A Multi-View-Assisted Semantic Segmentation Network on LiDAR via Multi-Level Mutual Learning Knowledge Distillation

1st Yun Zhang
School of Automation
Southeast University,
Nanjing, China
230208688@seu.edu.cn

2nd Kun Qian*
School of Automation
Southeast University,
Nanjing, China
kqian@seu.edu.cn

3rd Yixin Fang
School of Automation
Southeast University,
Nanjing, China
220211930@seu.edu.cn

4th Tong Shi
School of Automation
Southeast University,
Nanjing, China
shitong_2001@163.com

5th Hai Yu
State Grid smart Grid
Research Institute
Co.,LTD, Nanjing,
China
School of Automation,
Southeast University,
Nanjing, China
230209100@seu.edu.cn

2024 9th International Conference on Automation, Control and Robotics Engineering (CACRE) | 979-8-3503-5030-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/CACRE62362.2024.10635070

***Abstract*— LiDAR-based semantic segmentation is crucial in many robotic perception systems. Considering the data of LIDAR has various views with diverse spatial features, more and more semantic segmentation methods have been proposed to fuse them for better segmentation accuracy. However, compared to single-view segmentation, these multi-view methods that directly fuse all features face inferior real-time performance and higher computation costs. Therefore, this paper proposes a multi-view-assisted semantic segmentation network via multi-level mutual learning knowledge distillation (KD) to implement a high real-time and accurate semantic segmentation at a lower cost. The keys of the multi-level mutual learning-based KD strategy are intra-view mutual learning and inter-view mutual learning. To accelerate the computation and improve accuracy, we also introduce a two-step fusion strategy to fuse features hierarchically. Finally, we evaluated our approach on the SemanticKITTI dataset. The experimental results demonstrate that the proposed method competently improves efficiency and accuracy.**

Keywords*— *LiDAR, Semantic Segmentation, Multi-View fusion, multi-level and mutual learning Knowledge distillation.

I. INTRODUCTION

Semantic segmentation is one of the crucial techniques in autonomous robotic systems for fine-grained perception of the environment and achieving specific task objectives [1]. As a reliable and standard sensor in the field of environmental sensing, Light Detection and Ranging (LiDAR) can offer accurate spatial information over a wide range with robustness to illumination changes. As the core module of environment sensing technology, fast and accurate semantic segmentation is a prerequisite for precise robot localization, reliable path planning, and safe driving.

Recently, various LiDAR-based semantic segmentation approaches have emerged, mainly including range-based methods [2], voxel-based methods [3], and point-based methods [4]. Nevertheless, each view has drawbacks, such as the point view having good local geometric features but its computation being relatively more significant. In contrast, the voxel view reduces computation through voxelization. However, the local details are lost, and the range view incurs low segmentation accuracy due to projection errors, even though features can be extracted fast through 2D convolution. To surmount the issues introduced above, the strategy of multi-view fusion has been proposed, which leverages the complementary advantages of multiple views and achieves more accurate segmentation.

However, making use of more views also brings several drawbacks. Firstly, for multi-view semantic segmentation on multi-channel LiDAR, such as HDL-64E, it is necessary to trade off the segmentation accuracy and speed. Fusing more views can improve the segmentation accuracy. However, it will also incur higher computation costs, reduce the training and inference speed, and may cause the network to be unable to apply for some time-sensitive scenarios. Secondly, since the multi-view data originates from the same sensor, they contain a large amount of redundant information, and the redundant information in each view branch not only reduces the real-time performance but also may cause the network to over-fit since this redundant data has been repeated many times in views. Thirdly, each view has its drawbacks, such as a lack of local details, and projection errors. The respective branch networks trained based on these data will be affected by these problems, resulting in limited semantic segmentation performance in each branch. To overcome these issues, specific methods focus on exploiting the potential of LIDAR data by fully utilizing multi-view data and compressing redundancy features by gated-based fusion, such as [5]. However, these methods typically assign an individual encoder-decoder network for each view and fuse multi-view features directly. Therefore, to achieve high real-time and high-accuracy semantic segmentation on LiDAR, exploring an efficient multiple-views fusion method that improves accuracy, reduces redundant features, and simplifies the network is necessary.

This paper proposes a multi-view-assisted semantic segmentation network via multi-level mutual learning

This work was supported by the National Natural Science Foundation of China (No.61573101) and the Science and Technology Project of State Grid Corporation (5700-202318305A-1-1-ZN)

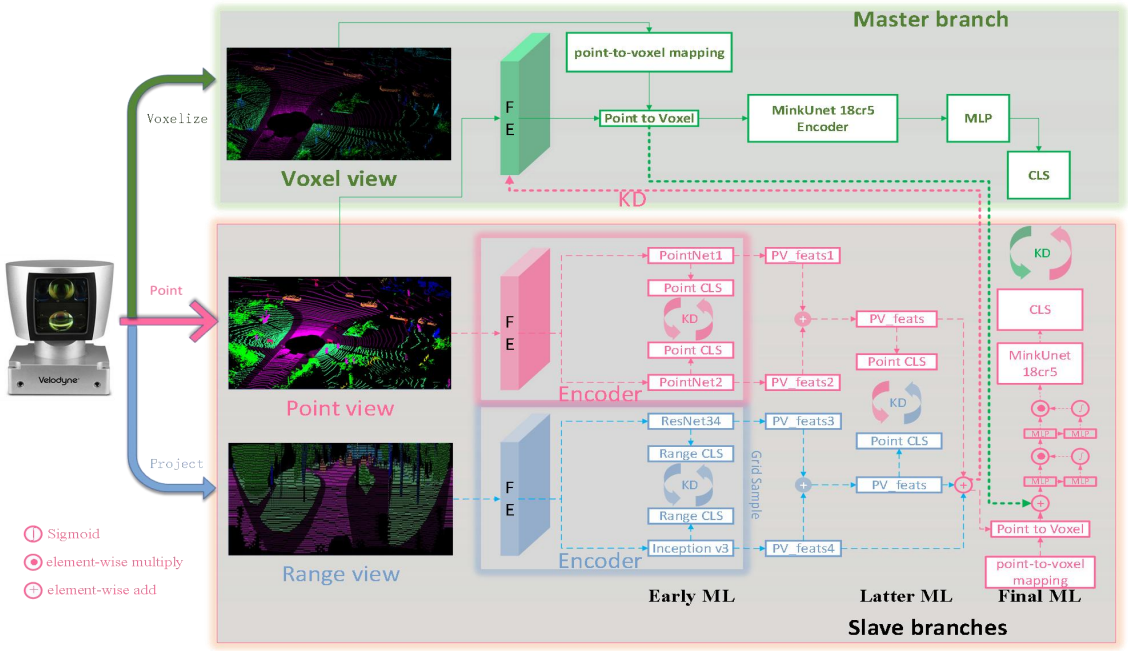

Fig. 1. The framework of the proposed method. It consists of a master branch (voxel view) and two slave branches (point view and range view). Each branch has an individual FE head and multiple encoders to extract features.

knowledge distillation. Due to the introduction of mutual learning, all branches play the roles of teacher and student. Therefore, the traditional training-inference branch and the training-only branch are regarded as master branch and slave branch. By trading off accuracy and real-time performance of the inference phase, we choose the voxel view as the master branch and select the point view and the range view as slave branches. The respective FE head first extracts all view features from view data. To achieve efficient fusion multi-view features, we propose the multi-level- and mutual-learning-based KD strategy, the keys of which are early intra-view mutual learning (Early ML) in the first encode phase and inter-view mutual learning (Latter ML) in the latter encoder phase for slave branches. The Early ML is implemented to enhance a single slave branch by combining the feature extract ability of different structure/parameter networks, and the Latter ML is used to achieve mutual boosting between slave branches by fusing spatial information of different views. To accelerate the computation and improve accuracy, we also import a two-step fusion strategy to fuse features hierarchically. The first-step fusion aggregates all slave branches' features, and the second further fuses the slave and the master branches while weighting the fusion features to suppress the redundant portion. This strategy not only comes true as a flexible fusion but also implements a modular design for view extension.

II. METHOD

Fig. 1 illustrates the detailed structure of the proposed network. The voxel view is depicted in green, and the slave branches are shown in blue (range view) and red (point view). The master branch is a baseline for training and inference, indicated by solid lines. Other branches only used for training are represented by dotted lines.

A. *Structures of Master Branch*

As shown in Fig. 1, for the point cloud $P = (x_i \quad y_i \quad z_i) \in \mathbb{R}^3$, we obtain a point-to-voxel mapping in the l-th layer through

$$D_l^{voxel} = (\lfloor x_i/r_l \rfloor \quad \lfloor y_i/r_l \rfloor \quad \lfloor z_i/r_l \rfloor)_i^N \in \mathbb{R}^{\mathrm{Nx3}} \tag{1}$$

where r_l is the voxelization resolution in the l-th layer and $\lfloor \cdot \rfloor$ is the floor operation. Point-to-voxel mapping is used to transform the point view into a voxel view. Then, a hash function is applied to calculate the unique hash value for each voxel and obtain a hash index, which is used for transforming the l-th layer voxel to the 0-th layer voxel. In this paper, we first use the MLP-based FE head to extract features from points. The whole slave branches can enhance the FE head through KD. The point-view features extracted from the FE head are converted to voxel view by the point-to-voxel mapping index, which is obtained by (1). Then, voxel features are processed by the encoder of MinkUNet-18cr5 and the MLP-based decoder; this variant has been implemented in the OpenPCSeg codebase [6].

B. *Structures of Slave Branches*

Due to the slave branches consisting of different view data, an appropriate encoder is crucial for efficiently extracting features. As shown in Fig. 1, there are two views: the point view obtained from LiDAR directly and the range view acquired by utilizing the projection formula (2) to project points to an image plane.

$$(u \quad v)^T = \left\{\frac{1}{2}\left[1 - \frac{arctan(y,x)}{\pi}\right] W_r, \left[1 - \frac{arcsin(zr^{-1}) + f_{up}}{f}\right] H_r\right\}^T, \tag{2}$$

To enhance each slave encoder in the early encoding phase, we extract features for each view by utilizing multiple encoders, the network structure of which can be different or

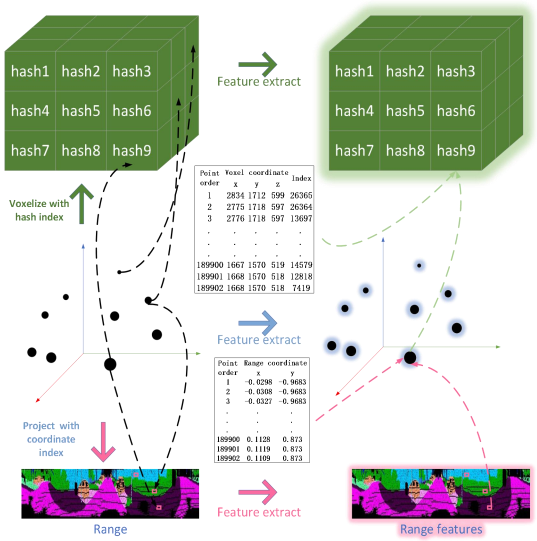

Fig. 2. The voxel view and range view are generated from the point view by voxelizing and projecting. To distill knowledge in an uniform view, the range features are transformed to point view through grid sampling and then transformed to voxel view by point-voxel mapping.

the same. The principle of this enhancement method is that the classification results are related to different features extracting ability [7] of different networks and different initial network parameters [8][9]. Therefore, we adopt mutual learning to enhance slave encoders in the early encoding phase. In this paper, we adopt two PointNet networks with different initial parameters for point view and adopt a deep network (ResNet34) and a wide network (Inception V3) for range view. By the way, considering that more encoders will lead to larger and larger parameters, we could use only one encoder for a view if we want faster training. Before distilling the knowledge from slave branches to the master branch, we transform range features from range view to point view for the first step by grid sample shown in Fig. 2. To make full use of knowledge of different branches, another mutual learning is implemented between the sum of range features that have been grid sampled and the sum of point features. Then, the range features and the point features are added as first-step fusion features, which play a teacher role in distilling beneficial knowledge to the FE head of the master; in other words, the entire encoders in slave branches before the first fusion are regarded as a large FE head for the master branch. Then, the first-step fusion features under point-view are voxelized by point-to-voxel mapping, which is added with voxel features of the master branch as second-step fusion features. We use MLP to generate the weight coefficient to compress the redundant features. Finally, the weighted second-step fusion voxel features are further encoded and decoded by MinkUNet-34cr10. We implement final mutual learning between the classification results of the master branch and the slave branch.

III. EXPERIMENTS

A. Experimental Setup

- Dataset

The SemanticKITTI dataset[10] consists of 43,551 LiDAR scans from 22 Velodyne HDL-64E LiDAR sequences, each containing approximately 130,000 points. These 22 sequences are categorized into three groups: the training set (sequences 00 to 10), the validation set (sequence 08), and the test set (sequences 11 to 21).

- Evaluation Metrics

mIoU (mean intersection-over-union) is a common evaluation metric to illustrate performance, which can be defined as:

$$mIoU = \frac{1}{C}\sum_{c}^{C}\frac{TP_c}{TP_c+FP_c+TN_c} \quad (3)$$

where TP_c, FP_c and TN_c represent true positive, false positive, and false negative predictions for the given class c, respectively, and C is the number of classes.

- Implementation Details

In this paper, the master branch adopts the baseline of Minkowski-UNet18-CR0.5. The range view slave branch employs the encoders of ResNet34 and Inception v3. The point-view slave branch directly employs two PointNet networks with different initial parameters. We only adopt common data augment methods: flip, scaling, rotation and transform. The voxel size is 0.5m, and the range size is [64, 2048]. We apply cross-entropy loss and Lovasz [11] loss for semantic segmentation and KL divergence loss for distillation. The batch sizes, learning rate, and epochs are 2, 0.16, and 64. Models were trained using SGD optimizer and CosineAnnealingWarmRestarts learning rate scheduler in an

TABLE I
CLASS-WISE AND MEAN IOU OF OUR PROPOSED METHOD AND SOME OTHER METHODS ON THE SEMANTICKITTI.

method	mIou(%)	speed(Fps)	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic sign
PointNet++ [8]	20.1	0.34	53.7	1.9	0.2	0.9	0.2	0.9	1	0	72	18.7	41.8	5.6	62.3	16.9	46.5	13.8	30	6	8.9
RangeNet53++ [3]	52.2	24	91.4	25.7	34.4	25.7	23	38.3	38.8	4.8	91.8	65	75.2	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9
PolarNet [9]	54.3	31.7	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90	61.3	84	65.5	67.8	51.8	57.5
SqueezeSegV3 [32]	55.9	8.1	92.5	38.7	36.5	29.6	33	45.6	46.2	20.1	91.7	63.4	74.8	26.4	89	59.4	82	58.7	65.4	49.6	58.9
RandLA-Net [10]	55.9	2.2	94.2	29.8	32.2	43.9	39.1	48.4	47.4	9.4	90.5	61.8	74	24.5	89.7	60.4	83.8	63.6	68.6	51	50.7
GFNet [17]	65.4	12.9	96	53.2	48.3	31.7	47.3	62.8	57.3	44.7	93.6	72.5	80.8	31.2	94	73.9	85.2	71.1	69.3	61.8	68
JS3C-Net [18]	66	4.3	95.8	59.3	52.9	54.3	46	69.5	65.4	39.9	88.9	61.9	72.1	31.9	92.5	70.8	84.5	69.8	67.9	60.7	68.7
SPVCNN [25]	59.31	20.5	93.5	11.1	53.9	80	26.8	63.9	88.1	0	93.6	51.1	80.3	0.3	90.4	56.3	88.5	64.9	76	62	46.1
MinkowskiNet[36]	59.35	22.5	94.8	14	38.7	85.3	40.5	60.8	89.7	0	93.2	51.3	80	0.1	90.2	55.6	88.2	64	75.3	60.3	45.6
RPVNet [15]	**70.3**	11.8	**97.6**	**68.4**	**68.7**	44.2	61.1	75.9	74.4	43.4	**93.4**	**70.3**	**80.7**	**33.3**	**93.5**	**72.1**	**86.5**	**75.1**	**71.7**	**64.8**	61.4
RVI-M2KD(ours)	70.21	**25.13**	96.5	61.9	65.7	**51.3**	**63.3**	**79.1**	**79.4**	**50**	89.7	67.7	74.7	31.4	91.8	67.3	86.1	74.9	70.9	63.7	**68.6**

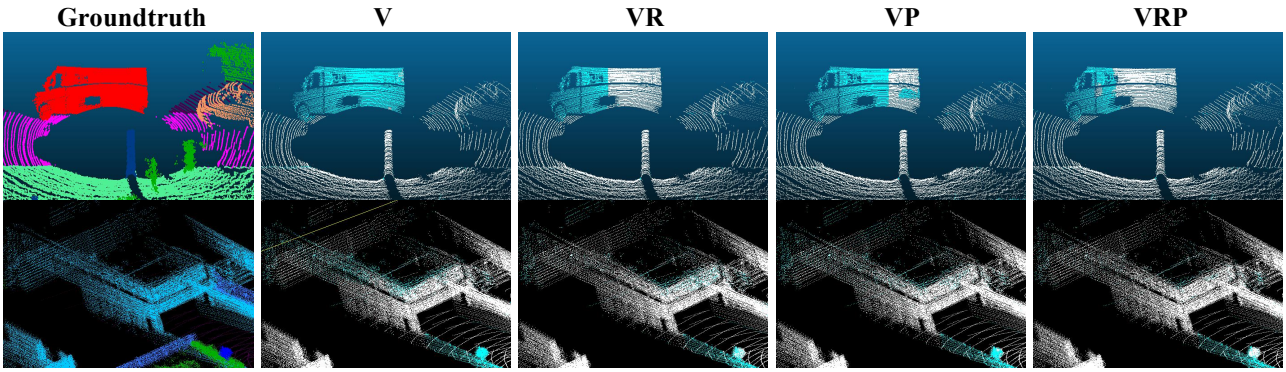

Fig. 3. Segmentation result. From left to right: ground truth, V(baseline), VR, VP, VRP segmentation results. Cyan and white points represent incorrectly and correctly classified points, respectively.

end-to-end manner on NVIDIA RTX 4090 GPU.

B. *Results on SemanticKITTI*

In this experiment, we compare the results of the proposed method with the existing LiDAR segmentation methods. The order of these methods in Table I is sorted by mIoU results. Table I shows that the mIoU result of our method reached 70.2%, which outperforms other methods except RPVNet. Although our method and RPVNet utilize point, voxel, and range views, our method is still 0.09% lower than RPVNet. We believe the reason why the mIoU is lower is that the RPVNet directly fuses point, voxel, and range view, which obtains all information of all views. However, we only adopt the voxel branch for inference and adopt KD to transform beneficial knowledge to this branch during training. Inevitably, information is lost during this KD process, resulting in a lower accuracy. But at the same time, thanks to KD, our proposed method obtains better real-time performance; the FPS of our method reaches 25.13, which is much higher than 11.8 of RPVNet. Therefore, our approach dramatically improves the real-time performance without much loss of accuracy. Considering both segmentation accuracy and real-time performance, our method is more competitive.

C. *Ablation Studies*

We conducted extensive comparative experiments, including voxel-view (V) segmentation, voxel-range-view (VR) fusion segmentation, voxel-point-view (VP) fusion segmentation, and voxel-point-range-view (VRP) fusion segmentation. It's important to note that we trained the networks with only 16 epochs to speed up the process. The mIoU was evaluated on the SemanticKITTI validation set (sequence 08). In Fig. 3, we present segmentation results for individual objects across different views. From left to right, these pictures are ground truth, V-based, VP-based, VR-based, and VPR-based methods. In the ground truth image, different colors illustrate different classifications, and cyan and white represent incorrect and correct reasons in the remaining pictures. From the Fig. 3, it can be seen that the segmentation accuracy gradually increases as the number of views increases. In addition, VR accuracy is better than VP, similar to the fact that range-based RangeNet++ accuracy is higher than PointNet-based PointNet++.

From the results in Table II, we can also observe that compared with the baseline (V), the fusion-distillation of VR, VP, and VRP is effective and can improve accuracy to varying degrees. The improvement depends on different combinations of views. The mIoU of the VR-based method is 65.6%, which is higher than the mIoU of VP (65.2%). For the VIR mode, it can reach the highest 66.5% mIoU. From this table, we can also find that the accuracy depends on different FE heads. For the VR and VRP fusion, we replace the MLP head of range with a proximity convolution[12]. The proximity convolution exploits range information to augment the spatial sampling locations and effectively improve the transformation modeling ability. It can be observed that employing proximity convolution results in a noticeable improvement in accuracy.

TABLE II
MEAN IOU OF DIFFERENT VIEWS, FE HEADS OF RANGE MODULE

V (FE:MLP)	P (FE:MLP)	R (FE:MLP)	FE of R (FE: proximity conv)	mIou (%)
√				63.5
√	√			64.6
√		√		65.2
√			√	65.6
√	√	√		66.0
√	√		√	66.5

TABLE III
PARAMETERS, TRAINING TIME, AND INFERENCE TIME OF DIFFERENT VIEWS.

mode	Params (M)	Training_Speed (it/s)	Inference_Speed (it/s)	mIou (%)
V	2.5	12.26	25.13	63.5
VP_Half	28.5	4.69	10.37	64.1
VP	28.6	4.25	9.61	64.6
VR_Inception	44.7	3.78	9.38	65.0
VR_ResNet	50.9	4.34	10.11	65.2
VR	67.3	3.35	8.6	65.6
VRP	67.4	2.84	7.41	66.5

We also compared the numbers of parameters, training speed, and inference speed for different view assembly modes in Table III. These modes are trained and inference separately. For the convenience of comparison, the inference phase in this table will contain all branches, unlike the actual case where only the voxel branch is available during inference. The mIoU is still obtained from all modes except the VRP. Here, we adopted the encoder of MinkUnet in the OpenPCSeg codebase as baseline (V). From Table III, it can be observed that when adopting the baseline alone, the number of parameters is only

TABLE IV
EFFECTIVENESS OF MUTUAL LEARNING UNDER DIFFERENT STAGES

Method	Early ML	Latter ML	Final ML	mIou (%)
VP	×	—	—	64.32
	√	—	—	64.6
VR	×	—	—	65.41
	√	—	—	65.6
VRP	×	×	×	66.15
	√	×	×	66.31
	×	√	×	66.26
	√	√	×	66.35
	√	√	√	66.5

2.5M. Due to the multi-view fusion of VR, VP, and VRP modes, their parameters increase gradually. From Table III, we can also find the VP mode with two same encoders has better mIoU than the VP_Half mode, which only has one encoder. The VR_ResNet and VR_Inception are VR-based modes that utilize the encoder of ResNet and Inception, respectively. Their mIoU illustrates that fusing two encoders with different structures can improve accuracy. In this table, all multi-view modes' training and inference speeds are vastly less than those of the voxel-view mode. TABLE II and TABLE III demonstrate that the proposed method combines the least parameters of 2.5M (the fastest inference speed of 25.13 FPS) and the best mIoU of 66.5.

Finally, we validate the effectiveness of mutual learning under different stages in TABLE IV. We only test the early mutual learning method (Early ML) for VP and VR modes. For VRP, we test all mutual learning stages. From FRP mode in this table, compared to the method without KD, it can be found that all ML stages can improve accuracy. The accuracy improvement of sequentially enabling Early ML, Late ML, and Final ML was 0.16%, 0.11%, and 0.15%, respectively,

IV. CONCLUSION

This paper introduces a multi-view-assisted semantic segmentation network via multi-level mutual learning knowledge distillation to implement a high real-time and accurate environmental perception at a lower cost. The innovations of this paper include multi-level mutual learning and two-step hierarchical fusion strategies. The experiments on SemanticKITTI illustrate the validation of the combination of these two strategies. Compared with other single-view or multi-view methods, our method trades off real-time performance and accuracy, maintaining high accuracy while achieving high real-time performance.

REFERENCES

[1] X. Liu, Y. Zhang and D. Shan, "Unseen Object Few-Shot Semantic Segmentation for Robotic Grasping," in IEEE Robotics and Automation Letters, vol. 8, no. 1, pp. 320-327, Jan. 2023.

[2] A. Milioto, I. Vizzo, J. Behley and C. Stachniss, "RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019, pp. 4213-4220.

[3] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In CVPR, pages 9224–9232, 2018. 1, 3

[4] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 77-85.

[5] Xu, Jianyun, et al. "RPVNet: A Deep and Efficient Range-Point-Voxel Fusion Network for LiDAR Point Cloud Segmentation." 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021): 16004-16013.

[6] Youquan Liu，Runnan Chen, et al.，"UniSeg: A Unified Multi-Modal LiDAR Segmentation Network and the OpenPCSeg Codebase," 2023, ArXiv: 2309.05573

[7] Z. Li, Y. Ming, L. Yang, J.-H. Xue, Mutual-learning sequence-level knowledge distillation for automatic speech recognition, Neurocomputing 428 (2021) 259–267.

[8] Y. Zhang, T. Xiang, T.M. Hospedales, H. Lu, Deep mutual learning, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4320–4328.

[9] Li, C., Li, G., Zhang, H. et al. Embedded mutual learning: A novel online distillation method integrating diverse knowledge sources. Appl Intell 53, 11524–11537 (2023). https://doi.org/10.1007/s10489-022-03974-7

[10] J. Behley et al., "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 9296-9306.

[11] M. Berman, A. R. Triki and M. B. Blaschko, "The Lovasz-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 4413-4421.

[12] K. Sirohi, R. Mohan, D. Büscher, W. Burgard and A. Valada, "EfficientLPS: Efficient LiDAR Panoptic Segmentation," in IEEE Transactions on Robotics, vol. 38, no. 3, pp. 1894-1914, June 2022.