TFNet: Exploiting Temporal Cues for Fast and Accurate LiDAR Semantic Segmentation

Rong Li[1]   Shijie Li[2†]   Xieyuanli Chen[2]   Teli Ma[1]   Juergen Gall[2,4]   Junwei Liang[1,3†]

[1] HKUST(GZ), China   [2] University of Bonn, Germany   [3] HKUST, China

[4] Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

rli335@connect.hkust-gz.edu.cn   lsj@uni-bonn.de   junweiliang@hkust-gz.edu.cn

Abstract

LiDAR semantic segmentation plays a crucial role in enabling autonomous driving and robots to understand their surroundings accurately and robustly. A multitude of methods exist within this domain, including point-based, range-image-based, polar-coordinate-based, and hybrid strategies. Among these, range-image-based techniques have gained widespread adoption in practical applications due to their efficiency. However, they face a significant challenge known as the "many-to-one" problem caused by the range image's limited horizontal and vertical angular resolution. As a result, around 20% of the 3D points can be occluded. In this paper, we present TFNet, a range-image-based LiDAR semantic segmentation method that utilizes temporal information to address this issue. Specifically, we incorporate a temporal fusion layer to extract useful information from previous scans and integrate it with the current scan. We then design a max-voting-based post-processing technique to correct false predictions, particularly those caused by the "many-to-one" issue. We evaluated the approach on two benchmarks and demonstrated that the plug-in post-processing technique is generic and can be applied to various networks.

1. INTRODUCTION

LiDAR (light detection and ranging) semantic segmentation enables a precise and fine-grained understanding of the environment for robotics and autonomous driving applications [2, 6, 56]. There are four categories of methods: point-based [19, 27, 28, 34, 35, 37, 40], range-image-based [13, 15, 16, 32, 46, 54], polar-based [53] and hybrid methods [24, 36]. Despite point-based methods achieving remarkable scores in metrics such as mean Intersection over Union (mIoU) and Accuracy, they tend to underper-
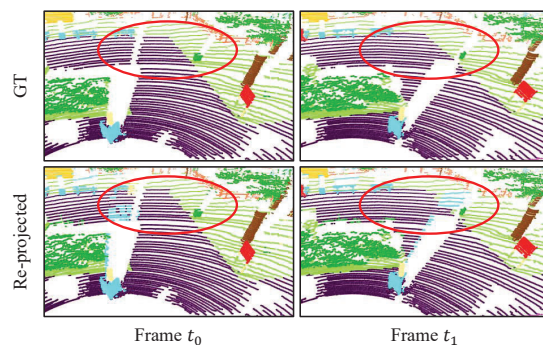
† Corresponding authors.

Figure 1. Range-image-based methods suffer from the "many-to-one" problem where multiple 3D points with the same angle are mapped to a single range pixel. Marked by the red circles of frame t_0, this can cause distant terrain points (purple) to receive erroneous predictions from nearby billboard points (blue) when the range image is re-projected to 3D. Furthermore, occluded points in frame t_0 become visible in t_1, offering an opportunity to refine the predictions.

form in terms of computational efficiency. In contrast, the range-image-based methods are orders of magnitude more efficient than the other methods as substantiated by studies [21, 41]. This efficiency is further enhanced by the direct applicability of well-optimized Convolutional Neural Network (CNN) models, which strike a balance between speed and accuracy. Given the requirement of real-time performance and computational efficiency for ensuring safety in practical applications, the distinctive advantages of range-image-based methods make them a suitable choice for LiDAR semantic segmentation in real-world scenarios.

However, the range view representation suffers from a boundary-blurring effect [32, 54]. This problem exists mainly because of the limited horizontal and vertical angular resolution: more than one point will be projected to the same range image pixel when these points share the same

vertical and horizontal angle. When multiple points share identical vertical and horizontal angles, they are projected onto the same pixel in the range image, giving rise to what is also referred to as the "many-to-one" problem [54]. Considering that the projection computes distant points first and near points later [32], the distant points will be occluded by the near points. Hence, when converting the range image back into 3D coordinates, which is essential for range-image-based methods, the farther points receive the same label as the overlapping points that are closer. This leads to inaccuracies in the semantic understanding of the scene.

Fig. 1 offers an illustration of this problem. Imagine that the LiDAR sensor is situated at the bottom-left of each range image. Close to the sensor, there is a billboard colored blue, and farther away is the terrain, displayed in purple. At time t_0, as marked by the red circle, even though the terrain and billboard are physically separate objects, some points on the terrain are incorrectly labeled as part of the billboard. This happens because these points, due to their similar angles relative to the LiDAR sensor, get projected onto the same pixel in the range image. Upon the movement of the car and the consequent change in the sensor's field of view, we see a different scenario at time t_1: the previously mislabeled terrain points are now accurately classified. This improvement is attributed to the fact that their angular positions relative to the LiDAR sensor have changed, allowing them to avoid being hidden or masked by the billboard. This example illustrates how the dynamic movement affects LiDAR-based semantic segmentation and underscores the possibility of developing reliable and adaptable methods to tackle the "many-to-one" issue in this context.

We quantitatively assess the effects of this phenomenon on the SemanticKITTI dataset [2, 3]. Under standard conditions, where the range image dimensions are set to 64 and 2048 for height and width, respectively, it is observed that more than 20% of the 3D points are occluded within the range image, i.e., more than one point is projected to the same pixel. As detailed in Tab. 3, this results in a substantial degradation of the accuracy if it is not addressed by an additional post-processing step. Therefore, various post-processing approaches like k-NN [32], CRF [46], or NLA [54] have been proposed. As an example, NLA [54] resorts to assigning the label of the closest non-occluded point to occluded points. Nonetheless, this process necessitates checking each individual point for occlusion, which undermines the inherent efficiency of range-image-based methods. A detailed discussion about these methodologies can be found in Section 2.

In this work, we propose to incorporate temporal information to address the "many-to-one" challenge for LiDAR semantic segmentation. This is inspired by human visual perception, where temporal information is crucial for understanding object motion and identifying occlusions.

This is also observed in LiDAR semantic segmentation, where heavily occluded points can be captured from adjacent range image scans, as shown in Fig. 1. Based on this intuition, we exploit the temporal relations of features in the range map via cross-attention [17, 22, 42]. As for the inference stage, we propose a max-voting-based post-processing scheme that effectively reuses the predictions of past frames. To this end, we transform the previous scans with predicted semantic class labels into the current ego car coordinate frame and then obtain the final segmentation by aggregating the predictions within the same voxel by max-voting. In summary, we make the following three contributions:

- We quantitatively and qualitatively analyze and explain the "many-to-one" issues existing in range-image-based methods.
- We propose TFNet, a range-based LiDAR semantic segmentation method. It utilizes a temporary cross-attention layer, which extracts informative features from previous LiDAR scans and combines them with current range features, to compensate for occluded objects.
- We design a temporal-based post-processing method to solve the "many-to-one" mapping issue in range images. Compared with previous post-processing steps, our method achieves better performance, which is verified for various networks.
- We evaluate the proposed method on two public benchmarks, namely SemanticKITTI [2] and Semantic-POSS [33], where our method achieves a good trade-off between accuracy and inference time.

2. RELATED WORK

LiDAR semantic segmentation. The LiDAR sensor captures high-fidelity 3D structural information, which can be represented by various formats, i.e., points [34, 35, 40], range view [13, 21, 32, 46, 54], voxels [14, 27, 55], bird's eye view (BEV) [8], hybrid [24, 36] and multi-modal representations [7, 51, 56]. There are also some works [51, 56] that fuse multi-sensor information. The point and voxel methods are prevailing, but their complexity is $\mathcal{O}(N \cdot d)$ where N is in the order of 10^5 [41]. Thus, most approaches are not suitable for robotics or autonomous driving applications. The BEV method [8] offers a more efficient choice with $\mathcal{O}(\frac{H \cdot W}{r^2} \cdot d)$ complexity, but the accuracy is subpar [21]. The multi-modal methods require additional resources to process the additional modalities. Among all representations, the range view reflects the LiDAR sampling process and it is much more efficient than other representations with $\mathcal{O}(\frac{H \cdot W}{r^2} \cdot d)$ complexity. We thus focus on the range-view as representation.

Multi-frame LiDAR data processing. Multi-frame information plays a crucial role in LiDAR data processing. For example, MOS [12] and MotionSeg3D [39] generate resid-

ual images from multiple LiDAR frames to explore the sequential information and use it for segmenting moving and static objects. Motivated by these approaches, Meta-RangeSeg [45] also uses residual range images for the task of semantic segmentation of LiDAR sequences. It employs a meta-kernel to extract the meta features from the residual images. SeqOT [29] exploits sequential LiDAR frames using yaw-rotation-invariant OverlapNets [10, 11] and transformer networks [30, 42] to generate a global descriptor for fast place recognition in an end-to-end manner. In addition, SCPNet [48] designs a knowledge distillation strategy between multi-frame LiDAR scans and a single-frame LiDAR scan for semantic scene completion. Recently, Mars3D [25] designed a plug-and-play motion-aware module for multi-scan 3D point clouds to classify semantic categories and motion states. Seal [26] proposes a temporal consistency loss to constrain the semantic prediction of super-points from multiple scans. Although the benefit of using multiple scans has been studied, these works address other tasks.

Post processing. Although range-view-based LiDAR segmentation methods are computationally efficient, they suffer from boundary blurriness or the "many-to-one" issue [32, 46] as discussed in Sec. 1. To alleviate this issue, most works use a conditional random field (CRF) [46] or k-NN [32] to smooth the predicted labels. [46] implements the CRF as an end-to-end trainable recurrent neural network to refine the predictions according to the predictions of the neighbors within three iterations. It does not address occluded points explicitly. k-NN [32] infers the semantics of ambiguous points by jointly considering its k closest neighbours in terms of the absolute range distance. However, finding a balance between under and over-smoothing can be challenging, and it may not be able to handle severe occlusions. Recently, NLA [54] assigns the nearest point's prediction in a local patch to the occluded point. However, it is required to iterate over each point to verify occlusions. In addition, RangeFormer [21] addresses this issue by creating sub-clouds from the entire point cloud and inferring labels for each subset. However, partitioning the cloud into sub-clouds ignores the global information. It can also not easily be applied to existing networks. Some methods [1, 20, 39] propose additional refinement modules for the networks to refine the initial estimate, which increases the runtime. In this work, we propose to tackle this issue by combining past predictions in an efficient max-voting manner. Our method complements existing approaches and can be applied to various networks.

3. PROPOSED METHOD

3.1. Network Overview

The overview of our proposed network is illustrated in Fig. 2. Our proposed network takes as input a point cloud P comprising N points represented by 3D coordinates x, y, z, and intensity i. The point cloud is projected onto a range image I of size $H \times W \times 5$ using a spherical projection technique employed in previous works [32, 46]. Here, H and W represent the height and width of the image, and the last dimension includes coordinates (x, y, z), range $r = \sqrt{x^2 + y^2 + z^2}$, and intensity i. Next, we feed the range image into our backbone model to obtain multi-scale features F with resolutions $\{1, 1/2, 1/4, 1/8\}$. We employ a **T**emporal **C**ross-**A**ttention (**TCA**) layer to integrate spatial features from the history frame. The aggregated features are then fed to the segmentation head, which predicts the range-image-based semantic segmentation logits O. For inference, we re-project the 2D semantic segmentation prediction to a 3D point-wise prediction S. Subsequently, we propose a **M**ax-**V**oting-based **P**ost-processing (**MVP**) strategy to refine the current prediction S_t by aggregating previous predictions. We describe the key components of our network in the following sections.

3.2. Temporal cross attention

Although the range image suffers from the "many-to-one" issue, the occluded points can be captured from adjacent scans. This observation motivates us to incorporate sequential scans into both the training and inference stages. First, we discuss how sequential data can be exploited during the training stage.

Inspired by the notable information extraction ability of the attention mechanism [42] verified by various other works [22, 31, 44, 49], we use the cross-attention mechanism to model the temporal connection between the previous range feature F_{t-1} and the current range feature F_t. The attended value is computed by:

$$\mathbf{x}_{in} = \text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{Q \cdot K^\mathsf{T}}{\sqrt{d_f}}\right) V. \tag{1}$$

where Q, K, V are obtained by $Q = Linear_q(F_t)$, $K = Linear_k(F_{t-1})$, $V = Linear_v(F_{t-1})$, and d_f is the dimension of the range features. We integrate a 3×3 convolution into the feed-forward module to encode positional information as in [49] as well as a residual connection [18]. The **feed-forward module** is defined as follows:

$$\mathbf{x}_{out} = \text{MLP}(\text{GELU}(\text{Conv}_{3 \times 3}(\text{MLP}(\mathbf{x}_{in})))) + \mathbf{x}_{in}. \tag{2}$$

The TCA module effectively exploits temporal dependencies in two ways. First, instead of using multiple stacked range features [12], our method extracts temporal information from the previous range features. This not only reduces computational costs but also minimizes the influence of moving objects, which can introduce noise into the data. Secondly, we only utilize the fusion module on the last feature level, which significantly decreases computation com-
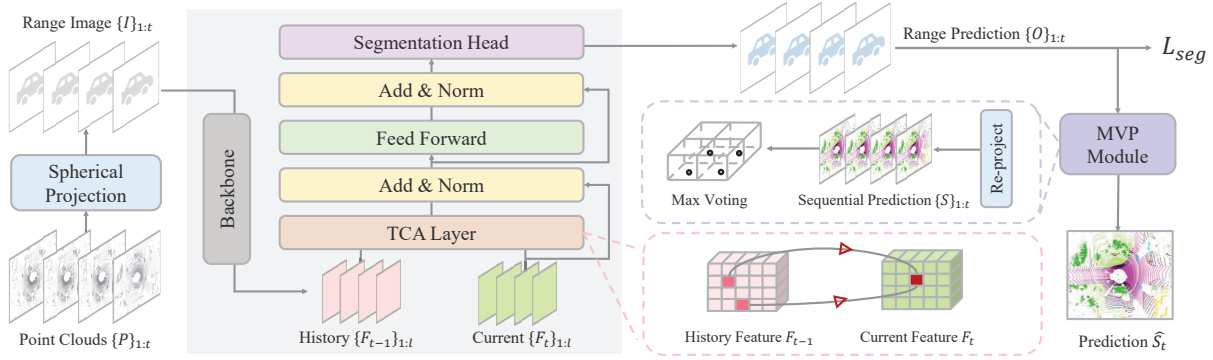
Figure 2. Architecture of TFNet. For a point cloud P_t, TFNet projects it onto range images I_t. It then uses a segmentation backbone to extract multi-scale features $\{F_t\}_{1:l}$, a **T**emporal **C**ross-**A**ttention (**TCA**) layer to integrate past features $\{F_{t-1}\}_{1:l}$, and a segmentation head to predict range-image-based logits O_t. In inference, it refines the re-projected prediction S_t by aggregating the current and past temporal predictions $\{S\}_{1:t}$ by a **M**ax-**V**oting-based **P**ost-processing (**MVP**) strategy.

plexity. Previous works [17, 22] have shown that the attention at shallower layers is not effective.
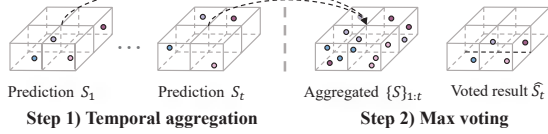
3.3. Max-voting-based post-processing

Figure 3. Illustration of the max voting post-processing strategy.

While temporal cross attention exploits temporal information at the feature level, it does not resolve the "many-to-one" issue during the re-projection process of a range-image-based method, which causes occluded far points to inherit the predictions of near points. We thus propose a max-voting-based post-processing (MVP) strategy, which is motivated by the observation that occluded points will be visible in the adjacent scans as shown in Fig. 1. As verified in Tab. 5, MVP is generic and can be added to various networks.

Temporal scan alignment. To initiate post-processing, it is essential to align a series of past LiDAR scans $(P_1, ..., P_t)$ to the viewpoint (i.e., coordinate frame) of P_t. The alignment is accomplished by utilizing the estimated relative pose transformations T_{j-1}^j between the scans P_{j-1} and P_j. These transformation matrices can be acquired from an odometry estimation approach such as SuMa++ [9]. The relative transformations between the scans $(T_1^2, ..., T_{t-1}^t)$ are represented by transformation matrices of $T_{j-1}^j \in \mathbb{R}^{4 \times 4}$. Further, we denote the jth scan transformed to the

viewpoint of P_t by

$$P^{j \to t} = \{T_j^t p_i\}_{p_i \in P_j} \quad \text{with } T_j^t = \prod_{k=j+1}^{t} T_{k-1}^k. \quad (3)$$

Sparse grid max voting. After applying the transformations, we aggregate the aligned scans. We quantize the aggregated scans into a voxel grid with a fixed resolution δ. In each grid, we use the max-voting strategy to use the most frequently predicted class label to represent the semantics of all points in the grid. We illustrate this process in Fig. 3 and evaluate the impact of the grid size in Fig. 4. To save computation and memory, we store only the non-empty voxels. This sparse representation allows our method to handle large scenes.

Sliding window update. We initialize a sliding window $W_{t-L+1:t}$ with the length of L to store the scans and use a FIFO (First In First Out) strategy to update the points falling in each grid. When the LiDAR sensor obtains a new point cloud scan, we add it to this sliding window and remove the oldest scan. We do not use different weights across frames due to the uncertain occlusion problem.

4. EXPERIMENTS

Datasets and evaluation metrics. We evaluate our proposed method on SemanticKITTI [2] and Semantic-POSS [33]. SemanticKITTI [2] is a popular benchmark for LiDAR-based semantic segmentation in driving scenes. It contains 19,130 training frames, 4,071 validation frames, and 20,351 test frames. Each point in the dataset is provided with a semantic label of 19 classes for semantic segmentation. We also evaluate our dataset on the SemanticPOSS [33] dataset, which contains 2988 scenes for training and testing. For evaluation, we follow previous

Table 1. Comparison with other range-image-based LiDAR segmentation methods with resolution (64, 2048) on SemanticKITTI test set.

	mean-IoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign
MINet [23]	55.2	90.1	41.8	34.0	29.9	23.6	51.4	52.4	25.0	90.5	59.0	72.6	25.8	85.6	52.3	81.1	58.1	66.1	49.0	59.9
FIDNet [54]	59.5	93.9	54.7	48.9	27.6	23.9	62.3	59.8	23.7	90.6	59.1	75.8	26.7	88.9	60.5	84.5	64.4	69.0	53.3	62.8
Meta-RangeSeg [45]	61.0	93.9	50.1	43.8	_43.9_	43.2	63.7	53.1	18.7	90.6	64.3	74.6	29.2	91.1	64.7	82.6	65.5	65.5	56.3	64.2
KPRNet [20]	63.1	**95.5**	54.1	47.9	23.6	42.6	65.9	65.0	16.5	**93.2**	**73.9**	**80.6**	30.2	_91.7_	_68.4_	**85.7**	69.8	**71.2**	58.7	64.1
Lite-HDSeg [38]	63.8	92.3	40.0	_55.4_	37.7	39.6	59.2	_71.6_	**54.1**	93.0	68.2	78.3	29.3	91.5	65.0	78.2	65.8	65.1	59.5	_67.7_
CENet [13]	_64.7_	91.9	58.6	50.3	40.6	42.3	_68.9_	65.9	_43.5_	90.3	60.9	75.1	_31.5_	91.0	66.2	84.5	69.7	70.0	61.5	67.6
RangeViT [1]	64.0	95.4	55.8	43.5	29.8	42.1	63.9	58.2	38.1	_93.1_	70.2	_80.0_	**32.5**	**92.0**	**69.0**	_85.3_	_70.6_	_71.2_	60.8	64.7
LENet [16]	64.5	93.9	57.0	51.3	**44.3**	_44.4_	66.6	64.9	36.0	91.8	68.3	76.9	30.5	91.2	66.0	83.7	68.3	67.8	58.6	63.2
TFNet (Ours)	**66.1**	_94.3_	60.7	**58.5**	38.4	**48.4**	**74.3**	**72.2**	35.5	90.6	_68.5_	75.3	29.0	91.6	67.3	83.8	**71.1**	67.0	_60.8_	**68.7**

works [13, 21, 46, 54], utilizing the class-wise Intersection over Union (IoU) and mean IoU (mIoU) metrics to evaluate and compare with others.

Implementation details. While we use CENet [13] as the main baseline method, our method demonstrates robust generalization across various backbones as shown in the following experiments. We train the proposed method using the Stochastic Gradient Descent (SGD) optimizer and set the batch size to 8 and 4 for SemanticKITTI and Semantic-POSS, respectively. We follow the baseline method [13] to supervise the training with a weighted combination of cross-entropy, Lovász softmax loss [4], and boundary loss [5]. The weights for the loss terms are set to $\beta_1 = 1.0$, $\beta_2 = 1.5$, $\beta_3 = 1.0$, respectively. All the models are trained on GeForce RTX 3090 GPUs. The inference latency is measured using a single GeForce RTX 3090 GPU. The backbone is trained from scratch on all the datasets.

4.1. Comparison with state of the art

Quantitative results on SemanticKITTI. Tab. 1 reports comparisons with representative models on the SemanticKITTI test set. Our method outperforms all range-image-based methods, including CNN-based architectures [13, 23, 54] and Transformer-based architectures [1] in terms of mean IoU. CENet [13] uses test time augmentation to improve the performance. We do not use test time augmentation for a fair comparison with previous methods [23, 32].

Tab. 1 presents a comprehensive comparison of the proposed TFNet method against several range-image-based LiDAR segmentation models on the SemanticKITTI test set. Specifically, TFNet excels in segmenting cars, bicycles, motorcycles, and pedestrians, showing significant improvements in IoU values over other methods. It registers particularly high IoU scores for bicycles (60.7%), motorcycles (58.5%), and persons (74.3%). Despite not always secur-

Table 2. Evaluation results on the SemanticPOSS test set.

	Sq.Seg [46]	Sq.SegV2 [47]	RangeNet [32]	MINet [23]	FIDNet [54]	CENet [13]	TFNet (Ours)
person	6.8	43.9	57.3	62.4	72.2	**75.5**	_72.4_
rider	0.6	7.1	4.6	12.1	**23.1**	_22.0_	20.5
car	6.7	47.9	35.0	63.8	72.7	_77.6_	**77.7**
truck	4.0	18.4	14.1	22.3	23.0	**25.3**	_24.8_
plants	2.5	40.9	58.3	68.6	68.0	**72.2**	_71.6_
traffic-sign	9.1	4.8	3.9	16.7	_22.2_	18.2	**29.1**
pole	1.3	2.8	6.9	30.1	28.6	_31.5_	**37.8**
trashcan	0.4	7.4	24.1	28.9	16.3	**48.1**	_46.3_
building	37.1	57.5	66.1	75.1	73.1	_76.3_	**79.9**
cone/stone	0.2	0.6	6.6	**58.6**	34.0	27.7	_34.5_
fence	8.4	12.0	23.4	32.2	40.9	**47.7**	_47.3_
bike	18.5	35.3	28.6	44.9	50.3	51.4	**53.9**
ground	72.1	71.3	73.5	76.3	_79.1_	**80.3**	78.4
mean-IoU	12.9	26.9	30.9	43.2	46.4	_50.3_	**51.9**

ing the top position in every class, TFNet consistently delivers strong results, especially in small and medium-sized object classes. TFNet falls slightly behind in the pole and traffic-sign categories, where it records IoU scores lower than some methods like CENet [13] and KPRNet [20]. Nevertheless, its ability to maintain balanced and above-average performance across most classes contributes to its overall leadership in mean-IoU.

Quantitative results on SemanticPOSS. We present a

Table 3. Comparison with different post-processing methods. Our MVP method is significantly better.

	mean-IoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign
w/o MVP	60.4	85.8	44.0	61.5	80.3	53.0	68.7	70.2	0.91	**94.8**	42.1	80.9	0.95	81.8	52.4	83.2	60.3	70.6	51.9	47.9
CRF [46]	58.2 (-2.2)	87.0	40.0	57.3	67.7	52.2	66.1	62.5	0.38	94.5	**46.4**	**81.1**	0.66	81.7	53.6	81.4	60.9	66.3	49.0	47.6
PointRefine [39]	59.2 (-1.2)	84.5	43.7	53.7	76.3	48.6	68.3	70.6	<u>7.5</u>	94.6	39.8	80.5	**11.8**	81.4	50.7	83.8	59.4	<u>72.2</u>	51.1	46.1
NLA [54]	64.4 (+4.0)	92.0	47.5	66.8	79.0	<u>55.9</u>	76.2	<u>85.7</u>	**12.4**	94.5	42.7	80.8	<u>10.6</u>	87.3	54.6	85.9	66.0	<u>72.2</u>	63.4	49.8
k-NN [32]	<u>64.5</u> (+4.1)	<u>91.4</u>	<u>50.7</u>	<u>66.9</u>	<u>81.2</u>	54.9	<u>76.8</u>	85.1	0.96	94.5	41.6	80.9	0.95	<u>88.5</u>	<u>55.6</u>	<u>86.2</u>	<u>66.8</u>	71.5	<u>64.5</u>	<u>50.2</u>
MVP (Ours)	**66.5** (+6.1)	**93.4**	**54.1**	**70.2**	**85.9**	**59.8**	**79.8**	**88.0**	0.58	<u>94.7</u>	<u>44.8</u>	**81.1**	0.46	**90.3**	**66.6**	**86.8**	**69.5**	**72.7**	**65.1**	**50.3**

quantitative evaluation of our TFNet method against several range-image-based LiDAR segmentation models on the SemanticPOSS test set [33] in Tab. 2. Our method achieves the highest mean Intersection-over-Union (mIoU) among all listed methods, indicating overall better segmentation accuracy. Notably, TFNet excels in detecting smaller objects. It significantly surpasses CENet in segmenting traffic signs and poles, improving the IoU score by 6.9 percentage points and 6.3 percentage points, respectively. Furthermore, TFNet performs competitively in identifying cone/stone, achieving the second-best IoU score, closely following MINet's performance. Moreover, TFNet ranks second in multiple categories such as rider, plants, fence, and bike, demonstrating its strong generalizability across diverse object classes.

4.2. Ablation Analysis

Effect of the temporal post-processing. Tab. 3 compares the proposed post-processing method with other post-processing approaches on the SemanticKITTI validation set. Using a CRF for post-processing has been used by SqueezeSegv2 [47]. We train the network with CRF from scratch using the same training pipeline as our method. The k-Nearest Neighbor (k-NN) method [32] is the most popular post-processing method. It is widely used in Lite-HDseg [38], SequeezeSegv3 [50], CENet [13], SalsaNext [15], and MiNet [23]. The Nearest Label Assignment (NLA) post-processing is used by FIDNet [54]. It iterates over each point to check if a point is occluded or not. We use the source code from the corresponding methods. For the Point Refine module proposed in MotionSeg3D [39], we follow its implementation. We use SPVCNN [27] as the Point Refine module and use the features before the classification layer as the input to the Point Refine module. We then fine-tune the network with the Point Refine module in a second stage with a 0.001 learning rate for ten epochs. The results show the "many-to-one" is-

Table 4. Comparison with other temporal fusion methods.

Fusion Strategies	mIoU
w/o TCA	66.9
TMA module [43]	67.8 (+0.9)
Residual images [45]	61.4 (-5.5)
Element-wise addition [25]	67.6 (+0.7)
Channel concatenation [52]	<u>68.0</u> (+1.1)
TCA module (ours)	**68.1** (+1.2)

sue harms the performance heavily. Without our proposed post-processing ('w/o MVP'), the mean IoU is 6.1 lower. That CRF can actually decrease the mean IoU has also been shown in [32]. While NLA and k-NN improve the results, the best mean IoU is achieved by our approach.

Effect of different fusion strategy. In Tab. 4, we replace the proposed temporal fusion layer with other strategies. Mars3D [25] adopts element-wise summation to aggregate temporal multi-scan point cloud embeddings and produce enhanced features. The temporal memory attention (TMA) module [43] validates its effectiveness on the video semantic segmentation task. BEVFormer v2 [52] uses a feature warp and concatenation strategy to incorporate temporal information and shows its effectiveness on the LiDAR detection task. We follow its implementation, which concatenates previous BEV features with the current BEV feature along the channel dimension and employs residual blocks for dimensionality reduction. We transform the scans to the same ego-car coordinates to implement the accurate alignment between temporal scans. For the LiDAR semantic segmentation task, Meta-RangeSeg [45] proposes to use three previous residual images as input and a meta-kernel module to incorporate temporal information. We follow its implementation and add to the five-channel input (x,y,z,r,i) three

Table 5. Performance on other range-image-based methods.

Backbone	Post-processing		
	-	k-NN [32]	MVP (Ours)
FIDNet [54]	55.4	58.6 (+3.2)	**61.5** (+6.1)
Meta-RangeSeg [45]	56.6	60.3 (+3.7)	**63.1** (+6.5)
CENet [13]	58.8	62.6 (+3.8)	**64.7** (+5.9)

(a) Effect of number of scans. (b) Effect of grid size.

Figure 4. Effect of window size and grid size resolution.

channels for the three residual images and a channel for the mask, which indicates whether the pixel is a projected 3D point or not. The residual images are calculated by first transforming the point clouds of previous frames into the coordinates of the current frame and then calculating the absolute differences between the range values of the current scan and the transformed one with normalization. A meta-kernel is followed to capture the spatial and temporal information. For a fair comparison, we keep the encoder and decoder of our architecture. We report the projection-based mIoU here because the loss function is applied directly to the range image. All strategies are trained with the same setting and pipeline. The results in Tab. 4 show that our temporal fusion approach performs best.

4.3. Generalization Ability

Tab. 5 presents the effectiveness of the proposed max-voting-based post-processing (MVP) technique when integrated with three different range-image-based semantic segmentation methods, specifically FIDNet [54], Meta-RangeSeg [45], and CENet [13]. Unlike results reported in Tab. 1, which reflect performances on the test set, this table displays the outcomes obtained on the validation set using publicly available pre-trained models with and without post-processing. For each backbone model, the table compares three post-processing scenarios: no post-processing (denoted as '-'), application of the k-NN method from [32], and our proposed MVP. Each row shows the mean Intersection-over-Union (IoU) scores resulting from these treatments.

It is evident from the table that employing the MVP consistently leads to notable improvements over the baseline scores (without any post-processing) and often surpasses the performance of k-NN post-processing. For instance, MVP increases the IoU score of FIDNet by 6.1 points compared to its base result, demonstrating superior refinement capabilities. Similarly, the IoU scores of Meta-RangeSeg and CENet also witness considerable boosts with the use of MVP, affirming its broad applicability and positive impact on various range-image-based semantic segmentation models.
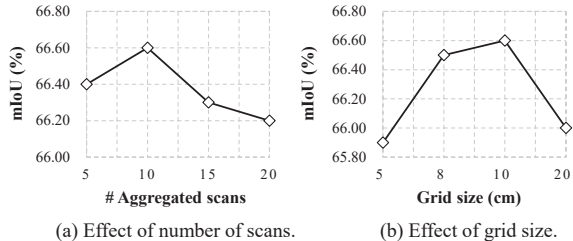
4.4. Further Analysis

Effect of frame numbers. In Fig. 4 (a), we delve into the effect of frame numbers, investigating the optimal length L of the sliding window used for temporal updates. This parameter determines the number of consecutive LiDAR frames that are combined to exploit temporal coherence in the scene as described in Sec. 3.3. Our analysis reveals that setting L to 10 frames achieves a desirable balance between capturing sufficient temporal context and avoiding excessive computational load or memory requirements. This optimal choice also enables the model to effectively leverage temporal dependencies while maintaining real-time performance and reducing potential noise introduced by distant past or future frames.

Effect of grid size resolution. As mentioned in Sec. 3.3, we convert the accumulated LiDAR scans into a voxel grid format with a fixed resolution. It is crucial to select an appropriate resolution because the fundamental assumption is that all points enclosed within a voxel belong to the same semantic category. Overestimating the voxel size can undermine this assumption, whereas selecting a resolution that is too fine can introduce noise into the estimates due to the inclusion of small-scale variations. To investigate the consequences of different voxel sizes, we perform an evaluation showcased in Fig. 4(b). The results clearly demonstrate that a voxel resolution of 0.10 meters yields the best semantic segmentation outcome. This finding underscores the significance of carefully tuning the grid size resolution to ensure that it neither oversimplifies nor overcomplicates the representation of the point cloud data, thereby preserving the integrity and accuracy of the semantic segmentation task.

Inference time comparison. We visualize popular methods' inference time and mIoU in Fig. 6. The results show that range-image-based methods are faster than point, polar, or hybrid methods. We measured the inference time of all the methods on the same hardware with a GeForce RTX 3090 GPU for a fair comparison.

Qualitative evaluation. The "many-to-one" issue becomes apparent in the absence of any post-processing technique, as depicted in Fig. 5(a). Here, we observe that points belonging to the tree trunk inadvertently adopt the predic-
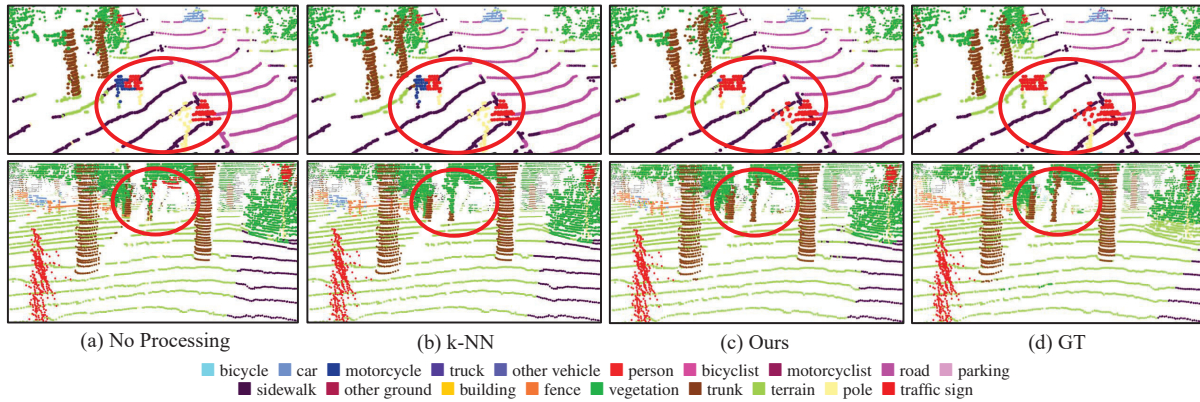
(a) No Processing	(b) k-NN	(c) Ours	(d) GT

■ bicycle ■ car ■ motorcycle ■ truck ■ other vehicle ■ person ■ bicyclist ■ motorcyclist ■ road ■ parking
■ sidewalk ■ other ground ■ building ■ fence ■ vegetation ■ trunk ■ terrain ■ pole ■ traffic sign

Figure 5. Qualitative analysis of the post-processing scheme. (a) The "many-to-one" issue is evident without post-processing, e.g., the trunk is partially segmented as traffic sign and vegetation as they project onto the same range pixel (row 2). (b) k-NN [32] smooths the semantic labels locally, but it cannot resolve ambiguities by objects that are close or prediction errors. (c) Our method exploits temporal information to resolve false predictions (row 1) or ambiguities due to occlusions (row 2). Best viewed in color.
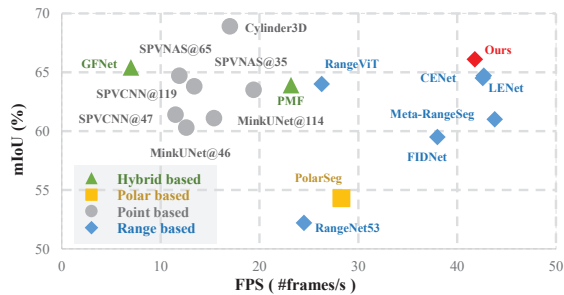
Figure 6. mIoU vs. runtime on SemanticKITTI. Our method balances mIoU and inference time better than other state-of-the-art methods. Best viewed in color.

tions intended for nearby points from the traffic sign and vegetation classes. This occurs because these points, despite being distinct physical entities, are projected onto the same range pixel in the LiDAR data. In Fig. 5(b), we illustrate the performance of the commonly-employed k-NN method [32]. While it does refine the initial predictions to some extent, it struggles to rectify false classifications when larger regions are occluded. This limitation highlights the inability of certain post-processing methods to handle complex scenarios where multiple points project to the same pixel. On the contrary, our proposed method effectively tackles this problem, as shown in Fig. 5(c). By incorporating temporal information across multiple scans, our approach consistently maintains the correct predictions for the tree trunk, even when the current scan is affected by the "many-to-one" issue. This capability showcases the merit of introducing temporal context in the post-processing phase, as it allows our method to discern and rectify errors

caused by occlusions and projection ambiguities in LiDAR data. Thus, our solution demonstrates improved robustness in handling the "many-to-one" problem, illustrating the potential gains achieved by leveraging temporal coherence in LiDAR semantic segmentation.

5. CONCLUSION

In this paper, we quantitatively and qualitatively analyzed the boundary blurriness, which is also called "many-to-one" problem, for range-image-based LiDAR segmentation, and introduced a novel solution named TFNet to tackle it. Our approach involves leveraging temporal information through the introduction of temporal fusion layers during the training process and a sequential max voting strategy during inference. The experiments on two benchmarks demonstrate the advantages of the proposed strategy. In particular, the incorporation of temporal data allows TFNet to maintain robust performance in environments with substantial occlusions, while still maintaining real-time performance. Additionally, we conducted comprehensive ablation studies to validate the design, as well as the broader adaptability of the proposed post-processing to other neural network architectures.

References

[1] Angelika Ando, Spyros Gidaris, Andrei Bursuc, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In *CVPR*, 2023. 3, 5

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019. 1, 2, 4

[3] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss. Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset. *The International Journal on Robotics Research*, 2021. 2

[4] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018. 5

[5] Alexey Bokhovkin and Evgeny Burnaev. Boundary loss for remote sensing imagery semantic segmentation. In *ISNN*, 2019. 5

[6] 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 *CVPR*, 2020. 1

[7] Jun Cen, Shiwei Zhang, Yixuan Pei, Kun Li, Hang Zheng, Maochun Luo, Yingya Zhang, and Qifeng Chen. Cmdfusion: Bidirectional fusion network with cross-modality knowledge distillation for lidar semantic segmentation. In *arXiv*, 2023. 2

[8] Qi Chen, Sourabh Vora, and Oscar Beijbom. Polarstream: Streaming lidar object detection and segmentation with polar pillars. In *NeurIPS*, 2021. 2

[9] Xieyuanli Chen, Andres Milioto Emanuele Palazzolo, Philippe Giguère, Jens Behley, and C. Stachniss. Suma++: Efficient lidar-based semantic slam. In *IROS*, 2019. 4

[10] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss. OverlapNet: Loop Closing for LiDAR-based SLAM. In *RSS*, 2020. 3

[11] X. Chen, T. Läbe, A. Milioto, T. Röhling, J. Behley, and C. Stachniss. OverlapNet: A Siamese Network for Computing LiDAR Scan Similarity with Applications to Loop Closing and Localization. In *Autonomous Robots*, 2021. 3

[12] Xieyuanli Chen, Shijie Li, Benedikt Mersch, Louis Wiesmann, Jürgen Gall, Jens Behley, and Cyrill Stachniss. Moving object segmentation in 3d lidar data: A learning-based approach exploiting sequential data. In *R-AL*, 2021. 2, 3

[13] Hui-Xian Cheng, Xian-Feng Han, and Guo-Qiang Xiao. Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving. In *ICME*, 2022. 1, 2, 5, 6, 7

[14] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 2

[15] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *ISVC*, 2020. 1, 6

[16] Ben Ding. Lenet: Lightweight and efficient lidar semantic segmentation using multi-scale convolution attention. In *arXiv*, 2023. 1, 5

[17] Peng Gao, Teli Ma, Hongsheng Li, Ziyi Lin, Jifeng Dai, and Yu Qiao. Mcmae: Masked convolution meets masked autoencoders. In *NeurIPS*, 2022. 2, 4

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[19] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *CVPR*, 2020. 1

[20] Deyvid Kochanov, Fatemeh Karimi Nejadasl, and Olaf Booij. Kprnet: Improving projection-based lidar semantic segmentation. In *arXiv*, 2020. 3, 5

[21] Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. In *ICCV*, 2023. 1, 2, 3, 5

[22] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. In *arXiv*, 2022. 2, 3, 4

[23] Shijie Li, Xieyuanli Chen, Yun Liu, Dengxin Dai, Cyrill Stachniss, and Juergen Gall. Multi-scale interaction for real-time lidar data segmentation on an embedded platform. In *R-AL*. IEEE, 2021. 5, 6

[24] Xiaoyan Li, Gang Zhang, Hongyu Pan, and Zhenhua Wang. Cpgnet: Cascade point-grid fusion network for real-time lidar semantic segmentation. In *ICRA*, 2022. 1, 2

[25] Jiahui Liu, Chirui Chang, Jianhui Liu, Xiaoyang Wu, Lan Ma, and Xiaojuan Qi. Mars3d: A plug-and-play motion-aware model for semantic segmentation on multi-scan 3d point clouds. In *CVPR*, 2023. 3, 6

[26] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. In *arXiv*, 2023. 3

[27] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *NeurIPS*, 2019. 1, 2, 6

[28] Xiaotong Luo, Yuan Xie, Yulun Zhang, Yanyun Qu, Cuihua Li, and Yun Fu. Latticenet: Towards lightweight image super-resolution with lattice block. In *ECCV*, 2020. 1

[29] Junyi Ma, Xieyuanli Chen, Jingyi Xu, and Guangming Xiong. Seqot: A spatial-temporal transformer network for place recognition using sequential lidar data. In *IEEE Transactions on Industrial Electronics*, 2022. 3

[30] Junyi Ma, Jun Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. Overlaptransformer: An efficient and yaw-angle-invariant transformer network for lidar-based place recognition. In *R-AL*, 2022. 3

[31] Teli Ma, Mengmeng Wang, Jimin Xiao, Huifeng Wu, and Yong Liu. Synchronize feature extracting and matching: A single branch framework for 3d object tracking. In *ICCV*, pages 9953–9963, 2023. 3

[32] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IROS*, 2019. 1, 2, 3, 5, 6, 7, 8

[33] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IV*, 2020. 2, 4, 6

[34] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 1, 2

[35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 1, 2

[36] Haibo Qiu, Baosheng Yu, and Dacheng Tao. GFNet: Geometric flow network for 3d point cloud semantic segmentation. In *Transactions on Machine Learning Research*, 2022. 1, 2

[37] Shi Qiu, Saeed Anwar, and Nick Barnes. Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In *CVPR*, 2021. 1

[38] Ryan Razani, Ran Cheng, Ehsan Taghavi, and Liu Bingbing. Lite-hdseg: Lidar semantic segmentation using lite harmonic dense convolutions. In *ICRA*, 2021. 5, 6

[39] Jiadai Sun, Yuchao Dai, Xianjing Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. Efficient spatial-temporal information fusion for lidar-based 3d moving object segmentation. In *IROS*, 2022. 2, 3, 6

[40] 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 *ICCV*, 2019. 1, 2

[41] Marc Uecker, Tobias Fleck, Marcel Pflugfelder, and J. Marius Zöllner. Analyzing deep learning representations of point clouds for real-time in-vehicle lidar perception. In *arXiv*, 2022. 1, 2

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3

[43] Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation. In *ICIP*, 2021. 6

[44] Mengmeng Wang, Teli Ma, Xingxing Zuo, Jiajun Lv, and Yong Liu. Correlation pyramid network for 3d single object tracking. In *CVPR*, 2023. 3

[45] Song Wang, Jianke Zhu, and Ruixiang Zhang. Metarangeseg: Lidar sequence semantic segmentation using multiple feature aggregation. In *R-AL*, 2022. 3, 5, 6, 7

[46] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*, 2018. 1, 2, 3, 5, 6

[47] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*, 2019. 5, 6

[48] Zhaoyang Xia, Youquan Liu, Xin Li, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, and Yu Qiao. Scpnet: Semantic scene completion on point cloud. In *CVPR*, 2023. 3

[49] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 3

[50] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient pointcloud segmentation. In *ECCV*, 2020. 6

[51] 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 *ECCV*, 2022. 2

[52] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *CVPR*, 2023. 6

[53] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *arXiv*, 2020. 1

[54] Yiming Zhao, Lin Bai, and Xinming Huang. Fidnet: Lidar point cloud semantic segmentation with fully interpolation decoding. In *IROS*, 2021. 1, 2, 3, 5, 6, 7

[55] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *CVPR*, 2021. 2

[56] Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In *ICCV*, 2021. 1, 2