism for the prediction L_{t-1} in the following training epoch. We randomly select the value of the prediction L_{t-1} in three different ways. Firstly, we set the ground truth of P_{t-1} as the prediction L_{t-1} for the model training. Secondly, similar to the first training epoch, we randomly set the previous frame P_{t-1} as an empty point cloud to approximate the P_t as the first frame P_0. Thirdly, we set the mid predictions as approximate predictions L_{t-1} of the previous frame. The data augmentation mechanism effectively prevents overfitting on the ground truth of $t-1$ and training with time-consuming sequential training pipelines.

Furthermore, we update the mid prediction every 10 epochs for simulating the predictions of the previous frame from the model trained in the respective training epochs. In the inference stage, we apply the segmentation sequentially and directly use the predictions from the previous point cloud frame as L_{t-1}.

B. Implementation Details

In the pre-processing phase, we align P_{t-1} with P_t so that P_{t-1} and P_t are in the same coordinate system. Then, we apply random sampling, rotation and shifting on P_{t-1} and P_t using the same setting. Then, we voxelize P_{t-1} and P_t with the scale unit $0.05m$, and the voxelized P_{t-1} and P_t are the input of FeatProp. Before training the FeatProp, we pre-train our backbone network using a single frame on the multiple scan segmentation task of SemanticKITTI. In the training phase, we use an Adam optimizer to train the FeatProp. The usage of GPU memory is around 8GB for batch size 2. The device of our experiment is one Intel Core i9-9900X Processor and one Nvidia RTX 2080Ti. To have a better performance, we use P_0 itself as P_{-1} and the prediction from the backbone network (single frame prediction) as L_{t-1}. In our experiment, it yields around 0.2% improvement for the result on the validation set.

C. Results on SmeanticKITTI Multi-Scan Task

As shown in Table I, we compare our method to existing methods on the multiple scan segmentation task of SemanticKITTI. Note that TangentConv [43] and DarkNet53Seg [4] combine 5 point cloud frames as one point cloud to predict the label of each point. The performance of our backbone network is 0.9% lower than that of TemporalLidarSeg [44]. For more details, our method has a significant improvement in the foreground objects, like car, bicycle and person. Simultaneously, the improvements in the

	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 [43]	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 [4]	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
SpSequenceNet [6]	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
TemporalLidarSeg [44]	47.0	92.1	47.7	40.9	39.2	35.0	14.4	0.0	0.0	91.8	59.6	75.8	23.2	89.8	63.8	82.3	62.5	64.7	52.6	60.4	58.2	42.8	40.4	12.9	12.4	2.1
TVSN [7]	52.5	93.5	45.6	42.8	36.9	35.7	15.2	0.0	0.0	89.9	67.9	73.5	21.4	92.3	69.5	85.0	70.3	68.5	61.2	65.4	74.2	65.5	55.6	63.0	14.8	5.2
PointMotionNet [34]	54.8	90.6	59.3	59.9	70.9	29.9	13.7	23.1	23.4	91.7	65.5	76.1	24.2	90.1	63.5	84.9	70.9	70.1	62.2	64.0	71.4	62.2	61.8	15.1	29.9	10.5
Backbone (MinkNet14 [5])	46.1	90.9	30.2	39.8	29.1	28.4	7.9	0.0	0.0	88.6	56.5	70.9	15.4	88.3	59.0	84.6	66.7	68.0	56.5	57.9	61.4	52.6	43.7	34.0	20.2	1.8
FeatProp (Ours)	52.7	93.8	56.2	33.9	49.0	36.7	17.4	0.0	0.0	88.4	61.5	72.2	32.8	91.3	68.2	84.2	70.8	67.5	60.1	66.5	80.3	63.5	51.7	39.9	27.1	4.9

TABLE I

OUR TEST RESULTS ON THE MULTIPLE SCAN SEGMENTATION TASK OF SEMANTICKITTI. AS OUR TASK IS A 4D SEGMENTATION TASK, THE RESULT IS BASED ON THE MULTIPLE SCAN SEGMENTATION TEST SET OF SEMANTICKITTI. FOR THE FOURTH ROW, MINKNET14 [5] IS THE BACKBONE NETWORK IN OUR IMPLEMENTATION. MINKNET14 [5] ONLY USES ONE FRAME AS THE INPUT. COMPARED TO THE BACKBONE NETWORK, FEATPROP HAS A 6.6% IMPROVEMENT, WHICH IS 52.7% ON MIOU. THE IMPROVEMENT FROM OUR PROPOSED METHODS SHOWS THE EFFICIENCY OF THE TEMPORAL FEATURE FUSION.

background objects, like road, parking and building, are not as significant as the improvement for moving objects. For our proposed method, FeatProp achieves absolute mIoU boosts of 6.6% over the backbone network and 5.2% over TemporalLidarSeg [44]. The performance of our previous work, TVSN [7] is 0.2% lower than FeatProp while TVSN uses a deeper backbone network (34-layer MinkUNet [5]) than FeatProp (14-layer MinkUNet). PointMotionNet [34] achieves an improvement of 2.1% over our work, mostly coming from improved predictions of bicyclists and motorcyclists. In contrast, all other compared approaches including our backbone network and our proposed method are unable to detect these two classes. PointMotionNet designs a point-based motion prediction branch to predict the motion statuses of the current frame with multiple consecutive frames and then trains one more branch for segmentation predictions. Using our voxel-based backbone network, our approach directly propagates and fuses the network outputs from the previous frame with the features of the current frame. Therefore, our method requires fewer computation resources and has lower computational complexity.

Furthermore, we also show qualitative results in Figure 8. The qualitative results is an evidence for the conclusion from the quantitative results. For the objects related to the motion status, our model has a high performance. In all five frames, the predictions of points on the road, which have a high probability of being a car or a moving object, are similar to the ground truth. For background objects, two failure cases happen in Frame a and Frame c, and the results for other cases also have a high quality.

To further investigate the effectiveness of global feature matching, we show matching qualitative results in Figure 9. We sample the associated pairs of points with high similarity scores in two corresponding frames. These associated pairs of points represent the matching results in global feature matching. In all three examples, global feature matching associates the points from the same objects with high quality even though there is no additional object-level supervision.

D. Ablation Experiements

We show a comparison result in Table II. The result is based on the validation dataset of SemanticKITTI. Without any proposed modules, the result of the backbone network is 46.8% in terms of mIoU. Local feature propagation improves performance by 2.3% compared to the backbone network alone. Local semantic propagation contributes around 6.0% improvement over the backbone network and a 4.9% improvement over the backbone network using local feature propagation. The overall improvement from our local approaches is 7.2%. These results demonstrate that temporal information extracted from local regions can effectively support the network in understanding the scene.

	mIoU
Backbone	46.8
Backbone+LFP	49.1
Backbone+LSP	52.8
Backbone+LFP+LSP	54.0
Backbone+LSP+GFM	54.7
Backbone+LFP+LSP+GFM_NO_A	54.2
Backbone+LFP+LSP+GFM	**55.5**

TABLE II

THE COMPARISON RESULTS OF EACH MODULE ON THE VALIDATION DATASET OF SEMANTICKITTI. HERE LFP IS LOCAL FEATURE PROPAGATION, LSP IS LOCAL SEMANTIC PROPAGATION AND GFM IS GLOBAL FEATURE MATCHING. GFM_NO_A IS A VERSION OF GLOBAL FEATURE MATCHING WITHOUT THE ADJUSTMENT MODULE.

The global feature matching module has a 1.5% improvement (from 54.0% to 55.5%) using local feature propagation, and a 1.9% improvement (from 52.8% to 54.7%) without local feature propagation. Furthermore, we evaluate the effectiveness of the adjustment function for global temporal feature matching. As stated in Section III-C, the adjustment function is designed to generate a sparse score map in global feature matching. We remove the adjustment function and the α parameter in global feature matching, which is GFM_NO_A in Table II. GFM_NO_A is similar to the matching mechanism in STM [42].

This naive version of global feature matching shows only 0.2% improvement over Backbone+LFP+LSP. In our observation, there are two reasons for the performance gap between the two versions of global feature matching. Firstly, the huge number of candidates causes difficulty in matching a

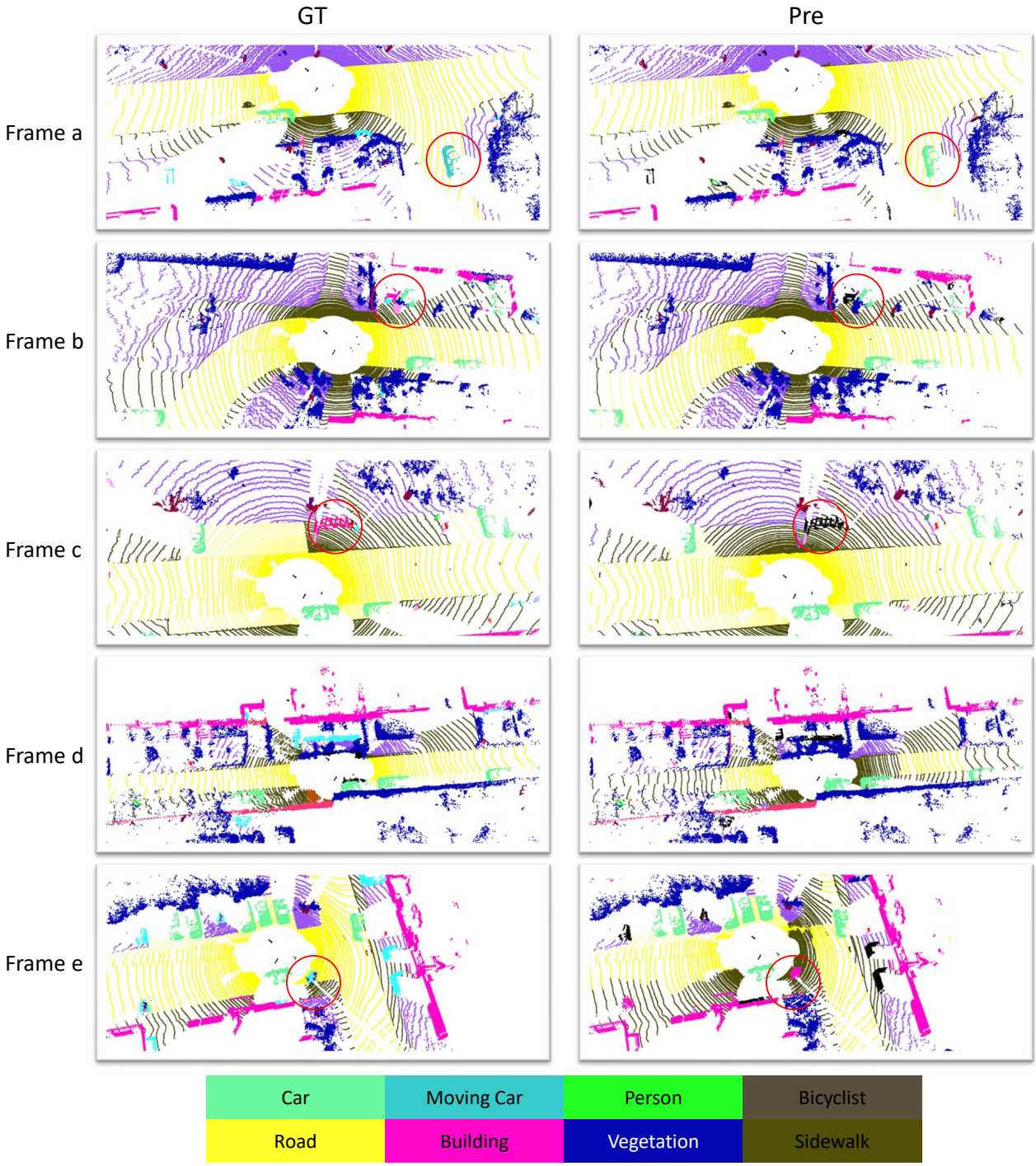


Fig. 8. **Qualitative Results on the validation dataset of SemanticKITTI.** The first column is the ground truth of five examples and the second column is the predictions. The yellow parts are the points for the road. There are several failure cases in the red circles.

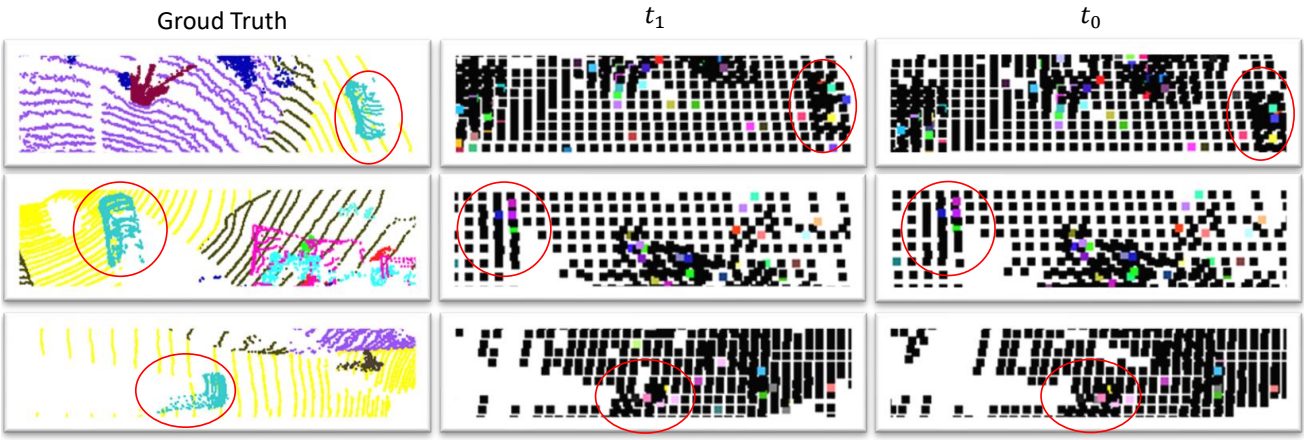


Fig. 9. **Matching Quality of Global Feature Matching.** We sample the pairs of points with high matching scores in frames t_0 and t_1. Note that coloured points represent matching results, and points with the same colour in frames t_0 and t_1 are associated as pairs of matching results with high scores. In each red area, global feature matching associates the points from the same object in frames t_0 and t_1

	mIoU	People	Rider	Car	Traffic Sign	Trunk	Plants	Pole	Fence	Building	Bike	Road
PointNet++ [1]	20.1	20.8	0.1	8.9	21.8	4.0	51.2	3.2	6.0	42.7	0.1	62.2
SqueezeSegV2 [45]	29.8	18.4	11.2	34.9	11.0	15.8	56.3	4.5	25.5	47.0	32.4	71.3
RandLA-Net [46]	53.5	69.2	26.7	77.2	24.3	27.3	73.4	30.9	51.4	82.5	43.9	81.2
KPConv [2]	55.2	77.3	29.4	78.2	23.0	28.1	71.6	32.4	51.5	81.8	53.2	80.6
JS3C-Net [47]	60.2	80.0	39.1	83.2	28.9	31.6	76.1	44.7	54.0	83.9	59.8	81.7
TVSN [7]	63.0	68.3	62.7	77.6	28.6	49.1	80.6	33.8	62.8	82.7	65.0	81.8
FeatProp (Ours) + $5cm$	61.9	74.0	70.0	44.2	47.5	40.2	82.3	35.1	63.3	79.4	64.2	81.6
FeatProp (Ours) + $10cm$	62.5	72.4	69.6	55.2	47.1	39.0	81.4	34.5	62.9	80.3	63.9	81.8

TABLE III
THE TEST RESULTS ON THE AREA 3 OF SEMANTICPOSS. THE PERFORMANCE OF FEATPROP ACHIEVES A 2.3% IMPROVEMENT OVER JS3C-NET [47].

useful feature globally in point cloud frames. The number of candidates is around 9 thousand for each point cloud frame in our framework and usually 576 for one 384*384 image frame at the end of a ResNet encoder. The quantitative gap limits the effectiveness of softmax-style soft matching approaches. Secondly, the difference between features in the LiDAR point cloud is lower than the feature for an RGB image, and it also increases the requirement of a matching algorithm.

E. Results on SemanticPOSS

In Table III, we show the results based on SemanticPOSS. $5cm$ and $10cm$ are the voxelization resolution of our network. Note that JS3C-Net [47] merges 70 consecutive frames together to predict the segmentation label and complete the scene. The merge process is one basic method to utilize temporal information, which usually requires high usage of computation recourses and memory. With our proposed modules, our FeatProp achieves an improvement of 2.3% over JS3C-Net. The performance of FeatProp is 0.5% lower than that of TVSN. As SemanticPOSS is obtained by using a 40-line LiDAR, the point clouds in SemanticPOSS are more sparse than those in SemanticKITTI. Hence, a lower resolution, like $10cm$, shows higher overall performance while a higher resolution, $5cm$, shows slight advantages in some classes.

V. CONCLUSION

We propose a novel method FeatProp to improve the performance in 4D point cloud segmentation. To fully explore effective temporal dependencies across two consecutive point clouds, we define different types of temporal information and design three novel modules to propagate the temporal information from the previous point cloud frame to the current inference stage. Experimental results on SemanticKITTI and SemanticPOSS demonstrate the superior effectiveness of our method in utilizing temporal information to improve the 4D semantic segmentation.

VI. ACKNOWLEDGEMENT

This research is supported by the MoE AcRF Tier 2 grant (MOE-T2EP20220-0007) and MoE AcRF Tier 1 grants (RG14/22, RG95/20), and the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-003). This research is also supported by the Agency for Science, Technology and Research (A*STAR), Singapore under its MTC Young Individual Research Grant (Grant No. M21K3c0130)

REFERENCES

[1] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[2] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6411–6420.

[3] Z. Zhang, B.-S. Hua, and S.-K. Yeung, "Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1607–1616.

[4] 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 *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9297–9307.

[5] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

[6] H. Shi, G. Lin, H. Wang, T.-Y. Hung, and Z. Wang, "Spsequencenet: Semantic segmentation network on 4d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4574–4583.

[7] S. Hanyu, W. Jiacheng, W. Hao, L. Fayao, and L. Guosheng, "Learning spatial and temporal variations for 4d point cloud segmentation," *arXiv preprint arXiv:2207.04673*, 2022.

[8] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao, "Semanticposs: A point cloud dataset with large quantity of dynamic instances," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 687–693.

[9] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[10] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg, "Deep projective 3d semantic segmentation," in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2017, pp. 95–107.

[11] A. Boulch, B. Le Saux, and N. Audebert, "Unstructured point cloud semantic labeling using deep segmentation networks." *3DOR*, vol. 2, p. 7, 2017.

[12] B. Wu, A. Wan, X. Yue, and K. 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)*. IEEE, 2018, pp. 1887–1893.

[13] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. 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)*. IEEE, 2019, pp. 4376–4382.

[14] C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka, "Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*. Springer, 2020, pp. 1–19.

[15] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "Splatnet: Sparse lattice networks for point cloud processing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2530–2539.

[16] B. Graham, "Spatially-sparse convolutional neural networks," *arXiv preprint arXiv:1409.6070*, 2014.

[17] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3577–3586.

[18] B. Graham, M. Engelcke, and L. van der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," *CVPR*, 2018.

[19] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu, "Pointsift: A sift-like network module for 3d point cloud semantic segmentation," *arXiv preprint arXiv:1807.00652*, 2018.

[20] D. Li, G. Shi, Y. Wu, Y. Yang, and M. Zhao, "Multi-scale neighborhood feature extraction and aggregation for point cloud segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 6, pp. 2175–2191, 2020.

[21] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[22] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in Neural Information Processing Systems*, 2018, pp. 820–830.

[23] H. Liu, Y. Guo, Y. Ma, Y. Lei, and G. Wen, "Semantic context encoding for accurate 3d point cloud segmentation," *IEEE Transactions on Multimedia*, vol. 23, pp. 2045–2055, 2020.

[24] L. Zhao and W. Tao, "Jsnet++: Dynamic filters and pointwise correlation for 3d point cloud instance and semantic segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[25] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9621–9630.

[26] C. Chen, S. Qian, Q. Fang, and C. Xu, "Hapgn: Hierarchical attentive pooling graph network for point cloud segmentation," *IEEE Transactions on Multimedia*, vol. 23, pp. 2335–2346, 2020.

[27] P. de Oliveira Rente, C. Brites, J. Ascenso, and F. Pereira, "Graph-based static 3d point clouds geometry coding," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 284–299, 2018.

[28] Z. Song, L. Zhao, and J. Zhou, "Learning hybrid semantic affinity for point cloud segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[29] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[30] G. Zhu, L. Zhang, P. Shen, J. Song, S. A. A. Shah, and M. Bennamoun, "Continuous gesture segmentation and recognition using 3dcnn and convolutional lstm," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1011–1021, 2018.

[31] X. Liu, M. Yan, and J. Bohg, "Meteornet: Deep learning on dynamic 3d point cloud sequences," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9246–9255.

[32] H. Fan, Y. Yang, and M. Kankanhalli, "Point 4d transformer networks for spatio-temporal modeling in point cloud videos," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 204–14 213.

[33] H. Fan, X. Yu, Y. Ding, Y. Yang, and M. Kankanhalli, "Pstnet: Point spatio-temporal convolution on point cloud sequences," in *International Conference on Learning Representations*, 2021.

[34] J. Wang, X. Li, A. Sullivan, L. Abbott, and S. Chen, "Pointmotionnet: Point-wise motion learning for large-scale lidar point clouds sequences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4419–4428.

[35] K. Wang, L. Sheng, S. Gu, and D. Xu, "Sequential point cloud upsampling by exploiting multi-scale temporal dependency," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4686–4696, 2021.

[36] J. Xiong, H. Gao, M. Wang, H. Li, and W. Lin, "Occupancy map guided fast video-based dynamic point cloud coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 813–825, 2021.

[37] Z. Zhou, L. Ren, P. Xiong, Y. Ji, P. Wang, H. Fan, and S. Liu, "Enhanced memory network for video segmentation," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[38] Q. Zhou, Z. Huang, L. Huang, Y. Gong, H. Shen, W. Liu, and X. Wang, "Motion-guided spatial time attention for video object segmentation," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[39] Z. Yang, Y. Wei, and Y. Yang, "Collaborative video object segmentation by foreground-background integration," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*. Springer, 2020, pp. 332–348.

[40] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. Van Gool, "Video object segmentation with episodic graph memory networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 661–679.

[41] Z. Zhang, D. Xu, W. Ouyang, and L. Zhou, "Dense video captioning using graph-based sentence summarization," *IEEE Transactions on Multimedia*, vol. 23, pp. 1799–1810, 2020.

[42] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9226–9235.

[43] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou, "Tangent convolutions for dense prediction in 3d," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3887–3896.

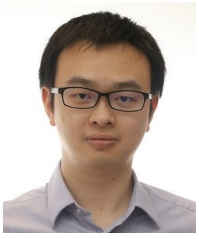
[44] F. Duerr, M. Pfaller, H. Weigel, and J. Beyerer, "Lidar-based recurrent 3d semantic segmentation with temporal memory alignment," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 781–790.

[45] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud," in *ICRA*, 2019.
[46] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
[47] X. Yan, J. Gao, J. Li, R. Zhang, Z. Li, R. Huang, and S. Cui, "Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, 2021, pp. 3101–3109.

Hanyu Shi is a PhD candidate at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received his B.Eng. degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 2016 and M.Sc degree in information system from Nanyang Technological University, Singapore, in 2017. His research interests include computer vision, 3D point cloud scene understanding and weakly point cloud semantic segmentation.

Ruibo Li is a PhD candidate with the School of Computer Science and Engineering, Nanyang Technological University (NTU). He received his B.Eng. degree and M.S. degree from Huazhong University of Science and Technology, Wuhan, China in 2016 and 2019 respectively. His research interests are in computer vision and machine learning.

Fayao Liu is a research scientist at Institute for Infocomm Research (I2R), A*STAR, Singapore. She received her PhD in computer science from the University of Adelaide, Australia in Dec. 2015. Before that, she obtained her B.Eng. and M.Eng. degrees from National University of Defense Technology, China in 2008 and 2010 respectively. She mainly works on machine learning and computer vision problems, with particular interests in self-supervised learning, few-shot learning and generative models.



Guosheng Lin is an Assistant Professor at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received his PhD degree from The University of Adelaide in 2014. His research interests are in computer vision and machine learning.