Point-to-Voxel Knowledge Distillation for LiDAR Semantic Segmentation

Yuenan Hou[1], Xinge Zhu[2], Yuexin Ma[3], Chen Change Loy[4], and Yikang Li[1]

[1]Shanghai AI Laboratory [2]The Chinese University of Hong Kong
[3]ShanghaiTech University [4]S-Lab, Nanyang Technological University
[1]{houyuenan, liyikang}@pjlab.org.cn, [2]zhuxinge123@gmail.com,
[3]mayuexin@shanghaitech.edu.cn, [4]ccloy@ntu.edu.sg

Abstract

*This article addresses the problem of distilling knowledge from a large teacher model to a slim student network for LiDAR semantic segmentation. Directly employing previous distillation approaches yields inferior results due to the intrinsic challenges of point cloud, i.e., sparsity, randomness and varying density. To tackle the aforementioned problems, we propose the Point-to-Voxel Knowledge Distillation (PVD), which transfers the hidden knowledge from both point level and voxel level. Specifically, we first leverage both the pointwise and voxelwise output distillation to complement the sparse supervision signals. Then, to better exploit the structural information, we divide the whole point cloud into several supervoxels and design a difficulty-aware sampling strategy to more frequently sample supervoxels containing less-frequent classes and faraway objects. On these supervoxels, we propose inter-point and inter-voxel affinity distillation, where the similarity information between points and voxels can help the student model better capture the structural information of the surrounding environment. We conduct extensive experiments on two popular LiDAR segmentation benchmarks, i.e., nuScenes [3] and SemanticKITTI [1]. On both benchmarks, our PVD consistently outperforms previous distillation approaches by a large margin on three representative backbones, i.e., Cylinder3D [36, 37], SPVNAS [25] and MinkowskiNet [5]. Notably, on the challenging nuScenes and SemanticKITTI datasets, our method can achieve roughly **75%** MACs reduction and **2×** speedup on the competitive Cylinder3D model and rank **1**st on the SemanticKITTI leaderboard among all published algorithms[1]. Our code is available at* https://github.com/cardwing/Codes-for-PVKD.

1. Introduction

LiDAR semantic segmentation plays a vital role in the perception of autonomous driving as it provides per-point semantic information of the surrounding environment. With the advent of deep learning, plenty of LiDAR segmentation models have been proposed [14, 25, 26, 37] and have dominated the leaderboard of many benchmarks [1, 3]. However, the impressive performance comes at the expense of heavy computation and storage, which impedes them from being deployed in resource-constrained devices.

To enable the deployment of these powerful LiDAR segmentation models on autonomous vehicles, knowledge distillation [10] is a prevailing technique to transfer the dark knowledge from the overparameterized teacher model to the slim student network to achieve model compression. However, directly applying previous distillation algorithms [9, 10, 17, 24, 28] to LiDAR semantic segmentation brings marginal gains due to the intrinsic difficulty of point cloud, *i.e.*, sparsity, randomness and varying density.

To address the aforementioned challenges, we propose the *Point-to-Voxel Knowledge Distillation* (PVD). As the name implies, we propose to distil the knowledge from both point-level and voxel-level. Specifically, to combat against the sparse supervision signals, we first propose to distil the pointwise and voxelwise probabilistic outputs from the teacher, respectively. The pointwise output contains fine-grained perceptual information while the voxelwise prediction embraces coarse but richer clues about the surrounding environment.

To effectively distil the valuable structural knowledge from the unordered point sequences, we propose to exploit the point-level and voxel-level affinity knowledge. The affinity knowledge is obtained via measuring the pairwise semantic similarity of the point features and voxel features. However, straightforwardly mimicking the affinity knowledge of the whole point cloud is intractable since there are tens of thousands of points and the affinity matrix of these point features have over ten billion elements. Consequently,
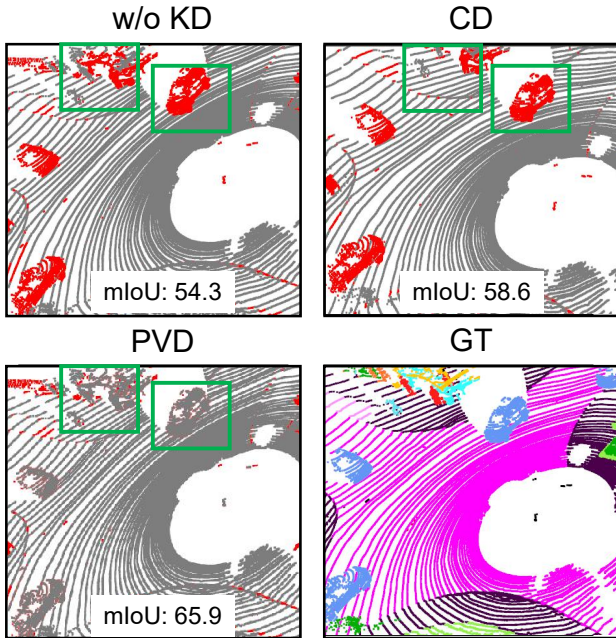
Figure 1. Comparison between the prediction of our method (PVD) and the competitive channel distillation (CD) algorithm [24] on SemanticKITTI validation set. Points that are mistakenly classified are painted red for better visualization. It is obvious that PVD can make the student model predict more accurately for those minority classes (person) and faraway objects (bicycles, highlighted with green rectangles) than the baseline distillation approach. Here, KD denotes knowledge distillation.

we put forward the supervoxel partition to divide the whole point cloud into a fixed number of supervoxels. At each distillation step, we only sample K supervoxels and distil the affinity knowledge computed from point features and voxel features in these supervoxels, thus significantly enhancing the learning efficiency. Considering that uneven number-of-points distribution exists among different classes and objects at distinct distances, we further introduce a difficulty-aware sampling strategy to more frequently sample supervoxels that contain minority classes and faraway objects, emphasizing the learning on hard cases.

We summarize our **contributions** as follow. To our knowledge, we are the first to study how to apply knowledge distillation to LiDAR semantic segmentation for model compression. To address the difficulty of distilling knowledge from point cloud, namely sparsity, randomness and varying density, we propose the point-to-voxel knowledge distillation. In addition, we put forward the supervoxel partition to make the affinity distillation tractable. A difficulty-aware sampling strategy is also employed to more frequently sample supervoxels that contain minority classes and distant objects, thus remarkably enhancing the distillation efficacy on these hard cases. As can be seen from Fig. 1, our method produces much more accurate predictions than the baseline distillation approaches especially for

those minority classes and faraway objects.

We conduct extensive experiments on the nuScenes [3] and SemanticKITTI [1] datasets and the results demonstrate that our algorithm consistently outperforms previous distillation approaches by a large margin on three contemporary models, *i.e.*, Cylinder3D [36, 37], SPV-NAS [25] and MinkowskiNet [5]. Notably, on the nuScenes and SemanticKITTI benchmarks, PVD achieves approximately **75%** MACs reduction and **2×** speedup on the top-performing Cylinder3D model with very minor performance degradation.

2. Related Work

LiDAR semantic segmentation: LiDAR semantic segmentation [5,6,8,14,18,21,22,25,26,29–31,35,37] is crucial for the navigation of autonomous vehicles. PointNet [21] is one of the pioneering work that uses Multi-Layer Perception (MLP) to process point cloud directly. Although effective in processing small-scale point cloud, PointNet and its variants [22] are extremely slow to handle large-scale outdoor point cloud. To cope with large-scale outdoor point cloud, Hu *et al.* [14] exploit random sampling for point selection and a local feature aggregation module is designed to further preserve the key features. Xu *et al.* [31] put forward the range-point-voxel fusion network to make use of the advantages of different views. Zhu *et al.* [37] propose the Cylinder3D method that adopts the cylindrical partition and asymmetrical convolution to better employ the valuable information in point cloud. Tang *et al.* [25] leverage neural architecture search to automatically find the optimal structure for the task at hand. Although these models have shown impressive performance on various benchmarks, a common drawback is that these networks are too cumbersome to be deployed on resource-constrained devices. To enable the deployment of these powerful yet lumbersome LiDAR semantic segmentation models on real-world applications, we propose the point-to-voxel knowledge distillation to achieve model compression.

Knowledge distillation: Knowledge distillation (KD) stems from the seminal work of G. Hinton *et al.* [10]. The primary objective of KD is to transfer the rich dark knowledge from a cumbersome teacher model to a compact student model to mitigate the performance gap between these two models. The majority of the KD approaches concentrate on image classification tasks and they take various forms of knowledge as the distillation targets, *e.g.*, intermediate outputs [13, 23], visual attention maps [12, 33], inter-layer similarity maps [32], sample-level similarity maps [20,27], *etc*. Recently, some researchers have adapted conventional KD techniques to distil knowledge for semantic segmentation tasks [9, 11, 17, 24, 28]. For instance, Liu *et al.* [17] propose to distil three levels of knowledge simultaneously, namely the pixel-level knowledge, the pairwise

similarity knowledge and the holistic knowledge. He *et al*. [9] make the student mimic the compressed knowledge as well as the affinity information of the teacher. Although previous distillation algorithms have shown excellent performance on 2D segmentation, straightforwardly deploying them on LiDAR segmentation tasks brings marginal gains owing to the inherent sparsity, randomness and varying density of point cloud. And to our best knowledge, we are the first to apply knowledge distillation to LiDAR semantic segmentation. The proposed PVD can effectively transfer both point-level and voxel-level knowledge to students and is suitable for distilling various LiDAR semantic segmentation models.

3. Methodology

Given an input point cloud $\mathbf{X} \in \mathbb{R}^{N \times 3}$, the objective of LiDAR semantic segmentation is to assign a class label $l \in \{0, 1, ..., C - 1\}$ to each point, where N is the number of points and C is the number of classes. Contemporary algorithms use CNNs for end-to-end prediction.

Considering that autonomous vehicles typically have limited computation and storage resources and call for real-time performance, efficient models are employed to fulfill the preceding requirement. Knowledge distillation [10] is widely adopted to achieve model compression via conveying the rich dark knowledge from the large teacher model to the compact student network. However, previous distillation methods [9, 17, 24, 28] are tailored for 2D semantic segmentation tasks. Directly applying these algorithms to distill knowledge for 3D segmentation tasks produces unsatisfactory results owing to the intrinsic difficulty of point cloud, *i.e.*, sparsity, randomness and varying density. To address the aforementioned challenges, we propose the Point-to-Voxel Knowledge Distillation (PVD) to transfer the knowledge from both the point level and voxel level.

3.1. Framework overview of Cylinder3D

We first have a brief review of the Cylinder3D model [37] and then introduce the point-to-voxel distillation algorithm based on it. Taking the point cloud as input, Cylinder3D first generates the corresponding feature for each point using a stack of MLPs and then reassigns the point features $F^p \in \mathbb{R}^{N \times C_f}$ based on the cylindrical partition where C_f is the dimension of point features. Point features that belong to the same voxel are aggregated together through the maxpooling operation to obtain the voxel features $F^v \in \mathbb{R}^{M \times C_f}$ where M is the number of non-empty voxels. Next, these voxel features are fed into the asymmetrical 3D convolution networks to produce the voxel-wise output $\mathbf{O}^v \in \mathbb{R}^{R \times A \times H \times C}$. A point-wise refinement module is further employed to produce the refined point-wise prediction $\mathbf{O}^p \in \mathbb{R}^{N \times C}$. Here, N, C, R, A and H denote the number of points, number of classes, radius, angle

and height, respectively. Eventually, we will use the argmax operation to process the pointwise prediction to obtain the classification result of each point.

3.2. Point-to-Voxel Output Distillation

The primary difference between 2D and 3D semantic segmentation lies in the input. Compared with images, point cloud is sparse and it is difficult to train the efficient student model using the sparse supervision signal. Previous distillation approaches [10, 17] typically resort to distilling the ultimate output of the teacher network, *i.e.*, the pointwise output of the teacher network for LiDAR semantic segmentation. Although the pointwise output contains fine-grained perceptual information of the environment, such knowledge is inefficient to learn as there are hundreds of thousands of points. To improve the learning efficiency, in addition to the pointwise output, we propose to distil the voxelwise output as the number of voxels is smaller and is easier to learn. The combination of both pointwise output distillation and voxelwise output distillation naturally form the coarse-to-fine learning process. The pointwise and voxelwise output distillation loss is given below:

$$\mathcal{L}_{out}^p(\mathbf{O}_S^p, \mathbf{O}_T^p) = \frac{1}{NC} \sum_{n=1}^{N} \sum_{c=1}^{C} \mathbf{KL}(\mathbf{O}_S^p(n, c) \| \mathbf{O}_T^p(n, c)),$$
(1)

$$\mathcal{L}_{out}^v(\mathbf{O}_S^v, \mathbf{O}_T^v) =$$
$$\frac{1}{RAHC} \sum_{r=1}^{R} \sum_{a=1}^{A} \sum_{h=1}^{H} \sum_{c=1}^{C} \mathbf{KL}(\mathbf{O}_S^v(r, a, h, c) \| \mathbf{O}_T^v(r, a, h, c)),$$
(2)

where $\mathbf{KL}(.)$ denotes the Kullback-Leibler divergence loss. **Labels for the voxelwise output:** Since a voxel may contain points from different classes, how to assign the proper label to the voxel is also crucial to the performance. Following [37], we adopt the majority encoding strategy that uses the class label having the maximum number of points inside a voxel as the voxel label.

3.3. Point-to-Voxel Affinity Distillation

Distilling the knowledge of the pointwise and voxelwise outputs is insufficient as it merely considers the knowledge of each element and fails to capture the structural information of the surrounding environment. Such structural knowledge is vital to the LiDAR-based semantic segmentation model as the input points are unordered. A natural remedy is to adopt the relational knowledge distillation [17] which calculates the pairwise similarity of all point features. However, there exist two shortcomings in this learn-
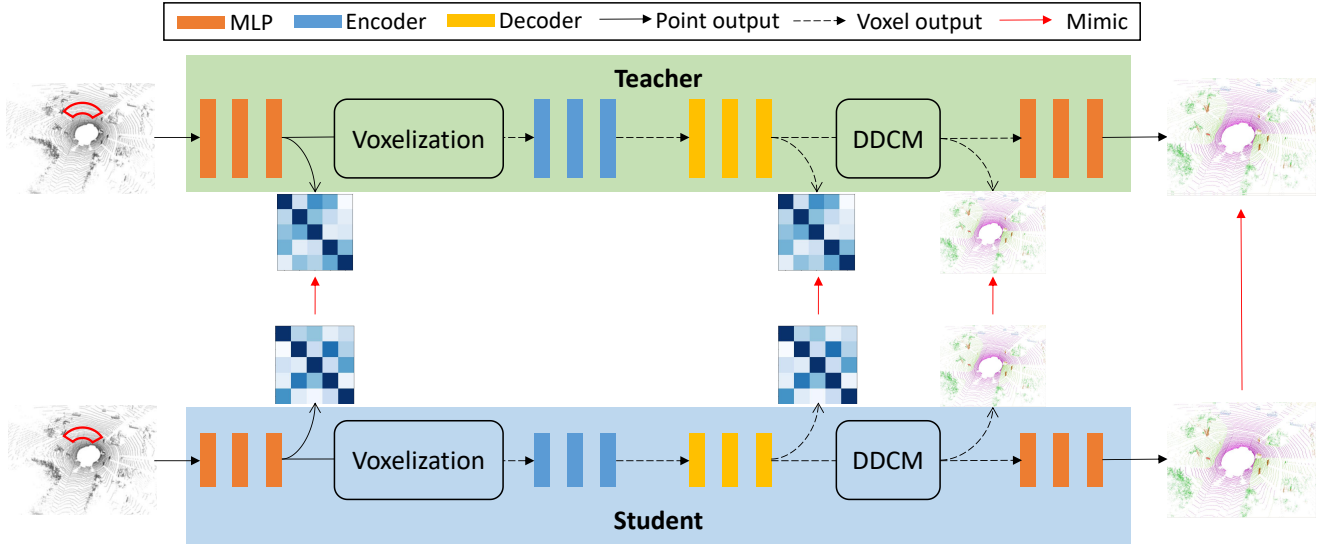
Figure 2. Framework overview. We take the Cylinder3D model [37] as example. There are two networks in our framework, one is the teacher and the other is the student. The student model is obtained via pruning 50% channels of each layer of the teacher. A teacher model is comprised of five parts, *i.e.*, point feature extraction module, point-to-voxel transformation module (voxelization), an encoder-decoder model (asymmetric 3D convolution network), the Dimension-Decomposition Contextual Modeling (DDCM) module, and the point refinement module. Given the input point cloud, we first divide it into a fixed number of supervoxels and sample K supervoxels according to the difficulty-aware sampling strategy (K=1 in this figure, denoted by the red sector). Then, the student is forced to mimic two levels of knowledge from the teacher. The first-level knowledge is the pointwise output and voxelwise output of the whole point cloud. The second-level knowledge is the inter-point affinity matrix and inter-voxel affinity matrix of the sampled supervoxels.

ing scheme: 1) since there are usually hundreds of thousands of points in an input point cloud, the similarity matrix which has over ten billion elements is computationally expensive to calculate and extremely difficult to learn. 2) There exist significant quantity differences between different classes and objects at different distances. The above-mentioned learning strategy ignores such difference and treats all classes and objects equally, thus making the distillation process sub-optimal.

Supervoxel partition: In order to more efficiently learn the relational knowledge, we divide the whole point cloud into several supervoxels whose size is $R_s \times A_s \times H_s$. Each supervoxel is comprised of a fixed number of voxels and the total number of supervoxels is $N_s = \lceil \frac{R}{R_s} \rceil \times \lceil \frac{A}{A_s} \rceil \times \lceil \frac{H}{H_s} \rceil$ where $\lceil . \rceil$ is the ceiling function. We will sample K supervoxels to perform the affinity distillation.

Difficulty-aware sampling: To make supervoxels that contain less frequent classes and faraway objects more likely to be sampled, we present the difficulty-aware sampling strategy. The weight for choosing the i-th supervoxel is:

$$W_i = \frac{1}{f_{class}} \times \frac{d_i}{R} \times \frac{1}{N_s}, \qquad (3)$$

where f_{class} is the class frequency, d_i is the distance of the outer arc of the i-th supervoxel to the origin in the X-Y plane. We treat the classes that have more than 1% of all points in the whole dataset as majority classes and the remaining classes are considered as minority classes. We
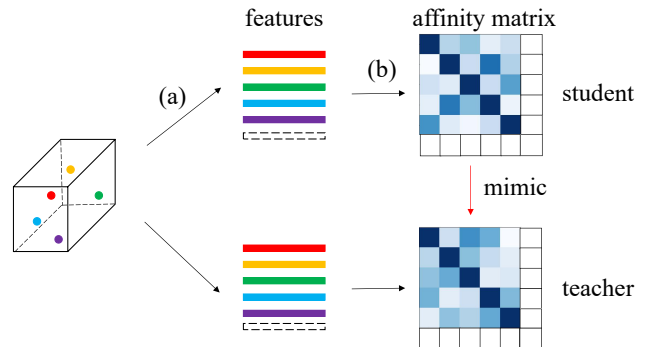

Figure 3. Computation of the inter-point affinity matrix within a supervoxel. Given the sampled supervoxel, we first (a) extract features from each point and then (b) obtain the affinity matrix via calculating the pairwise point features. Finally, the produced affinity matrix of the student is forced to mimic that of the teacher. Note that all-zeros features (the dashed rectangle) will be appended to current features if the number of point features is smaller than N_p.

empirically set the class frequency of the supervoxel as: $f_{class} = 4\exp(-2N_{minor}) + 1$, where N_{minor} is the number of voxels of minority classes in the supervoxel. If there is no voxel of minority class, the f_{class} will be 5. And as the the number of voxels of minority classes increases, f_{class} will be close to 1 quickly. Then, we normalize the weight and obtain the probability of the i-th supervoxel being sampled is: $P_i = \frac{W_i}{\sum_{i=1}^{N_s} W_i}$.

Point/voxel features processing: Note that for each point cloud, the number of input points is different and the den-

sity is varying, thus making the number of point features and voxel features variable in a supervoxel. As to the calculation of loss function, it is desirable to keep the number of features fixed. Hence, we set the number of retained point features and non-empty voxel features as N_p and N_v, respectively. If the number of point features is larger than N_p, then we will retain N_p point features by randomly discarding additional point features of majority class. If the number of point features is smaller than N_p, we will append all-zero features to the current features to obtain N_p features, as is shown in Fig. 3 (a). The voxel features are processed in a similar way.

Here we have N_p point features $\hat{F}_r^p \in \mathbb{R}^{N_p \times C_f}$ and N_v voxel features $\hat{F}_r^v \in \mathbb{R}^{N_v \times C_f}$ in the r-th supervoxel, respectively. Then, for each supervoxel, we calculate the inter-point affinity matrix according to the following equation:

$$\mathbf{C}^p(i,j,r) = \frac{\hat{F}_r^p(i)^\top \hat{F}_r^p(j)}{\|\hat{F}_r^p(i)\|_2 \|\hat{F}_r^p(j)\|_2}, r \in \{1,...,K\} \quad (4)$$

The affinity score captures the similarity of each pair of point features and it can be taken as the high-level structural knowledge to be learned by the student. The inter-point affinity distillation loss is given as below:

$$\mathcal{L}_{aff}^p(\mathbf{C}_S^p, \mathbf{C}_T^p) = \frac{1}{KN_p^2} \sum_{r=1}^{K} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \|\mathbf{C}_S^p(i,j,r) - \mathbf{C}_T^p(i,j,r)\|_2^2. \quad (5)$$

The inter-voxel affinity matrix is computed similarly. Eventually, we make the student mimic the generated affinity matrices of the teacher model. The inter-voxel affinity distillation loss is presented as follows:

$$\mathcal{L}_{aff}^v(\mathbf{C}_S^v, \mathbf{C}_T^v) = \frac{1}{KN_v^2} \sum_{r=1}^{K} \sum_{i=1}^{N_v} \sum_{j=1}^{N_v} \|\mathbf{C}_S^v(i,j,r) - \mathbf{C}_T^v(i,j,r)\|_2^2. \quad (6)$$

Visualization of the learned affinity maps: From Fig. 4, we can see that PVD causes a closer affinity map between student and teacher. With PVD , features that belong to the same class are pulled closer while those of different classes are pushed apart in the feature space, resulting in a much clear affinity map. Compared with the rival channel distillation approach [24], PVD can better transfer the structural knowledge from teacher to student, which strongly validates the superiority of PVD in distilling LiDAR segmentation models.
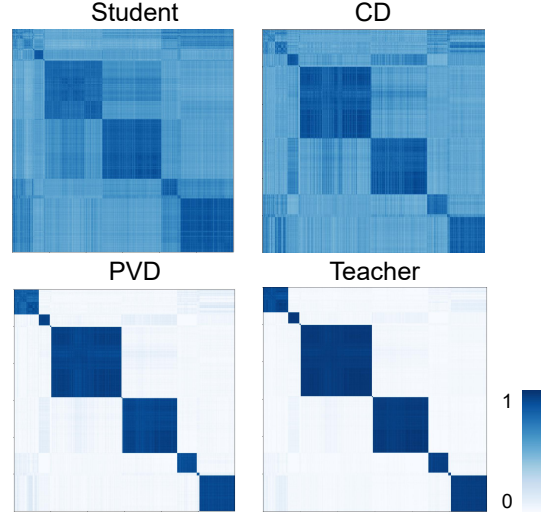

Figure 4. Comparison between inter-voxel affinity maps of different algorithms. The affinity value is obtained via normalizing the cosine similarity score to [0, 1].

3.4. Final objective

Our final loss function is composed of seven terms, *i.e.*, the weighted cross entropy loss for the pointwise output and voxelwise output, the lovasz-softmax loss [2], the point-to-voxel output distillation loss and the point-to-voxel affinity distillation loss:

$$\begin{aligned} \mathcal{L} =& \mathcal{L}_{wce}^p + \mathcal{L}_{wce}^v + \mathcal{L}_{lovasz} + \alpha_1 \mathcal{L}_{out}^p(\mathbf{O}_S^p, \mathbf{O}_T^p) \\ &+ \alpha_2 \mathcal{L}_{out}^v(\mathbf{O}_S^v, \mathbf{O}_T^v) + \beta_1 \mathcal{L}_{aff}^p(\mathbf{C}_S^p, \mathbf{C}_T^p) \\ &+ \beta_2 \mathcal{L}_{aff}^v(\mathbf{C}_S^v, \mathbf{C}_T^v), \end{aligned} \quad (7)$$

where α_1, α_2, β_1 and β_2 are the loss coefficients to balance the effect of the distillation losses on the main task loss.

4. Experiments

Datasets. Following the practice of Cylinder3D [37], we conduct experiments on two popular LiDAR semantic segmentation benchmarks, *i.e.*, nuScenes [3] and SemanticKITTI [1]. For nuScenes, it consists of 1000 driving scenes, in which 850 scenes are selected for training and validation, and the remaining 150 scenes are chosen for testing. 16 classes are utilized for LiDAR semantic segmentation after merging similar classes and eliminating infrequent classes. For SemanticKITTI, it is comprised of 22 point cloud sequences, where sequences 00 to 10, 08 and 11 to 21 are used for training, validation and testing, respectively. A total number of 19 classes are chosen for training and evaluation after merging classes with distinct moving status and discarding classes with very few points.

Evaluation metrics. Following [37], we adopt the intersection-over-union (IoU) of each class and mIoU of all classes as the evaluation metric. The calculation of IoU is:

$IoU_i = \frac{TP_i}{TP_i+FP_i+FN_i}$, where TP_i, FP_i and FN_i represent the true positive, false positive and false negative of class i.

Implementation details. Following [37], we leverage Adam [15] as the optimizer and the initial learning rate is set as 2e-3. Batch size is set as 4 and the number of training epochs is 40. α_1, α_2, β_1 and β_2 are set as 0.1, 0.15, 0.15 and 0.25, respectively. We take the rival and open sourced Cylinder3D[2] [37] approach as the backbone since the top-performing RPVNet [31] and AF2S3Net [4] do not release their codes. Random flipping, rotation, scaling and transformation are taken as the data augmentation strategy. The size of the voxel output is $480\times360\times32$, where the three dimensions denote the radius, angle and height, respectively. The size of the supervoxel is set as $120\times60\times8$. N_v and N_p are set as 3000 and 6000, respectively. The number of sampled supervoxels K is set as 4. The inter-point affinity distillation is performed on the output of the point feature extraction module and the inter-voxel affinity distillation is conducted on the output of the encoder-decoder backbone. For nuScenes, Cylinder3D_0.5$\times$ is produced from the original Cylinder3D model by pruning 50% channels for each layer of the whole network. For SemanticKITTI, Cylinder3D_0.5$\times$ is obtained by merely pruning 50% channels for each layer of the asymmetrical 3D convolution network and we keep the point feature extraction module unchanged as it is vital to extract rich information from the input point cloud. We also apply our method to compress SPVNAS[3] [25] and MinkowskiNet [5] to verify the scalability of our algorithm. More details are provided in the supplementary material.

Baseline distillation algorithms. In addition to the state-of-the-art methods in each benchmark, we also compare our method with classical KD methods and contemporary distillation approaches tailored for 2D semantic segmentation, including vanilla KD [10], SKD [17], CD [24], IFV [28] and KA [9]. Here, SKD takes the output probability maps and pairwise similarity maps as mimicking targets. We remove the original holistic distillation loss for SKD as incorporating GANs into current framework will cause severe training instability; CD utilizes the intermediate feature maps and score maps as knowledge; IFV transfers the intra-class feature variation from the teacher to the student; KA makes the student distil the compressed knowledge and the affinity information of the whole output from the teacher.

4.1. Results

Comparison with state-of-the-art LiDAR segmentation models: We compare our model with contemporary LiDAR semantic segmentation models, *e.g.*, KPConv [26], TORNADONet [7] and SPVNAS [25]. From Table 1, we

can see that Cylinder3D_0.5$\times$+PVD (the penultimate row) achieves comparable performance with the original Cylinder3D model on the SemanticKITTI test set. Compared to KPConv and SPVNAS, our Cylinder3D_0.5$\times$+PVD not only achieves better performance, *e.g.*, **3.8%** higher than SPVNAS in mIoU, but also has much lower latency than the SPVNAS method (259 ms v.s. 76 ms). Specifically, on minority classes such as bicycle, motocycle and bicyclist, the IoU of Cylinder3D_0.5$\times$+PVD is at least **10.5%** higher than the SPVNAS method. And with some engineering tricks like finetuning and flip & rotation test ensemble, our Cylinder3D_0.5$\times$+PVD (the last row) can obtain **71.2** mIoU, which is **2.3** mIoU higher than the original Cylinder3D model. Impressive performance is also observed in nuScenes validation set. Our Cylinder3D_0.5$\times$+PVD exhibits similar performance with the original Cylinder3D network in terms of the overall mIoU and the IoU on each class.

Comparison with previous distillation methods: From Table 1 and 2, we can see that PVD significantly outperforms baseline distillation algorithms in both benchmarks. The performance gap between PVD and the most competitive KD method is larger than 1.8. For instance, on SemanticKITTI test set, our PVD is **2.8** mIoU higher than the CD method. And on both majority classes and minority classes, our PVD significantly outperforms traditional distillation algorithms. For instance, on nuScenes dataset, PVD is at least **2** mIoU higher than SKD in classes such as bicycle, bus, car, trailer and sidewalk. The aforementioned results strongly demonstrate the effectiveness of PVD in transferring knowledge for teacher-student learning.

Generalization to more architectures: To verify the generalization of our method, we also apply PVD to compress SPVNAS [25] and MinkowskiNet [5]. Since SPVNAS does not provide the training code for the NAS-based architecture, we conduct experiments on its manually designed architecture. As can be seen from Table 3, our PVD still brings more gains to the student model than baseline distillation algorithms. For instance, our PVD outperforms the SKD algorithm by **2.6** mIoU in terms of mIoU on the SPVNAS backbone. It is noteworthy that PVD can safely achieve 75% MACs reduction without causing severe performance drop. The above results strongly demonstrate the good scalability of our method.

Qualitative results: As can be seen from Fig. 5, compared with the SKD approach, our PVD greatly improves the prediction of the student model. The prediction errors of PVD on those minority classes, *e.g.*, person and bicycle, are significantly smaller than those of SKD. Besides, on objects that are faraway from the origin, *e.g.*, the car highlighted by the green rectangle, PVD also yields more accurate predictions than SKD. And PVD has lower inter-class similarity and higher intra-class similarity, which explicitly showcases

---

[2]https://github.com/xinge008/Cylinder3D
[3]https://github.com/mit-han-lab/spvnas

Table 1. Quantitative results of our proposed method and state-of-the-art LiDAR semantic segmentation methods as well as previous distillation approaches on SemanticKITTI test set. Cylinder3D_0.5× is abbreviated as C3D_0.5× to save space. * means that finetuning and flip & rotation test ensemble are applied. All results can be found in the online leaderboard.

Methods	mIoU	Latency (ms)	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic
Salsanext [6]	59.5	–	91.9	48.3	38.6	38.9	31.9	60.2	59.0	19.4	91.7	63.7	75.8	29.1	90.2	64.2	81.8	63.6	66.5	54.3	62.1
KPConv [26]	58.8	263	96.0	32.0	42.5	33.4	44.3	61.5	61.6	11.8	88.8	61.3	72.7	31.6	**95.0**	64.2	84.8	69.2	69.1	56.4	47.4
FusionNet [34] *	61.3	–	95.3	47.5	37.7	41.8	34.5	59.5	56.8	11.9	91.8	68.8	77.1	30.8	92.5	**69.4**	84.5	69.8	68.5	60.4	66.5
KPRNet [16] *	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
TORNADONet [7] *	63.1	–	94.2	55.7	48.1	40.0	38.2	63.6	60.1	34.9	89.7	66.3	74.5	28.7	91.3	65.6	85.6	67.0	71.5	58.0	65.9
SPVNAS [25] *	66.4	259	**97.3**	51.5	50.8	59.8	58.8	65.7	65.2	43.7	90.2	67.6	75.2	16.9	91.3	65.9	86.1	73.4	71.0	64.2	66.9
Cylinder3D [37] *	68.9	170	97.1	67.6	63.8	50.8	58.5	73.7	69.2	48.0	92.2	65.0	77.0	32.3	90.7	66.5	85.6	72.5	69.8	62.4	66.2
C3D_0.5×	65.3		93.4	62.3	59.2	48.3	56.4	72.3	66.3	21.0	91.2	61.3	75.3	30.4	89.8	65.4	84.2	71.4	67.3	60.2	64.2
C3D_0.5× + KD	65.6		93.8	62.5	59.4	48.6	55.3	72.9	66.5	21.9	91.8	61.3	75.7	30.5	90.4	65.5	84.3	71.7	67.6	60.3	64.8
C3D_0.5× + CD	66.1		94.5	62.7	59.8	49.3	57.2	72.1	67.1	22.7	92.1	61.4	74.9	30.8	90.9	67.3	84.6	72.2	68.3	61.1	65.1
C3D_0.5× + IFV	65.5	76	93.7	62.4	59.1	48.7	56.7	72.4	66.6	21.4	91.5	62.0	75.6	30.4	90.3	66.2	84.7	71.5	67.5	60.6	64.3
C3D_0.5× + SKD	65.8		93.6	62.7	59.6	48.5	57.4	72.8	66.7	24.3	91.6	61.4	75.9	30.8	90.1	65.6	84.8	71.7	67.5	60.7	65.2
C3D_0.5× + KA	65.8		93.4	62.6	58.9	48.5	56.5	72.7	66.5	20.7	91.6	61.5	75.3	30.1	89.9	65.6	84.4	71.3	67.8	60.3	65.8
C3D_0.5× + PVD	68.9		96.7	66.4	61.0	**60.0**	59.3	73.2	72.1	25.0	91.4	66.5	76.2	37.1	93.0	70.5	85.9	72.7	69.8	64.1	**67.8**
C3D_0.5× + PVD *	**71.2**		97.0	**67.9**	**69.3**	53.5	**60.2**	**75.1**	**73.5**	**50.5**	91.8	70.9	77.5	**41.0**	92.4	**69.4**	**86.5**	**73.8**	**71.9**	**64.9**	65.8

Table 2. Quantitative results of our proposed method and state-of-the-art LiDAR semantic segmentation methods as well as previous distillation approaches on nuScenes validation set.

Methods	mIoU	barrier	bicycle	bus	car	construction	motorcycle	pedestrian	traffic-cone	trailer	truck	driveable	other	sidewalk	terrain	manmade	vegetation
RangeNet++ [19]	65.5	66.0	21.3	77.2	80.9	30.2	66.8	69.6	52.1	54.2	72.3	94.1	66.6	63.5	70.1	83.1	79.8
PolarNet [35]	71.0	74.7	28.2	85.3	90.9	35.1	77.5	71.3	58.8	57.4	76.1	96.5	71.1	74.7	74.0	87.3	85.7
Salsanext [6]	72.2	74.8	34.1	85.9	88.4	42.2	72.4	72.2	63.1	61.3	76.5	96.0	70.8	71.2	71.5	86.7	84.4
Cylinder3D [37]	**76.1**	**76.4**	**40.3**	**91.2**	93.8	**51.3**	**78.0**	**78.9**	**64.9**	**62.1**	**84.4**	**96.8**	**71.6**	**76.4**	75.4	**90.5**	**87.4**
C3D_0.5×	73.6	74.6	36.2	88.2	87.3	47.9	76.4	77.0	63.4	58.8	82.3	95.1	70.0	73.5	73.6	88.7	85.2
C3D_0.5× + KD	73.9	75.2	35.4	88.3	88.2	47.6	76.8	77.2	63.6	57.3	83.1	95.7	70.1	75.2	73.1	89.2	85.3
C3D_0.5× + CD	74.1	75.4	36.1	88.4	89.3	46.9	76.1	77.6	62.9	58.0	84.3	96.0	70.3	74.8	74.6	90.1	85.6
C3D_0.5× + IFV	73.8	74.7	36.6	88.3	88.6	47.2	76.7	77.1	63.1	58.2	83.5	95.1	70.2	73.4	73.8	88.9	84.3
C3D_0.5× + SKD	74.2	74.9	37.3	87.6	89.1	47.5	76.2	77.4	63.2	59.3	83.4	95.9	70.4	73.9	74.3	90.3	87.1
C3D_0.5× + KA	73.9	74.2	36.3	88.5	87.6	47.1	76.9	78.3	63.5	57.6	83.4	94.9	70.3	73.8	73.2	88.4	86.3
C3D_0.5× + PVD	76.0	76.2	40.0	90.2	**94.0**	50.9	77.4	78.8	64.7	62.0	84.1	96.6	71.4	76.4	**76.3**	90.3	86.9

Table 3. Performance of different algorithms on compressing SPV-NAS and MinkowskiNet on SemanticKITTI validation set.

Algorithm	mIoU	MACs (G)
SPVNAS [25]	**63.8**	118.6
SPVNAS_0.5×	60.4	
SPVNAS_0.5× + CD	60.9	29.7
SPVNAS_0.5× + SKD	61.2	
SPVNAS_0.5× + PVD	**63.8**	
MinkowskiNet [5]	61.9	114.0
MinkowskiNet_0.5×	58.9	
MinkowskiNet_0.5× + CD	59.6	28.5
MinkowskiNet_0.5× + SKD	59.4	
MinkowskiNet_0.5× + PVD	61.8	

Table 4. Influence of each component on the final performance.

$\mathcal{L}_{out\_p}$	$\mathcal{L}_{out\_v}$	$\mathcal{L}_{aff\_p}$	$\mathcal{L}_{aff\_v}$	mIoU
				63.1
√				63.4
√	√			64.1
√	√	√		64.7
√	√	√	√	66.4

the efficacy of PVD in distilling structural knowledge from the teacher model.

4.2. Ablation studies

In this section, we perform comprehensive ablation studies to examine the efficacy of each component, super-voxel size as well as the sampling strategy on the final

performance. The experiments are conducted in the SemanticKITTI validation set. More ablation studies are put in the supplementary material.

Effect of each component. From Table 4, we have the following observations: 1) Combining both point-to-voxel output distillation and affinity distillation brings the most performance gains. 2) The voxel-level distillation brings more gains than the point-level distillation, suggesting the necessity of introducing the voxel-level mimicking loss. 3) The affinity distillation yields more gains than the output distillation, demonstrating the importance of leveraging the relational knowledge to better capture the structural information.
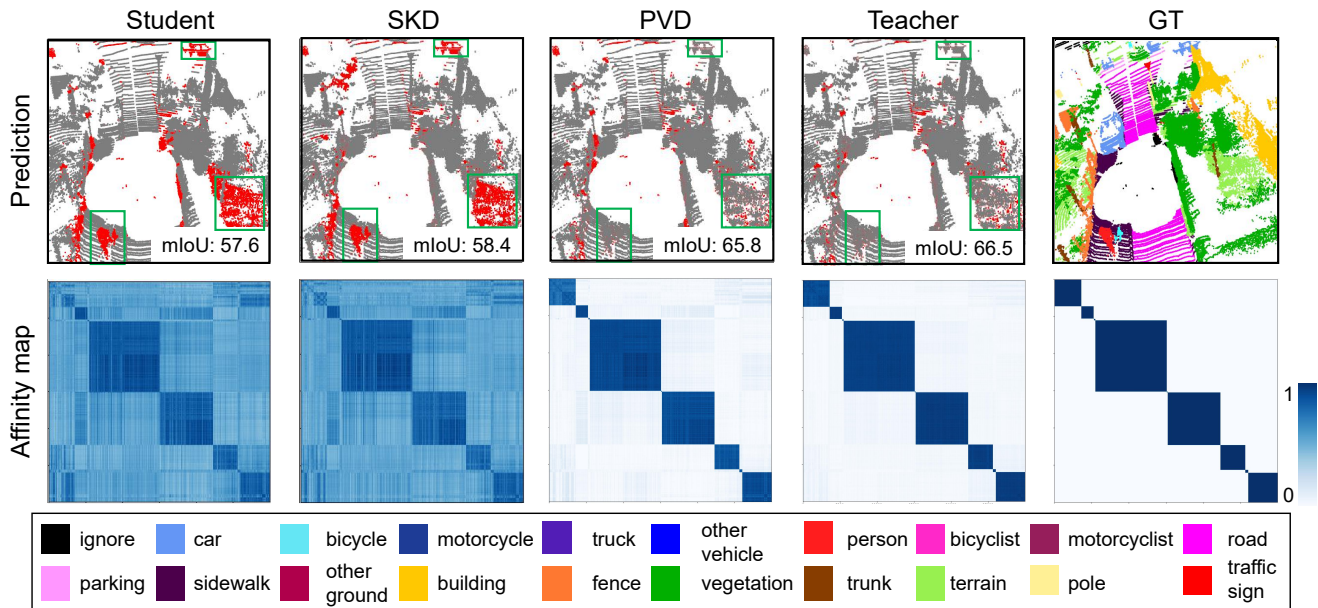
Figure 5. Visual comparison of different methods on the SemanticKITTI validation set. Here, the ground-truth for the inter-voxel affinity map is the ideal map where the intra-class similarity score is 1 and inter-class similarity score is 0.

Table 5. Effect of supervoxel size on the performance.

supervoxel size	mIoU
(60, 30, 4)	65.3
(90, 45, 6)	65.7
(120, 60, 8)	66.4
(180, 90, 12)	65.6
(240, 180, 16)	65.2
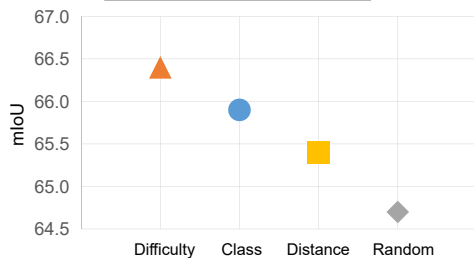
Figure 6. Comparison between different sampling strategies.

Supervoxel size. Note that we perform the inter-point and inter-voxel distillation on the sampled supervoxels. Supervoxel size has a non-negligible effect on the efficacy of PVD since a overly small size will make student learn little from the affinity distillation loss while a large supervoxel size will weaken the learning efficiency of PVD. Here, we keep the number of sampled supervoxels as 4 to remove the effect of this factor. From Table 5, we can see that setting the supervoxel size to (120, 60, 8) yields the best performance. Remarkably increasing or decreasing the supervoxel size will harm the distillation efficacy.

Sampling strategy. We compare four different sampling strategies, *i.e.*, the original difficulty-aware sampling, distance-aware sampling, category-aware sampling and random sampling. Here, distance-aware sampling is to only

more frequently sample distant points while category-aware sampling will more likely sample points belonging to rare classes. From Fig. 6, it is apparent that difficulty-aware sampling brings more gains than the other three strategies. Specifically, difficulty-aware sampling outperforms both distance-aware and category-aware sampling, suggesting that both distance and categorical awareness are crucial to the distillation effect. The large gap between difficulty-aware sampling and random sampling validates the necessity of difficulty-aware sampling strategy.

5. Conclusion

In this paper, we propose a novel point-to-voxel knowledge distillation approach (PVD) tailored for LiDAR semantic segmentation. PVD is comprised of the point-to-voxel output distillation and affinity distillation. The supervoxel partition and difficulty-aware sampling strategy are further proposed to improve the learning efficiency of affinity distillation. We perform experiments on two LiDAR semantic segmentation benchmarks and show that PVD significantly outperforms baseline distillation algorithms on distilling Cylinder3D, SPVNAS and MinkowskiNet. The impressive results indicate that there still exists large redundancy in 3D segmentation models and our approach can serve as a strong baseline to compress these cumbersome models.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of Lidar Sequences. In *IEEE International Conference on Computer Vision*, pages 9297–9307, 2019. 1, 2, 5

[2] Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. The Lovasz-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2018, pages 4413–4421, 2018. 5

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. NuScenes: A Multimodal Dataset for Autonomous Driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 1, 2, 5

[4] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. (af)2-S3Net: Attentive Feature Fusion with Adaptive Feature Selection for Sparse Semantic Segmentation Network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12547–12556, 2021. 6

[5] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1, 2, 6, 7

[6] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. SalsaNext: Fast, Uncertainty-Aware Semantic Segmentation of lidar Point Clouds. In *International Symposium on Visual Computing*, pages 207–222. Springer, 2020. 2, 7

[7] Martin Gerdzhev, Ryan Razani, Ehsan Taghavi, and Liu Bingbing. Tornado-net: Multiview Total Variation Semantic Segmentation with Diamond Inception Module. In *IEEE International Conference on Robotics and Automation*, pages 9543–9549. IEEE, 2021. 6, 7

[8] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, 2018. 2

[9] Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun, and Youliang Yan. Knowledge Adaptation for Efficient Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 578–587, 2019. 1, 2, 3, 6

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *Statistics*, 1050:9, 2015. 1, 2, 3, 6

[11] Yuenan Hou, Zheng Ma, Chunxiao Liu, Tak-Wai Hui, and Chen Change Loy. Inter-Region Affinity Distillation for Road Marking Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12486–12495, 2020. 2

[12] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning Lightweight Lane Detection CNNs by Self Attention Distillation. In *IEEE International Conference on Computer Vision*, pages 1013–1021, 2019. 2

[13] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning to Steer by Mimicking Features from Heterogeneous Auxiliary Networks. In *Association for the Advancement of Artificial Intelligence*, volume 33, pages 8433–8440, 2019. 2

[14] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 1, 2

[15] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. 6

[16] Deyvid Kochanov, Fatemeh Karimi Nejadasl, and Olaf Booij. KPRNet: Improving Projection-based Lidar Semantic Segmentation. *arXiv preprint arXiv:2007.12668*, 2020. 7

[17] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured Knowledge Distillation for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2604–2613, 2019. 1, 2, 3, 6

[18] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-Voxel CNN for Efficient 3D Deep Learning. *arXiv preprint arXiv:1907.03739*, 2019. 2

[19] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. RangeNet++: Fast and Accurate Lidar Semantic Segmentation. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4213–4220. IEEE, 2019. 7

[20] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational Knowledge Distillation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019. 2

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 2

[22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[23] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for Thin Deep Nets. In *International Conference on Learning Representations*, 2015. 2

[24] Changyong Shu, Yifan Liu, Jianfei Gao, Lin Xu, and Chunhua Shen. Channel-wise Distillation for Semantic Segmentation. *IEEE International Conference on Computer Vision*, 2020. 1, 2, 3, 5, 6

[25] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In *European Conference on Computer Vision*, pages 685–702. Springer, 2020. 1, 2, 6, 7

[26] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J

Guibas. Kpconv: Flexible and Deformable Convolution for Point Clouds. In *IEEE International Conference on Computer Vision*, pages 6411–6420, 2019. 1, 2, 6, 7

[27] Fred Tung and Greg Mori. Similarity-Preserving Knowledge Distillation. In *IEEE International Conference on Computer Vision*, pages 1365–1374, 2019. 2

[28] Yukang Wang, Wei Zhou, Tao Jiang, Xiang Bai, and Yongchao Xu. Intra-class Feature Variation Distillation for Semantic Segmentation. In *European Conference on Computer Vision*, pages 346–362, 2020. 1, 2, 3, 6

[29] 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 *IEEE International Conference on Robotics and Automation*, pages 1887–1893. IEEE, 2018. 2

[30] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeeze-segv3: Spatially-Adaptive Convolution for Efficient Point-Cloud Segmentation. In *European Conference on Computer Vision*, pages 1–19. Springer, 2020. 2

[31] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. RPVnet: A Deep and Efficient Range-Point-Voxel Fusion Network for Lidar Point Cloud Segmentation. In *IEEE International Conference on Computer Vision*, pages 16024–16033, October 2021. 2, 6

[32] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017. 2

[33] Sergey Zagoruyko and Nikos Komodakis. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *International Conference on Learning Representations*, 2017. 2

[34] Feihu Zhang, Jin Fang, Benjamin Wah, and Philip Torr. Deep FusionNet for Point Cloud Semantic Segmentation. In *European Conference on Computer Vision*, pages 644–663. Springer, 2020. 7

[35] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An Improved Grid Representation for Online Lidar Point Clouds Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020. 2, 7

[36] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Wei Li, Yuexin Ma, Hongsheng Li, Ruigang Yang, and Dahua Lin. Cylindrical and Asymmetrical 3D Convolution Networks for Lidar-based Perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 2

[37] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9939–9948, 2021. 1, 2, 3, 4, 5, 6, 7