electronics

MDPI

Article

Dual-Branch Dynamic Object Segmentation Network Based on Spatio-Temporal Information Fusion

Fei Huang [1], Zhiwen Wang [1], Yu Zheng [1], Qi Wang [2], Bingsen Hao [2,*] and Yangkai Xiang [2]

[1] China Road & Bridge Corporation, Beijing 100011, China; huangf@crbc.com (F.H.)
[2] School of Mechatronics and Vehicle Engineering, Chongqing Jiaotong University, Chongqing 400074, China; 622220990112@mails.cqjtu.edu.cn (Q.W.); xiangyangkai@mails.cqjtu.edu.cn (Y.X.)
* Correspondence: 622220040024@mails.cqjtu.edu.cn

Abstract: To address the issue of low accuracy in the segmentation of dynamic objects using semantic segmentation networks, a dual-branch dynamic object segmentation network has been proposed, which is based on the fusion of spatiotemporal information. First, an appearance–motion feature fusion module is designed, which characterizes the motion information of objects by introducing a residual graph. This module combines a co-attention mechanism and a motion correction method to enhance the extraction of appearance features for dynamic objects. Furthermore, to mitigate boundary blurring and misclassification issues when 2D semantic information is projected back into 3D point clouds, a majority voting strategy based on time-series point cloud information has been proposed. This approach aims to overcome the limitations of post-processing in single-frame point clouds. By doing this, this method can significantly enhance the accuracy of segmenting moving objects in practical scenarios. Test results from the semantic KITTI public dataset demonstrate that our improved method outperforms mainstream dynamic object segmentation networks like LMNet and MotionSeg3D. Specifically, it achieves an Intersection over Union (IoU) of 72.19%, representing an improvement of 9.68% and 4.86% compared to LMNet and MotionSeg3D, respectively. The proposed method, with its precise algorithm, has practical applications in autonomous driving perception.

Keywords: dynamic object segmentation; co-attention; feature fusion; post-processing

Citation: Huang, F.; Wang, Z.; Zheng, Y.; Wang, Q.; Hao, B.; Xiang, Y. Dual-Branch Dynamic Object Segmentation Network Based on Spatio-Temporal Information Fusion. *Electronics* **2024**, *13*, 3975. https://doi.org/10.3390/electronics13203975

Academic Editor: Beiwen Li

Received: 14 August 2024
Revised: 29 September 2024
Accepted: 30 September 2024
Published: 10 October 2024

1. Introduction

In dynamic scenes, distinguishing between dynamic and static objects is key to improving semantic segmentation accuracy. Dynamic object segmentation can effectively enhance point cloud map construction accuracy [1], scene flow estimation [2,3], and avoid dynamic interference in planning tasks [4]. Current research mainly focuses on two approaches: geometric-based methods and deep learning-based methods.

Geometric-based dynamic object segmentation algorithms primarily include viewpoint visibility methods [5] and ray-casting methods [6]. For instance, Kim et al. [7]. used a multi-resolution depth map visibility mechanism to match local keyframes with static maps, achieving dynamic object removal. This method requires pre-constructed point cloud maps and a recovery strategy to restore static points from incorrectly classified dynamic points, which can be challenging for real-time applications. Schauer et al. [8], using ray-casting principles, determine whether a grid cell has been passed through by LIDAR to achieve dynamic filtering. However, this method struggles with updating the probability values of grids in open scenes and requires traversing all grid cells, leading to high computational demands.

With the development of deep learning, many excellent algorithms have emerged in the field of semantic segmentation, such as PointNet++ [9], VoxelNet [10], and SalsaNeXt [11]. However, these algorithms can only segment objects with motion attributes that are temporarily stationary (e.g., parked vehicles) as well as static objects. In complex

dynamic scenes, the capability of dynamic object segmentation is especially important. Dynamic object segmentation from input data can be categorized into point cloud-based methods, voxel-based methods, and depth map-based methods [12]. Using raw point cloud data as network input avoids preprocessing, effectively preserving the 3D information of the point cloud. The 4DMOS network proposed by Mersch et al. [13] uses sparse 4D convolution to extract spatial and temporal features from the point cloud, enabling online dynamic object prediction and then processes new predictions with a Bayesian filter. Wang et al. [14] introduced InsMOS based on 4DMOS, which not only performs point-by-point dynamic object prediction but also detects the instance information of major traffic participants. Point-based prediction methods achieve good accuracy but increase computational resource consumption. To effectively address the irregularity of point clouds, Graham [15] and Yan et al. [16] adopted voxel-based representation methods, which still introduce considerable memory overhead and computational cost. Depth maps, as an intermediate representation method from 3D point clouds to 2D space, significantly enhance network training and inference speed without sacrificing much accuracy. For instance, Chen et al. [17] proposed the LMNet network, which transforms point clouds into depth map representations. To capture inter-frame motion information, residual images are added as additional channels to the network input, leveraging mainstream semantic segmentation networks for dynamic object segmentation. This method simply concatenates depth maps and residual images, without fully utilizing temporal information, and has noticeable boundary blur issues during the reprojection process. Kim et al. [18] also used depth maps and residual information as network input and further improved dynamic object segmentation performance with data augmentation. Sun et al. [19] proposed a dual-branch structure MotionSeg3D network, which separately processes spatial and temporal information and introduces a motion-guided attention module for feature fusion while adopting a coarse-to-fine approach for processing prediction results. This method effectively improves dynamic object segmentation accuracy but has relatively high training costs for the two-stage network.

To address the boundary blur issue in semantic segmentation networks based on depth map representation [20], most works [21,22] employ post-processing methods such as Conditional Random Fields (CRF) or K-Nearest Neighbor (KNN) to smooth the predicted label results. For instance, Squeezeseg [21] utilizes CRF to refine predictions based on neighboring results after three iterations but fails to effectively handle occluded points. RangeNet++ [22] applies a KNN approach to search for K-nearest neighbor points within a certain region to infer the semantic information of ambiguous points. This method often leads to either insufficient or excessive smoothing and performs poorly on severely occluded points. MotionSeg3D introduces a refinement module to replace traditional post-processing methods, achieving some improvement in accuracy but increasing training costs.

To address these issues, this paper proposes a dual-branch dynamic object segmentation network based on spatiotemporal information fusion, built upon the MotionSeg3D network, to enhance the segmentation accuracy of dynamic objects in semantic segmentation tasks. The semantic segmentation model proposed in this paper makes the following main contributions:

- Inspired by video object segmentation tasks [23], an appearance–motion fusion (AMF) module is designed, which consists of a shared attention mechanism and motion correction method, to enhance the extraction capability of appearance features with motion information;
- A majority voting strategy (MVS) post-processing method is proposed, which integrates temporal point cloud semantic information to update the current predictions, addressing the boundary blur and semantic label misclassification issues caused by re-projection;

- In the test set, the IoU of this proposed method reaches 72.19%, exceeding top dynamic object segmentation networks such as LMNet and MotionSeg3D by 9.68% and 4.86%, respectively.

2. Method

2.1. Overall Network Structure

The Dual-Branch Dynamic Object Semantic Segmentation Network employs an encoder–decoder structure to separately extract appearance and motion features from spatial and temporal dimensions. It utilizes the feature fusion module (AMF) to aggregate these features, thereby enhancing the segmentation capability for dynamic objects. In the post-processing stage, a majority voting strategy is proposed to reduce boundary blurring and label misclassification issues.

The overall network framework is shown in Figure 1. A spherical projection is used to convert 3D point clouds into depth maps [24], serving as input data for the appearance feature extraction branch. This avoids the disorder associated with directly processing raw point cloud data [25] and improves data processing efficiency. To aggregate contextual information from different regions, residual dilated convolutions are introduced, stacking dilated convolutions with a receptive field of 5 after standard convolutions to capture richer spatial information from different receptive fields. The Meta-Kernel [26] module is used to dynamically learn weights from the relative Cartesian coordinate system, enabling the network to extract more spatial geometric information from the depth maps. To capture motion information, residual representations are obtained by computing depth maps generated from consecutive frames of point clouds and are input into the motion branch for feature extraction. The first layer uses residual dilated convolutions to capture contextual information. To avoid a significant increase in the number of parameters due to larger receptive fields, subsequent layers use combinations of dilated convolutions with receptive fields of 3, 5, and 7 and convolutional residual connections at the output positions to gather more information from different features. For each residual dilated convolution in this branch, dropout layers and Adaptive Exponential Weighted Pooling (AdaPool) [27] are used for downsampling. Features extracted by the two branches interact through the appearance–motion fusion module, dynamically assigning feature weights, and are fused to output enhanced appearance features. The feature results from each fusion module are concatenated with the aforementioned residual dilated convolutions, while residual structures connect to the upsampling modules in the decoder. The network's predicted results are post-processed using a majority voting strategy to reduce boundary blur and misclassification issues.

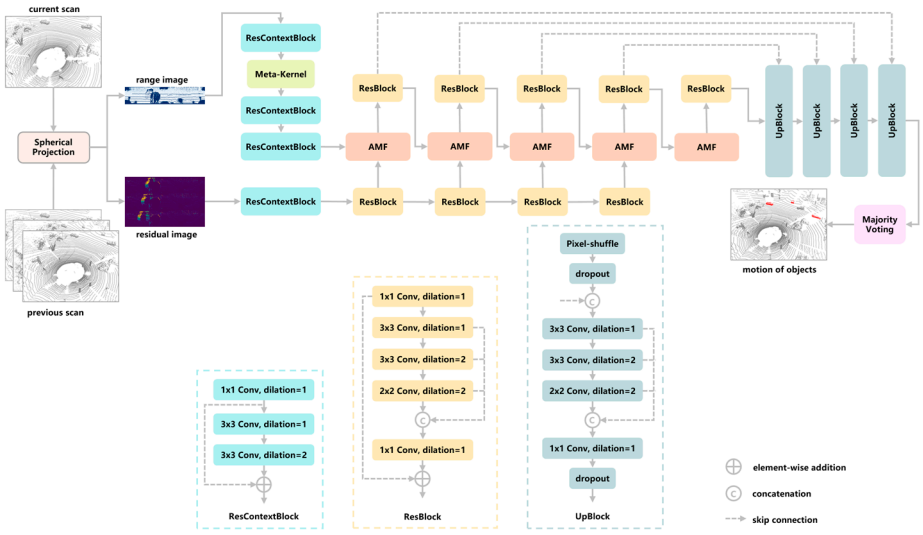


Figure 1. Dual-branch dynamic object segmentation network.

2.2. Motion Feature Representation

The range image representation maps the point cloud in three-dimensional space to two-dimensional space, avoiding the complexity of processing unordered point cloud data. For three-dimensional point clouds in Cartesian coordinates, they are projected onto the image coordinate system through spherical projection to obtain the corresponding range image, and a two-dimensional convolutional neural network is used to extract appearance features from it. However, relying solely on a single semantic segmentation network cannot effectively identify moving objects in the scene. To enhance the network's ability to recognize dynamic objects, this paper adopts the residual image calculation method used in LMNet [17] and introduces temporal information of dynamic objects during the training process.

Using the coordinate system of the current frame point cloud as the reference, historical n frame point clouds are transformed to the current coordinate system using a transformation matrix, resulting in the generation of a range image. In the range image, each pixel represents the distance value r of the corresponding pixel coordinate point (u, v). The normalized absolute difference between the current frame and historical frames is calculated according to Formula (1) to obtain the residual $d^c_{k,i}$, resulting in the generation of the residual map as shown in Figure 2.

$$d^c_{k,i} = \frac{\left| r_i - r_i^{k \to c} \right|}{r_i} \tag{1}$$

where $r_i^{k \to c}$ is the distance value obtained by transforming the point cloud c of the historical frame k.

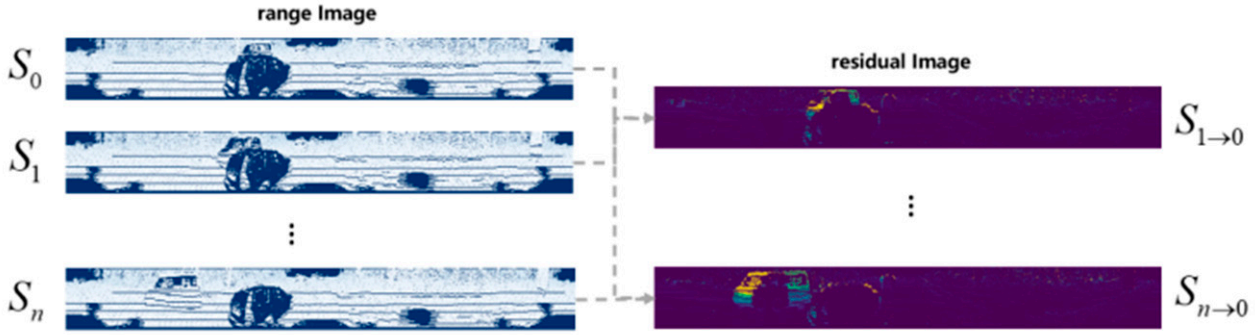


Figure 2. Generation of residual images.

2.3. Meta-Kernel Convolution

Reducing 3D point clouds to 2D space results in 2D convolutions being unable to fully utilize the 3D geometric information of the point clouds. As shown in Figure 3, this paper introduces a meta-kernel convolution module, which selects the relative Cartesian coordinates of the 3×3 neighborhood at the center of the feature map and inputs them into a multi-layer perceptron (MLP) with two fully connected layers, generating a total of 9 weight vectors w_i, $i = (1, 2, \ldots, 9)$ that adapt to the local 3D structure. These weight vectors are then element-wise multiplied with the corresponding 9 feature vectors f_i. Finally, the resulting 9 neighborhood feature vectors are concatenated and passed through a 1×1 convolution. This aggregates information from different channels and different sampling positions to update the central feature vector.

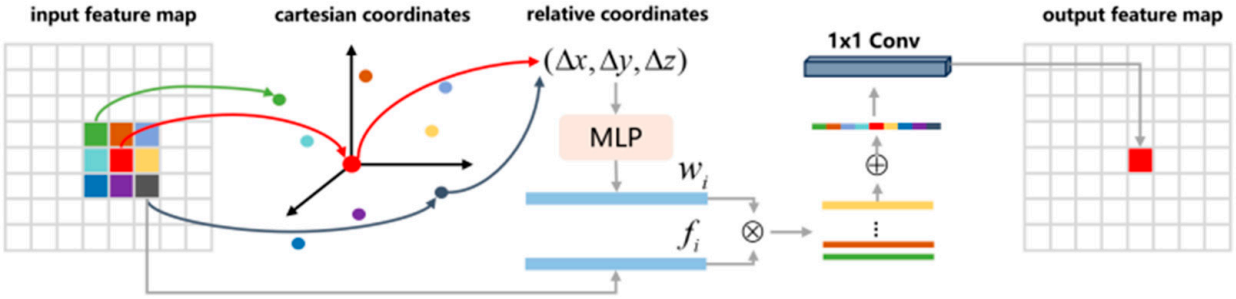


Figure 3. Meta-kernel convolution module.

2.4. Appearance–Motion Feature Fusion Module

The unidirectional attention guidance module in MotionSeg3D depends solely on the primary moving objects in the scene, neglecting the inherent noise in the range image. This paper proposes an appearance–motion fusion (AMF) module composed of co-attention and motion-guided attention mechanisms. By adaptively allocating feature weights and cross-modal feature fusion, the AMF module enhances the representation capability of appearance features for dynamic objects. Its structure is shown in Figure 4.

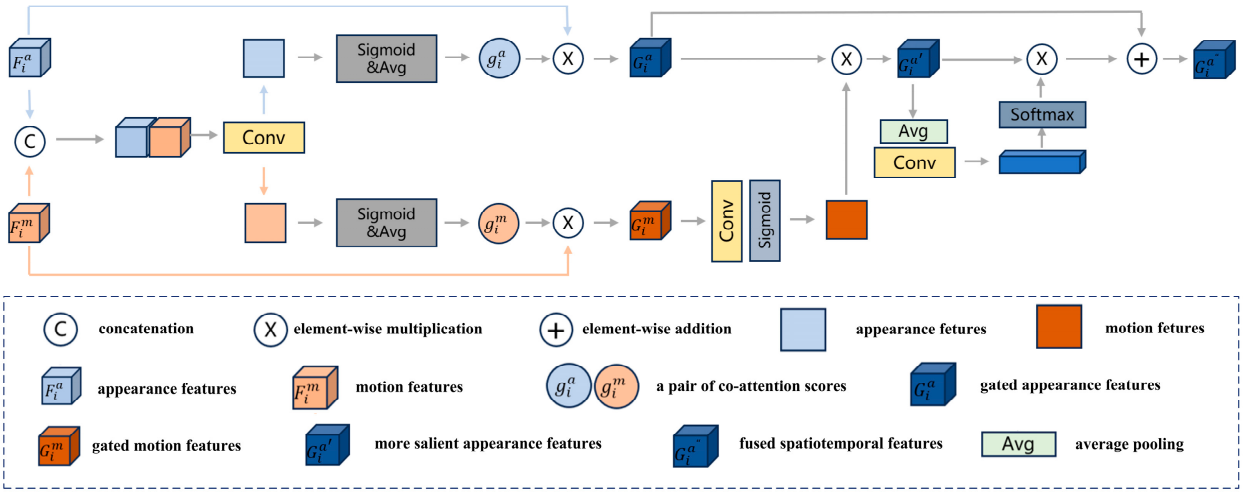


Figure 4. Appearance–motion features fusion module.

First, using cross-channel concatenation and convolution operations in the ith layer, the appearance features F_i^a and motion features F_i^m are aligned to capture the relative relationships between multimodal features. The aligned and fused features $H \in \mathbb{R}^{h \times w \times 2}$ are divided into two sub-branches, and for each channel, a Sigmoid function and global average pooling are performed to obtain a pair of co-attention scores, g_i^a and g_i^m. Higher scores indicate that the corresponding modality features contain more accurate and effective segmentation information. In contrast, lower scores suggest that the modality features may contain noise that affects performance. The co-attention gating function composed of appearance features and motion features can be expressed as

$$g_i = \text{Avg}(\text{Sigmoid}(\text{Conv}(\text{Cat}(F_i^a, F_i^m)))) \tag{2}$$

where g_i represents a pair of co-attention scores, including g_i^a and g_i^m. $\text{Avg}(\cdot)$ denotes the global average pooling operation. $\text{Sigmoid}(\cdot)$ denotes the activation function with a range of $(0, 1)$. $\text{Conv}(\cdot)$ denotes a convolutional layer with an output channel of 2. $\text{Cat}(\cdot)$ represents concatenation operation across each channel.

The scores generated by the co-attention gating module are applied to the corresponding features to obtain the updated gated appearance features, G_i^a and G_i^m, which are defined as

$$G_i^a = F_i^a \otimes g_i^a, G_i^m = F_i^m \otimes g_i^m \tag{3}$$

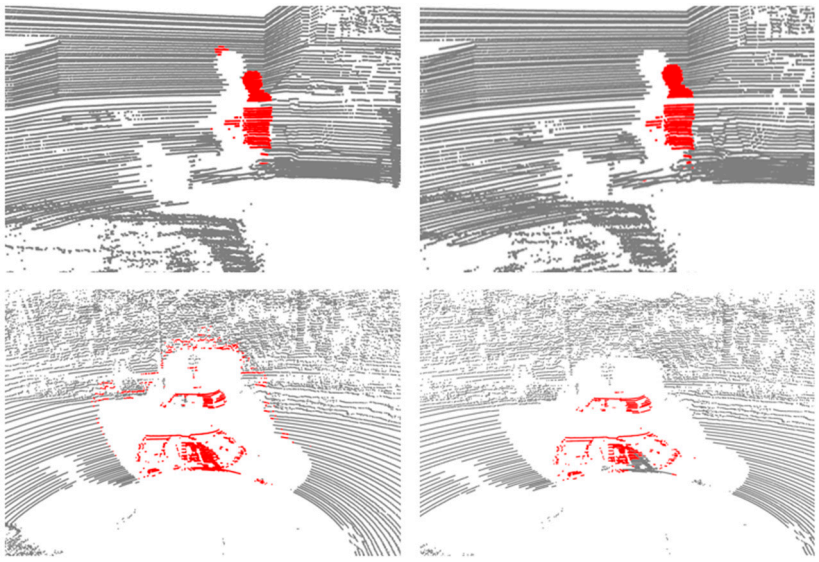
In the motion-guided attention module, firstly, the spatial attention mechanism is utilized on the motion features G_i^m to augment the spatial positional information of the appearance features G_i^a, yielding more salient appearance features $G_i^{a'}$. Subsequently, a channel attention mechanism is applied to enhance critical attributes, resulting in fused $G_i^{a''}$ spatiotemporal features with dimensions $C \times h \times w$, where

$$G_i^{a'} = G_i^a \otimes \text{Sigmoid}(\text{Conv}_{1\times1}(G_i^m)) \tag{4}$$

$$G_i^{a''} = G_i^{a'} \otimes [\text{Softmax}\Big(\text{Conv}_{1\times1}(\text{Avg}(G_i^{a'}))) \cdot C] + G_i^a \tag{5}$$

2.5. Majority Voting Strategy

Although the moving object segmentation network achieved through the AMF module efficiently segments dynamic objects, there remains an issue during the range image reprojection back into 3D space. Distant points occluded by closer ones inherit the predictive attributes of the latter. Figure 5 illustrates boundary ambiguity, with Figure 5a showing an image with blurred boundaries and Figure 5b showing the ground truth (GT). Among them, red points represent dynamic attribute semantic labels, and black points represent static attribute semantic labels. As shown in Figure 5, background objects with static attributes are misclassified as dynamic semantic labels due to occlusion by dynamic objects, particularly at their edges.



(**a**) Boundary blur (**b**) GT

Figure 5. Boundary blur problem.

To address this issue, traditional semantic segmentation networks typically employ k-Nearest Neighbor (k-NN) post-processing, where the semantic information of a point is determined by searching its K-nearest neighbors. While this method alleviates the problem of boundary blur to some extent, it is sensitive to the choice of k value and distance, and its performance remains suboptimal when a large area of the neighborhood is misclassified. This paper proposes a post-processing method based on a majority voting strategy, utilizing different perspectives of adjacent keyframes in various temporal sequences to refine the

classification of predicted results. This approach circumvents the limitations of single-scan frames, as illustrated in the overall process depicted in Figure 6.

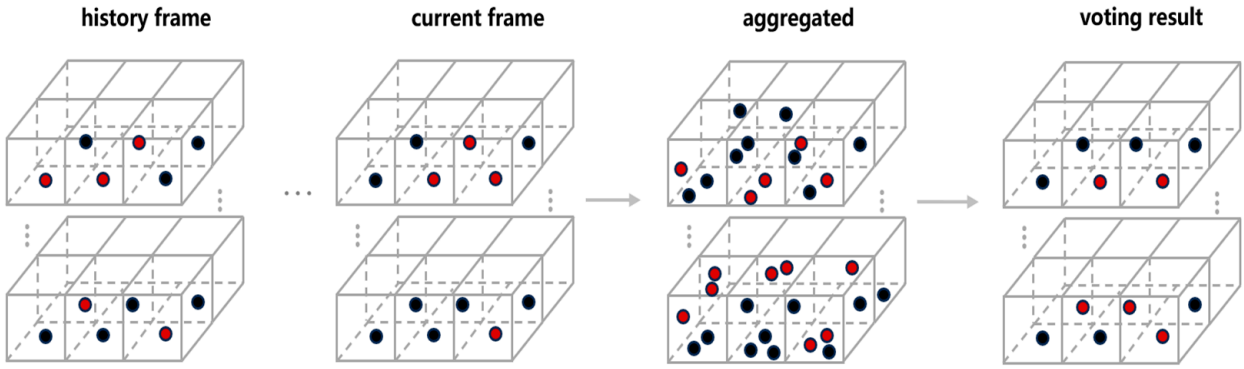


Figure 6. Majority voting strategy.

Increasing the number of keyframes can provide richer contextual information, thereby enhancing segmentation accuracy. The historical n frames of LiDAR point clouds $(p_{t-n}, \ldots, p_{t-1})$ are aligned to the current frame point cloud p_t coordinate system using a transformation matrix. The transformation matrix T_{t-i}^{t} between point clouds p_t and p_{t-1} can be obtained from odometry estimation approaches such as LIO-SAM [28], where T_{t-i}^{t} represents the homogeneous transformation matrix $\left(T_{i-1}^{i} \in \mathbb{R}^{4\times4},\ i \in n\right)$. The point cloud sequence is aligned to the current coordinate system, and a voxel grid with resolution σ is generated based on the coordinate range of the current frame point cloud. The semantic labels of each point in the current and historical frames are sequentially mapped into the voxel grid, discarding points that fall outside the range. For each voxel cell, the most frequent semantic label is selected, and redundant labels are filtered out. Finally, the semantic labels in the voxel grid are re-mapped to the corresponding point cloud channel. In Figure 6, red points represent dynamic attribute semantic labels, and black points represent static attribute semantic labels. To achieve efficient post-processing, a sliding window of length is used to update the points falling within the grid. When a new LiDAR scan frame is received, it is added to the sliding window, and old scan frames are removed.

3. Experiments

3.1. Experiment Setups

The SemanticKITTI-MOS [18] dataset is utilized for training and testing. SemanticKITTI-MOS [29] is a popular benchmark for LiDAR-based moving object segmentation in driving scenes. It consists of 22 sequences, with sequences 00–07 and 09–10 used for the training set, sequence 08 used for the validation set, and sequences 11–21 used for the test set.

To quantify the performance of the dynamic object segmentation network, standard metrics including Intersection-over-Union (IoU) and accuracy are employed. R_{IoU} represents the ratio of the intersection to the union between predicted and ground truth categories, while D_{acc} indicates the proportion of correctly predicted dynamic points among the true dynamic points.

$$R_{\mathrm{IoU}} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FP}} + N_{\mathrm{FN}}} \tag{6}$$

$$D_{acc} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP\_truth}}} \tag{7}$$

where N_{TP}, N_{FP}, and N_{FN} represent the number of true positives, false positives, and false negative predictions for the moving class, respectively. $N_{\mathrm{TP\_truth}}$ represents the number of true dynamic object points.

To reflect the spatial complexity of the network, the number of parameters is used in this section. The specific formula is as follows:

$$P_{total} = \sum_{l=1}^{L} (W_l + B_l) \tag{8}$$

where P_{total}, L, W_l, and B_l represent the total parameter number, the number of layers of the network model, the number of weights for layer l, and the number of offsets in layer l, respectively.

3.2. Implementation Details

PyTorch is used to implement the proposed method, which is trained on an NVIDIA RTX 4090 GPU with a batch size of 5. The network input data are represented by range images with a size of 64×2048. As for hyperparameters, the proposed method uses stochastic gradient descent with a momentum of 0.9 and weight decay of 0.0001 to minimize the loss function. The maximum number of training epochs is set to 150, and inference is performed on the validation set at the end of each epoch.

3.3. Analysis of Experimental Results

Based on the SemanticKITTI dataset, this paper compares state-of-the-art dynamic object segmentation networks such as LMNet, MotionSeg3D-v1 (where v1 denotes the use of KNN post-processing), MotionSeg3D-v2 (where v2 indicates the use of a refinement module), RVMOS, and SalsaNext; point-based methods like InsMOS; and BEV-based methods like LiMoSeg. The accuracy and comparative performance on the validation set (Seq08) are shown in Table 1. The proposed method achieves an IoU of 72.19% in dynamic object segmentation, improving by 9.68% over the baseline network LMNet and 4.86% over MotionSeg3D-v1, and outperforming MotionSeg3D-v2 which uses a refinement module. In the RVMOS network, which consumes fewer computational resources, a similar segmentation accuracy is achieved compared to our method, though this network uses moving object labels and semantic labels during training. Similarly, the InsMOS network, which utilizes instance label information and additional training data, achieves an IoU improvement of 1.01% over our method but has an inference time of 2.6 times longer. In terms of dynamic point cloud prediction accuracy, our method shows a 4.26% improvement over the pre-improvement MotionSeg3D network and achieves an accuracy of 81.76%, similar to the InsMOS network. Therefore, when balancing inference time and segmentation accuracy, our method demonstrates superior dynamic object segmentation performance using only single-moving object labels.

Table 1. Comparison of Intersection over Union, accuracy, parameters, and inference time.

Methods	$R_{\mathbf{IoU}}$	D_{acc}	Params [M]	Inference Time [ms]
LMNet	62.51	74.48	6.71	35
SalsaNext	46.6	52.69	6.73	41.67
LiMoSeg	52.6	-	-	8
MotionSeg3D-v1	67.33	77.50	10.41	42.53
MotionSeg3D-v2	71.42	79.58	21.77	112
RVMOS	71.2	-	2.63	29
InsMOS	73.2	82.12	25.35	127
Ours	72.19	81.76	12.18	48.23

The segmentation results are visualized in Figure 7, where dynamic objects are marked in red, blue circles indicate dynamic objects misclassified as static, and green rectangles denote static objects misclassified as dynamic. In scene 1 of Figure 7, a vehicle in the rear, which was obscured by a vehicle in front in the previous frame, appears in the current scan. Both LMNet and MotionSeg3D algorithms incorrectly classify the rear-moving vehicle as static. Thanks to the AMF appearance–motion feature fusion module proposed in this

paper, the semantic segmentation network does not rely solely on single-motion features but integrates a joint attention mechanism and motion guidance module to dynamically adjust the feature weight distribution between appearance and motion features. This enables the correct classification of moving object attributes, even when dynamic targets are occluded in the previous frame, by learning their continuous motion states and adaptively distributing feature weights. The InsMOS network, which incorporates instance label information, also correctly predicts the dynamic attributes of the rear vehicle. In the remaining two scenes, the proposed network and InsMOS network show better visualization results than LMNet and MotionSeg3D. Moreover, the post-processing recovery strategy implemented in this paper reduces the occurrence of false positives and false negatives in the scene compared to the InsMOS network.

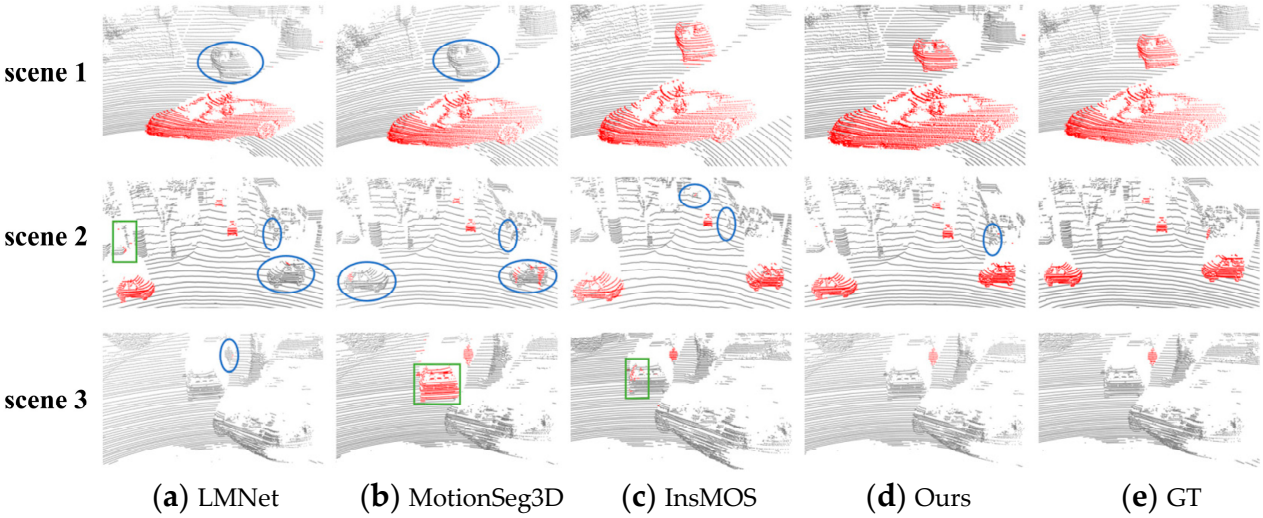
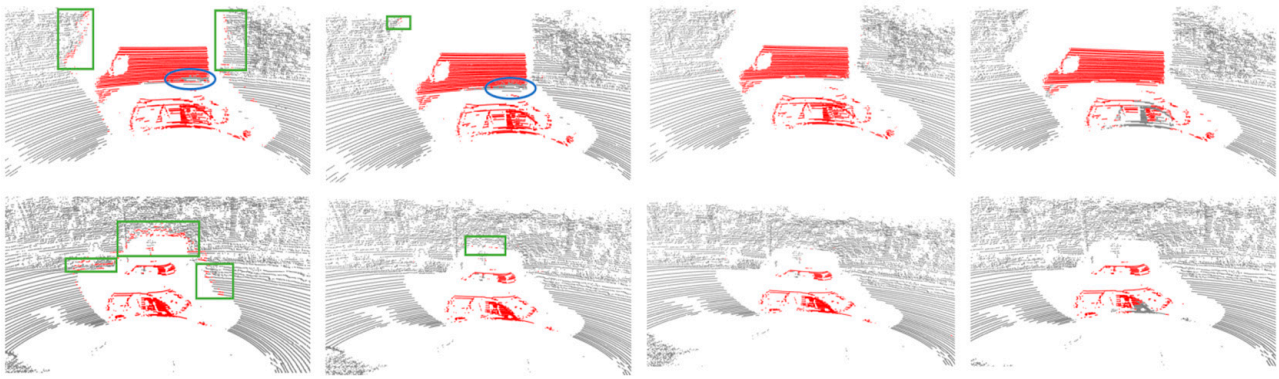


Figure 7. Visualization of segmentation for dynamic objects. Three scenarios were chosen for visualizing analysis results. Rectangular frames indicate the misclassification of boundary points as dynamic attributes, while ellipses show the misclassification of dynamic points due to occlusion.

To address the issue of boundary blur resulting from semantic segmentation, this chapter designs a majority voting strategy for post-processing and compares it with mainstream CRF and KNN post-processing methods, as shown in Figure 8. Among them, red points represent dynamic attribute semantic labels, and black points represent static attribute semantic labels. It can be observed that CRF and KNN algorithms do not handle the boundary blur problem in depth maps effectively, leading to some boundary points being misclassified as dynamic attributes (indicated by the rectangular frames). The MVS majority voting strategy post-processing method designed in this chapter, by integrating semantic prediction information from historical frame point clouds, determines the semantic category of the current state based on consistent attributes of static objects observed in adjacent frames. The results indicate that the MVS majority voting strategy effectively reduces boundary blur. Additionally, the motion features of moving objects extracted from temporal information partly address the issue of dynamic point misclassification due to occlusion (shown in the ellipses).

In deep learning, parameters (Params) and floating point operations (FLOPs) represent the model's computational space complexity and time complexity, respectively. To achieve more accurate moving object segmentation, the AMF module proposed in this chapter increases both the parameter count and computational load, but the sliding window and parallel processing in the post-processing stage result in overall algorithm processing time being close to that of the baseline model (as shown in Table 2).



(**a**) CRF (**b**) k-NN (**c**) Ours (**d**) GT

Figure 8. Comparison of post-processing. Two scenarios were chosen for visualizing analysis results. Rectangular frames indicate the misclassification of boundary points as dynamic attributes, while ellipses show the misclassification of dynamic points due to occlusion.

Table 2. Comparison of operation efficiency.

	Params [M]	FLOPs [G]	Inference Time [ms]
Baseline Model	10.41	523.28	40.78
Baseline Model + CRF			51.05
Baseline Model + k-NN			42.53
Baseline Model + MVS	12.18	553.54	48.23

3.4. Ablation Study

To verify the effectiveness of semantic segmentation performance on dynamic objects, ablation experiments were conducted on the appearance–motion feature fusion module and the MVS majority voting strategy post-processing method using the MotionSeg3D network as the baseline model. The experimental results are shown in Table 3.

Table 3. Ablation experiment results.

Baseline Models	AMF	MVS	R_{IoU}	Δ
√			61.93	Baseline
√	√		65.69	+3.76
√		√	68.43	+6.5
√	√	√	72.19	+10.26

Table 3 shows that both our proposed AMF module and the MVS post-processing method effectively improve the accuracy of dynamic object segmentation. The AMF module with shared attention effectively balances the multimodal features of dynamic objects, resulting in a 3.76% improvement compared to the baseline model. The post-processing method based on the MVS majority voting strategy fully utilizes temporal information, leading to a 6.5% improvement over the baseline model. By combining these two proposed methods, the network model achieves an IoU accuracy of 72.19%, representing a 10.26% improvement over the baseline network model. Table 4 compares the results with existing post-processing methods, all based on the network improved with AMF. The results indicate that the CRF method reduces accuracy in dynamic object segmentation tasks and fails to address boundary ambiguity issues, while the k-NN method partially alleviates this problem, resulting in a 1.97% improvement compared to no post-processing. The MVS method proposed in this paper outperforms the above two mainstream post-processing methods, achieving a 6.5% improvement.

Table 4. Comparison of post-processing.

Post-Processing	R_{IoU}	Δ
No Post-Processing	65.69	Baseline
CRF	64.53	−1.16
k-NN	67.66	+1.97
MVS	72.19	+6.5

The MVS is based on point cloud voxelization, wherein a fixed grid size voxel grid is set up to store and filter semantic labels. Smaller grids can better capture objects with fine boundaries, while larger grids are beneficial for handling larger dynamic objects. The impact of grid size settings on segmentation accuracy is shown in Figure 9. The results indicate that the setting with a fixed resolution of 20 cm achieves optimal accuracy. Although a voxel size of 20 cm performs well on the KITTI dataset, different datasets may have varying requirements for voxel size. In future work, the adaptability of different voxel sizes across various datasets will be planned for exploration, and these findings will be incorporated into further discussions and analyses.

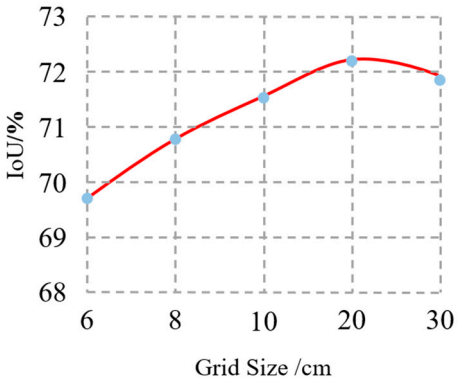


Figure 9. Effect of voxel grid size.

4. Conclusions

In this paper, a dual-branch dynamic object segmentation network based on the fusion of spatio-temporal information is proposed. The proposed method designs an appearance–motion fusion module with shared attention, which dynamically adjusts feature weights to achieve cross-modal feature fusion. To enhance the segmentation accuracy of the network model and reduce boundary ambiguity, a post-processing method based on a majority voting strategy has been further proposed, which utilizes temporal information between keyframes to recover misclassified semantic labels. The experimental results on the SemanticKITTI semantic segmentation dataset show that the method proposed in this paper outperforms state-of-the-art dynamic object segmentation networks such as LMNet and MotionSeg3D, achieving an IoU of 72.19%. This represents an improvement of 9.68% and 4.86% over LMNet and MotionSeg3D, respectively. These findings support the effectiveness of the appearance–motion feature fusion module and the majority voting strategy in dynamic object segmentation.

Although this method has demonstrated excellent performance on the KITTI dataset, different datasets may impose varying requirements on the model. Future work will explore the adaptability of the model to multiple datasets and incorporate these explorations into further discussions and analyses. Additionally, the current method has not yet been deployed on mobile edge computing devices or validated through real-vehicle testing, so future research will focus on further optimizing and validating the algorithm's performance.

Author Contributions: Conceptualization, F.H. and Z.W.; methodology, Y.Z.; software, Q.W.; validation, Q.W. and B.H.; formal analysis, B.H.; investigation, B.H. and Q.W.; resources, Y.X.; data curation, B.H.; writing—original draft preparation, Q.W.; writing—review and editing, B.H. and Y.X.; visualization, Y.Z.; supervision, F.H. and Z.W.; project administration, Y.X.; funding acquisition, Y.X. and B.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Innovation Key R&D Program of Chongqing, grant number CSTB2022TIAD-STX0003, and the Research and Innovation Program for Graduate Students in Chongqing Jiaotong University (YYK202405).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Authors Fei Huang, Zhiwen Wang and Yu Zheng were employed by the company China Road & Bridge Corporation. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Chen, X.; Milioto, A.; Palazzolo, E.; Giguere, P.; Behley, J.; Stachniss, C. Suma++: Efficient lidar-based semantic slam. In Proceedings of the2019 IEEE/2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: New York, NY, USA, 2019; pp. 4530–4537.
2. Baur, S.A.; Emmerichs, D.J.; Moosmann, F.; Pinggera, P.; Ommer, B.; Geiger, A. Slim: Self-supervised lidar scene flow and motion segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; IEEE: New York, NY, USA, 2021; pp. 13126–13136.
3. Tishchenko, I.; Lombardi, S.; Oswald, M.R.; Pollefeys, M. Self-supervised learning of non-rigid residual flow and ego-motion. In Proceedings of the 2020 International Conference on 3D Vision (3DV), Fukuoka, Japan, 25–28 November 2020; IEEE: New York, NY, USA, 2020; pp. 150–159.
4. Chen, P.; Pei, J.; Lu, W.; Li, M. A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance. *Neurocomputing* **2022**, *497*, 64–75. [CrossRef]
5. Pomerleau, F.; Krüsi, P.; Colas, F.; Furgale, P.; Siegwart, R. Long-term 3D map maintenance in dynamic environments. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; IEEE: New York, NY, USA, 2014; pp. 3712–3719.
6. Underwood, J.P.; Gillsjö, D.; Bailey, T.; Vlaskine, V. Explicit 3D change detection using ray-tracing in spherical coordinates. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; IEEE: New York, NY, USA, 2013; pp. 4735–4741.
7. Kim, G.; Kim, A. Remove, then revert: Static point cloud map construction using multiresolution range images. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; IEEE: New York, NY, USA, 2020; pp. 10758–10765.
8. Schauer, J.; Nüchter, A. The peopleremover-removing dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1679–1686. [CrossRef]
9. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5105–5114.
10. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3D object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lak City, UT, USA, 18–23 June 2018; IEEE: New York, NY, USA, 2018; pp. 4490–4499.
11. Cortinhal, T.; Tzelepis, G.; Erdal Aksoy, E. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In Proceedings of the Advances in Visual Computing: 15th International Symposium, San Diego, CA, USA, 5–7 October 2020; Springer: Berlin, Germany, 2020; pp. 207–222.
12. Wang, D.F.; Shang, H.; Cao, J.; Wang, T.; Xia, X.; Han, Y. Semantic segmentation of point clouds in autonomous driving scenes based on self-attention mechanism. *Automot. Eng.* **2022**, *44*, 1656–1664.
13. Mersch, B.; Chen, X.; Vizzo, I.; Nunes, L.; Behley, J.; Stachniss, C. Receding moving object segmentation in 3d lidar data using sparse 4d convolutions. *IEEE Robot. Autom. Lett.* **2022**, *7*, 7503–7510. [CrossRef]
14. Wang, N.; Shi, C.; Guo, R.; Lu, H.; Zheng, Z.; Chen, X. InsMOS: Instance-Aware Moving Object Segmentation in LiDAR Data. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; IEEE: New York, NY, USA, 2023; pp. 7598–7605.
15. Graham, B.; Engelcke, M.; Van Der Maaten, L. 3D Semantic segmentation with submanifold sparse convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: New York, NY, USA, 2018; pp. 9224–9232.

16. Yan, X.; Gao, J.; Li, J.; Zhang, R.; Li, Z.; Huang, R.; Cui, S. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; AAAI: Menlo Park, CA, USA, 2021; Volume 35, pp. 3101–3109.
17. Chen, X.; Li, S.; Mersch, B.; Wiesmann, L.; Gall, J.; Behley, J.; Stachniss, C. Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6529–6536. [CrossRef]
18. Kim, J.; Woo, J.; Im, S. Rvmos: Range-view moving object segmentation leveraged by semantic and motion features. *IEEE Robot. Autom. Lett.* **2022**, *7*, 8044–8051. [CrossRef]
19. Sun, J.; Dai, Y.; Zhang, X.; Xu, J.; Ai, R.; Gu, W.; Chen, X. Efficient spatial-temporal information fusion for lidar-based 3d moving object segmentation. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; IEEE: New York, NY, USA, 2022; pp. 11456–11463.
20. Wang, N.; Hou, Z.Q.; Zhao, M.Q.; Yu, W.; Ma, S. Semantic segmentation algorithm combined with edge detection. *Comput. Eng.* **2021**, *47*, 257–265.
21. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; IEEE: New York, NY, USA, 2018; pp. 1887–1893.
22. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. Rangenet++: Fast and accurate lidar semantic segmentation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: New York, NY, USA, 2019; pp. 4213–4220.
23. Yang, S.; Zhang, L.; Qi, J.; Lu, H.; Wang, S.; Zhang, X. Learning motion-appearance co-attention for zero-shot video object segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; IEEE: New York, NY, USA, 2021; pp. 1564–1573.
24. Wang, T.; Wang, W.J.; Cai, Y. Research on semantic segmentation methods for 3D point clouds based on deep learning. *Comput. Eng. Appl.* **2021**, *57*, 18–26.
25. Xia, X.T.; Wang, D.F.; Cao, J.; Zhang, G.; Zhang, J. Semantic segmentation of vehicle-mounted LiDAR point clouds based on sparse convolutional neural networks. *Automot. Eng.* **2022**, *44*, 26–35.
26. Fan, L.; Xiong, X.; Wang, F.; Wang, N.; Zhang, Z. Rangedet: In defense of range view for lidar-based 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; IEEE: New York, NY, USA, 2021; pp. 2918–2927.
27. Stergiou, A.; Poppe, R. Adapool: Exponential adaptive pooling for information-retaining downsampling. *IEEE Trans. Image Process.* **2022**, *32*, 251–266. [CrossRef] [PubMed]
28. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020.
29. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: New York, NY, US, 2019; pp. 9297–9307.