TRANSUPR: A TRANSFORMER-BASED UNCERTAIN POINT REFINER FOR LIDAR POINT CLOUD SEMANTIC SEGMENTATION

Zifan Yu[⋆] *Meida Chen*[†] *Zhikang Zhang* [⋆] *Suya You*[‡] *Fengbo Ren*[⋆]

[⋆] Arizona State University, Tempe, AZ, [‡]US Army Research Laboratory, Los Angeles, CA
[†]USC Institute for Creatives Technologies, Los Angeles, CA

ABSTRACT

In this work, we target the problem of uncertain points refinement for image-based LiDAR point cloud semantic segmentation (LiDAR PCSS). This problem mainly results from the boundary-blurring problem of convolution neural networks (CNNs) and quantitation loss of spherical projection, which are often hard to avoid for common image-based LiDAR PCSS approaches. We propose a plug-and-play transformer-based uncertain point refiner (TransUPR) to address the problem. Through local feature aggregation, uncertain point localization, and self-attention-based transformer design, TransUPR, integrated into an existing range image-based LiDAR PCSS approach (e.g., CENet), achieves the state-of-the-art performance (68.2% mIoU) on Semantic-KITTI benchmark, which provides a performance improvement of 0.6% on the mIoU. The code is available at `https://figshare.com/s/3db86ab1d05fe6f3afab` for review.

Index Terms— LiDAR point cloud segmentation, convolutional neural networks, deep learning

1. INTRODUCTION

Range image-based convolution neural networks (CNNs) [1, 2, 3, 4, 5] are a critical alternative to point-based [6, 7, 8] and partition-based methods [9, 10, 11] for LiDAR point cloud semantic segmentation (LiDAR PCSS) when real-time inference is needed for target applications. [5, 12] achieve higher segmentation accuracy by applying domain-specific network architecture modifications and loss functions to CNNs borrowed from other 2D vision tasks. [13, 14] alleviate the boundary-blurring problem of CNNs by adding a boundary loss. The GPU-accelerated KNN refiner [2, 5, 12, 13] is a common post-processing technique of image-based methods to reclassify points blocked by foreground points during spherical projection, i.e., quantitation loss of spherical projection. However, image-based methods still fall behind state-of-the-art partition-based [10] and hybrid methods [15] in terms of segmentation accuracy, and the widely-used KNN refiner may incorrectly rectify points (see Fig. 1).

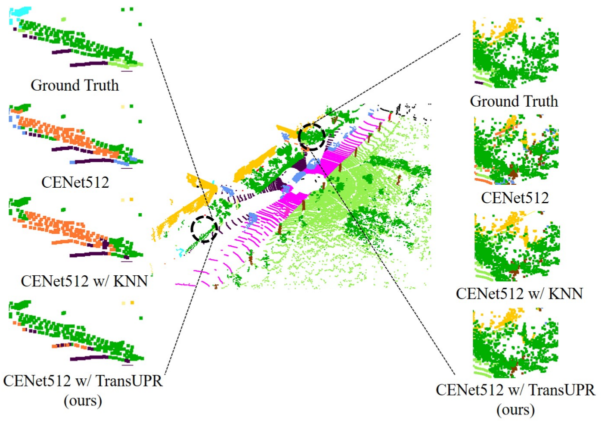

Fig. 1: The KNN refiner may rectify some points that are correctly predicted by CNN backbones.

In this paper, we propose a plug-and-play transformer-based refiner for image-based LiDAR PCSS to improve the segmentation accuracy by reclassifying uncertain points that are misclassified due to the boundary-blurring problem of CNNs and quantitation loss of spherical projection. Our pipeline combines the image-based CNNs with the point-based methods sequentially to benefit from each method efficiently via semantic-augmented point representation. As shown in Fig 2, the image-based CNNs first generate coarse semantic segmentation features. Following that, the uncertain points are selected in two ways: (i) choosing the points with lower probability of coarse semantic segmentation labels [16]. (ii) selecting background points, i.e., points blocked by foreground points during spherical projection and are far away from foreground points. We then use the geometry (e.g., coordinates) along with the locally augmented semantic segmentation features to train our designed learnable transformer-based refiner, which contains concatenated self-attention layer and fully connected layers for uncertain point classification.

The contributions of this paper are as follows: 1) We propose a plug-and-play transformer-based uncertain point refiner for image-based LiDAR PCSS, which is generic and can be easily integrated into existing image-based CNN backbones without any network architecture modifications. 2) the existing image-based LiDAR PCSS can be further improved using our TransUPR and achieves new state-of-the-art performance, i.e., 68.2% mIoU, on Semantic-KITTI[17].

2. METHODOLOGY

As shown in Fig. 2, for a LiDAR point cloud $P = \mathbb{R}^{N\times 3}$, we first convert each point $p_i = (x, y, z, remission)$ of P into a 2D representation R, i.e., range image, by spherical projection. Each pixel in R contains five channels $(x, y, z, r, remission)$, which only presents the geometry feature of the foreground point that has the closest range to the origin. We then utilize existing range image segmentation CNNs as our backbone that takes R as the input to generate coarse semantic segmentation features. Following that, our TransUPR module selects the uncertain points and locally aggregates the point features for refinement, and produces point-wise fine-grained semantic labels. We explain the mechanism of our proposed TransUPR module from the following three aspects: 1) the local feature aggregation of geometry features and coarse semantic segmentation features; (2) the strategies of selecting uncertain points for training and testing; (3) the architecture of the learnable transform-based network and self-attention layers.

2.1. Local Feature Aggregation

We hypothesize that the output of the softmax layer in CNN backbones is the pixel-wise class probabilities, i.e., coarse semantic segmentation features [16]. The points projected into the same 2D pixel have duplicated coarse semantic segmentation features. Thus, we augment the features of each point by averaging the features of its neighbor points in the 3D space to make the features distinguishable. Our implementation is similar to the original KNN [2], which means GPU can accelerate the process without adding harsh computation burdens. At the same time, the local feature aggregation module have a identical KNN module with [2] to generate refined p_c (see Fig. 2).

In addition, we concatenate the point geometry attributes $(x, y, z, range, remission)$ and the augmented coarse semantic features as the input feature vectors for the following transformer. Therefore, a finalized feature vector of a 3D points have a size of $N_{geometry} + N_{classes}$, where $N_{geometry} = 5$ and $N_{classes} = 20$ in Semantic-KITTI [17].

2.2. Strategies of Selecting Uncertain Points

Applying the typical transformer [18] to the large-scale LiDAR point cloud severely burdens the GPU memory footprint [19]. However, we think the image-based methods are enough for comparable semantic segmentation labels, and only a small amount of points are misclassified due to the boundary-blurring problem and quantitation loss of projection. Thus, we localize the uncertain points to reduce the input size for the following transformer network.

One strategy is to identify uncertain points in 2D representation, where points are misclassified due to the boundary-blurring problem of CNNs. First, we rank the 2D pixels in ascending order of top-2 class probability differences, where the class probability is the output of the Softmax layer in CNNs. The 3D points projected into the same 2D pixel have identical top-2 class probability differences. Lower top-2 class probability differences imply that the assigned semantic labels are low confidence. Then, we randomly select the top 8192 points with the lowest top-2 class probability differences into the uncertain point pool.

Another uncertain point localization way is viewing all the background points of spherical projection as the uncertain points in 3D space. However, we noticed that most of the background points close to the foreground points belong to the same semantic label as the foreground points. Therefore, we set a cutoff c_u to filter out the background points that are close to the foreground points as the uncertain points. If the distance between a background point and its corresponding foreground points is less than the c_u, i.e., $c_u = 1$ for Semantic-KITTI[17], we put the background point into the uncertain point pool. We randomly select N_u points from the uncertain point pool for the refiner training, where $N_u = 4096$ for Semantic-KITTI. All points in the uncertain point pool are refined during the model inference via the transformer network.

2.3. Transformer Network and Attention Layers

Given an input uncertain point $p_u \in \mathbb{R}^{N_{geometry}+N_{classes}}$, a d_e-dimensional feature embedding $F_e \in \mathbb{R}^{d_e}$ is first learned via two fully connected layers, where $d_e = 256$. We modify the typical transformer [18], i.e., naïve point cloud transformer [19], with four self-attention layers, where each uncertain point is viewed as a word. As shown in figure 2, the inputs of each self-attention layer are the output of the precedent self-attention layer, and the outputs of each self-attention layer are concatenated together for the following uncertain point classification. Let $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ be the ***query***, ***key***, and ***value***. For each attention layer, the $\mathbf{Q}$ and $\mathbf{K}$ are first learned by a shared fully connected layer W_p, and the $\mathbf{V}$ is learned via another fully connected layer W_v,

$$\begin{aligned}(\mathbf{Q}, \mathbf{K}) &= W_p(F_{in}) \\ \mathbf{V} &= W_v(F_{in}),\end{aligned} \tag{1}$$

where F_{in} is the input feature embedding. The $\mathbf{Q}$ and $\mathbf{K}$ are utilized to calculate the correlation, i.e., attention weights, by the matrix dot-product, and the attention weights are normalized,

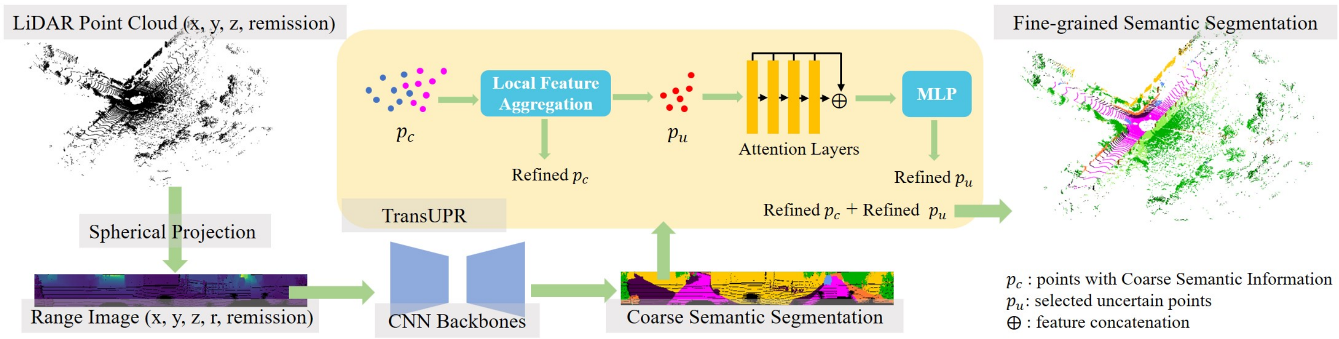
LiDAR Point Cloud (x, y, z, remission)
Local Feature Aggregation
p_c
Refined p_c
p_u
Attention Layers
MLP
Refined p_u
Fine-grained Semantic Segmentation
Spherical Projection
TransUPR
Refined p_c + Refined p_u
Range Image (x, y, z, r, remission)
CNN Backbones
Coarse Semantic Segmentation
p_c : points with Coarse Semantic Information
p_u: selected uncertain points
⊕ : feature concatenation

Fig. 2: The Pipeline of TransUPR.

$$\bar{a}_{i,j} = \frac{\tilde{a}_{i,j}}{\sqrt{d}} = \frac{\mathbf{Q} \cdot \mathbf{K}^{\mathbf{T}}}{\sqrt{d}}, \tag{2}$$

where d is the dimension of the learned feature embedding of the $\mathbf{Q}$ and $\mathbf{K}$. The output feature of a attention layer is the sum of the $\mathbf{V}$ using the attention weights,

$$F_{out} = \mathbf{A} \cdot \mathbf{V} = softmax(\bar{a}_{i,j}) \cdot \mathbf{V} = \frac{\exp \bar{a}_{i,j}}{\sum_k \exp \bar{a}_{i,k}} \cdot \mathbf{V}. \tag{3}$$

The dimension of the output features is 256 for all four self-attention layers. The concatenated features are fed into the following three fully-connected layers for point classification. The loss function of our TransUPR is the sum of a softmax cross-entropy loss and a *Lovász-Softmax* [20] loss,

$$L = L_{wce}(y, \hat{y}) + L_{ls}(C), \tag{4}$$

where y is the ground truth label of an uncertain point, and $\hat{y}$ is the predicted label. C is the number of semantic classes for model training. After reclassifying the semantic label of selected uncertain points, we replace the label of uncertain points in refined p_c (see Fig. 2) with the label of refined p_u.

3. EXPERIMENTS

3.1. Dataset and Experiment Settings

We evaluate the performance of our proposed TransUPR on the large-scale challenging Semantic-KITTI dataset[17] that contains 22 sequences of different scenarios. Over 19k scans (sequences between 00 and 10 except for 08) are used for training, and sequence 08 is used as a validation set. Note that we do not use sequence 08 for model fine-tuning to achieve higher performance on the benchmark. We combine our TransUPR with existing state-of-the-art image-based methods to demonstrate the efficiency and generality of our proposed uncertain point refiner. For the existing state-of-the-art methods, we utilize the available pre-trained models without fine-tuned by validation set and freeze the model weights when we train our refiner. We resubmit the predictions of the baseline methods and methods with our TransUPR to the Semantic KITTI benchmark for comparison.

We train our TransUPR for 50 epochs regardless of the CNN backbones. We quantitatively evaluate the results of the methods involved via mean intersection-over-union (mIoU), which can be formulated as $mIoU = \frac{1}{C}\sum_{c=1}^{C} \frac{TP_c}{FN_c+FP_c+TP_c}$, where TP_c is the number of the true positive cases of the c^{th} class, FN_c is the number of the false negative cases of the c^{th} class, and FP_c is the number of the false positive cases of the c^{th} class. The minor evaluation metric overall accuracy (oACC) is defined as $oACC = \frac{TP}{TP+FP}$.

3.2. Experiment Results

As results shown in Table 1, existing image-based PCSS methods can achieve obvious improvements of mIoU and oACC without any architecture modifications via our proposed TransUPR. By combining CENet-512 [13] with our TransUPR, we achieve the state-of-the-art performance, i.e., 68.2% mIoU, for image-based methods, which is +0.6% in mIoU and +0.4% in oACC over the original state-of-the-art CENet-512[13]. For the detailed class-wise IoU, our method improves the IoU in most of the classes except for *motorcyclist* and *traffic-sign*. At the same time, the CENet-512 with our TransUPR still can achieve about 19 FPS on a single Nvidia RTX 3090, which is much faster than the scanning frequency, i.e., 10Hz, of LiDAR sensors. Fig. 3 shows the visualization results of CENet512 with our TransUPR on the validation sequence 08.

3.3. Ablation Study

Other than the geometry features discussed in 2.1, FIDNet [12] calculates a normal vector for each point and concatenates the five-channel range image and normal vectors as inputs for CNN backbones. Thus, we explore the impact of normal vectors for training our TransUPR, and the quantitative results are shown in Table 2. The normal vectors for

Methods	*car*	*bicycle*	*motorcycle*	*truck*	*other-vehicle*	*person*	*bicyclist*	*motorcyclist*	*road*	*parking*	*sidewalk*	*other-ground*	*building*	*fence*	*vegetation*	*trunk*	*terrain*	*pole*	*traffic-sign*	***oACC***	***mIoU***
SqueezeSeg[1]	68.8	16.0	4.1	3.3	3.6	12.9	13.1	0.9	85.4	26.9	54.3	4.5	57.4	29.0	60.0	24.3	53.7	17.5	24.5	-	29.5
SqueezeSeg-CRF[1]	68.3	18.1	5.1	4.1	4.8	16.5	17.3	1.2	84.9	28.4	54.7	4.6	61.5	29.2	59.6	25.5	54.7	11.2	36.3	-	30.8
SequeezeSegV2 [3]	81.8	18.5	17.9	13.4	14.0	20.1	25.1	3.9	88.6	45.8	67.6	17.7	73.7	41.1	71.8	35.8	60.2	20.2	36.3	-	39.7
SequeezeSegV2-CRF [3]	82.7	21.0	22.6	14.5	15.9	20.2	24.3	2.9	88.5	42.4	65.5	18.7	73.8	41.0	68.5	36.9	58.9	12.9	41.0	-	39.6
SequeezeSegV3 [4]	92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9	-	55.9
RangeNet++ [2]	91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	91.8	65.0	75.2	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9	-	52.2
SalsaNext [5]	91.9	46.0	36.5	31.6	31.9	60.9	61.7	19.4	91.9	63.6	**76.1**	28.9	89.8	63.1	82.1	63.5	66.4	54.0	61.4	90.0	59.0
FIDNet + KNN [12]	92.3	45.6	41.9	28.0	32.6	62.1	56.8	30.6	**90.9**	58.4	74.9	20.5	88.6	59.1	**83.1**	64.6	67.8	53.1	60.1	89.6	58.5
CENet512[13]	93.0	51.2	67.2	58.4	62.4	69.1	70.0	**54.3**	90.3	70.0	74.5	40.8	88.6	61.3	81.6	63.5	69.2	55.5	**63.1**	89.9	67.6
SalsaNext [5] + **TransUPR**	92.4	45.3	37.0	30.9	31.5	61.9	63.2	16.2	92.0	64.0	76.2	28.9	**90.3**	**63.9**	83.0	66.0	67.2	55.2	60.3	**90.4**	59.2
FIDNet[12] + **TransUPR**	92.6	43.6	42.0	29.7	32.7	63.3	58.8	27.0	90.8	58.5	74.5	20.2	88.7	59.9	82.9	**66.5**	67.7	54.1	61.0	90.1	58.7
CENet512[13] + **TransUPR**	**93.3**	**53.6**	**68.1**	**58.6**	**63.7**	**70.0**	**70.6**	52.3	90.4	**70.2**	74.8	**42.1**	89.1	62.8	82.3	64.6	**69.4**	**56.4**	63.0	90.3	**68.2**

Table 1: Quantitative Results Comparison on Semantic KITTI Test Set (Sequences 11 to 21). Scores are given in percentage (%). The input size is 64×2048 except for CENet-512, which takes range images with a size of 64×512 as inputs.

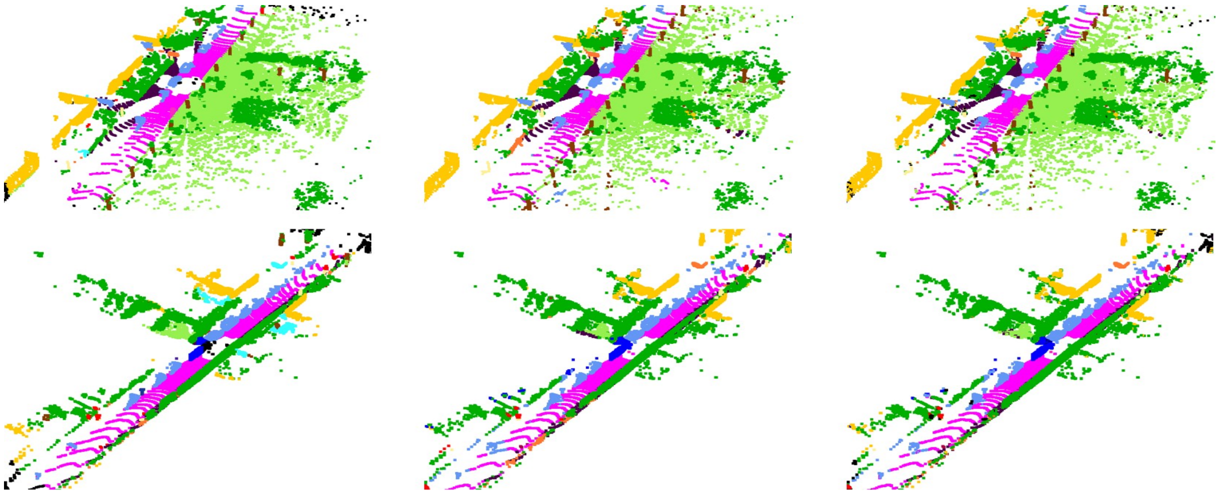

Fig. 3: Visualization results on sequence 08. Column 1: point clouds with ground truth labels. Column 2: point clouds with predicted labels of CENet 512 with a KNN refiner. Column 3: point clouds with predicted labels of CENet with our TransUPR.

selected uncertain points do not influence the performance of our TransUPR. Both experiments with TransUPR can achieve a 1.3% improvement in mIoU on sequence 08 compared to the original FIDNet [12].

Then, we explore the influence of the distance cutoff c_u defined in 2.2 during inference with our TransUPR. Note that we make the $c_u = 1$ for the training of TransUPR to generate more uncertain points. As Table 3 shows, as the c_u increases, the resulted mIoU of the methods with TransUPR decreases, which means when a background point has a distance over 1 between its corresponding foreground point, the back-projected label of the background point has low confidence.

Methods	oACC	mIoU
FIDNet + KNN	90.2	58.6
FIDNet + TransUPR w/o normal	90.5	59.8
FIDNet + TransUPR w/ normal	90.3	59.8

Table 2: Comparison of FIDNet + TransUPR with or without normal vectors on sequence 08.

Methods	oACC	mIoU
SalsaNext	89.3	59.0
SalsaNext + TransUPR $c_u = 1$	89.9	60.6
SalsaNext + TransUPR $c_u = 3$	89.9	60.5
SalsaNext + TransUPR $c_u = 4$	89.9	60.4

Table 3: Quantitative Results to explore influence of c_u for uncertain point selection on sequence 08.

4. CONCLUSION

We propose a plug-and-play transformer-based uncertain point refiner for LiDAR PCSS, i.e., TransUPR, to improve the existing range image-based CNNs. Our TransUPR outperforms the existing methods and achieves state-of-the-art performance, i.e., 68.2% mIoU, on challenging Semantic-KITTI. Extensive experiments demonstrate that our method is generic and easy to use to combine with other existing image-based LiDAR PCSS approaches.

5. REFERENCES

[1] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1887–1893.

[2] A. Milioto, I.o Vizzo, J. Behley, and C. Stachniss, "Rangenet ++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4213–4220.

[3] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4376–4382.

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

[5] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds," in *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II*, 2020, p. 207–222.

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, vol. 30.

[7] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11105–11114.

[8] H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6410–6419.

[9] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9939–9948.

[10] Y. Hou, X. Zhu, Y. Ma, C. C. Loy, and Y. Li, "Point-to-voxel knowledge distillation for lidar semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8479–8488.

[11] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3070–3079.

[12] Y. Zhao, L. Bai, and X. Huang, "Fidnet: Lidar point cloud semantic segmentation with fully interpolation decoding," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4453–4458.

[13] H. Cheng, X. Han, and G. Xiao, "Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, 2022, pp. 01–06.

[14] R. Razani, R. Cheng, E. Taghavi, and L. Bingbing, "Lite-hdseg: Lidar semantic segmentation using lite harmonic dense convolutions," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9550–9556.

[15] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, "Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16004–16013, 2021.

[16] A. Kirillov, Y. Wu, K. He, and R. Girshick, "Pointrend: Image segmentation as rendering," in *2020 CVPR*, 2020, pp. 9796–9805.

[17] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L . Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, vol. 30.

[19] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *2021 ICCV*, 2021, pp. 16239–16248.

[20] M. Berman, A. R. Triki, and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4413–4421.