MarS3D: A Plug-and-Play Motion-Aware Model for Semantic Segmentation on Multi-Scan 3D Point Clouds

Jiahui Liu[1]* Chirui Chang[1]* Jianhui Liu[1] Xiaoyang Wu[1] Lan Ma[2] Xiaojuan Qi[1]†
[1]The University of Hong Kong [2]TCL AI Lab
{liujh, xjqi}@eee.hku.hk, crchang@hku.hk, jhliu0212@gmail.com, xywu3@cs.hku.hk, rubyma@tcl.com

Abstract

3D semantic segmentation on multi-scan large-scale point clouds plays an important role in autonomous systems. Unlike the single-scan-based semantic segmentation task, this task requires distinguishing the motion states of points in addition to their semantic categories. However, methods designed for single-scan-based segmentation tasks perform poorly on the multi-scan task due to the lacking of an effective way to integrate temporal information. We propose MarS3D, a plug-and-play motion-aware module for semantic segmentation on multi-scan 3D point clouds. This module can be flexibly combined with single-scan models to allow them to have multi-scan perception abilities. The model encompasses two key designs: the Cross-Frame Feature Embedding module for enriching representation learning and the Motion-Aware Feature Learning module for enhancing motion awareness. Extensive experiments show that MarS3D can improve the performance of the baseline model by a large margin. The code is available at https://github.com/CVMI-Lab/MarS3D.

1. Introduction

3D semantic segmentation on multi-scan large-scale point clouds is a fundamental computer vision task that benefits many downstream problems in autonomous systems, such as decision-making, motion planning, and 3D reconstruction, to name just a few. Compared with the single-scan semantic segmentation task, this task requires understanding not only the semantic categories but also the motion states (*e.g.*, moving or static) of points based on multi-scan point cloud data.

In the past few years, extensive research has been conducted on single-scan semantic segmentation with significant research advancements [4, 5, 12, 25, 31, 33, 36]. These approaches are also applied to process multi-scan point clouds, wherein multiple point clouds are fused to form a single point cloud before being fed to the network for processing. Albeit simple, this strategy may lose temporal information and make distinguishing motion states a challenging problem. As a result, they perform poorly in classifying the motion states of objects. As shown in Figure 1, the simple point cloud fusion strategy cannot effectively enable the model to distinguish the motion states of cars even with a state-of-the-art backbone network SPVCNN [25]. Recently, there have been some early attempts [7, 23, 24, 28] to employ attention modules [24] and recurrent networks [7,23,28] to fuse information across different temporal frames. However, these approaches do not perform well on the multi-scan task due to the insufficiency of temporal representations and the limited feature extraction ability of the model.

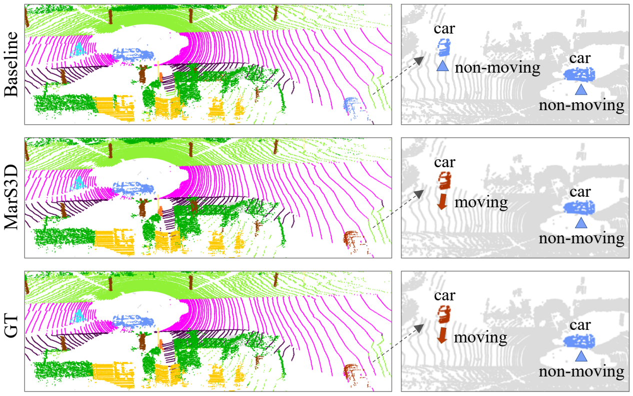

Figure 1. Comparison of our proposed method, MarS3D, with baseline method using SPVCNN [25] as the backbone on SemanticKITTI [1] dataset. MarS3D achieves excellent results in the classification of semantic categories and motion states, while the baseline method can not distinguish motion well from static.

In sum, a systematic investigation of utilizing the rich spatial-temporal information from multiple-point cloud scans is still lacking. This requires answering two critical questions: (1) how can we leverage the multi-scan information to improve representation learning on point clouds for

*Equal contribution.
†Corresponding author.

better semantic understanding? and (2) how can the temporal information be effectively extracted and learned for classifying the motion states of objects?

In this paper, we propose a simple plug-and-play **M**otion-**a**wa**r**e **S**egmentation module for **3D** multi-scan analysis (**MarS3D**), which can seamlessly integrate with existing single-scan semantic segmentation models and endow them with the ability to perform accurate multi-scan 3D point cloud semantic segmentation with negligible computational costs. Specifically, our method incorporates two core designs: First, to enrich representation learning of multi-frame point clouds, we propose a Cross-Frame Feature Embedding (CFFE) module which embeds time-step information into features to facilitate inter-frame fusion and representation learning. Second, inspired by the observation that objects primarily move along the horizontal ground plane (*i.e.*, *xy*-plane) in large-scale outdoor scenes, *i.e.*, minimal motion along the *z*-axis, we propose a Motion-Aware Feature Learning (MAFL) module based on Bird's Eye View (BEV), which learns the motion patterns of objects between frames to facilitate effectively discriminating the motion states of objects.

We extensively evaluate our approach upon several mainstream baseline frameworks on SemanticKITTI [1] and nuScenes [3] dataset. It consistently improves the performance of the baseline approaches, *e.g.*, MinkUnet [5], by 6.24% in mIoU on SemanticKITTI with a negligible increase in model parameters, *i.e.*, about 0.2%. The main contributions are summarized as follows:

- We are the first to propose a plug-and-play module for large-scale multi-scan 3D semantic segmentation, which can be flexibly integrated with mainstream single-scan segmentation models without incurring too much cost.
- We devise a Cross-Frame Feature Embedding module to fuse multiple point clouds while preserving their temporal information, thereby enriching representation learning for multi-scan point clouds.
- We introduce a BEV-based Motion-Aware Feature Learning module to exploit temporal information and enhance the model's motion awareness, facilitating the prediction of motion states.
- We conduct extensive experiments and comprehensive analyses of our approach with different backbone models. The proposed model performs favorably compared to the baseline methods while introducing negligible extra parameters and inference time.

2. Related Work

Single-scan Outdoor 3D Semantic Segmentation: 3D single-scan outdoor semantic segmentation is indispensable for autonomous driving. In early work, PointNet [19] uses Multi-Layer Perception (MLP) to extract features from input point clouds directly, and PointNet++ [20] tries to incorporate multi-scale designs for dense prediction tasks. Later, various literature [19, 20, 26, 29, 30, 32] works on designing point-based convolution on either geometric or semantic neighborhoods. To handle the large-scale dataset, some works [4, 5, 10, 12, 25, 31, 36] focus on volumetric features and use 3D convolution to achieve a balance between accuracy and efficiency. SparseConv [10] and MinkUNet [5] are representative works and demonstrate good performance. Later, SPVNAS [25] combines voxel and point representations and designs a neural architecture search method to find the optimal model structure. Recently, Cylinder3D [36] introduces a cylindrical partition to leverage the properties of LiDAR point clouds for enriching the feature information.

The remarkable feature extraction capability enables the above methods to achieve high performance on single-scan tasks. To solve the multi-scan task, most of these methods [4, 12, 31, 34, 36] first fuse multiple point clouds into one and treat the fused point cloud as a single point cloud for processing. Albeit simple, this fusion strategy overlooks important temporal information and entangles moving and non-moving objects, leading to performance degradation.

Multi-scan Outdoor 3D Semantic Segmentation: Compared to single-scan semantic segmentation, the multi-scan task needs to discriminate the moving and stationary states of the objects based on temporal information. In addition to the simple fusion strategy discussed above, another stream of approaches [7, 23, 24, 28] attempts to process each point cloud in a sequence separately and fuse the feature representations for temporal modeling. For instance, SpSequenceNet [24] proposes a U-Net-based architecture to extract per-frame features and combine features of two consecutive frames to gather temporal information. The fused feature is further fed into the prediction head to produce results. Duerr *et al*. [7] design a recurrent architecture with a temporal memory alignment module for sequential processing of multiple point clouds. TemporalLatticeNet [23] proposes to match similar feature patterns between adjacent frames and fuse them temporally. However, these approaches cannot fully leverage multiple point clouds to enrich temporal representation learning as the feature extraction is still conducted on each frame separately.

BEV-based 3D Point Cloud Perception: Recently, BEV-based representation [11, 14, 16, 35, 37, 38] has emerged as an effective way to process 3D point cloud due to its efficiency and ease of deployment using 2D convolution operations. By projecting a 3D point cloud into the bird's eye view (BEV), BEV-based representation converts 3D representations into 2D to avoid heavy processing in 3D and improve computation and memory efficiency. Some representative works include PointPillars [14] for 3D object de-

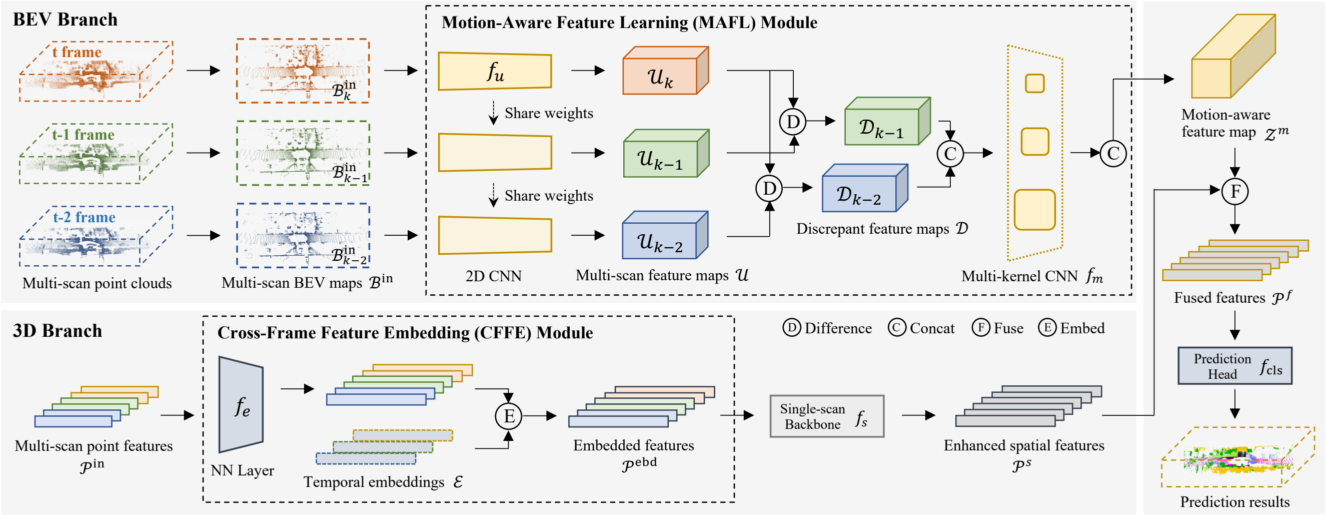
BEV Branch
t frame
t-1 frame
t-2 frame
Multi-scan point clouds
$\mathcal{B}_k^{\text{in}}$
$\mathcal{B}_{k-1}^{\text{in}}$
$\mathcal{B}_{k-2}^{\text{in}}$
Multi-scan BEV maps $\mathcal{B}^{\text{in}}$
Motion-Aware Feature Learning (MAFL) Module
f_u
Share weights
Share weights
2D CNN
$\mathcal{U}_k$
$\mathcal{U}_{k-1}$
$\mathcal{U}_{k-2}$
Multi-scan feature maps $\mathcal{U}$
D
D
$\mathcal{D}_{k-1}$
$\mathcal{D}_{k-2}$
Discrepant feature maps $\mathcal{D}$
C
C
Multi-kernel CNN f_m
Motion-aware feature map $\mathcal{Z}^m$
F
Fused features $\mathcal{P}^f$
Prediction Head f_{cls}
Prediction results
3D Branch
Multi-scan point features $\mathcal{P}^{\text{in}}$
Cross-Frame Feature Embedding (CFFE) Module
f_e
NN Layer
Temporal embeddings $\mathcal{E}$
E
Embedded features $\mathcal{P}^{\text{ebd}}$
Single-scan Backbone f_s
Enhanced spatial features $\mathcal{P}^s$
D Difference
C Concat
F Fuse
E Embed

Figure 2. The proposed framework of MarS3D. We take three scan point clouds inputs as illustration. MarS3D contains two branches. One is the BEV branch with 2D BEV representations as input, and it employs a Motion-Aware Feature Learning (MAFL) module to enhance the motion-aware feature learning. The other is the 3D branch that takes multi-scan 3D point clouds as input, enriching the feature representations with our Cross-Frame Feature Embedding (CFFE) module. The fused features of the above two modules are fed into the prediction head and make final predictions on semantic categories and motion states.

tection, BEVfusion [16] for multi-sensor fusion, BEV projection [38] for 3D segmentation. The above work demonstrates the potential of BEV representations in outdoor 3D scene analysis. Here, we explore utilizing BEV-based representation to extract temporal information for analyzing the motion states of objects.

2D Video Semantic Segmentation: In video segmentation, many efforts [2, 6, 13, 15, 17, 18, 21] have been made to conduct temporal feature extraction. Oh *et al.* [18] introduce a space-time memory network to integrate features from adjacent frames for video object segmentation and attains significant performance gains. TDNet [13] exploits temporal redundancy for fast video segmentation by developing an attention propagation method to propagate features to adjacent frames. Although they achieve promising results on video segmentation, we experimentally demonstrate these designs are not optimal for multi-scan point cloud segmentation. We focus on designing new methods tailored to point clouds for better performance and efficiency.

3. Problem Statement

Given a sequence of LiDAR point clouds as inputs, the multi-scan 3D point cloud semantic segmentation task aims to assign a semantic category to each point and predict their motion states (*i.e.*, moving or static). Specifically, a LiDAR point cloud frame contains a set of unordered points that are annotated with labels for training. We denote a pair of training data as $(\mathcal{P}_i, \mathcal{L}_i) = \{p_j, l_j\}_{j=1}^N$ with $p_j \in \mathbb{R}^{D_{\text{in}}}$, where N denotes the number of points. Each point p_j contains input descriptors with D_{in} dimensions, including point coordinates (x, y, z) and other features such as intensities (bm). The corresponding label l_j incorporates both semantic categories and motion states of p_j. Therefore, points belonging to the same semantic category but possessing distinct motion states are allocated distinct labels. For a sequence of point clouds $\{\mathcal{P}_i, \mathcal{L}_i\}_{i=1}^M$ that contains M frames, all frames are scanned sequentially in time order, and scanning poses and timestamps are used to align different frames into the same world-coordinate system. In the following, we omit the alignment process for simplicity, where the point clouds in a sequence are calibrated to the same coordinate system by default.

4. Method

4.1. Overview

An overview of our motion-aware model for multi-scan semantic segmentation, namely MarS3D, is shown in Figure 2. MarS3D contains a 3D branch for multi-scan spatial representation learning and a BEV branch for motion-aware feature learning. First, the BEV branch takes as inputs k BEV representations $\mathcal{B}^{\text{in}} = \{\mathcal{B}_i^{\text{in}}\}_{i=1}^k$ that are derived by point cloud polarization (see Section 4.2) and outputs motion-aware feature map $\mathcal{Z}^m$. The core is a Motion-Aware Feature Learning (MAFL) module (see Section 4.2) that extracts and leverages BEV features through a dedicated design to produce a motion-aware feature map $\mathcal{Z}^m$. Second, the 3D branch takes as inputs the fused k point clouds $\mathcal{P}^{\text{in}} = \{\mathcal{P}_i^{\text{in}}\}_{i=1}^k$ and outputs enriched 3D enhanced spatial features $\mathcal{P}^s$. The 3D branch incorporates a cross-frame feature embedding (CFFE) module (see Section 4.3) to inject temporal information. It outputs embedded features, denoted as $\mathcal{P}^{\text{ebd}} = \left\{\mathcal{P}_i^{\text{ebd}}\right\}_{i=1}^k$. These features are further processed by a single-scan backbone f_s to yield multi-scan enhanced 3D spatial representations $\mathcal{P}^s$ as:

$$\mathcal{P}^s = f_s(\mathcal{P}^{\text{ebd}}), \tag{1}$$

where $\mathcal{P}^s \in \mathbb{R}^{D_z}$ and D_z is the dimension of the output feature of the single-scan backbone.

Finally, the motion-aware feature map $\mathcal{Z}^m$ and enhanced spatial features $\mathcal{P}^s$ are combined to produce the fused features $\mathcal{P}^f$ by aligning the coordinates of the features with the pixels of the motion-aware feature map. The fused features are then fed to the prediction head to produce the final outputs (see Section 4.4).

4.2. BEV Branch

In the following, we elaborate on the BEV branch, which targets extracting motion-friendly features for motion prediction (see Figure 2). Before delving into the details, we first introduce BEV mapping, which maps a 3D point cloud into a 2D BEV image. Then, we introduce our key Motion-Aware Feature Learning (MAFL) module for motion extraction.

BEV Mapping: Each point cloud $\mathcal{P}$ for the BEV branch is pillarized into a three-channel BEV map $\mathcal{B}$ of size $H \times W$. Inspired by [14, 16], for a pillar located at (x^*, y^*) with a pixel grid size of l_B, its three-channel feature $b_{(x^*,y^*)}$ consists of the average translation along x and y axis regarding the grid center, and total intensity features (taking SemanticKITTI as an example) among all points inside the pillar, which is formulated as:

$$b_{(x^*,y^*)} = \left[\frac{1}{N^*}\sum_i^{N^*}\frac{2\Delta x_i}{l_B}, \frac{1}{N^*}\sum_i^{N^*}\frac{2\Delta y_i}{l_B}, \sum_i^{N^*}\text{bm}_i\right], \tag{2}$$

where N^* denotes the number of points in current pillar, while Δx_i and Δy_i denote the translation of the ith point along *x-axis* and *y-axis* respectively.

Motion-Aware Feature Learning Module: Our Motion-Aware Feature Learning (MAFL) module is illustrated in the BEV branch in Figure 2 which is designed to extract motion-aware features for better distinguishing moving/static objects. The input to this module consists of k multi-scan BEV representations $\mathcal{B}^{\text{in}} = \left\{\mathcal{B}_i^{\text{in}}\right\}_{i=1}^k$, where $\mathcal{B}_k^{\text{in}}$ represents the target frame. A lightweight 2D CNN f_u with a UNet-like architecture [22] is used to extract features $\mathcal{U} = \{\mathcal{U}_i\}_{i=1}^k$ from multi-scan inputs as:

$$\{\mathcal{U}_i\}_{i=1}^k = \left\{f_u(\mathcal{B}_i^{\text{in}})\right\}_{i=1}^k. \tag{3}$$

Furthermore, to identify moving objects, we take the difference between the target frame k and the remaining $k-1$ reference frames which outputs $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{k-1}$ as:

$$\{\mathcal{D}_i\}_{i=1}^{k-1} = \{\mathcal{U}_k - \mathcal{U}_i\}_{i=1}^{k-1}. \tag{4}$$

By doing so, the static objects can be erased, and the dynamic objects are highlighted with a large feature magnitude. Then, $\{\mathcal{D}_i\}_{i=1}^{k-1}$ are channel-wise concatenated to form a new 2D map. Note that objects may have different moving patterns and velocities. Therefore, we design a multi-kernel convolutional network f_m with multiple branches of various kernel sizes to capture objects with various movement patterns. Finally, the outputs from f_m are concatenated to output a motion-aware feature map $\mathcal{Z}^m$.

4.3. 3D Branch

The 3D branch uses temporal information to enhance spatial representation learning on 3D point clouds. The core component is the Cross-Frame Feature Embedding (CFFE) module, whose output is fed into the single-scan backbone network to produce enhanced spatial features $\mathcal{P}^s$ (see Figure 2). In the following, we elaborate on the CFFE module to improve spatial representation learning on multi-scan point clouds.

Cross-Frame Feature Embedding Module: When the multi-frame point clouds are fused as discussed in the previous sections, points from different time steps are mixed, making it challenging for the final recognition. Inspired by positional embedding [27], we propose a Cross-Frame Feature Embedding (CFFE) module to generate a time-aware embedding and produce consistent features for each point across different timestamps. Given k point cloud frames, $\left\{\mathcal{P}_i^{\text{in}}\right\}_{i=1}^k$, we design an embedding neural network layer f_e that maps point clouds into intermediate latent features and k learnable temporal embeddings $\mathcal{E} = \{e_i\}_{i=1}^k$ corresponding to k frames respectively. The dimension of e_i is the same as the output dimension of the f_e. The point-level embedded features $\mathcal{P}^{\text{ebd}}$ are obtained by:

$$\left\{\mathcal{P}_i^{\text{ebd}}\right\}_{i=1}^k = \left\{e_i + f_e(\mathcal{P}_i^{\text{in}})\right\}_{i=1}^k, \tag{5}$$

where element-wise summation is conducted on e_i and each point of the corresponding $f_e(\mathcal{P}_i^{\text{in}})$. The obtained features with different temporal embeddings are represented in a point cloud $\{\mathcal{P}_i^{\text{ebd}}\}_{i=1}^k$ and subsequently fed into the single-scan backbone network f_s, following Eq. (1), to produce a set of enhanced spatial features $\mathcal{P}^s$.

B.	Method	mIoU	#param	latency	*car*	*bic.*	*mot.*	*tru.*	*ove.*	*per.*	*bil.*	*mol.*	*roa.*	*par.*	*sid.*	*ogr.*	*bui.*	*fen.*	*veg.*	*trn.*	*ter.*	*pol.*	*tra.*	*mca.*	*mbi.*	*mpe.*	*mmo.*	*mov.*	*mtr.*
SPVCNN [25]	*Baseline*	49.70	21.8M	206ms	93.9	34.4	64.7	68.0	33.0	19.7	0.0	0.0	93.6	45.2	80.1	0.2	90.3	59.7	88.4	63.5	75.6	64.1	51.9	74.3	86.7	55.0	0.0	0.0	0.0
	Ours	54.66	21.9M	225ms	95.6	52.7	77.8	79.4	51.5	27.9	0.0	0.0	94.2	51.3	82.0	0.1	91.6	65.1	89.0	68.4	76.2	65.2	51.5	80.6	94.9	68.0	0.0	3.6	0.0
	Δ	+4.96	+0.1M	+19ms	1.7	18.3	13.1	11.4	18.5	8.2	0.0	0.0	0.6	6.1	1.9	0.1	1.3	5.4	0.6	4.9	0.6	1.1	0.4	6.3	8.2	13.0	0.0	3.6	0.0
SparseConv [9]	*Baseline*	48.99	39.2M	239ms	94.7	24.1	54.1	69.6	43.4	17.3	0.2	0.0	93.2	45.1	79.8	0.2	89.5	61.7	87.7	62.9	74.6	63.8	50.0	73.9	85.4	53.6	0.0	0.0	0.0
	Ours	54.64	39.3M	253ms	96.6	35.2	69.0	83.3	64.8	26.9	0.0	0.0	94.0	61.2	82.5	0.1	90.9	65.8	88.1	67.8	75.2	66.2	51.4	83.5	94.4	68.9	0.0	0.0	0.0
	Δ	+5.65	+0.1M	+14ms	1.9	11.1	14.9	13.7	21.4	9.6	0.2	0.0	0.8	16.1	2.7	0.1	1.4	4.1	0.4	4.9	0.6	2.4	1.4	9.6	9.0	15.3	0.0	0.0	0.0
MinkUNet [5]	*Baseline*	48.47	37.9M	295ms	93.8	23.7	48.9	90.3	41.3	18.0	0.0	0.0	92.2	32.2	78.4	0.0	89.8	55.5	88.8	63.7	77.0	63.6	50.0	69.2	83.1	52.5	0.0	0.0	0.0
	Ours	54.71	38.0M	323ms	96.4	28.4	70.0	93.9	62.7	31.6	0.0	0.0	93.8	58.5	81.7	0.1	92.6	67.6	89.0	66.7	76.4	66.5	51.6	82.6	93.1	64.4	0.0	0.1	0.0
	Δ	+6.24	+0.1M	+28ms	2.6	4.7	21.1	3.6	21.4	13.6	0.0	0.0	1.6	26.3	3.3	0.1	2.8	12.1	0.2	3.0	0.6	2.9	1.6	13.4	10.0	11.9	0.0	0.1	0.0

Table 1. Quantitative results of the proposed method, MarS3D, on SemanticKITTI [1] multi-scan public validation set. Combined with different mainstream single-scan 3D point cloud semantic segmentation backbones, MarS3D has a large performance improvement over the corresponding baseline methods without introducing excessive parameters and each-frame inference time. (**B.** indicates Backbone, full names of the categories are in the supplementary material, blue indicates degradation.)

4.4. Feature Fusion and Prediction

As illustrated in Figure 2, equipped with the motion-aware feature map ($\mathcal{Z}^m$) and enhanced spatial features ($\mathcal{P}^s$), the next step is to fully integrate the representation information from both branches and make predictions.

Feature Fusion: The motion-aware feature map $\mathcal{Z}^m \in \mathbb{R}^{H\times W\times D_z}$ (H: height; W: width; D_z: number of channels) is obtained from the BEV branch, while the enhanced spatial features $\mathcal{P}^s \in \mathbb{R}^{N\times D_p}$ (N: number of points; D_p: number of channels) are the outputs of the 3D branch. The feature fusion module aggregates information from the above two representations to make subsequent predictions. For each point feature in $\mathcal{P}^s$ with D_p dimension, a corresponding pixel from $\mathcal{Z}^m$ can be queried based on the point's 3D location (*x* and *y* coordinates). This pixel serves as an index to extract a D_z-dimensional motion-aware feature along the channel dimension from $\mathcal{Z}^m$. Subsequently, the motion-aware feature is concatenated with the point feature, resulting in the fused features denoted as $\mathcal{P}^f \in \mathbb{R}^{N\times(D_z+D_p)}$.

Prediction: Based on the correspondence between 2D and 3D, $\mathcal{P}^f$ is then fed into an MLP classifier f_{cls} to obtain the output s_{pred}:

$$s_{\text{pred}} = f_{\text{cls}}(\mathcal{P}^f), \tag{6}$$

where s_{pred} is the logits for the prediction result of the semantic segmentation task of the input point cloud.

4.5. Model Training and Inference

During training, the obtained fused features $\mathcal{P}^f$ are fed into two classification heads: the category-aware classification head and the motion-aware classification head (a binary classifier), which outputs the predicted logits s^c_{pred} and s^m_{pred} (more details are included in the supplementary material). First, with the ground truth labels of semantic categories $\mathcal{L}^c_{\text{GT}}$, back-propagation is performed for parameters optimization using Cross Entropy (CE) loss L_c for semantic categories classification:

$$L_c = \text{CE}(s^c_{\text{pred}}, \mathcal{L}^c_{\text{GT}}), \tag{7}$$

where $\text{CE}(\cdot,\cdot)$ is the cross-entropy loss, and s^c_{pred} is the prediction probability for each point. Then, the Binary Cross Entropy (BCE) loss L_m is used for motion states classification with motion states ground truth $\mathcal{L}^m_{\text{GT}}$:

$$L_m = \text{BCE}(s^m_{\text{pred}}, \mathcal{L}^m_{\text{GT}}), \tag{8}$$

where $\text{BCE}(\cdot,\cdot)$ indicates the binary cross-entropy loss.

The final objective function L of the optimization is:

$$L = \omega_c \cdot L_c + \omega_m \cdot L_m, \tag{9}$$

where ω_c and ω_m are the weights of the two losses (L_c and L_m) respectively.

During inference, the final prediction result is determined using the logits produced by both classification heads. When presented with an input sample, the motion-aware classification head will identify the motion state of the input point only if the category-aware classification head recognizes the point as belonging to a class with the potential to move.

5. Experiments

Datasets and Evaluation Metric: We evaluate our method on SemanticKITTI [1] and nuScenes [3]. For SemanticKITTI, the multi-scan setting is fully supervised and contains 25-category (6 moving categories and 19 static categories) with high-quality semantic annotations. The annotations are based on the KITTI dataset [8]. It comprises 22 point cloud sequences. For nuScenes, we propose a new

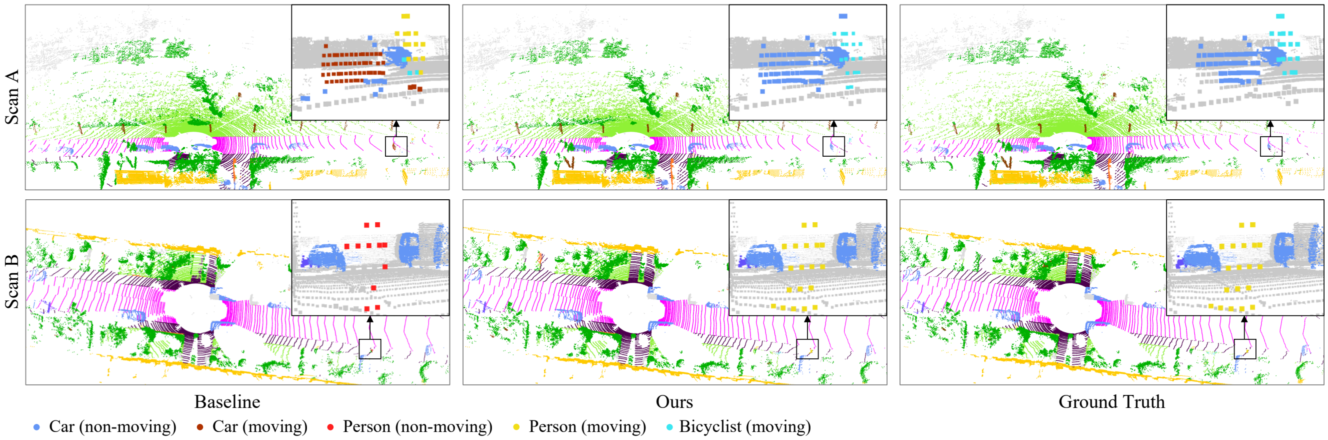

Figure 3. Qualitative results on SemanticKITTI [1] public validation dataset. With SPVCNN [25] as the backbone, the segmentation results on the SemanticKITTI multi-scan task of the baseline model and our model are shown together with ground truth. At the same time, a specific area containing moving points is magnified and displayed at the top right of each sub-figure.

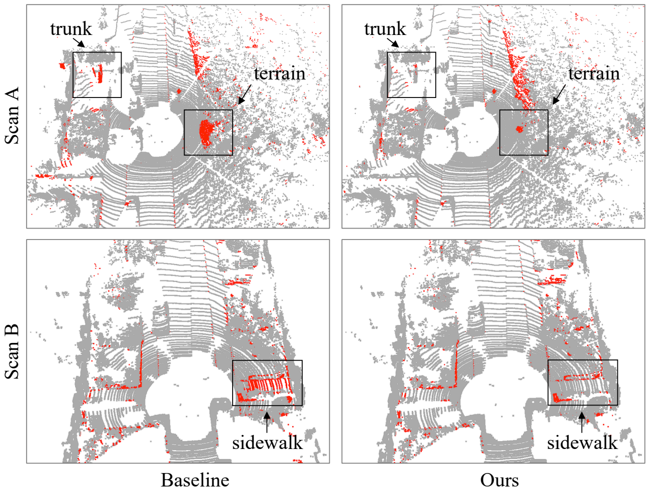

Figure 4. Evaluation errors (shown in red) by the baseline methods and MarS3D (using SPVCNN [25] as the backbone) on SemanticKITTI [1] public validation dataset. MarS3D significantly outperforms the baseline. Contrasting regions in the evaluation errors are highlighted with boxes and corresponding categories.

multi-scan setting based on the 'lidar-seg' task (16 semantic categories) without reference frame supervision. We use its object-level velocity to construct a multi-scan segmentation dataset with 24 categories (8 moving and 16 static categories). More details are provided in the supplementary material. To assess the effectiveness of our proposed method and make comparisons with baselines and other methods, we use the mean Intersection over Union (mIoU) as the evaluation metric.

Implementation Details: Our model is designed as a plug-and-play module that provides motion-aware features to enhance the backbone learned features. For the backbone model, we consider SPVCNN [25][*], SparseConv [9][†], and MinkUNet [5][‡]. Following previous works [12, 23, 36], we use the current and its previous two frames as input. The size of the BEV representation is set to 501×301, and the multiple kernel sizes in the MAFL module are set as 1, 3, and 5, respectively. We set the embedded feature dimension in the CFFE module to 18, and the data augmentations are the same as the standard settings. All the models are trained on GeForce RTX 3090 GPUs, and the inference latency is recorded using a single GeForce RTX 3090 GPU.

5.1. Main Results

Comparison with Baseline Methods: We evaluate the performance of our proposed method on multi-scan benchmarks of SemanticKITTI [1] and nuScenes [3]. The baseline method uses the same backbone to process multi-scan point clouds. We then compare this baseline approach to the same backbone augmented with MarS3D for a fair evaluation. As shown in Table 1, MarS3D significantly improves the performance over baseline methods on the public validation set of SemanticKITTI. With the most lightweight network, SPVCNN [25], MarS3D brings a 4.96% improvement while introducing less than 0.5% additional parameters. Particularly, consistent performance gains are observed on the dynamic object classes with non-moving/moving properties (*i.e.*, car & moving car, person & moving person). This shows that the proposed BEV branch is both lightweight and powerful. Further, to verify the generalizability of the model, we offer a multi-scan task based on nuScenes 'lidar-seg' dataset [3]. Our method outperforms the baseline (using MinkUNet [5] as the backbone) by a significant margin, achieving 64.83%

[*] https://github.com/mit-han-lab/spvnas
[†] https://github.com/traveller59/spconv
[‡] https://github.com/NVIDIA/MinkowskiEngine

Method	mIoU	*car*	*bic.*	*mot.*	*tru.*	*ove.*	*per.*	*bil.*	*mol.*	*roa.*	*par.*	*sid.*	*ogr.*	*bui.*	*fen.*	*veg.*	*trn.*	*ter.*	*pol.*	*tra.*	*mca.*	*mbi.*	*mpe.*	*mmo.*	*mov.*	*mtr.*
SpSequenceNet [24]	43.1	88.5	24.0	26.2	29.2	22.7	6.3	0.0	0.0	90.1	57.9	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 [7]	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	68.2	42.8	40.4	12.9	12.4	2.1
TemporalLatticeNet [23]	47.1	91.6	35.4	36.1	26.9	23.0	9.4	0.0	0.0	91.5	59.3	75.3	27.5	89.6	65.3	84.6	66.7	70.4	57.2	60.4	59.7	41.7	51.0	48.8	5.9	0.0
Meta-RangeSeg [28]	49.5	90.1	52.7	43.9	30.3	35.4	14.3	0.0	0.0	90.7	63.3	74.7	26.9	90.5	63.5	83.0	67.0	67.7	56.4	64.4	64.5	56.1	55.0	24.4	20.3	3.4
KPConv [26]	51.2	93.7	44.9	47.2	43.5	38.6	21.6	0.0	0.0	86.5	58.4	70.5	26.7	90.8	64.5	84.6	70.3	66.0	57.0	53.9	69.4	67.4	67.5	47.2	4.7	5.8
Baseline	49.2	89.8	39.4	34.0	39.4	21.0	8.9	1.8	0.0	89.1	62.0	72.4	12.9	90.5	63.9	84.6	68.4	68.7	58.9	60.1	69.3	63.5	58.7	56.5	9.5	3.6
Ours	52.7	95.1	49.2	49.5	39.7	36.6	16.2	1.2	0.0	89.9	66.8	74.3	26.4	92.1	68.2	86.0	72.1	70.5	62.8	64.8	78.4	67.3	58.0	36.3	10.0	5.1

Table 2. Comparison with the state-of-the-art models on SemanticKITTI multi-scan benchmark (official test set). MarS3D (with SPVCNN [25] as the backbone) significantly outperforms these models for multi-scan tasks. (full names of the categories are in the supplementary material.)

mIoU compared to the baseline's 61.90%. This improvement is achieved with a negligible increase in each-frame inference time from 53ms to 58ms.

Comparison with State-of-the-Art Methods: Compared to various models applied to multi-scan tasks, MarS3D (with SPVCNN as the backbone) is evaluated on the SemanticKITTI multi-scan benchmark[§]. As shown in Table 2, the proposed approach has demonstrated superior performance, with a 1.5% increase in mIoU compared to the current state-of-the-art method, KPConv [26]. Furthermore, our method performs similarly or better than other state-of-the-art models across nearly all categories.

Qualitative Comparisons: Quantitative results are shown in Figure 3. The baseline model suffers from mistaking the status of static cars (Figure 3: Scan A) and moving persons (Figure 3: Scan B). It even recognizes the moving bicyclist as a moving person (Figure 3: Scan A). In contrast, MarS3D can circumvent such category and motion state discrimination errors. In addition to achieving better qualitative results in motion states, MarS3D also outperforms the baseline in semantic categories. We visualize the error maps (errors are shown in red) of the baseline and our method in Figure 4, where bounding boxes indicating specific categories highlight the differences between our method and the baseline. These improvements indicate that MarS3D has stronger semantic feature extraction capabilities compared to the baseline model, resulting in better segmentation performance even for immobile objects.

5.2. Ablation Studies

In this section, we conduct comprehensive ablation experiments on the SemanticKITTI multi-scan validation set to examine the effects of each component in our proposed method. As shown in Table 3, we gradually add three components to the baseline method, including the CFFE module, vanilla BEV branch (introduce only BEV representations and 2D CNN without the MAFL module), and the MAFL module to illustrate the effectiveness of our designs.

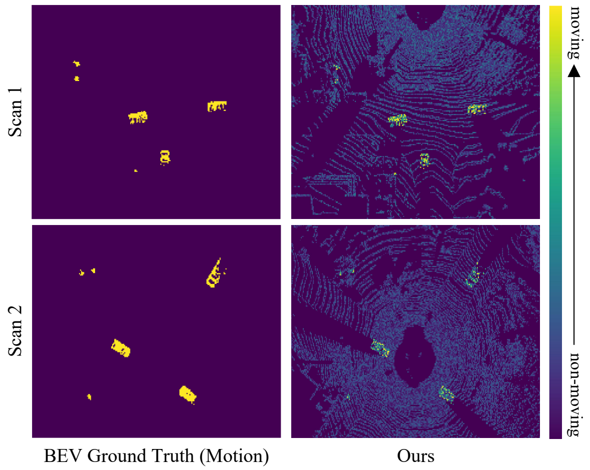

Figure 5. Sampled discrepant feature maps demonstrate that the region that contains moving points are represented higher activation values than other regions.

Effiectiveness of CFFE Module: As shown in Table 3, we first employ only the CFFE module with the baseline model and observe a significant performance improvement. For instance, with SPVCNN [25] as the backbone, the introduction of the CFFE module led to a boost in performance by 1.1% in terms of mIoU. This confirms the efficacy of the CFFE module for multi-frame representation learning.

Effiectiveness of BEV Representation: We also observe a significant performance improvement when we add the vanilla BEV branch alone to the baseline model. Using SPVCNN [25] as the backbone, the BEV branch results in a 3.7% increase in mIoU. This confirms the importance of BEV representations in extracting motion-aware information. Furthermore, when both the CFFE module and the vanilla BEV branch are added to the baseline model, there is a further increase in performance, demonstrating that the two components complement each other.

[§] http://www.semantic-kitti.org/tasks - Semantic Segmentation - Multiple Scans

Method	CFFE	BEV	MAFL	mIoU(%)
baseline	-	-	-	49.7 / 49.0 / 48.5
proposed	✓	-	-	50.8 / 51.3 / 52.1
	-	✓	-	53.4 / 53.9 / 53.1
	✓	✓	-	53.6 / 54.2 / 54.1
	✓	✓	✓	54.7 / 54.6 / 54.7

Table 3. Ablation studies on different backbones (SPVCNN / SparseConv / MinkUNet) on SemanticKITTI [1] public validation dataset. The effectiveness of different designs is demonstrated step-by-step and our method is marked with a gray box.

Method	Vanilla BEV	TDNet [13]	STM [18]	*Ours*
mIoU(%)	53.36	53.56 (+0.20)	53.64 (+0.28)	54.66 (+1.30)

Table 4. Comparing our method (using SPVCNN [25] as the backbone) with different 2D temporal semantic segmentation approaches the public validation dataset of SemanticKITTI [1].

Effiectiveness of MAFL Module: In the final experiment presented in Table 3, we study the impact of our proposed MAFL module on the BEV representations, resulting in an increase of around 5.0% from the baseline model (using SPVCNN [25] as the backbone). By stacking all the proposed components together, our final solution (marked with a grey box in Table 3) reaches its pinnacle in performance. The improvement in other backbones is also significant, and more statistical results are provided in the supplementary material. In addition, the multi-channel discrepant feature map represents the differences between the extracted feature maps of BEV representations from two point clouds. For a discrepant feature map generated during inference, we randomly sample the channels pixel by pixel. The average absolute activation values of pixels with the same motion state in the sampled channels are re-scaled and aggregated into a single-channel feature map as shown in Figure 5. The sampled discrepant feature map has higher activation on pixels containing moving points compared to other regions. This clearly demonstrates that the MAFL module clearly distinguishes between regions that contain moving points and those that do not.

5.3. More Comparison on Temporal Segmentation

As for BEV representation learning, the proposed MAFL module is utilized on extracted 2D feature maps for capturing inter-frame temporal information of point clouds. We further conduct comparative experiments by replacing the MAFL module with its 2D counterparts, *i.e.*, STM [18] and TDNet [13]. These 2D modules have been widely utilized for preserving temporal information in 2D tasks. The results of the comparative experiments using SPVCNN [25] as the backbone are presented in Table 4. According to the evaluation results, it can be concluded that the MAFL module shows superior performance compared to the other models, thereby confirming its remarkable effectiveness. This shows that the proposed MAFL module is better suited for the specific task of handling motion-aware semantic segmentation in 3D point clouds.

5.4. Limitations and Failure Cases Analysis

Although MarS3D demonstrates impressive overall performance on the SemanticKITTI multi-scan benchmark, some limitations can be identified from the quantitative results in Table 1 and Table 2. The SemanticKITTI training dataset has an imbalanced distribution of point categories (more details are included in the supplementary material), which causes MarS3D to perform poorly on long-tailed categories due to insufficient training data on these categories. As shown in Table 2, MarS3D fails to show effects on several long-tail categories (*i.e.*, bicyclist, motorcyclist, moving-other-vehicle, and moving-truck). Exploring solutions to address the long-tail problem is a promising research direction for the work. Since our model assumes planar motion, it may not perform well in scenarios where objects move in non-planar ways, such as on steep terrains or on non-planar surfaces.

6. Conclusion

In this paper, we propose MarS3D, a novel plug-and-play motion-aware model for 3D multi-scan point cloud semantic segmentation. The Motion-Aware Feature Learning (MAFL) module, based on BEV representations, is designed to facilitate extracting motion-aware representations. Additionally, the Cross-Frame Feature Embedding (CFFE) module is introduced to improve representation learning by embedding time-step information into features, thus preserving rich temporal information. Extensive experiments and ablation studies demonstrate that MarS3D significantly improves multiple 3D semantic segmentation baselines while introducing minimal overheads. In comparison to state-of-the-art methods designed for multi-scan tasks, MarS3D achieves superior performance and offers significant improvements over baseline methods. The proposed MarS3D model demonstrates the potential for effectively incorporating motion-awareness into 3D point cloud semantic segmentation tasks, providing a strong foundation for further research and development in this area.

Acknowledgments: This work has been supported by Hong Kong Research Grant Council - Early Career Scheme (Grant No. 27209621), General Research Fund Scheme (Grant no. 17202422), and RGC matching fund scheme (RMGS). Part of the described research work is conducted in the JC STEM Lab of Robotics for Soft Materials funded by The Hong Kong Jockey Club Charities Trust.

References

[1] 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. 1, 2, 5, 6, 8

[2] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *European Conference on Computer Vision*, pages 777–794. Springer, 2020. 3

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 5, 6

[4] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12547–12556, 2021. 1, 2

[5] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1, 2, 5, 6

[6] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. Every frame counts: Joint learning of video segmentation and optical flow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10713–10720, 2020. 3

[7] Fabian Duerr, Mario Pfaller, Hendrik Weigel, and Jürgen Beyerer. Lidar-based recurrent 3d semantic segmentation with temporal memory alignment. In *2020 International Conference on 3D Vision (3DV)*, pages 781–790. IEEE, 2020. 1, 2, 7

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5

[9] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 5, 6

[10] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 2

[11] Noureldin Hendy, Cooper Sloan, Feng Tian, Pengfei Duan, Nick Charchut, Yuesong Xie, Chuang Wang, and James Philbin. Fishing net: Future inference of semantic heatmaps in grids. *arXiv preprint arXiv:2006.09917*, 2020. 2

[12] Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8479–8488, 2022. 1, 2, 6

[13] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8827, 2020. 3, 8

[14] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 2, 4

[15] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. In *European Conference on Computer Vision*, pages 352–368. Springer, 2020. 3

[16] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multitask multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 2, 3, 4

[17] Jiaxu Miao, Yunchao Wei, and Yi Yang. Memory aggregation networks for efficient interactive video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10366–10375, 2020. 3

[18] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019. 3, 8

[19] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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*, pages 652–660, 2017. 2

[20] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[21] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7406–7415, 2020. 3

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[23] Peer Schutt, Radu Alexandru Rosu, and Sven Behnke. Abstract flow for temporal semantic segmentation on the permutohedral lattice. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5139–5145. IEEE, 2022. 1, 2, 6, 7

[24] Hanyu Shi, Guosheng Lin, Hao Wang, Tzu-Yi Hung, and Zhenhua Wang. Spsequencenet: Semantic segmentation network on 4d point clouds. In *Proceedings of the IEEE/CVF*

conference on computer vision and pattern recognition, pages 4574–4583, 2020. 1, 2, 7

[25] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European conference on computer vision*, pages 685–702. Springer, 2020. 1, 2, 5, 6, 7, 8

[26] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 2, 7

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[28] Song Wang, Jianke Zhu, and Ruixiang Zhang. Meta-rangeseg: Lidar sequence semantic segmentation using multiple feature aggregation. *arXiv preprint arXiv:2202.13377*, 2022. 1, 2, 7

[29] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2

[30] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 2

[31] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16024–16033, 2021. 1, 2

[32] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3173–3182, 2021. 2

[33] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang 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*, volume 35, pages 3101–3109, 2021. 1

[34] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In *European Conference on Computer Vision*, pages 677–695. Springer, 2022. 2

[35] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020. 2

[36] Hui Zhou, Xinge Zhu, Xiao Song, Yuexin Ma, Zhe Wang, Hongsheng Li, and Dahua Lin. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv preprint arXiv:2008.01550*, 2020. 1, 2, 6

[37] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 2

[38] Zhenhong Zou and Yizhe Li. Efficient urban-scale point clouds segmentation with bev projection. *arXiv preprint arXiv:2109.09074*, 2021. 2, 3