2DPASS: 2D Priors Assisted Semantic Segmentation on LiDAR Point Clouds

Xu Yan[1†], Jiantao Gao[2†], Chaoda Zheng[1†],
Chao Zheng[3], Ruimao Zhang[1], Shuguang Cui[1], Zhen Li[1⋆]

[1]The Future Network of Intelligence Institute, The Chinese University of Hong Kong (Shenzhen), Shenzhen Research Institute of Big Data,
[2]Shanghai University, [3]Tencent Map, T Lab

Abstract. As camera and LiDAR sensors capture complementary information in autonomous driving, great efforts have been made to conduct semantic segmentation through multi-modality data fusion. However, fusion-based approaches require paired data, *i.e.*, LiDAR point clouds and camera images with strict point-to-pixel mappings, as the inputs in both training and inference stages. It seriously hinders their application in practical scenarios. Thus, in this work, we propose the 2D Priors Assisted Semantic Segmentation (**2DPASS**) method, a general training scheme, to boost the representation learning on point clouds. The proposed 2DPASS method fully takes advantage of 2D images with rich appearance during training, and then conduct semantic segmentation without strict paired data constraints. In practice, by leveraging an auxiliary modal fusion and multi-scale fusion-to-single knowledge distillation (MSFSKD), 2DPASS acquires richer semantic and structural information from the multi-modal data, which are then distilled to the pure 3D network. As a result, our baseline model shows significant improvement with only point cloud inputs once equipped with the 2DPASS. Specifically, it achieves the state-of-the-arts on two large-scale recognized benchmarks (*i.e.*, SemanticKITTI and NuScenes), *i.e.*, ranking the top-1 in both single and multiple scan(s) competitions of SemanticKITTI. Code will be made available at https://github.com/yanx27/2DPASS.

Keywords: Semantic Segmentation, Multi-Modal, Knowledge Distillation, LiDAR Point Clouds

1 Introduction

Semantic segmentation plays a crucial role in large-scale outdoor scene understanding, which has broad applications in autonomous driving and robotics [1–3]. In the past few years, the research community has devoted significant effort to understanding natural scenes using either camera images [4–7] or LiDAR point clouds [2, 8–12] as the input. However, these single-modal methods inevitably face challenges in complex environments due to the inherent limitations of the

⋆ Corresponding author: Zhen Li. † Equal first authorship.

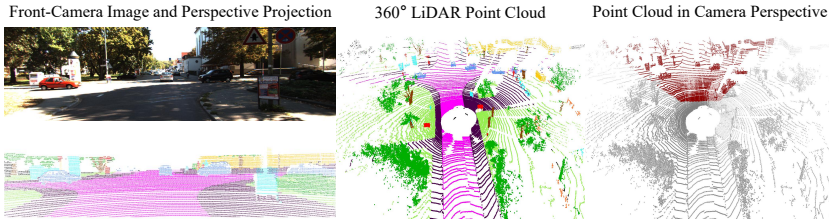


Fig. 1. Limitation of fusion-based methods. When the self-driving car only has front-cameras with limited perspective such as SemanticKITTI [16] dataset while the 360-degree LiDAR has a much larger sensing range, fusion-based methods that require strict alignment between camera and LiDAR can only identify a small proportion of the point cloud (see the red region).

input sensors. Concretely, cameras provide dense color information and fine-grained texture, but they are ambiguous in depth sensing and unreliable in low light conditions. In contrast, LiDARs robustly offer accurate and wide-ranging depth information regardless of lighting variances but only capture sparse and textureless data. Since cameras and LiDARs complement each other, it is better to perceive the surrounding with both sensors.

Recently, many commercial cars have been equipped with both cameras and LiDARs. This excites the research community to improve the semantic segmentation by fusing the information from two complementary sensors [13–15]. These approaches first establish the mapping between 3D points and 2D pixels by projecting the point clouds onto the image planes using the sensor calibrations. Based on the point-to-pixel mapping, the models fuse the corresponding image features into the point features, which are further processed to obtain the final semantic scores. Despite the improvements, fusion-based methods have the following unavoidable limitations: **1)** Due to the difference of FOVs (field of views) between cameras and LiDARs, the point-to-pixel mapping cannot be established for points that are out of the image planes. Typically, the FOVs of LiDAR and cameras only overlap in a small portion (see Fig. 1), which significantly limits the application of fusion-based methods. **2)** Fusion-based methods consume more computational resources since they process both images and point clouds (through multitask or cascade manners) at runtime, which introduces a great burden on real-time applications.

To address the above two issues, we focus on improving semantic segmentation by leveraging both images and point clouds through an effective design in this work. Considering the sensors are moving in the scenes, the non-overlap part of the 360-degree LiDAR point clouds corresponding to image in the same time-stamp (see the gray region of the right part in Fig. 1) can be covered by images from other time-stamp. Besides, the dense and structural information of images provides useful regularization for both seen and unseen point cloud regions. Based on these observations, we propose a "model-independent" training scheme, namely 2D Priors Assisted Semantic Segmentation (**2DPASS**), to enhance the representation learning of any 3D semantic segmentation networks with minor

structure modification. In practice, on the one hand, for above-mentioned non-overlap regions, 2DPASS takes pure point clouds as the inputs to train the segmentation model. On the other hand, for subregions with well-aligned point-to-pixel mappings, 2DPASS adopts an auxiliary multi-modal fusion to aggregate image and point features in each scale, and then aligns the 3D predictions with the fusion predictions. Unlike previous cross-modal alignment [17] apt to contaminate the modal-specific information, we design a multi-scale fusion-to-single knowledge distillation (MSFSKD) strategy to transfer extra knowledge to the 3D model as well as retaining its modal-specific ability. Compared with fusion-based methods, our solution has the following preferable properties: **1) Generality**: It can be easily integrated with any 3D segmentation model with minor structural modification; **2) Flexibility**: The fusion module is only used during the training to enhance the 3D network. After training, the enhanced 3D model can be deployed without image inputs. **3) Effectively**: Even with only a small section of overlapped multi-modality data, our method can significantly boost the performance. As a result, we evaluate 2DPASS with a simple yet strong baseline implemented with sparse convolutions [3]. The experiments show 2DPASS brings noticeable improvements even over this strong baseline. Equipped with 2DPASS using multi-modal data, our model achieves the **top-1** results on the single and multiple-scan leaderboards of SemanticKITTI [16]. The state-of-the-art results on the NuScenes [18] dataset further confirm the generality of our method.

In general, the main contributions are summarized as follows.

- We propose 2D Priors Assisted Semantic Segmentation (2DPASS) that assists 3D LiDAR semantic segmentation with 2D priors from cameras. To the best of our knowledge, 2DPASS is the first method that distills multi-modal knowledge to single point cloud modality for semantic segmentation.
- Equipped with the proposed multi-scale fusion-to-single knowledge distillation (MSFSKS) strategy, 2DPASS achieves the significant performance gains on SemanticKITTI and NuScenes benchmarks, ranking the **1st** on single and multiple tracks of SemanticKITTI.

2 Related Work

2.1 Single-Sensor Methods

Camera-Based Methods. Camera-based semantic segmentation aims to predict the pixel-wise labels for input 2D images. FCN [19] is the pioneer in semantic segmentation, which proposes an end-to-end fully convolutional architecture based on image classification networks. Recent works have achieved significant improvements via exploring multi-scale features learning [4,20,21], dilated convolution [5,22], and attention mechanisms [7,23]. However, camera-only methods are ambiguous in depth sensing and not robust in low light conditions.

LiDAR-Based Methods. The LiDAR data is generally represented as point clouds. There are several mainstreams to process point clouds with different representations. **1) Point-based** methods approximate a permutation-invariant

set function using a per-point Multi-Layer Perceptron (MLP). PointNet [24] is the pioneer in this field. Later on, many studies design point-wise MLP [25, 26], adaptive weight [27, 28] and pseudo grid [29, 30] based methods to extract local features of point clouds or exploit nonlocal operators [31–33] to learn long distance dependency. However, point-based methods are not efficient in the LiDAR scenario since their sampling and grouping algorithms are generally time-consuming. **2) Projection-based** methods are very efficient approaches for LiDAR point clouds. They project point clouds onto 2D pixels so that traditional CNN can play a normal role. Previous works project all points scanned by the rotating LiDAR onto 2D images by plane projection [34–36], spherical projection [37, 38] or both [39]. However, the projection inevitably causes information loss. And the projection-based methods currently meet the bottleneck of the segmentation accuracy. **3)** Most recent works adopt **voxel-based** frameworks since they balance the efficiency and effectiveness, where sparse convolution (SparseConv) [3] are most commonly utilized. Compared to traditional voxel-based methods (*i.e.*, 3DCNN) directly transforming all points into the 3D voxel grids, SparseConv only stores non-empty voxels in a Hash table and conducts convolution operations only on these non-empty voxels in a more efficient way. Recently, many studies have used SparseConv to design more powerful network architectures. Cylinder3D [40] changes original grid voxels to cylinder ones and designs an asymmetrical network to boost the performance. AF^2-S3Net [41] applies multiple branches with different kernel sizes, aggregating multi-scale features via an attention mechanism. **4)** Very recently, there is a trend of exploiting **multi-representation fusion** methods. These methods combine multiple representations above (*i.e.*, points, projection images, and voxels) and design feature fusion among different branches. Tang *et.al.* [10] combines point-wise MLPs in each sparse convolution block to learn a point-voxel representation and uses NAS to search for a more efficient architecture. RPVNet [42] proposes range-point-voxel fusion network to utilizes information from three representations. Nevertheless, these methods only take sparse and textureless LiDAR point clouds as inputs, thus appearance and texture in the camera images have not been fully utilized.

2.2 Multi-Sensor Methods

Multi-sensor methods attempt to fuse information from two complementary sensors and leverage the benefits of both camera and LiDAR [14, 15, 43, 44]. RGBAL [14] converts RGB images to a polar-grid mapping representation and designs early and mid-level fusion strategies. PointPainting [15] exploits the segmentation logits of images and projects them to the LiDAR space by bird's-eye projection [23] or spherical projection [45] for LiDAR network performance improvement. Recently, PMF [13] exploits a collaborative fusion of two modalities in camera coordinates. However, these methods require multi-sensor inputs in both training and inference phases. Moreover, the paired multi-modality data is usually computation-intensive and unavailable in practical application.

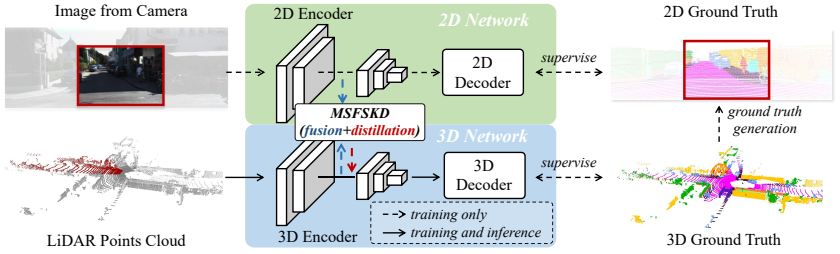


Fig. 2. 2D Priors Assisted Semantic Segmentation (2DPASS). It first crops a small patch from the original camera image as the 2D input. Then the cropped image patch and LiDAR point cloud independently pass through the 2D and 3D encoders to generate multi-scale features in parallel. Afterwards, for each scale, complementary 2D knowledge is effectively transferred to the 3D network via the multi-scale fusion-to-single knowledge distillation (MSFSKD). The feature maps (in the form of either pixel grid or point set) are used to generate the final semantic scores using modal-specific decoders, which are supervised by pure 3D labels.

2.3 Cross-modal Knowledge Transfer

Knowledge distillation was initially proposed for compressing the large teacher network to a small student one [46]. Over the past few years, several subsequent studies enhanced knowledge transferring through matching feature representations in different manners [47–50]. For instance, aligning attention maps [49] and Jacobean matrixes [50] were independently applied. With the development of multi-modal computer vision, recent research apply knowledge distillation to transfer priors across different modalities, *e.g.*, exploiting extra 2D images in the training phase and improving the performance in the inference [51–55]. Specifically, [56] introduces the 2D-assisted pre-training, [57] inflates the kernels of 2D convolution to the 3D ones, and [58] applies well-designed teacher-student framework. Inspired but different from the above, we transfer 2D knowledge through a multi-scale fusion-to-single manner, which additionally takes care of the modal-specific knowledge.

3 Method

3.1 Framework Overview

This paper focuses on improving the LiDAR point cloud semantic segmentation, which aims to assign the semantic label to each point. To handle difficulties in large-scale outdoor LiDAR point clouds, *i.e.*, sparsity, varying density, and lack of texture, we introduce the strong regularization and priors from 2D camera images through a **fusion-to-single** knowledge transferring.

The workflow of our 2D Priors Assisted Semantic Segmentation (2DPASS) is shown in Fig. 2. Since the camera images are pretty large (*e.g.*, 1242×512), sending the original ones to our multi-modal pipeline is intractable. Therefore,

we randomly sample a small patch (480×320) from the original camera image as the 2D input [17], accelerating the training processing without performance drop. Then the cropped image patch and LiDAR point cloud independently pass through independent 2D and 3D encoders, where multi-scale features from the two backbones are extracted in parallel. Afterwards, multi-scale fusion-to-single knowledge distillation (MSFSKD) is conducted to enhance the 3D network using multi-modal features, *i.e.*, fully utilizing texture and color-aware 2D priors as well as retaining the original 3D-specific knowledge. Finally, all the 2D and 3D features at each scale are used to generate semantic segmentation predictions, which are supervised by pure 3D labels. During inference, the 2D-related branch can be discarded, which effectively prevents extra computational burden in real application compared with fusion-based approaches.

3.2 Modal-Specific Architectures

Multi-Scale Feature Encoders. As shown in Fig. 2, we use two different networks to independently encode multi-scale features from 2D image and 3D point cloud. We apply ResNet34 [59] encoder with 2D convolution as the 2D network. For the 3D network, we adopt sparse convolution [3] to construct the 3D network. One merit of sparse convolution lies in the sparsity, with which the convolution operation only considers the non-empty voxels. Specifically, we design a hierarchical point-voxel encoder as that used in the decoder of [10], and adopt the ResNet bottleneck [59] in each scale while replacing the ReLU with Leaky ReLU [60]. In both network, we extract L feature maps from different scales, obtaining the 2D and 3D features, *i.e.*, $\{F_l^{2D}\}_{l=1}^{L}$ and $\{F_l^{3D}\}_{l=1}^{L}$.

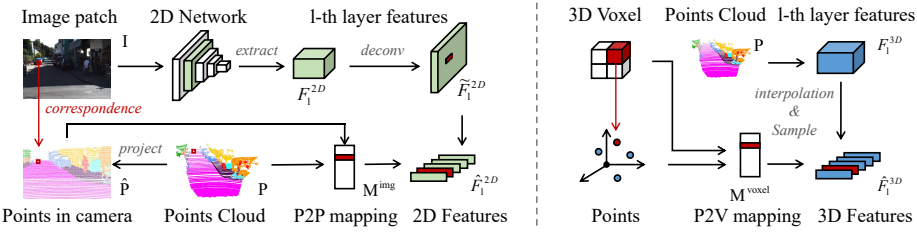
Prediction Decoders. After processing the features from images and point clouds at each scale, two modal-specific prediction decoders are independently applied to restore the down-sampled feature maps to their original sizes.

For the 2D network, we adopt FCN [19] decoder to up-sample the features from each encoder layer. Specifically, the feature map D_l^{2D} from the l-th decoder layer can be gained by up-sampling the feature map from the $(L - l + 1)$-th encoder layer, where all the up-sampled feature maps will be merged through element-wise addition. Finally, the semantic segmentation of the 2D network is obtained by passing the fused feature map through a linear classifier.

For the 3D network, we do not adopt the U-Net decoder used in previous methods [10, 40, 41]. In contrast, we up-sample the features from different scales to the original size and concatenate them together before feeding them into the classifier. We find out that such a structure can better learn hierarchical information while gaining the prediction in a more efficient way.

3.3 Point-to-Pixel Correspondence

Since the 2D features and 3D features are generally represented as pixels and points, respectively, it is difficult to directly transfer information between two modalities. In this section, we aim to generate paired features of two modalities



(a) 2D Feature Generation *(b) 3D Feature Generation*

Fig. 3. 2D and 3D feature generation. Part (a) demonstrates the 2D feature generation, where the point cloud will first be projected onto the image patch and generate the point-to-pixel (P2P) mapping. After that, it transfers the 2D feature map to the point-wise 2D features according to P2P mapping. Part (b) shows the 3D feature generation. The point-to-voxel (P2V) mapping is easy to obtain, and the voxel features will be interpolated onto the point cloud.

for further knowledge distillation, using the point-to-pixel correspondence. The details of paired feature generation in two modalities are demonstrated in Fig. 3.

2D Features. The process of 2D feature generation is illustrated in Fig. 3 (a). By cropping a small patch $I \in \mathbb{R}^{H\times W\times 3}$ from the original image and passing it through a 2D network, multi-scale features can be extracted in the hidden layers with different resolution. Taking the feature map $F_l^{2D} \in \mathbb{R}^{H_l\times W_l\times D_l}$ from l-th layer as an example, we first conduct a decovolution operation to upscale its resolution to the original one $\tilde{F}_l^{2D}$. Similar to the recent multi-sensor method [13], we adopt perspective projection and calculate a point-to-pixel mapping between point clouds and images. Specifically, given a LiDAR point cloud $P = \{p_i\}_{i=1}^N \in \mathbb{R}^{N\times 3}$, the projection of each 3D point $p_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ to a point $\hat{p}_i = (u_i, v_i) \in \mathbb{R}^2$ in the image plane is given as:

$$[u_i, v_i, 1]^T = \frac{1}{z_i} \times K \times T \times [x_i, y_i, z_i, 1]^T, \tag{1}$$

where $K \in \mathbb{R}^{3\times 4}$ and $T \in \mathbb{R}^{4\times 4}$ are the camera intrinsic and extrinsic matrices respectively. K and T are directly provided in KITTI [61]. Since the lidar and cameras operate at different frequencies in NuScenes [18], we need to transform the LiDAR frame at timestamp t_l to camera frame at timestamp t_c via the global coordinate system. The extrinsic matrix T in NuScenes dataset [18] is given as:

$$T = T_{\text{camera}\leftarrow\text{ego}_{t_c}} \times T_{\text{ego}_{t_c}\leftarrow\text{global}} \times T_{\text{global}\leftarrow\text{ego}_{t_l}} \times T_{\text{ego}_{t_l}\leftarrow\text{lidar}} \tag{2}$$

After the projection, the point-to-pixel mapping is represented as

$$M^{img} = \{(\lfloor v_i \rfloor, \lfloor u_i \rfloor)\}_{i=1}^N \in \mathbb{R}^{N\times 2}, \tag{3}$$

where $\lfloor\cdot\rfloor$ is the floor operation. According to the point-to-pixel mapping, we extract a point-wise 2D feature $\hat{F}^{2D} \in \mathbb{R}^{N^{img}\times D_l}$ from the original feature map

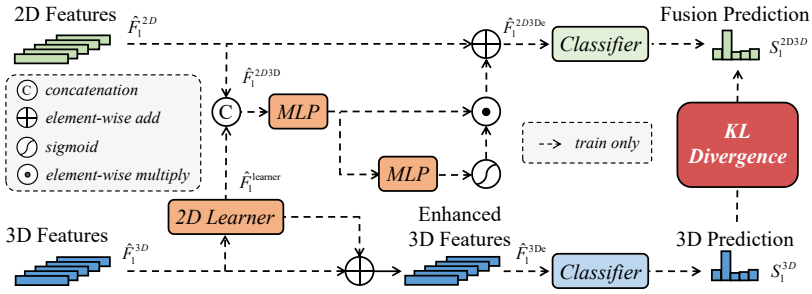


Fig. 4. Internal structure of **Multi-Scale Fusion-to-Single Knowledge Distillation (MSFSKD)**, which consists of the modality fusion and Modality-Preserving KD. For each scale, modality fusion is first ultilized to achieve an enhanced multi-modality feature $\hat{F}_l^{2D3D_e}$. Afterwards, the enhanced feature $\hat{F}_l^{2D3D_e}$ promotes the 3D representation $\hat{F}_l^{3D_e}$ through the uni-directional Modality-Preserving KD.

F^{2D} if any pixel on the feature map is included in M^{img}. Here $N^{img} < N$ represents the number of points that are included in M^{img}.

3D Features. The process of 3D features is relatively straightforward (as shown in Fig. 3 (b)). Specifically, for the point cloud $P = \{(x_i, y_i, z_i)\}_{i=1}^{N}$, we obtain a point-to-voxel mapping in the l-th layer through

$$M_l^{voxel} = \{(\lfloor x_i/r_l \rfloor, \lfloor y_i/r_l \rfloor, \lfloor z_i/r_l \rfloor)\}_{i=1}^{N} \in \mathbb{R}^{N\times 3}, \tag{4}$$

where r_l is the voxelization resolution in the l-th layer. After that, given the 3D feature $F_l^{3D} \in \mathbb{R}^{N_l' \times D_l}$ from a sparse convolution layer, we gain a point-wise 3D feature $\tilde{F}_l^{3D} \in \mathbb{R}^{N\times D_l}$ through nearest interpolation on the original feature map F_l^{3D} according to M_l^{voxel}. Finally, we filter the points by discarding points outside the image FOV:

$$\hat{F}_l^{3D} = \{f_i | f_i \in \tilde{F}_l^{3D}, M_{i,1}^{img} \le H, M_{i,2}^{img} \le W\}_{i=1}^{N} \in \mathbb{R}^{N^{img}\times D_l}, \tag{5}$$

2D Ground Truths. Considering only 2D images is provided, the 2D ground-truths are obtained by projecting the 3D point labels to the corresponding image plane using above point-to-pixel mapping. Afterwards, the projected 2D ground truths can work as the supervision for the 2D branch.

Features Correspondence. Since both 2D and 3D feature use the same point-to-pixel mapping, 2D features $\hat{F}_l^{2D}$ and 3D features $\hat{F}_l^{3D}$ in arbitrary l-th layer have the same number of point N^{img} and point-to-pixel correspondence.

3.4 Multi-Scale Fusion-to-Single Knowledge Distillation (MSFSKD)

As the key of 2DPASS, MSFSKD aims at improving the 3D representation in each scale using auxiliary 2D priors through a fusion-then-distillation manner. The knowledge distillation (KD) design of MSFSKD is partially inspired by [17]. However, [17] conducts KD in a naive cross-modal manner, *i.e.*, simply

aligning the outputs from two sets of single modal features (*i.e.* either 2D or 3D), which inevitably pushes the features from two modals to their overlapped space. Therefore, such a manner actually discards the modal-specific information, which is crucial in multi-sensor segmentation. Although this issue can be relieved by introducing extra segmentation heads [17], it is inherent for the cross-modal distillation, resulting in biased predictions. To this end, we propose multi-scale fusion-to-single knowledge distillation (MSFSKD) module as shown in Fig. 4, which first fuses features of both images and point clouds and then conducts unidirectional alignment between the fused and the point cloud features. In our fusion-then-distillation manner, the fusion well retains the complete information from multi-modal data. Besides, the unidirectional alignment ensures boosted point cloud features from fusion without losing modal-specific information.

Modality Fusion. For each scale, considering the 2D and 3D feature gaps owing to different backbones, it is ineffective to directly fuse the raw 3D features $\hat{F}_l^{3D}$ into their 2D counterparts $\hat{F}_l^{2D}$. Thus, we firstly transform $\hat{F}_l^{3D}$ to $\hat{F}_l^{\text{learner}}$ through a "2D learner" MLP, which struggles to narrow the feature gap. Afterwards, the $\hat{F}_l^{\text{learner}}$ not only flows into the subsequent concatenation with 2D features $\hat{F}_l^{2D}$ to gain the fused features $\hat{F}_l^{2D3D}$ through another MLP, but also goes back into the original 3D features via a skip connection to yield enhanced 3D features $\hat{F}_l^{3D_e}$. Besides, similar to attention mechanism, the final enhanced fused features $\hat{F}_l^{2D3D_e}$ is obtained by:

$$\hat{F}_l^{2D3D_e} = \hat{F}_l^{2D} + \sigma(\texttt{MLP}(\hat{F}_l^{2D3D})) \odot \hat{F}_l^{2D3D}, \tag{6}$$

where σ denotes Sigmoid activation function.

Modality-Preserving KD. Although the $\hat{F}_l^{\text{learner}}$ is generated from pure 3D features, it is influenced by the segmentation loss of the 2D decoder as well, which takes enhanced fused feature $\hat{F}_l^{2D3D_e}$ as inputs. Acting like a residual between fused and point features, the 2D learner feature $\hat{F}_l^{\text{learner}}$ well prevents the distillation from contaminating the modal-specific information in $\hat{F}_l^{3D}$, achieving a Modality-Preserving KD. Finally, two independent classifiers (fully-connected layers) are respectively applied on top of $\hat{F}_l^{2D3D_e}$ and $\hat{F}_l^{3D_e}$ to obtain the semantic scores S_l^{2D3D} and S_l^{3D}. We choose KL divergence as the distillation loss L_{xM} as follows:

$$L_{xM} = D_{KL}(S_l^{2D3D} || S_l^{3D}). \tag{7}$$

Through such an implementation, it enforces the uni-directional distillation by pushing S_l^{3D} closer to S_l^{2D3D}.

By taking such a knowledge distillation scheme, there are several advantages in our framework: **1)** The 2D learner and the fusion-to-single distillation provides rich texture information and structural regularization to enhance the 3D feature learning without losing any modal-specific information in 3D. **2)** The fusion branch is only adopted in the training phase. Therefore, the enhanced model can almost run without extra computational cost during the inference.

Table 1. Semantic segmentation results on the *SemanticKITTI* test benchmark. Only approaches published before 03/08/2022 are compared.

Method	mIoU	road	sidewalk	parking	other-ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic sign	speed (ms)
SqueezeSegV2 [38]	39.7	88.6	67.6	45.8	17.7	73.7	81.8	13.4	18.5	17.9	14.0	71.8	35.8	60.2	20.1	25.1	3.9	41.1	20.2	26.3	-
DarkNet53Seg [16]	49.9	91.8	74.6	64.8	27.9	84.1	86.4	25.5	24.5	32.7	22.6	78.3	50.1	64.0	36.2	33.6	4.7	55.0	38.9	52.2	-
RangeNet53++ [45]	52.2	91.8	75.2	65.0	27.8	87.4	91.4	25.7	25.7	34.4	23.0	80.5	55.1	64.6	38.3	38.8	4.8	58.6	47.9	55.9	83.3
3D-MiniNet [62]	55.8	91.6	74.5	64.2	25.4	89.4	90.5	28.5	42.3	42.1	29.4	82.8	60.8	66.7	47.8	44.1	14.5	60.8	48.0	56.6	-
SqueezeSegV3 [8]	55.9	91.7	74.8	63.4	26.4	89.0	92.5	29.6	38.7	36.5	33.0	82.0	58.7	65.4	45.6	46.2	20.1	59.4	49.6	58.9	238
PointNet++ [25]	20.1	72.0	41.8	18.7	5.6	62.3	53.7	0.9	1.9	0.2	0.2	46.5	13.8	30.0	0.9	1.0	0.0	16.9	6.0	8.9	5900
TangentConv [36]	40.9	83.9	63.9	33.4	15.4	83.4	90.8	15.2	2.7	16.5	12.1	79.5	49.3	58.1	23.0	28.4	8.1	49.0	35.8	28.5	3000
PointASNL [31]	46.8	87.4	74.3	24.3	1.8	83.1	87.9	39.0	0.0	25.1	29.2	84.1	52.2	70.6	34.2	57.6	0.0	43.9	57.8	36.9	-
RandLA-Net [1]	55.9	90.5	74.0	61.8	24.5	89.7	94.2	43.9	29.8	32.2	39.1	83.8	63.6	68.6	48.4	47.4	9.4	60.4	51.0	50.7	880
KPConv [29]	58.8	90.3	72.7	61.3	31.5	90.5	95.0	33.4	30.2	42.5	44.3	84.8	69.2	69.1	61.5	61.6	11.8	64.2	56.4	47.4	-
PolarNet [63]	54.3	90.8	74.4	61.7	21.7	90.0	93.8	22.9	40.3	30.1	28.5	84.0	65.5	67.8	43.2	40.2	5.6	61.3	51.8	57.5	**62**
JS3C-Net [2]	66.0	88.9	72.1	61.9	31.9	92.5	95.8	54.3	59.3	52.9	46.0	84.5	69.8	67.9	69.5	65.4	39.9	70.8	60.7	68.7	471
SPVNAS [10]	67.0	90.2	75.4	67.6	21.8	91.6	97.2	56.6	50.6	50.4	58.0	86.1	73.4	71.0	67.4	67.1	50.3	66.9	64.3	67.3	259
Cylinder3D [40]	68.9	92.2	77.0	65.0	32.3	90.7	97.1	50.8	67.6	63.8	58.5	85.6	72.5	69.8	73.7	69.2	48.0	66.5	62.4	66.2	131
RPVNet [42]	70.3	**93.4**	**80.7**	**70.3**	33.3	**93.5**	**97.6**	44.2	**68.4**	68.7	61.1	**86.5**	**75.1**	**71.7**	75.9	74.4	43.4	72.1	64.8	61.4	168
$(AF)^2$-S3Net [41]	70.8	92.0	76.2	66.8	**45.8**	92.5	94.3	40.2	63.0	**81.4**	40.0	78.6	68.0	63.1	76.4	**81.7**	**77.7**	69.6	64.0	**73.3**	-
Baseline	67.4	89.8	73.8	62.1	33.5	91.9	96.3	54.9	51.1	55.8	51.6	**86.5**	72.3	71.3	76.8	79.8	30.3	68.7	63.7	70.2	**62**
2DPASS(Ours)	**72.9**	89.7	74.7	67.4	40.0	**93.5**	97.0	**61.1**	63.6	63.4	**61.5**	86.2	73.9	71.0	**77.9**	81.3	74.1	**72.9**	**65.0**	70.4	**62**

4 Experiments

4.1 Experiment Setups

Datasets. We extensively evaluate 2DPASS on two large-scale outdoor benchmarks: SemanticKITTI [16] and Nuscenes [18]. **SemanticKITTI** provides dense semantic annotations for each individual scan of sequences 00-10 in KITTI dataset [61]. According to the official setting, sequence 08 is the validation split, while the remaining are the train split. SemanticKITTI uses sequences 11-21 in KITTI as the test set, whose labels are held on for blind online testing[1]. **NuScenes** contains 1000 scenes which show a great diversity in inner cities traffic and weather conditions. It officially divides the data into 700/150/150 scenes for train/val/test. Similar to SemanticKITTI, the test set of NuScenes is used for online benchmarking[2]. For 2D sensors, KITTI has only two front-view cameras, while NuScenes has six cameras covering the full 360° fields of view.

Evaluation Metrics. We evaluate methods mainly using mean intersection over union (mIoU), which is defined as the average IoU over all classes. Additionally, we report the overall accuracy (Acc)/ frequency-weighted IOU (FwIOU) provided by the online leaderboard of two benchmarks. FwIoU is similar to mIoU except that each IoU is weighted by the point-level frequency of its class.

Network Setup. We apply ResNet34 [59] encoder with 2D convolution as the 2D network, where features after each down-sampling layers are extracted to generate 2D features. The 3D encoder is a modified SPVCNN [10] (voxel size 0.1) with fewer parameters, whose hidden dimensions are 64 for SemanticKITTI and 128 for NuScenes to speed up the network. The number of layers L for MSFSKD

[1] https://competitions.codalab.org/competitions/20331

[2] https://eval.ai/web/challenges/challenge-page/720/leaderboard/1967

Table 2. Comparison to the state-of-the-art methods on the test set of SemanticKITTI multiple scans challenge. *-s* indicates static and *-m* stands for moving.

Method	**mIoU**	**Acc**	car-s	car-m	truck-s	truck-m	other-s	other-m	person-s	person-m	bicyclist-s	bicyclist-m	motorcyclist-s	motorcyclist-m
LatticeNet [64]	45.2	89.3	91.1	54.8	29.7	3.5	23.1	0.6	6.8	49.9	0.0	44.6	0.0	64.3
TemporalLidarSeg [65]	47.0	89.6	92.1	68.2	39.2	2.1	35.0	**12.4**	14.4	40.4	0.0	42.8	0.0	12.9
KPConv [29]	51.2	89.3	93.7	69.4	42.5	5.8	38.6	4.7	21.6	67.5	0.0	67.4	0.0	47.2
Cylinder3D [40]	52.5	91.0	94.6	74.9	41.3	0.0	38.8	0.1	12.5	65.7	1.7	68.3	0.2	11.9
$(AF)^2$-S3Net [41]	56.9	88.1	91.8	65.3	15.7	5.6	27.5	3.9	16.4	67.6	**15.1**	66.4	**67.1**	59.6
2DPASS(Ours)	**62.4**	**91.4**	**96.2**	**82.1**	**48.2**	**16.1**	**52.7**	3.8	**35.4**	**80.3**	7.9	**71.2**	62.0	**73.1**

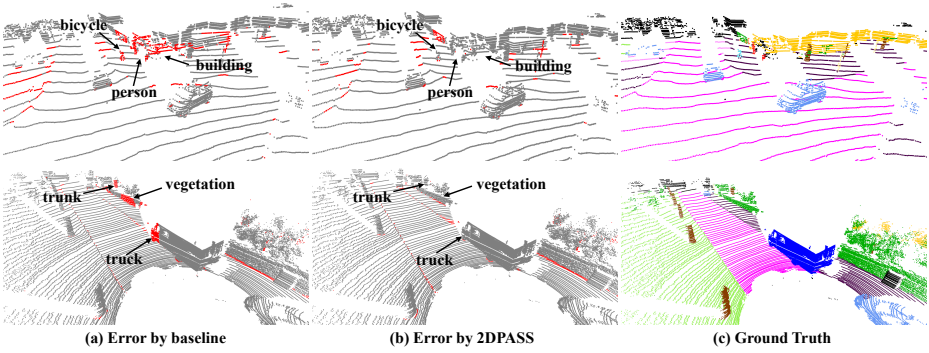


Fig. 5. Qualitative results of 2DPASS on the validation set of SemanticKITTI. Our baseline has a higher error recognizing small objects and region boundaries, while 2DPASS recognizes small objects better thanks to the prior of 2D modality.

is set to 4 and 6 for SemanticKITTI and NuScenes, respectively. In each scale of knowledge distillation, 2D and 3D features are reduced to 64 dimensions through deconvolution or MLPs. Similarly, the hidden size of MLPs and 2D learner in MSFSKD are identically 64.

Training and Inference Details. We employ the cross-entropy and Lovasz losses as [40] for semantic segmentation. For the knowledge distillation, we set the proportion of segmentation loss and KL divergence as 1 : 0.05. Test-time augmentation [40] is applied during the inference. Training details will be introduced in supplementary material.

4.2 Benchmark Results

SemanticKITTI. SemanticKITTI evaluates segmentation performance using two settings: single scan and multiple scans. For methods using a single scan as input, moving and non-moving are mapped to a single class. While methods using multiple scans as inputs should distinguish between moving and non-moving

Table 3. Semantic segmentation results on the *Nuscenes* test benchmark. Only approaches published before 03/08/2022 are compared. L and C stand for LiDAR and camera, respectively. (*) The speed reported in PMF [13] is accelerated by TensorRT, and we test their model without such technique in the same environment.

Method	Input	**mIoU**	**FW mIoU**	barrier	bicycle	bus	car	construction	motorcycle	pedestrian	traffic cone	trailer	truck	driveable	other flat	sidewalk	terrain	manmade	vegetation	speed (ms)
PolarNet [63]	L	69.4	87.4	72.2	16.8	77.0	86.5	51.1	69.7	64.8	54.1	69.7	63.5	96.6	67.1	77.7	72.1	87.1	84.5	-
JS3C-Net [2]	L	73.6	88.1	80.1	26.2	87.8	84.5	55.2	72.6	71.3	66.3	76.8	71.2	96.8	64.5	76.9	74.1	87.5	86.1	-
Cylinder3D [40]	L	77.2	89.9	82.8	29.8	84.3	89.4	63.0	79.3	77.2	73.4	84.6	69.1	**97.7**	**70.2**	**80.3**	75.5	90.4	87.6	63
AMVNet [39]	L	77.3	90.1	80.6	32.0	81.7	88.9	67.1	84.3	76.1	73.5	**84.9**	67.3	97.5	67.4	79.4	75.5	91.5	88.7	85
SPVCNN [10]	L	77.4	89.7	80.0	30.0	91.9	90.8	64.7	79.0	75.6	70.9	81.0	74.6	97.4	69.2	80.0	76.1	89.3	87.1	63
$(AF)^2$-S3Net [41]	L	78.3	88.5	78.9	52.2	89.9	84.2	**77.4**	74.3	77.3	72.0	83.9	73.8	97.1	66.5	77.5	74.0	87.7	86.8	270
PMF [13]	L+C	77.0	89.0	82.0	40.0	81.0	88.0	64.0	79.0	80.0	**76.0**	81.0	67.0	97.0	68.0	78.0	74.0	90.0	88.0	125*
2D3DNet [66]	L+C	80.0	**90.1**	**83.0**	**59.4**	88.0	85.1	63.7	84.4	**82.0**	**76.0**	84.8	71.9	96.9	67.4	79.8	**76.0**	**92.1**	**89.2**	-
Baseline	L	77.6	88.5	80.8	37.9	92.7	90.5	65.4	77.6	71.5	70.9	83.1	75.3	97.0	69.3	78.1	75.6	89.1	86.8	**44**
2DPASS(Ours)	L	**80.8**	**90.1**	81.7	55.3	**92.0**	**91.8**	73.3	**86.5**	78.5	72.5	84.7	**75.5**	97.6	69.1	79.9	75.5	90.2	88.0	**44**

objects, which is more challenging. All the reported results are from the official blind test competition website of SemanticKITTI.

Tab. 1 shows our performance under the single scan setting. Our baseline without 2DPASS already performs on par with a strong model Cylinder3D [40] while runs at a faster speed. Even so, the application of 2DPASS still brings a significant improvement over the baseline. Thanks to the auxiliary knowledge distillation, 2DPASS does not put any extra burden on the original model and thus does not sacrifice the running speed of the baseline. Overall, 2DPASS achieves the best result in terms of mIoU and running speed, outperforming the state-of-the-art (*i.e.*, $(AF)^2$-S3Net [41]) by **2.1%**. The visualization results on SemanticKITTI single scan are shown in Fig. 5.

Tab. 2 reports the results under the multiple scans setting. The mIoU and overall accuracy are calculated over all 25 classes. Due to the limited space, we only report the per-class IOUs for dynamic objects with non-moving/moving properties. Under this challenge setting, 2DPASS surprisingly surpasses previous approaches with even larger margins, *i.e.*, achieving better mIoU (5.5% improvement over $(AF)^2$-S3Net [41]) and overall accuracy.

NuScenes. The results on NuScenes are reported in Tab. 3, where 2DPASS achieves the 1st place as well. Note that we only include published works in Tab. 3 and the results are directly taken from the official leaderboard of NuScenes, where our model also ranks the 3rd place with slight disadvantage when considering unpublished works. Besides surpassing all single-modal methods, 2DPASS surprisingly outperforms those fusion-based approaches (the last two rows in Tab. 3). Note that NuScenes provides images covering the whole FOV of the LiDAR, and fusion-based approaches achieve such results by using both point clouds and image features during the inference. In contrast, our method only takes point clouds as input.

Table 4. Comparison with different knowledge distillation.

Method	SemanticKITTI
Hinton *et.al.* [46]	66.34
Huang *et.al.* [67]	66.46
Yang *et.al.* [68]	66.75
xMUDA [17]	67.88
2DPASS	**69.32**

Table 5. Ablation study on the SemanticKITTI validation set.

baseline	MSFSKD			SemanticKITTI
	KL Div	Modality Fusion	2D Learner	
✓				65.58
✓	✓			66.34
✓	✓	✓		69.13
✓	✓	✓	✓	**69.32**

4.3 Comprehensive Analysis

Comparing with Other Knowledge Distillation. To further verify the effectiveness of our fusion-to-single knowledge distillation paradigm upon common teach-student architecture and other cross-modal manners, we compare 2DPASS with typical approaches of knowledge transfer in Tab. 4, where we utilize these methods in each scale for fair comparison. Among all the methods, Hinton *et.al.* [46], Huang *et.al.* [67] and Yang *et.al.* [68] are pure knowledge distillation designs, where the former is the pioneer for the research field and the latter is newly proposed. As shown in the Tab. 4, pure knowledge distillation manners cannot be directly adopted on the LiDAR semantic segmentation, and their improvement upon the baseline model is limited. Recently, [17] adopts cross-modal feature alignment technique in the task of domain adaptation on semantic segmentation. However, their improvement is still marginal. To the end, in the Tab. 4, 2DPASS significantly performs better, which illustrates the effectiveness of our multi-scale fusion-to-single knowledge distillation (MSFSKD).

Design Analysis of MSFSKD. Tab. 5 demonstrates the ablation study on SemanticKITTI validation set. As shown in the table, our baseline only achieves a lower result of 65.58 mIoU. Note that simply using feature alignment between two modalities cannot effectively improve the result, where the metric of mIoU will be only increased to 66.34. After using 2D-3D fusion in each knowledge distillation scale, there is a significant improvement to 69.13. This improvement mainly comes from the knowledge provided by the stronger fusion prediction. Finally, we find out that 2D learner design can slightly improve the performance by about 0.2%. Note that the results on SemanticKITTI validation set is lower than that on benchmark since small object category (*i.e.*, motocyclist) only occupies a small proportion.

Distance-based Evaluation. We investigate how segmentation is affected by distance of the points to the ego-vehicle, and compare 2DPASS, current state-of-the-art and the baseline on the SemanticKITTI validation set. Fig. 6 (a) illustrates the mIoU of 2DPASS as opposed to the baseline and $(AF)^2$-S3Net. The results of all the methods get worse by increasing the distance since points are relatively sparse in the long distance. 2DPASS improves the performance greatly within $10m$, *i.e.*, from 61.2 to 89.1, which is the best distance for the camera to capture objects' color and texture. There is also a significant improvement upon $(AF)^2$-S3Net within this distance, *i.e.*, 84.4 v.s. 89.1.

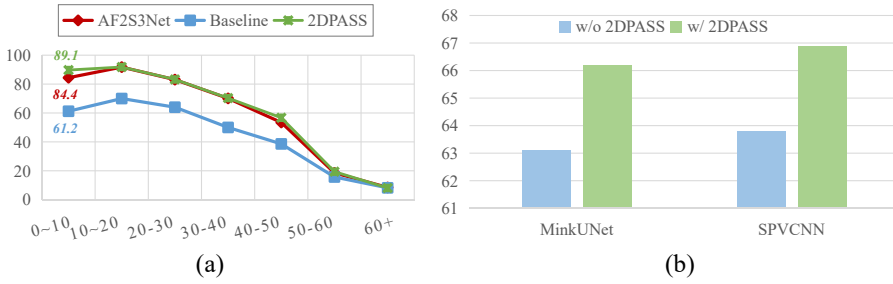


Fig. 6. Extensive experiment results. The part (a) shows the results on SemanticKITTI validation set with different distance-range. Part (b) demonstrates the results before and after exploiting 2DPASS on MinkowskiNet [10] and SPVCNN [10].

Generality. We show our 2DPASS can be a "model-independent" training scheme that boosts the performance of other networks. We additionally trained two open-sourced baselines, *i.e.*, MinkowskiNet and SPVCNN implemented in [10] with 2DPASS. During the experiment, we keep all the setups the same except for the 2D-related components. As shown in Fig. 6 (b), 2DPASS improves the former one from 63.1 to 66.2 and the latter from 63.8 to 66.9. These results sufficiently demonstrate the effectiveness and generality of 2DPASS.

5 Conclusion

This work proposes the 2D Priors Assisted Semantic Segmentation (**2DPASS**), a general training scheme, to boost the performance of LiDAR point cloud semantic segmentation via 2D prior-related knowledge distillation. By leveraging an auxiliary modal fusion and knowledge distillation in a multi-scale manner, 2DPASS acquires richer semantic and structural information from the multi-modal data, effectively enhancing the performance of a pure 3D network. Eventually, it achieves the state-of-the-arts on two large-scale benchmarks (*i.e.*, SemanticKITTI and NuScenes). We believe that our work can be applied to a wider range of other scenarios in the future, such as 3D detection and tracking.

Acknowledgment. This work was supported in part by NSFC-Youth 61902335, by the Basic Research Project No. HZQB-KCZYZ-2021067 of Hetao Shenzhen HK S&T Cooperation Zone, by the National Key R&D Program of China with grant No.2018YFB1800800, by Shenzhen Outstanding Talents Training Fund, by Guangdong Research Project No. 2017ZT07X152 and No. 2019CX01X104, by the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001), by the NSFC 61931024&8192 2046, by NSFC-Youth 62106154, by zelixir biotechnology company Fund, by Tencent Open Fund, and by ITSO at CUHKSZ.

References

1. 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. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
2. Yan, X., Gao, J., Li, J., Zhang, R., Li, Z., Huang, R., Cui, S.: Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 35. (2021) 3101–3109
3. Graham, B., Engelcke, M., van der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 9224–9232
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4) (2017) 834–848
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
6. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 1746–1754
7. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 603–612
8. Xu, C., Wu, B., Wang, Z., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In: European Conference on Computer Vision, Springer (2020) 1–19
9. 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2021) 9939–9948
10. Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S.: Searching efficient 3d architectures with sparse point-voxel convolution. In: European conference on computer vision, Springer (2020) 685–702
11. Zheng, C., Yan, X., Zhang, H., Wang, B., Cheng, S., Cui, S., Li, Z.: Beyond 3d siamese tracking: A motion-centric paradigm for 3d single object tracking in point clouds. arXiv preprint arXiv:2203.01730 (2022)
12. Zheng, C., Yan, X., Gao, J., Zhao, W., Zhang, W., Li, Z., Cui, S.: Box-aware feature enhancement for single object tracking on point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 13199–13208
13. Zhuang, Z., Li, R., Jia, K., Wang, Q., Li, Y., Tan, M.: Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 16280–16290
14. El Madawi, K., Rashed, H., El Sallab, A., Nasr, O., Kamel, H., Yogamani, S.: Rgb and lidar fusion based 3d semantic segmentation for autonomous driving. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE (2019) 7–12
15. Vora, S., Lang, A.H., Helou, B., Beijbom, O.: Pointpainting: Sequential fusion for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2020) 4604–4612

16. 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: Proceedings of the IEEE International Conference on Computer Vision. (2019) 9297–9307
17. Jaritz, M., Vu, T.H., Charette, R.d., Wirbel, E., Pérez, P.: xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 12605–12614
18. 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. (2020)
19. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3431–3440
20. Lin, G., Shen, C., Van Den Hengel, A., Reid, I.: Efficient piecewise training of deep structured models for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 3194–3203
21. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 2881–2890
22. Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., Cottrell, G.: Understanding convolution for semantic segmentation. In: 2018 IEEE winter conference on applications of computer vision (WACV), Ieee (2018) 1451–1460
23. Yuan, Y., Huang, L., Guo, J., Zhang, C., Chen, X., Wang, J.: Ocnet: Object context network for scene parsing. arXiv preprint arXiv:1809.00916 (2018)
24. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 652–660
25. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. (2017) 5099–5108
26. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (TOG) **38**(5) (2019) 1–12
27. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 9621–9630
28. Liu, Y., Fan, B., Xiang, S., Pan, C.: Relation-shape convolutional neural network for point cloud analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 8895–8904
29. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: The IEEE International Conference on Computer Vision (ICCV). (October 2019)
30. Hua, B.S., Tran, M.K., Yeung, S.K.: Pointwise convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 984–993
31. Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S.: Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 5589–5598

32. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 16259–16268
33. Engel, N., Belagiannis, V., Dietmayer, K.: Point transformer. IEEE Access **9** (2021) 134826–134840
34. Lawin, F.J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F.S., Felsberg, M.: Deep projective 3d semantic segmentation. In: International Conference on Computer Analysis of Images and Patterns, Springer (2017) 95–107
35. Boulch, A., Le Saux, B., Audebert, N.: Unstructured point cloud semantic labeling using deep segmentation networks. 3DOR **2** (2017) 7
36. Tatarchenko, M., Park, J., Koltun, V., Zhou, Q.Y.: Tangent convolutions for dense prediction in 3d. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 3887–3896
37. 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: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE (2018) 1887–1893
38. 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: 2019 International Conference on Robotics and Automation (ICRA), IEEE (2019) 4376–4382
39. 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. arXiv preprint arXiv:2012.04934 (2020)
40. Zhou, H., Zhu, X., Song, X., Ma, Y., Wang, Z., Li, H., Lin, D.: Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. arXiv preprint arXiv:2008.01550 (2020)
41. Cheng, R., Razani, R., Taghavi, E., Li, E., Liu, B.: Af2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2021) 12547–12556
42. Xu, J., Zhang, R., Dou, J., Zhu, Y., Sun, J., Pu, S.: Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 16024–16033
43. Krispel, G., Opitz, M., Waltner, G., Possegger, H., Bischof, H.: Fuseseg: Lidar point cloud segmentation fusing multi-modal data. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. (2020) 1874–1883
44. Meyer, G.P., Charland, J., Hegde, D., Laddha, A., Vallespi-Gonzalez, C.: Sensor fusion for joint 3d object detection and semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. (2019) 0–0
45. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: Rangenet++: Fast and accurate lidar semantic segmentation. In: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS). (2019)
46. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. NeurIPS Workshops (2014)
47. Ba, L.J., Caruana, R.: Do deep nets really need to be deep? NeurIPS (2014) 2654–2662

48. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. (2017) 742–751
49. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. ICLR (2017)
50. Srinivas, S., Fleuret, F.: Knowledge transfer with jacobian matching. In: International Conference on Machine Learning, PMLR (2018) 4723–4731
51. Gupta, S., Hoffman, J., Malik, J.: Cross modal distillation for supervision transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2827–2836
52. Wang, L., Wu, J., Huang, S.L., Zheng, L., Xu, X., Zhang, L., Huang, J.: An efficient approach to informative feature extraction from multimodal data. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 5281–5288
53. Yuan, S., Stenger, B., Kim, T.K.: Rgb-based 3d hand pose estimation via privileged learning with depth images. arXiv preprint arXiv:1811.07376 (2018)
54. Liu, Z., Qi, X., Fu, C.W.: 3d-to-2d distillation for indoor scene parsing. CVPR (2021)
55. Zhao, L., Peng, X., Chen, Y., Kapadia, M., Metaxas, D.N.: Knowledge as priors: Cross-modal knowledge generalization for datasets without superior knowledge. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 6528–6537
56. Liu, Y.C., Huang, Y.K., Chiang, H.Y., Su, H.T., Liu, Z.Y., Chen, C.T., Tseng, C.Y., Hsu, W.H.: Learning from 2d: Pixel-to-point knowledge transfer for 3d pretraining. arXiv preprint arXiv:2104.04687 (2021)
57. Xu, C., Yang, S., Zhai, B., Wu, B., Yue, X., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: Image2point: 3d point-cloud understanding with pretrained 2d convnets. arXiv preprint arXiv:2106.04180 (2021)
58. Yuan, Z., Yan, X., Liao, Y., Guo, Y., Li, G., Cui, S., Li, Z.: X-trans2cap: Cross-modal knowledge transfer using transformer for 3d dense captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 8563–8573
59. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
60. Maas, A.L., Hannun, A.Y., Ng, A.Y., et al.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. Volume 30., Citeseer (2013) 3
61. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2012) 3354–3361
62. 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. arXiv preprint arXiv:2002.10893 (2020)
63. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 9601–9610
64. Rosu, R.A., Schütt, P., Quenzel, J., Behnke, S.: Latticenet: Fast point cloud segmentation using permutohedral lattices. arXiv preprint arXiv:1912.05905 (2019)
65. Duerr, F., Pfaller, M., Weigel, H., Beyerer, J.: Lidar-based recurrent 3d semantic segmentation with temporal memory alignment. In: 2020 International Conference on 3D Vision (3DV), IEEE (2020) 781–790

66. Genova, K., Yin, X., Kundu, A., Pantofaru, C., Cole, F., Sud, A., Brewington, B., Shucker, B., Funkhouser, T.: Learning 3d semantic segmentation with only 2d image supervision. In: 2021 International Conference on 3D Vision (3DV), IEEE (2021) 361–372
67. Huang, Z., Shen, X., Xing, J., Liu, T., Tian, X., Li, H., Deng, B., Huang, J., Hua, X.S.: Revisiting knowledge distillation: An inheritance and exploration framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 3579–3588
68. Jing Yang, Brais Martinez, A.B.G.T.: Knowledge distillation via softmax regression representation learning. In: ICLR2021. (2021)

Supplementary Material

A Training and Inference Details

For the 3D input, we utilize the widely used data augmentation strategy for semantic segmentation, including global scaling with a random scaling factor sampled from [0.95, 1.05], and global rotation around the Z axis with a random angle. For the 2D input, we employ horizontal flipping and color jitter. Each 2D image is cropped to the size 480 × 320 (width × height) for faster training. The 2DPASS is trained in an end-to-end manner with the SGD optimizer. For the SemanticKITTI validation set, our model was trained with batch size 8 and learning rate 0.24 for 64 epochs, which is kept the same as SPVCNN [10] for fair comparison. For the SemanticKITTI online benchmark, we conduct instance CutMix as [42], and fine-tune the last checkpoint with additional 48 epochs. As for the NuScenes dataset, we trained the model with batch size 16 for 80 epochs since the number of points per scene in NuScenes is generally smaller. During the inference, following [10, 40], we apply the voting test-time augmentation, *i.e.*, rotating the input scene with 12 angles around the Z axis and averaging the prediction scores. All experiments are on Nvidia Tesla V100 GPUs.

B Additional Experiments

B.1 Comparing with Multi-Sensor Architecture

To further demonstrate the advantages of our 2DPASS upon multi-sensor methods, we set several multi-sensor baselines and compare against them.

- **PointPainting**: We follow the setup of previous work [15], which exploits the segmentation logits of images and projects them to the LiDAR space by bird's-eye projection [23] or spherical projection [45]. Here, we use several pre-trained backbones, *i.e.*, FCN [19] with ResNet34 [59] and DeepLab_v3 [5], to achieve the 2D semantic segmentation logits. After that, we use outputs of 2D backbones as the inputs of our 3D network.
- **Multi-branch Baseline**: As shown in Fig. 1 (a), we design an ensemble architecture through concatenating the output logits from the two modalities.
- **Multi-branch with Interaction**: Instead of only concatenating the predictions, we also concatenate the 2D features from each layer into the corresponding layers in the 3D network, as illustrated in Fig. 1 (b).
- **2DPASS (light)**: Since above multi-sensor manners are trained with the entire 2D image as input, they are time-consuming and GPU memory cost expensive. So we set all of hidden dimensions as 64 in the 3D network due to GPU memory limitation. This design is different from our manuscript with hidden dimensions 128 due to our light memory cost.

The experiment results are shown in Table 1, where we illustrate the results on NuScenes validation set and inference time (speeds), respectively. As shown

Table 1. Comparison with different multi-sensor manners.

Method	mIoU (%)	Speed (ms)
PointPainting-FCN-ResNet34	76.54	2330
PointPainting-DeepLabV3	76.56	3347
Multi-branch Baseline	77.25	2353
Multi-branch with Interaction	**79.12**	2374
Basline (light)	76.04	40
2DPASS (light)	78.87	40

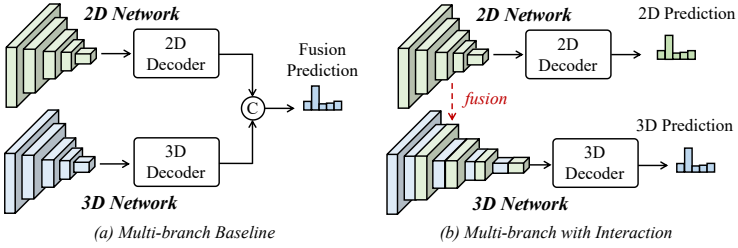


Fig. 1. The illustration of multi-sensor methods.

in Table 1, using naive combination such as PointPainting [15] and concatenation (*i.e.*, Multi-branch Baseline) of prediction cannot improve the segmentation results obviously while introducing huge computational burden (*i.e.*, there are six 1600×900 camera images corresponding to each point cloud). Exploiting feature combination in each scale can slightly improve the performance, but leads to much slower network compared with the pure 3D network. On the contrary, 2DPASS (light) achieves the second-best performance in term of mIoU criterion while $60\times$ speed faster than multi-sensor methods.

B.2 Concrete Results

In this section, we give our detailed results on the NuScenes dataset in Table 2 as a benchmark for future work.

Table 2. Semantic segmentation results on the NuScenes valid set.

Method	Input	**mIoU**	barrier	bicycle	bus	car	construction	motorcycle	pedestrian	traffic cone	trailer	truck	driveable	other flat	sidewalk	terrain	manmade	vegetation
$(AF)^2$-S3Net [41]	L	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
AMVNet [39]	L	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 [40]	L	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
RPVNet [42]	L	77.6	78.2	43.4	92.7	93.2	49.0	**85.7**	80.5	66.0	66.9	84.0	96.9	73.5	75.9	76.0	90.6	88.9
PMF [13]	L+C	76.9	74.1	46.6	89.8	92.1	57.0	77.7	80.9	70.9	64.6	82.9	95.5	73.3	73.6	74.8	89.4	87.7
2D3DNet [66]	L+C	79.0	78.3	**55.1**	95.4	87.7	59.4	79.3	80.7	70.2	68.2	86.6	96.1	**74.9**	75.7	75.1	**91.4**	**89.9**
Baseline	L	76.2	75.3	43.5	95.3	91.2	54.5	78.9	72.8	62.1	70.0	83.2	96.3	73.2	74.2	74.9	88.1	85.9
2DPASS(Ours)	L	**79.4**	78.8	49.6	**95.6**	93.6	60.0	84.1	**82.2**	67.5	72.6	**88.1**	96.8	72.8	76.2	**76.5**	89.4	87.2