Self-Distillation for Robust LiDAR Semantic Segmentation in Autonomous Driving

Jile Li[1], ✉Hang Dai[2], and ✉Yong Ding[1]

[1] Zhejiang University, Hangzhou, China
[2] MBZUAI, Abu Dhabi, United Arab Emirates
hang.dai@mbzuai.ac.ae dingy@vlsi.zju.edu.cn

Abstract. We propose a new and effective self-distillation framework with our new Test-Time Augmentation (TTA) and Transformer based Voxel Feature Encoder (TransVFE) for robust LiDAR semantic segmentation in autonomous driving, where the robustness is mission-critical but usually neglected. The proposed framework enables the knowledge to be distilled from a teacher model instance to a student model instance, while the two model instances are with the same network architecture for jointly learning and evolving. This requires a strong teacher model to evolve in training. Our TTA strategy effectively reduces the uncertainty in the inference stage of the teacher model. Thus, we propose to equip the teacher model with TTA for providing privileged guidance while the student continuously updates the teacher with better network parameters learned by itself. To further enhance the teacher model, we propose a TransVFE to improve the point cloud encoding by modeling and preserving the local relationship among the points inside each voxel via multi-head attention. The proposed modules are generally designed to be instantiated with different backbones. Evaluations on SemanticKITTI and nuScenes datasets show that our method achieves state-of-the-art performance. Our code is publicly available at `https://github.com/jialeli1/lidarseg3d`.

Keywords: Semantic Segmentation, LiDAR, Self-Distillation

1 Introduction

LiDAR point cloud semantic segmentation network as a visual recognition module in autonomous driving system, which is vital for driving scenario understanding [4,6]. The previous works [38,17,18] show that slight disturbances to the input data may impair the prediction results of neural networks, such as noise, missing part, and so on [38]. As shown in Fig. 1, the LiDAR semantic segmentation network gets undesirable performance degradation when imposing the point-wise random noise and the random dropping on the input point cloud. The unexpected conditions, such as weather changes and unstable data transmission, may cause the disturbances. Thus, boosting robustness is mission-critical but neglected in LiDAR segmentation for autonomous driving.

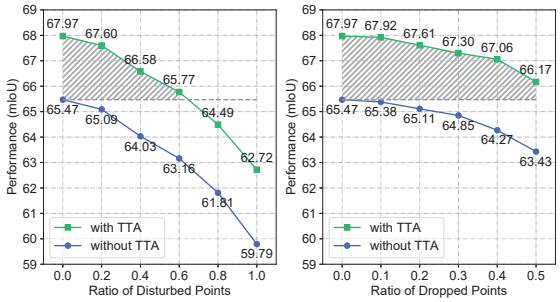

Fig. 1. Robustness test by disturbing point cloud samples in SemanticKITTI [4] dataset using the following settings: 1) add point-wise random noise uniformly distributed in $[-0.05, +0.05]$ meters, 2) randomly drop points. We plot the performance mIoU (%) of the models with and without TTA against disturbing.

Training a robust model can be achieved by means of data augmentation [11,15] as well as knowledge distillation [19,41,45]. The standard types of data augmentation transformations for LiDAR point cloud include random flip, random rotation, random scaling, and random translation [59,56,39,8], which have been widely applied. Recently, the knowledge distillation [19] is broadly developed for model compression purposes in 2D semantic segmentation [30,45,2], showing a promising way to robustly train a compact student model with the guidance from a cumbersome teacher model. But there are three main limitations: (**i**) A cumbersome teacher model with higher performance is required to be designed and trained. (**ii**) The differences of the feature distributions between the heterogeneous teacher model and student model are detrimental for distillation [45]. Although many efforts are made to adapt the feature map for alleviating the distribution gap in 2D semantic segmentation [45,30,2], the gap still exists in such two heavy and lightweight models. (**iii**) The student model yields a better performance with distillation than individually training, but not a new state-of-the-art performance due to the limited distillation efficiency.

Inspired by 2D semi-supervised learning works [41,40] that can train a model with the unlabeled image mainly by the consistency regularization between outputs of two instances of the same model, we propose to perform self-distillation to learn from the segmentation model itself without the cumbersome teacher model. We follow these 2D works [41,29,40] to instantiate a teacher model to provide predictions as the soft labels for training a student model, and update the teacher model with the temporal assembly of student model. In such a self-distillation, the above limitations (**i**) and (**ii**) can be effectively avoided. Unlike the purpose of training on extra unlabeled data [29,40], we aim to further address the limitation (**iii**) above by self-distillation to achieve robust LiDAR semantic segmentation with stronger performance. As privileged guidance from the teacher is critical for transferring useful knowledge to a student, the quality guarantee should be considered for further enhancing the teacher. Thus, two additional components on robustness boost in the inference stage and the point

cloud encoding aspect, which are further proposed as follows and tailored on LiDAR semantic segmentation field.

As shown in Fig. 1, the robustness in the inference stage is susceptible to external disturbances, thus the robustness boost is necessary to be addressed when the soft labels are inferred from the teacher. Test-Time Augmentation (TTA) is a effective and general idea for boosting the models in image-based 2D computer vision [60,20,7] by averaging the predictions of input variants to reduce the uncertainty. It is feasible to equip the teacher model with TTA when inferring predictions as distillation guidance. But TTA is barely used for distillation in 2D semantic segmentation and not well-investigated on LiDAR 3D semantic segmentation yet. Only the flip and multi-scale tests are independently used in 2D semantic segmentation [60,20,48], since the other transformations like rotation and translation will make the pixel-wise results overflow the image boundaries and spatially un-aligned. Differently, as long as the order of points is unchanged in a point cloud, the point-wise results of different input variants are naturally aligned for merging so that more types of transformations should account for LiDAR semantic segmentation. Thus, we firstly introduce the TTA into this field by reusing a proposed compound transformation instead of any individual transformations multiple times. The compound transformation can provide more diversity and flexibility than individual transformations in point cloud variants generation. In Fig. 1, for the input samples with the ratio of disturbance within the shaded area, the LiDAR segmentation model with TTA can still achieve better performance than the model without TTA on clean input samples. The proposed TTA is potential to improve the soft labels from the teacher model for better self-distillation.

To enhance the point cloud encoding in a large-scale autonomous driving scene, we mainly focus on the voxel methods since they [59,39,56] are significantly more effective and efficient than point methods [50,42,21] due to the better structured representation and the deeper convolutional network architecture. The major concern for voxel-based methods is the quantization error introduced in the voxelization process, where a cluster of local points inside a voxel are encoded as the average of their input features (e.g. 3D coordinates and reflection intensity) [39,12,37,51,57]. The average operation encoded voxel features can also be treated as introducing some noise to the initial point features. Encoding a cluster of points as the average reduces the consumption resource of feature extraction, it is also equivalent to losing object details as well as dropping points. The larger voxel size worsens these cases and weakens the robustness of point cloud encoding. To address this, we propose a novel Voxel Feature Encoder (VFE) with Transformer [43] on the local points inside a voxel, termed as TransVFE. Transformer can naturally accommodate the unordered sequence data like point clouds and model the relationship among local points via multi-head attention [14]. Thus, the proposed TransVFE can serve as a performance enhancement module in our teacher and student models, which models and preserves the local geometric relationship during the conversion from points to voxels at the point cloud encoding level.

Notably, teacher and student models in our method are designed with the same network architectures for jointly learning and evolving without requiring the additional cumbersome teacher model. The model equipped with the proposed TTA as the teacher can provide privileged guidance while the student continuously updates the teacher with better network parameters learned by itself. The proposed TransVFE also are integrated into the models to enhance the voxel feature learning. In such a manner, even the teacher and student are homogeneous, the robust model training can proceed with the self-distillation.

Our main **contributions** are 4-fold: (**i**) We propose a novel method for robust LiDAR semantic segmentation in autonomous driving, achieving new state-of-the-art performance on SemanticKITTI and nuScenes datasets; (**ii**) we propose to perform self-distillation for LiDAR semantic segmentation, which enables the homogeneous models of teacher and student to jointly learn and evolve; (**iii**) We propose a simple yet efficient LiDAR semantic segmentation TTA strategy with a compound transformation, which can improve the teacher model for better self-distillation as well as be utilized independently in the inference stage with mIoU improvements; (**iv**) We propose a novel TransVFE that can enhance the robust point cloud voxel feature learning in our teacher and student models by modeling and preserving the local relationship among the points in each voxel.

2 Related Work

2.1 LiDAR Semantic Segmentation

Semantic segmentation on large-scale point clouds [4,6] measured by the LiDAR sensors are of more challenges than the synthetic and indoor point clouds [52,3]. The LiDAR semantic segmentation methods mostly follow the network architecture of U-Net [36] with skip-connections incorporated symmetrical encoder and decoder, but are differently designed with point cloud representations of point, 2D image and 3D voxel.

The **point** representation usually takes large computation costs on gathering the disordered neighbors for feature extraction. To trade off the computation burden and segmentation performance, PointASNL [50] and RandLA-Net [21] propose the learnable adaptive and the efficient random down-sampling algorithms to improve the classic farthest point sampling [34], respectively. The expressive local feature extractors developed in KPConv [42], BAAFNet [35] and others perform well on small point clouds but not that well on LiDAR point clouds. Besides, the **2D images** methods of PolarNet-series [56,58] and others [46,47,49,4,32] project the 3D point cloud as 2D images in Bird's-Eye-View (BEV) and range view, achieving the most efficient LiDAR semantic segmentation with the mature 2D Convolutional Neural Networks (CNNs) on GPUs. But the 3D-to-2D projection inevitably suffers from the loss of the 3D structure information of objects, resulting in unsatisfactory segmentation performance. The recent **3D voxel** methods Cylinder3D [59] and SPVNAS [39] yield top performances by designing deeper 3D sparse CNNs to explicitly explore the 3D structure information in the cylindrical [59] or cartesian [39] coordinate

Table 1. Comparing with other distillation related semantic segmentation methods.

Task	2D Semantic Segmantation	3D Semantic Segmantation	
Method	He *et al.*[16], SKD[30], IFVD[45], CSC[33], An *et al.* [2]	PSD [55]	Ours
Purpose	Model compression	Weakly supervised learning	Achieving higher performance
Cumbersome Teacher	√	×	×
Parameters of Teacher Model	Pretrained & Fixed	Independently trained	Updated from student itself
Equip Teacher with TTA	×	×	√

system. The 3D sparse convolutions are performed only on the non-empty voxels with acceptable memory consumption and significant computational acceleration [9,39,13]. As a fundamental module in 3D voxel methods, the VFE implemented by the average operation [12,51,39] or a PointNet [34,24] ignore the relationship of local points in voxel at the point cloud encoding level.

2.2 Semantic Segmentation with Knowledge Distillation

Knowledge distillation [19] is recently researched for compressing a cumbersome teacher as a compact student model in 2D semantic segmentation. An additional auto-encoder is employed to translate the high-level features for distillation in a latent domain by He *et al.*[16]. SKD [30] proposes to structurally transfer the pairwise relation on features, pixel-wise outputs, and holistic representations to the student. Unlike the dense distillation, IFVD [45] computes the intra-class feature variation to guide the student to mimic the class-wise prototype. The long-range dependence, spatial and channel correlations also are extracted as the knowledge for distillation by An *et al.* [2] and CSC [33], respectively. The above methods make efforts to adapt the feature map for alleviating the differences of feature distributions between the heterogeneous teacher model and student model, but the distribution gap still exists in such two heavy and lightweight models. Instead, we aim at achieving higher LiDAR semantic segmentation performance by performing self-distillation without any cumbersome teacher.

In LiDAR semantic segmentation, only the recently published weakly-supervised PSD [55] shares the closest distillation manner in terms of using two model instances of the same network architecture. However, the differences between PSD and ours still appear in Tab. 1 as follows. (**i**) Different model training strategies. Given a point cloud with only a tiny fraction of point-wise labels provided, PSD has two branches of disturbed and undisturbed input point clouds, where the undisturbed branch is termed as the teacher for providing robust feature representation as the knowledge to guide the disturbed branch termed as the student. The teacher and student are independently trained and only the consistency regularization effectiveness can aid the training. Instead, our teacher and student models are designed to be jointly evolved, where the more privileged guidance from the teacher can be transferred to the student while the student continuously updates the teacher with better network parameters learned by itself. (**ii**) Our teacher model is equipped with the proposed TTA strategy for the quality grantee of the distillation guidance, while no such quality grantee is considered in PSD. (**iii**) Different purposes. PSD mainly focuses on

indoor point clouds and relies on consistency regularization to achieve weakly-supervised segmentation for reducing the labeling burden on an input point cloud, while our self-distillation aims to achieve more robust LiDAR semantic segmentation with stronger performance in large-scale driving scenarios.

2.3 Test-time Augmentation

Since few efforts have been made on the TTA for LiDAR semantic segmentation, we mainly review the TTA applied in image-based 2D vision works [60,20,7,31,22]. The pixel-wise segmentation and object-wise axis-aligned boxes of the flipped and scaled input images can be averaged easily by the corresponding inverse transformation in 2D segmentation [60,48,20] and detection [7,54]. But the translation and rotation can cause some content pixels to overflow the image boundaries and coordinate quantization errors, resulting in unacceptable misalignment among transformed images. Image recognition tasks with image-wise classifications can additionally employ the rotation and translation transformations to perform TTA. For the point cloud with unstructured and unordered points, the point-wise semantic predictions can be easily averaged across the input variants from different types of transformations, as long as the order of points is unchanged. Some greedy [31] and learnable [22] policies search the combination of different TTA input variants in image recognition with carefully tuned search parameters and loss functions. However, as a pioneer of introducing the TTA into the LiDAR segmentation field, we employ a compound transformation based on the four types of transformations in TTA. It is simple yet effective, demonstrating the beauty of science.

3 Method

This section describes the proposed LiDAR semantic segmentation method. Since the teacher and student models in our method are related to the TransVFE and TTA, we begin with the overview of our network architecture in Sec. 3.1 followed by the description of a novel TTA strategy for LiDAR segmentation in Sec 3.2. The proposed self-distillation framework and the loss function are presented in Sec. 3.3 and Sec. 3.4.

3.1 Network

TransVFE. Let $\left\{\left(x_i, f_i^{\text{in}}\right) : i = 1, \cdots, N_{\text{P}}\right\}$ denote a input point cloud X within the range of $[x_{\text{Min}}, x_{\text{Max}}]$, where $x \in \mathbb{R}^{3\times 1}$ represents the 3D point coordinates, and $f^{\text{in}} \in \mathbb{R}^{C_{\text{in}}\times 1}$ represents the C_{in}-dimensional point-wise input features such as coordinates x and reflection intensity r. We rearrange X as the structural non-empty voxels by voxelization. The point coordinates x are discretized into integer values $\bar{v} = \left\lfloor \frac{x - x_{\text{Min}}}{d} \right\rfloor$ by a step d. The points with the same $\bar{v}$ are gathered into a voxel and denoted as $\mathcal{N} = \{(\bar{v}_i, f_i^{\text{in}}) : i = 1, \cdots, T\}$. The unique values of

$\bar{v}$ are set as the voxel indices v. The VFE is defined on the $\mathcal{N}$ to encode the all the local point features $F^{\text{in}} \in \mathbb{R}^{C_{\text{in}} \times T}$ in $\mathcal{N}$ as the voxel-wise feature.

Given a voxel with T local points inside it, we use a VFE with Transfomer [43], termed as TransVFE, to model the relationship of local points in voxel via Multi-Head Self-Attention $MHSA(\cdot)$ and Feed-Forward Network $FFN(\cdot)$:

$$F' = Norm(W^{\text{in}} F^{\text{in}}), \tag{1}$$

$$F^{\text{att}} = F' + MHSA(F'), \tag{2}$$

$$F^{\text{trans}} = F^{\text{att}} + FFN(F^{\text{att}}), \tag{3}$$

where the input features F^{in} are initially projected to C_{trans} dimension by a linear layer with learnable parameters W^{in}. $Norm$ indicates LayerNorm operation. The $MHSA(\cdot)$ can be decomposed into N_{H} heads with features H as

$$MHSA(F') = \left[H_j : j = 1, \cdots N_{\text{H}}\right], \tag{4}$$

$$H_j^T = Softmax(\frac{Q_j^T K_j}{\sqrt{C^{\text{trans}}/N_{\text{H}}}}) V_j^T, \tag{5}$$

$$Q_j, K_j, V_j = W_j^{\text{q}} F', W_j^{\text{k}} F', W_j^{\text{v}} F'. \tag{6}$$

The $W_j^{\text{q}}, W_j^{\text{k}}, W_j^{\text{v}}$ are the learnable parameters in linear layers for feature projection, and each local point in F' interacts with others as defined in Eq. 5. After stacking three blocks of Transformer, we follow [14] to employ a max-pooling operation along the point-axis to get the expressive voxel-wise feature $f^{\text{trans}} \in \mathbb{R}^{C_{\text{trans}} \times 1}$ from $F^{\text{trans}} \in \mathbb{R}^{C_{\text{trans}} \times T}$, and apply another linear layer to compress $f^{\text{trans}} \in \mathbb{R}^{C_{\text{trans}} \times 1}$ to $f^{\text{vfe}} \in \mathbb{R}^{C_0 \times 1}$ with less channels for saving computation in the subsequent network. Thus, our TransVFE explicitly encodes local points into the voxel features as $\mathcal{V} = \left\{\left(v_i, f_i^{\text{vfe}}\right) : i = 1, \cdots, N_{\text{V}}\right\}$, where each item (v, f^{vfe}) indicates the non-empty voxel located at v with the corresponding voxel-wise feature f^{vfe}.

3D Sparse U-Net. Without loss of generalization, we leverage the universal U-Net from [37] as our backbone, which is implemented with the computationally efficient 3D sparse convolutions [13,51] following [37,59]. We rearrange the voxels in $\mathcal{V}$ as a sparse tensor for feature learning with the 3D sparse convolutional blocks. The details are in the supplementary material.

Voxel Head and Point Head. Voxelization makes feature extraction efficient [25,27], but LiDAR segmentation requires point-wise outputs. The existing methods [56,59] reverse the pair-wise mapping from the voxel-wise outputs back to the points, and associate all the points in the same voxel with the same category. This inevitably has the risk of classifying points of different categories into the same category, especially for object boundaries. Thus, we devoxelize the voxel-wise features into point-wise features to predict point-wise output $\hat{Y}_{\text{P}}$ following [39]. For each point, we interpolate the point feature from its K nearest neighboring voxels [26]. We set the K to 3 for computation efficiency.

We construct segmentation heads composed of fully connected layers on voxel-wise and point-wise features, respectively. The voxel-wise prediction $\hat{Y}_{\text{V}}$ is

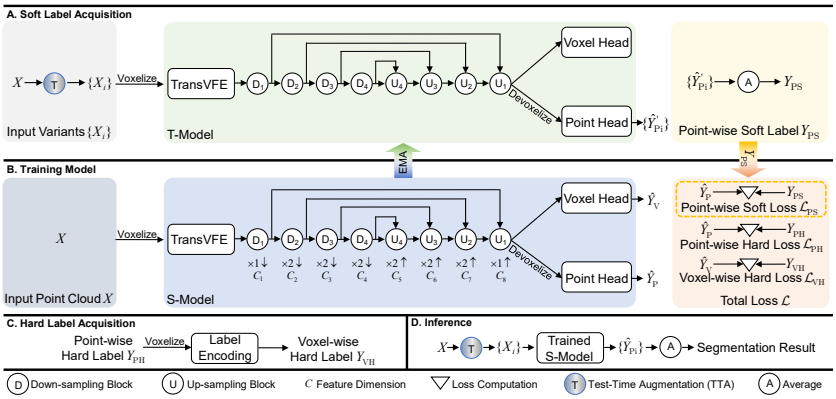

Fig. 2. Overview of self-distillation for robust LiDAR semantic segmentation.

for auxiliary supervision only, while we apply an *argmax* function to the point-wise prediction $\hat{Y}_{\mathrm{P}}$ for obtaining the predicted classes as the segmentation results. To avoid ambiguity in voxel-wise supervision, we ignore the voxels containing points of multiple classes in the auxiliary voxel-wise loss.

3.2 Test-time Augmentation for LiDAR Semantic Segmentation

We choose four standard types of data augmentation transformations with the common hyper-parameters widely used in the training phase of a LiDAR segmentation network [59,56,39,8]: global scaling (τ_{scale}) with a random scaling factor in $[0.95, 1.05]$, random flipping (τ_{flip}) along the X, Y axis, global rotation (τ_{rot}) around the Z axis with a random angle in $\left[-\frac{\pi}{4}, +\frac{\pi}{4}\right]$, global translation ($\tau_{\mathrm{tran}}$) with a random vector $(\Delta x, \Delta y, \Delta z)$ sampled from a Gaussian distribution with mean zero and the standard deviation 0.5. The random value controls the magnitude of applying each transformation.

We can apply the above transformations in test-time and assemble the predictions from the input and its augmented samples to boost the robustness of LiDAR segmentation model. Given a model with weights θ', the assembled prediction $\hat{Y}'$ from the naive version of point cloud TTA can be formulated as $\hat{Y}' = \frac{1}{M}\sum_{i=1}^{M} \theta'(X_i)$, where $\mathcal{X} = \{X_i\}$ is the set of input variants and M denotes the number of samples in $\mathcal{X}$. The naive strategy is to generate the $\mathcal{X}$ as a set of $\{X, X_{\mathrm{scale}}, X_{\mathrm{flip}}, X_{\mathrm{rot}}, X_{\mathrm{tran}}\}$, which consist of the identical input X and the augmented input samples generated by the four types of transformations. Although we can apply each type of transformation multiple times with different magnitudes to generate more input variants, the flexibility of data augmentation is still within the individual transformation.

Instead, we develop a more effective strategy for increasing the diversity and flexibility of the augmented samples. We define a compound transformation $\tau_{\mathrm{comp}}(X)$ as $\tau_{\mathrm{tran}}(\tau_{\mathrm{rot}}(\tau_{\mathrm{flip}}(\tau_{\mathrm{scale}}(X))))$, which combines individual transformations. The magnitude of τ_{comp} can be independently controlled by each individual

transformation. The X is augmented into a set of $\mathcal{X}^* = \{X, X_{\text{comp},j}\}$ with different magnitudes, where "j" indexes the augmented samples in the set. The diversity and the flexibility in the input point cloud variants help reduce the uncertainty with the assembled prediction from multiple input point cloud variants in the inference stage.

3.3 Self-distillation

For training a robust model with stronger performance without a cumbersome teacher model (T-Model) providing the distillation guidance, we propose to perform self-distillation on two model instances of the same network architecture.

Teacher Model Configuration. The quality of the guidance from the T-Model is critical to training a good student model (S-Model) [41,16,29,57]. Hence, we decide the configuration of the T-Model in two terms of the performance boost and the network parameters updating. Since TTA always improves the performance of a model robustly, it is feasible to equip the T-Model with TTA when inferring predictions as distillation guidance. But it is barely used for distillation in 2D semantic segmentation, and we show that our TTA strategy can be involved in the T-Model configuration for better self-distillation. With the aid of TTA, the parameters of the T-Model can naively be copied from the current S-Model or loaded from the pretrained parameters. Besides these, updating T-Model with the successive network parameters of S-Model is widely used to provide the predictions as the reliable soft labels on unlabeled images in the semi-supervised 2D vision works [41,29,40]. Inspired by this, we can also update the T-Model with weights θ' as the Exponential Moving Average (EMA) of the S-Model with weights θ in successive training step t [41], which can be formulated as $\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t$, where $\alpha = \min(1-\frac{1}{t}, 0.999)$. The smoothing coefficient α makes the T-Model a temporal assembly of S-Models from different training steps, so that the T-Model is more likely to have better soft labels to regularize the learning process [41].

As the combination of different manners in updating T-Model and TTA, we investigate five strategies for self-distillation. Specifically, we describe all the five optional configurations for T-Model as follows. (**i**) We copy the T-Model from the current S-Model at each training step t, but its prediction is improved by our TTA[1]. (**ii**) The T-Model is a pre-trained model, then frozen when training the S-Model following [19,44]. (**iii**) We pre-train and freeze the T-Model, and boost it with TTA in distillation[1]. (**iv**) The T-Model is the EMA of the S-Model following [41,57]. (**v**) The T-Model is the EMA of the S-Model with our TTA[1].

We finally employ configuration (**v**) to excavate instructive knowledge in our self-distillation with the most significant improvements achieved.

Training with Soft Labels. Since there are different numbers of non-empty voxels but the same number of points between the augmented point clouds in TTA, we only consider the point-wise outputs for self-distillation. As shown at the 2^{nd} row of Fig. 2, we use the generated soft label Y_{PS} to

[1] The new soft label acquisition strategies proposed in this paper.

compute an additional soft loss term $\mathcal{L}_{\mathrm{PS}}$ between the soft label Y_{PS} and the S-Model prediction $\hat{Y_{\mathrm{P}}}$ as a useful knowledge transfer for helping the S-Model learn better. It is a self-distillation procedure between the T-Model from the EMA of the S-Model with the proposed TTA and the S-Model in our case where the T-Model and the S-Model are with the same network architecture, excavating instructive knowledge from the S-Model itself. After self-distillation, only the S-Model is used for inference, which avoids the computation consumption from the T-Model. In the inference stage, we can also apply TTA to the S-Model to get the final output for further performance improvement. More discussions on the differences of other knowledge distillation related semantic segmentation methods and ours can be retrieved in Sec. 2.2 and Tab. 1.

3.4 Loss Function

We can train our model individually with voxel-wise hard label Y_{VH} and point-wise hard label Y_{PH} only, or jointly with the additional point-wise soft label Y_{PS} for achieving self-distillation. When training without self-distillation, the total loss $\mathcal{L}$ is the sum of the hard label loss $\mathcal{L}_{\mathrm{VH}}$ on voxel-wise prediction and the hard label loss $\mathcal{L}_{\mathrm{PH}}$ on point-wise prediction as $\mathcal{L} = \mathcal{L}_{\mathrm{VH}} + \mathcal{L}_{\mathrm{PH}}$. Let $\alpha \in \{\mathrm{V}, \mathrm{P}\}$ denote the voxel-wise and point-wise, respectively. The hard label loss term $\mathcal{L}_{\alpha\mathrm{H}}$ in $\mathcal{L}$ is a combination of the commonly used cross-entropy loss $\mathcal{L}_{\mathrm{ce}}$ and lovasz-softmax loss $\mathcal{L}_{\mathrm{lovasz}}$ [5] as $\mathcal{L}_{\alpha\mathrm{H}} = \mathcal{L}_{\mathrm{ce}}(\hat{Y_{\alpha}}, Y_{\alpha\mathrm{H}}) + \mathcal{L}_{\mathrm{lovasz}}(\hat{Y_{\alpha}}, Y_{\alpha\mathrm{H}})$.

When training the full framework of self-distillation, we formulate the overall loss $\mathcal{L}$ as $\mathcal{L}_{\mathrm{VH}} + \mathcal{L}_{\mathrm{PH}} + \mathcal{L}_{\mathrm{PS}}$. Inspired by knowledge distillation [19,44] in image recognition, the soft loss $\mathcal{L}_{\mathrm{PS}}$ is implemented with the cross-entropy loss between the point-wise prediction $\hat{Y_{\mathrm{P}}}$ and the point-wise soft label Y_{PS} as $\gamma\mathcal{L}_{\mathrm{ce}}(\hat{Y_{\mathrm{P}}}, Y_{\mathrm{PS}})$. When the mIoU of the soft label Y_{PS} is larger, the dynamic coefficient γ of $\exp(\mathrm{mIoU}(Y_{\mathrm{PS}}, Y_{\mathrm{PH}}))$ assigns a larger weight value to the soft loss $\mathcal{L}_{\mathrm{PS}}$.

4 Experiments

4.1 Dataset and Metric

SemanticKITTI Dataset. SemanticKITTI [4] dataset for LiDAR semantic segmentation is collected in Germany with the Velodyne-HDL64E LiDAR. It contains 22 sequences: sequences 00 to 10 (excluding 08) containing 19,130 point clouds as the training set, sequence 08 containing 4,071 point clouds as the validation set, and the remaining sequences 11 to 21 with 20,351 point clouds as the testing set. For the setting of single scan input, the official evaluation protocol merges classes with different motion states and ignores classes with only a few points, so 19 valid classes are preserved from 28 annotated classes.

nuScenes Dataset. nuScenes [6] dataset for LiDAR semantic segmentation is collected in different areas of Boston and Singapore with the Velodyne-HDL32E LiDAR. It officially splits 28,130 point clouds for training, 6,019 point clouds for validation. After merging similar classes and ignoring rare classes, 16 valid classes remain for semantic segmentation evaluation.

Table 2. Performance comparison on IoU (%) between our method and state-of-the-art LiDAR Segmentation methods on SemanticKITTI testing set [4]. The methods are divided into the branches of point, 2D image, and 3D voxel, according to their main point cloud representations (PC Rep.). Bold and underlined indicate the best and the second best results, respectively.

PC Rep.	Methods	**mIoU**	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign
Point	PointASNL [50]	46.8	87.4	0.0	25.1	39.0	29.2	34.2	57.6	0.0	87.4	24.3	74.3	1.8	83.1	43.9	84.1	52.2	70.6	57.8	36.9
	RandLA-Net [21]	53.9	94.2	26.0	25.8	40.1	38.9	49.2	48.2	7.2	90.7	60.3	73.7	20.4	86.9	56.3	81.4	61.3	66.8	49.2	47.7
	KPConv [42]	58.8	96.0	32.0	42.5	33.4	44.3	61.5	61.6	11.8	88.8	61.3	72.7	31.6	90.5	64.2	84.8	69.2	69.1	56.4	47.4
	BAAF-Net [35]	59.9	95.4	31.8	35.5	48.7	46.7	49.5	55.7	**53.0**	90.9	62.2	74.4	23.6	89.8	60.8	82.7	63.4	67.9	53.7	52.0
2D Image	RangeNet++ [32]	52.2	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
	PolarNet [56]	54.3	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5
	SqueezeSegv3 [49]	55.9	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
	SalsaNext [10]	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
	KPRNet [23]	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
3D Voxel	Darknet53 [4]	49.9	86.4	24.5	32.7	25.5	22.6	36.2	33.6	4.7	91.8	64.8	74.6	27.9	84.1	55.0	78.3	50.1	64.0	38.9	52.2
	MinkNet42 [9]	54.3	94.3	23.1	26.2	26.1	36.7	43.1	36.4	7.9	91.1	63.8	69.7	29.3	92.7	57.1	83.7	68.4	64.7	57.3	60.1
	3D-MiniNet [1]	55.8	90.5	42.3	42.1	28.5	29.4	47.8	44.1	14.5	91.6	64.2	74.5	25.4	89.4	60.8	82.8	60.8	66.7	48.0	56.6
	FusionNet [53]	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
	SPVNAS-Lite [39]	63.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	SPVNAS [39]	66.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Cylinder3D [59]	67.8	97.1	**67.6**	64.0	**59.0**	58.6	73.9	67.9	36.0	91.4	65.1	75.5	32.3	91.0	66.5	85.4	71.8	68.5	62.6	65.6
	$(AF)^2$-S3Net [8]	69.7	94.5	65.4	**86.8**	39.2	41.1	**80.7**	**80.4**	**74.3**	91.3	68.8	72.5	**53.5**	87.9	63.2	70.2	68.5	53.7	61.5	**71.0**
	Ours (w.o. TTA)	68.0	97.0	54.4	48.1	55.9	61.6	65.5	69.4	51.1	91.3	67.0	77.0	35.6	92.2	67.8	84.9	72.2	69.3	63.4	68.0
	Ours (w. TTA)	**70.4**	**97.4**	58.7	54.2	54.9	**65.2**	70.2	74.4	52.2	90.9	69.4	76.7	41.9	**93.2**	**71.1**	**86.1**	**74.3**	71.1	**65.4**	70.6

Table 3. Performance comparison on IoU (%) between our method and other LiDAR Segmentation methods on nuScenes [6] validation set. Bold and underlined indicate the best and the second best results, respectively.

Methods	**mIoU**	barrier	bicycle	bus	car	construction	motorcycle	pedestrian	traffic-cone	trailer	truck	driveable	other	sidewalk	terrain	manmade	vegetation
$(AF)^2$-S3Net [8]	62.2	60.3	12.6	82.3	80.0	20.1	62.0	59.0	49.0	42.2	67.4	94.2	68.0	64.1	68.6	82.9	82.4
RangeNet++ [32]	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 [56]	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 [10]	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
AMVNet [28]	76.1	**79.8**	32.4	82.2	86.4	**62.5**	81.9	75.3	**72.3**	**83.5**	65.1	**97.4**	67.0	**78.8**	74.6	**90.8**	**87.9**
Cylinder3D [59]	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
Ours (w.o. TTA)	77.7	77.5	49.4	93.9	92.5	54.9	86.7	80.1	67.8	65.7	86.0	96.4	74.0	74.9	74.5	86.0	82.8
Ours (w. TTA)	**78.7**	78.2	**52.8**	**94.5**	93.1	54.5	**88.1**	**82.2**	69.4	67.3	**86.6**	96.4	**74.5**	75.2	75.3	87.1	84.1

Evaluation Metric. We adopt the mean Intersection-over-Union (mIoU) over all classes as the evaluation metric defined in [4,6], which can be formulated as $\text{mIoU} = \frac{1}{C}\sum_{c=1}^{C}\frac{\text{TP}_c}{\text{TP}_c+\text{FP}_c+\text{FN}_c}$, where TP_c, FP_c, FN_c correspond to the number of true positive, false positive, and false negative predictions for the c-th class in C classes.

4.2 Implementation Details

We configure TransVFE with the number of multi-head N_{H} of 4, the hidden feature dimension C_{trans} of 64, the compressed feature dimension C_0 of 16, which are mentioned in Eq. 5. In the 3D sparse U-Net, the feature dimensions C_1 - C_8 are set as 32, 64, 128, 128, 128, 128, 64, 32, respectively. We only use the point-wise outputs as the inferred segmentation results, while the voxel-wise outputs

Table 4. Latency analysis on SemanticKITTI testing set.

Methods	RandLA-Net[21]	SqueezeSegV3 [49]	SPVNAS-Lite[39]	Ours
Latency (s)	0.416	0.113	0.150	0.148
mIoU (%)	53.9	55.9	63.7	68.0

are used in the training phase only. More details on point cloud voxelization and model training are provided in the supplementary material.

4.3 Evaluation

Results on SemanticKITTI. Tab. 2 presents the performance of our method without (w.o.) and with (w.) TTA after self-distillation against other methods on the testing set of SemanticKITTI dataset. Our full pipeline achieves the best result on mIoU. The efficient 3D convolution facilitates the exploration of 3D object structures, so 3D voxel methods achieve better performance than point and 2D image methods. Our method achieves the largest number of Top-1 and Top-2 results (8/20 & 13/20) among the overall 19 object classes and mIoU, which shows the reliable and robust LiDAR segmentation performance of various object categories. Qualitative visualizations are in supplementary material.

Results on nuScenes. We also evaluate our method on the nuScenes dataset. In Tab. 3, our method outperforms the Cylinder3D [59] and AMVNet [28] in mIoU of 2.6%. In terms of all metrics, our full method still achieves the 7/17 Top-1 results and the largest number of Top-2 results (12/17). Considering the different collection environments and collection LiDAR sensors between the SemanticKITTI and nuScenes, our method generalizes well in different datasets.

Table 5. Effects of VFE. All models are trained without self-distillation and inferred without TTA. "Improv." is the improvement compared to average-based model.

VFE Module	SemanticKITTI		nuScenes	
	mIoU	Improv.	mIoU	Improv.
Average	64.90	-	75.94	-
PointNet	65.08	+0.18	75.96	+0.02
TransVFE (Ours)	65.47	+0.57	76.42	+0.48

Latency. Tab. 4 shows the latency analysis under the same experimental settings with the same machine. Our method achieves much higher mIoU with close latency against SqueezeSegV3 [49] and SPVNAS-Lite [39].

4.4 Ablation study

We conduct extensive ablation experiments on SemanticKITTI and nuScenes datasets following the official evaluation protocol on the validation sets. Since the T-Model in our method is related to the TransVFE and TTA, Tab. 5 and 6 first demonstrate the effects of the TransVFE and TTA without the self-distillation followed by the further analyses on our full framework in Tab. 7 and Fig. 3.

Table 6. Effects of the different augmentation strategies for TTA. M and j are the number and index of input variants in $\mathcal{X}$, respectively. "Row" and "Improv." denote the row index and the mIoU improvements compared with the baseline in the first row.

Row	$\mathcal{X}$	M	SemanticKITTI		nuScenes	
			mIoU	Improv.	mIoU	Improv.
1	$\{X\}$	1	65.47	-	76.42	-
2	$\{X, X_{\text{scale},j} : j = 1, \cdots, 4\}$	5	66.32	+0.85	77.13	+0.71
3	$\{X, X_{\text{flip},j} : j = 1, \cdots, 4\}$	5	66.49	+1.02	76.87	+0.45
4	$\{X, X_{\text{rot},j} : j = 1, \cdots, 4\}$	5	66.53	+1.06	77.04	+0.62
5	$\{X, X_{\text{tran},j} : j = 1, \cdots, 4\}$	5	66.57	+1.10	77.06	+0.64
6	$\{X, X_{\text{scale}}, X_{\text{flip}}, X_{\text{rot}}, X_{\text{tran}}\}$	5	66.64	+1.17	77.27	+0.85
7	$\{X, X_{\text{comp},j} : j = 1, \cdots, 4\}$	5	67.75	+2.28	77.54	+1.12
8	$\{X, X_{\text{comp},j} : j = 1, \cdots, 5\}$	6	67.97	+2.50	77.70	+1.28
9	$\{X, X_{\text{comp},j} : j = 1, \cdots, 6\}$	7	67.82	+2.35	77.67	+1.25
10	$\{X, X_{\text{comp},j} : j = 1, \cdots, 7\}$	8	68.04	+2.57	77.72	+1.30

Table 7. Effects of self-distillation. "Exp.", "EMA", and "Improv." indicate experiment tags, Exponential Moving Average, and Improvements compared to the baseline in Exp. A. The setting of the baseline (Exp. A) is mentioned in **TransVFE**. Note that TTA is only used in self-distillation for this experiment setting.

Exp.	S-Model Initialization	T-Model Configuration		SemanticKITTI		nuScenes	
		T-Model Parameter	TTA	mIoU	Improv.	mIoU	Improv.
A	Scratch	-	-	65.47	-	76.42	-
B		Copy	√	65.61	+0.14	76.62	+0.20
C		Pre-trained & fixed	×	65.89	+0.42	76.83	+0.41
D		Pre-trained & fixed	√	66.09	+0.62	76.98	+0.56
E		EMA	×	66.18	+0.71	76.93	+0.51
F		EMA	√	66.64	+1.17	77.31	+0.89
G	Pre-trained	Pre-trained & fixed	√	66.81	+1.34	77.50	+1.08
H		EMA	√	**67.11**	**+1.64**	**77.74**	**+1.32**

TransVFE. Tab. 5 shows that our TransVFE of modeling the local relationship can improve the performance compared with the VFE modules implemented by average operation in [51,39] and PointNet in [56,59] in terms of point cloud encoding. Thus, we use this model shown in the last row of Tab. 5 as the **baseline model** for the ablation studies of TTA and self-distillation.

TTA strategy. In Tab. 6, the improvements in rows 2 - 5 first demonstrate that all the four types of transformations can be taken into account for TTA. Rows 6 and 2 - 5 then show that combining the four individual TTA transformations is slightly better than reusing any individual transformations multiple times, when an input variant set $\mathcal{X}$ with a fixed capacity ($M = 5$) is given. Reusing compound transform in row 7 achieves much more significant improvements than combining the four individual TTA transformations in row 6 on both datasets, indicating that our compound transform is simple yet effective enough. The last three rows validate that the performance can be further improved with the increased capacity of $\mathcal{X}$ but gradually saturated after $M = 6$. Thus, we set M to 6 in this work.

Self-distillation. The Exp. B-F in Tab. 7 inspect the effectiveness of the five optional T-Model configurations as described in Sec. 3.3. The mIoU of Exp. B, D, and F is consistently higher than the mIoU of Exp. A, C, and E, showing that applying the TTA to acquire soft labels with higher quality is effective for better self-distillation. Among different manners of updating the T-Model network parameters, the EMA shows the most significant improvements. Especially, the Exp. F that applies both the EMA and TTA achieves the best mIoU on both datasets. The dual assemblies of the S-Model network parameters and the T-Model predictions guarantee the quality of the soft labels thus enhancing the self-distillation. Finally, the last two rows suggest that we can initialize the parameters of both the S-Model and T-Model from the pre-trained baseline model with higher overall performances. The Exp. H that applies both the EMA

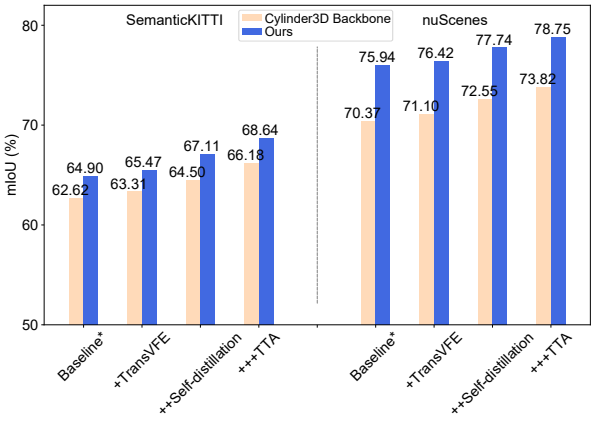

Fig. 3. Continuous improvements of our method and **Cylinder3D backbone** [59] in our framework. Baseline* is the model with average-based VFE.

and our TTA with pre-trained model initialization yields the best mIoU on two datasets. We thus use this self-distillation strategy in the proposed framework.

Backbone. The proposed method is agnostic to the backbone architectures under voxel representation of LiDAR point cloud. Since the codes of Cylinder3D [59] and SPVNAS [39] are not completely released yet, we choose our backbone as a general U-Net from the LiDAR perception work [37]. The experiments in Fig. 3 validate that our framework can also achieve significant improvements when using Cylinder3D backbone. Therefore, all the proposed components can be instantiated with other backbones for LiDAR semantic segmentation.

5 Conclusions

In this paper, we propose a novel self-distillation framework and firstly use it for robust LiDAR semantic segmentation in autonomous driving. The proposed framework enables the self-distillation from a teacher model instance to a student model instance with the same network architecture for jointly learning and evolving in training. Our method achieves state-of-the-art performance on SemanticKITTI and nuScenes datasets, demonstrating the effectiveness of the overall self-distillation framework. Extensive experiments validate that all the proposed components are effective with significant improvements and general to be instantiated with different backbones for LiDAR semantic segmentation.

Acknowledgement

This work was supported by the National Key Research and Development Program of China (2018YFE0183900) and YUNJI Technology Co. Ltd.

References

1. Alonso, I., Riazuelo, L., Montesano, L., Murillo, A.C.: 3D-MiniNet: Learning a 2D representation from point clouds for fast and efficient 3D LiDAR semantic segmentation. IEEE Robotics Autom. Lett. **5**(4), 5432–5439 (2020)
2. An, S., Liao, Q., Lu, Z., Xue, J.H.: Efficient semantic segmentation via self-attention and self-distillation. IEEE Transactions on Intelligent Transportation Systems pp. 1–11 (2022)
3. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I.K., Fischer, M., Savarese, S.: 3D semantic parsing of large-scale indoor spaces. In: CVPR. pp. 1534–1543 (2016)
4. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In: ICCV. pp. 9296–9306 (2019)
5. Berman, M., Triki, A.R., Blaschko, M.B.: The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: CVPR. pp. 4413–4421 (2018)
6. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11618–11628 (2020)
7. Chen, Y., Zhang, Z., Cao, Y., Wang, L., Lin, S., Hu, H.: RepPoints v2: Verification meets regression for object detection. In: NeurIPS (2020)
8. Cheng, R., Razani, R., Taghavi, E., Li, E., Liu, B.: (AF)2-S3Net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In: CVPR. pp. 12547–12556 (2021)
9. Choy, C.B., Gwak, J., Savarese, S.: 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In: CVPR. pp. 3075–3084 (2019)
10. Cortinhal, T., Tzelepis, G., Aksoy, E.E.: SalsaNext: fast, uncertainty-aware semantic segmentation of LiDAR point clouds for autonomous driving. arXiv preprint arXiv:2003.03653 (2020)
11. Cubuk, E.D., Zoph, B., Mané, D., Vasudevan, V., Le, Q.V.: AutoAugment: Learning augmentation strategies from data. In: CVPR. pp. 113–123 (2019)
12. Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H.: Voxel R-CNN: towards high performance voxel-based 3D object detection. In: AAAI. pp. 1201–1209 (2021)
13. Graham, B., Engelcke, M., van der Maaten, L.: 3D semantic segmentation with submanifold sparse convolutional networks. In: CVPR. pp. 9224–9232 (2018)
14. Guo, M., Cai, J., Liu, Z., Mu, T., Martin, R.R., Hu, S.: PCT: point cloud transformer. Computational Visual Media **7**(2), 187–199 (2021)
15. Hataya, R., Zdenek, J., Yoshizoe, K., Nakayama, H.: Faster AutoAugment: Learning augmentation strategies using backpropagation. In: ECCV. vol. 12370, pp. 1–16 (2020)
16. He, T., Shen, C., Tian, Z., Gong, D., Sun, C., Yan, Y.: Knowledge adaptation for efficient semantic segmentation. In: CVPR. pp. 578–587 (2019)
17. Hendrycks, D., Dietterich, T.G.: Benchmarking neural network robustness to common corruptions and perturbations. In: ICLR (2019)
18. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: AugMix: A simple data processing method to improve robustness and uncertainty. In: ICLR (2020)
19. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR **abs/1503.02531** (2015), `http://arxiv.org/abs/1503.02531`

20. Hu, H., Cui, J., Wang, L.: Region-aware contrastive learning for semantic segmentation. In: ICCV. pp. 16271–16281 (2021)
21. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: RandLA-Net: Efficient semantic segmentation of large-scale point clouds. In: CVPR. pp. 11105–11114 (2020)
22. Kim, I., Kim, Y., Kim, S.: Learning loss for test-time augmentation. In: NeurIPS (2020)
23. Kochanov, D., Nejadasl, F.K., Booij, O.: KPRNet: Improving projection-based LiDAR semantic segmentation. In: ECCV (2020)
24. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: PointPillars: Fast encoders for object detection from point clouds. In: CVPR. pp. 12689–12697 (2019)
25. Li, J., Dai, H., Shao, L., Ding, Y.: Anchor-free 3D single stage detector with mask-guided attention for point cloud. In: ACM MM. pp. 553–562 (2021)
26. Li, J., Dai, H., Shao, L., Ding, Y.: From voxel to point: IoU-guided 3D object detection for point cloud with voxel-to-point decoder. In: ACM MM (2021)
27. Li, J., Sun, Y., Luo, S., Zhu, Z., Dai, H., Krylov, A.S., Ding, Y., Shao, L.: P2V-RCNN: point to voxel feature learning for 3D object detection from point clouds. IEEE Access **9**, 98249–98260 (2021)
28. Liong, V.E., Nguyen, T.N.T., Widjaja, S., Sharma, D., Chong, Z.J.: AMVNet: Assertion-based multi-view fusion network for LiDAR semantic segmentation. CoRR **abs/2012.04934** (2020), `https://arxiv.org/abs/2012.04934`
29. Liu, Y., Ma, C., He, Z., Kuo, C., Chen, K., Zhang, P., Wu, B., Kira, Z., Vajda, P.: Unbiased teacher for semi-supervised object detection. In: ICLR (2021)
30. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured knowledge distillation for semantic segmentation. In: CVPR. pp. 2604–2613 (2019)
31. Lyzhov, A., Molchanova, Y., Ashukha, A., Molchanov, D., Vetrov, D.P.: Greedy policy search: A simple baseline for learnable test-time augmentation. In: Adams, R.P., Gogate, V. (eds.) UAI. vol. 124, pp. 1308–1317 (2020)
32. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: RangeNet++: Fast and accurate LiDAR semantic segmentation. In: IROS. pp. 4213–4220 (2019)
33. Park, S., Heo, Y.S.: Knowledge distillation for semantic segmentation using channel and spatial correlations and adaptive cross entropy. Sensors **20**(16), 4616 (2020)
34. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: CVPR. pp. 77–85 (2017)
35. Qiu, S., Anwar, S., Barnes, N.: Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In: CVPR. pp. 1757–1767 (2021)
36. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., III, W.M.W., Frangi, A.F. (eds.) MICCAI. vol. 9351, pp. 234–241 (2015)
37. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. IEEE TPAMI **43**(8), 2647–2664 (2021)
38. Taghanaki, S.A., Luo, J., Zhang, R., Wang, Y., Jayaraman, P.K., Jatavallabhula, K.M.: Robustpointset: A dataset for benchmarking robustness of point cloud classifiers. In: ICLR (2021)
39. Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S.: Searching efficient 3D architectures with sparse point-voxel convolution. In: ECCV. vol. 12373, pp. 685–702 (2020)
40. Tang, Y., Chen, W., Luo, Y., Zhang, Y.: Humble teachers teach better students for semi-supervised object detection. In: CVPR. pp. 3132–3141 (2021)

41. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: NeurIPS. pp. 1195–1204 (2017)
42. Thomas, H., Qi, C.R., Deschaud, J., Marcotegui, B., Goulette, F., Guibas, L.J.: KPConv: Flexible and deformable convolution for point clouds. In: ICCV. pp. 6410–6419 (2019)
43. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 5998–6008 (2017)
44. Wang, H., Zhao, H., Li, X., Tan, X.: Progressive blockwise knowledge distillation for neural network acceleration. In: IJCAI. pp. 2769–2775 (2018)
45. Wang, Y., Zhou, W., Jiang, T., Bai, X., Xu, Y.: Intra-class feature variation distillation for semantic segmentation. In: ECCV. vol. 12352, pp. 346–362 (2020)
46. Wu, B., Wan, A., Yue, X., Keutzer, K.: SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. In: ICRA. pp. 1887–1893 (2018)
47. Wu, B., Zhou, X., Zhao, S., Yue, X., Keutzer, K.: SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud. In: ICRA. pp. 4376–4382 (2019)
48. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and efficient design for semantic segmentation with transformers. CoRR **abs/2105.15203** (2021)
49. Xu, C., Wu, B., Wang, Z., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In: ECCV. pp. 1–19 (2020)
50. Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S.: PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: CVPR. pp. 5588–5597 (2020)
51. Yan, Y., Mao, Y., Li, B.: SECOND: sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
52. Yi, L., Kim, V.G., Ceylan, D., Shen, I., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L.J.: A scalable active framework for region annotation in 3D shape collections. ACM TOG **35**(6), 210:1–210:12 (2016)
53. Zhang, F., Fang, J., Wah, B.W., Torr, P.H.S.: Deep fusionnet for point cloud semantic segmentation. In: ECCV. vol. 12369, pp. 644–663 (2020)
54. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: CVPR. pp. 9756–9765 (2020)
55. Zhang, Y., Qu, Y., Xie, Y., Li, Z., Zheng, S., Li, C.: Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation. In: ICCV. pp. 15520–15528 (2021)
56. Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B., Foroosh, H.: PolarNet: An improved grid representation for online LiDAR point clouds semantic segmentation. In: CVPR. pp. 9598–9607 (2020)
57. Zheng, W., Tang, W., Jiang, L., Fu, C.: SE-SSD: self-ensembling single-stage object detector from point cloud. In: CVPR. pp. 14494–14503 (2021)
58. Zhou, Z., Zhang, Y., Foroosh, H.: Panoptic-PolarNet: Proposal-free LiDAR point cloud panoptic segmentation. In: CVPR. pp. 13194–13203 (2021)
59. Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., Lin, D.: Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation. In: CVPR. pp. 9939–9948 (2021)

60. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: ICCV. pp. 593–602. IEEE (2019)