Perception-Aware Multi-Sensor Fusion for 3D LiDAR Semantic Segmentation

Zhuangwei Zhuang[1,2] Rong Li[1,2] Kui Jia[1] Qicheng Wang[3] Yuanqing Li[2,1†] Mingkui Tan[1,2†]

[1]South China University of Technology [2]Pazhou Lab [3]Shenzhen Youjia Innov Tech Co., Ltd

{z.zhuangwei, selirong}@mail.scut.edu.cn, wangqicheng@minieye.cc

{auyqli, kuijia, mingkuitan}@scut.edu.cn

Abstract

*3D LiDAR (light detection and ranging) semantic segmentation is important in scene understanding for many applications, such as auto-driving and robotics. For example, for autonomous cars equipped with RGB cameras and LiDAR, it is crucial to fuse complementary information from different sensors for robust and accurate segmentation. Existing fusion-based methods, however, may not achieve promising performance due to the vast difference between the two modalities. In this work, we investigate a collaborative fusion scheme called perception-aware multi-sensor fusion (PMF) to exploit perceptual information from two modalities, namely, appearance information from RGB images and spatio-depth information from point clouds. To this end, we first project point clouds to the camera coordinates to provide spatio-depth information for RGB images. Then, we propose a two-stream network to extract features from the two modalities, separately, and fuse the features by effective residual-based fusion modules. Moreover, we propose additional perception-aware losses to measure the perceptual difference between the two modalities. Extensive experiments on two benchmark data sets show the superiority of our method. For example, on nuScenes, our PMF outperforms the state-of-the-art method by **0.8%** in mIoU.*

1. Introduction

Semantic scene understanding is a fundamental task for many applications, such as auto-driving and robotics [16, 34, 47, 48]. Specifically, in the scenes of auto-driving, it provides fine-grained environmental information for high-level motion planning and improves the safety of autonomous cars [3, 18]. One of the important tasks in semantic scene understanding is semantic segmentation, which assigns a class label to each data point in the input data, and helps autonomous cars to better understand the environment.

According to the sensors used by semantic segmenta-

(a) Input RGB Image

(b) Input Point Cloud

(c) Results of Spherical Projection
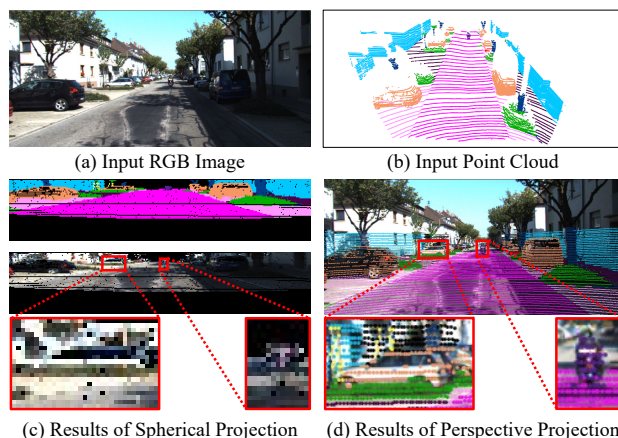
(d) Results of Perspective Projection

Figure 1. Comparisons of spherical projection [40, 54] and perspective projection. With spherical projection, most of the appearance information from RGB images is lost. Instead, we preserve the information of images with perspective projection. To distinguish different classes, we colorize the point clouds using semantic labels from SemanticKITTI.

tion methods, recent studies can be divided into three categories: camera-only methods [2, 9, 10, 35, 58], LiDAR-only methods [1, 13, 26, 54, 62] and multi-sensor fusion methods [30, 37, 39, 52, 59]. Camera-only methods have achieved great progress with the help of a massive amount of open-access data sets [6, 12, 14]. Since images obtained by a camera are rich in appearance information (*e.g.*, texture and color), camera-only methods can provide fine-grained and accurate semantic segmentation results. However, as passive sensors, cameras are susceptible to changes in lighting conditions and are thus unreliable [50].[1] To address this problem, researchers conduct semantic segmentation on point clouds from LiDAR. Compared with camera-only approaches, LiDAR-only methods are more robust to different light conditions, as LiDAR provides reliable and accurate spatio-depth information on the physical world. Unfortunately, LiDAR-only semantic segmentation is challenging due to the sparse and irregular distribution of point clouds.

[†]Corresponding authors.

[1]See Section 4.5 for more details.

(a) RGB Image   (b) Predictions of Car from Image

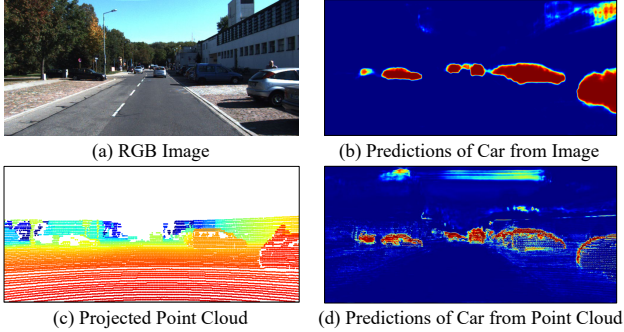(c) Projected Point Cloud   (d) Predictions of Car from Point Cloud

Figure 2. Comparisons of the predictions from images and point clouds. Deep neural networks capture different perceptual information from RGB images and point clouds. Red indicates predictions with higher scores.

In addition, point clouds lack texture and color information, resulting in high classification error in the fine-grained segmentation task of LiDAR-only methods. A straightforward solution for addressing both drawbacks of camera-only and LiDAR-only methods is to fuse the multimodal data from both sensors, *i.e.*, multi-sensor fusion methods. Nevertheless, due to the large domain gap between RGB cameras and LiDAR, multi-sensor fusion is still a nontrivial task.

In multi-sensor fusion methods, fusing multimodal data from different sensors is an important problem. Existing fusion-based methods [37, 52] mainly project dense image features to the LiDAR coordinates using spherical projection [40] and conduct feature fusion in the sparse LiDAR domain. However, these methods suffer from a critical limitation: as the point clouds are very sparse, most of the appearance information from the RGB images is missing after projecting it to the LiDAR coordinates. For example, as shown in Figure 1 (c), the car and motorcycle in the image become distorted with spherical projection. As a result, existing fusion-based methods have difficulty capturing the appearance information from the projected RGB images.

In this paper, we aim to exploit an effective multi-sensor fusion method. Unlike existing methods [37, 52], we assume and highlight that the perceptual information from both RGB images and point clouds, *i.e.*, appearance information from images and spatio-depth information from point clouds, is important in fusion-based semantic segmentation. Based on this intuition, we propose a perception-aware multi-sensor fusion (PMF) scheme that conducts collaborative fusion of perceptual information from two modalities of data in three aspects. **First**, we propose a perspective projection to project the point clouds to the camera coordinate system to obtain additional spatio-depth information for RGB images. **Second**, we propose a two-stream network (TSNet) that contains a camera stream and a LiDAR stream to extract perceptual features from multimodal sensors separately. Considering that the information from images is unreliable in an outdoor environment, we fuse the image features to

the LiDAR stream by effective residual-based fusion (RF) modules, which are designed to learn the complementary features of the original LiDAR modules. **Third**, we propose perception-aware losses to measure the vast perceptual difference between the two data modalities and boost the fusion of different perceptual information. Specifically, as shown in Figure 2, the perceptual features captured by the camera stream and LiDAR stream are different. Therefore, we use the predictions with higher confidence to supervise those with lower confidence.

Our contributions are summarized as follows. First, we propose a perception-aware multi-sensor fusion (PMF) scheme to effectively fuse the perceptual information from RGB images and point clouds. Second, by fusing the spatio-depth information from point clouds and appearance information from RGB images, PMF is able to address segmentation with undesired light conditions and sparse point clouds. More critically, PMF is robust to adversarial samples of RGB images by integrating the information from point clouds. Third, we introduce perception-aware losses into the network and force the network to capture the perceptual information from two different-modality sensors. The extensive experiments on two benchmark data sets demonstrate the superior performance of our method. For example, on nuScenes [7], PMF outperforms Cylinder3D [64], a state-of-the-art LiDAR-only method, by **0.8%** in mIoU.

2. Related Work

In this section, we revisit the existing literature on 2D and 3D semantic segmentation, *i.e.*, camera-only methods, LiDAR-only methods and multi-sensor fusion methods.

2.1. Camera-Only Methods

Camera-only semantic segmentation aims to predict the pixel-wise labels of 2D images. FCN [35] is a fundamental work in semantic segmentation, which proposes an end-to-end fully convolutional architecture based on image classification networks. In addition to FCN, recent works have achieved significant improvements via exploring multi-scale information [9, 31, 63], dilated convolution [10, 38, 53], and attention mechanisms [27, 58]. However, camera-only methods are easily disturbed by lighting (*e.g.*, underexposure or overexposure) and may not be robust to outdoor scenes.

2.2. LiDAR-Only Methods

To address the drawbacks of cameras, LiDAR is an important sensor on an autonomous car, as it is robust to more complex scenes. According to the preprocessing pipeline, existing methods for point clouds mainly contains two categories, including direct methods [26, 44, 45, 64] and projection-based methods [13, 54, 55, 56]. Direct methods perform semantic segmentation by processing the raw 3D point clouds directly. PointNet [44] is a pioneering work in this category
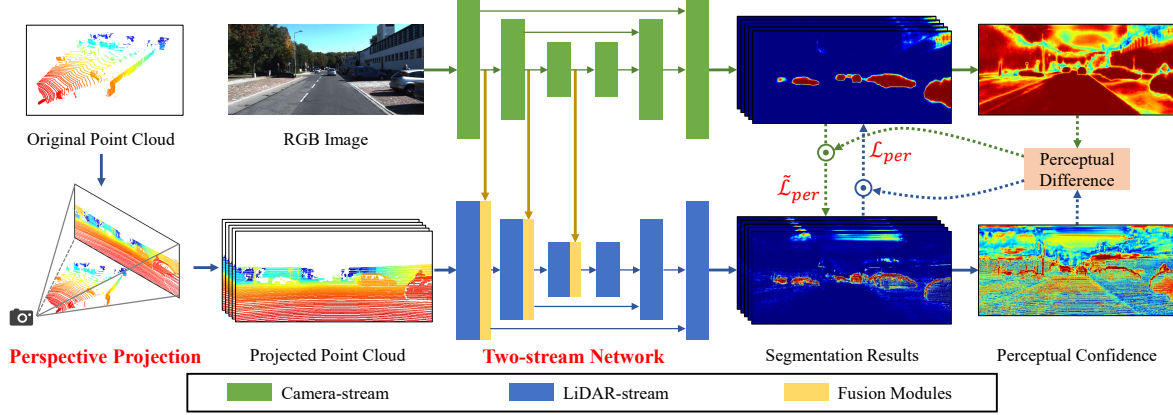
Figure 3. Illustration of perception-aware multi-sensor fusion (PMF). PMF consists of three components: (1) perspective projection; (2) a two-stream network (TSNet) with feature fusion modules; and (3) perception-aware losses $\mathcal{L}_{per}, \widetilde{\mathcal{L}}_{per}$ w.r.t. the camera stream and the LiDAR stream. We first project the point clouds to camera coordinate with perspective projection and learn the features from both the RGB images and point clouds using TSNet. The image features are fused into the LiDAR stream network by fusion modules. Last, we use perception-aware losses to help the network focus on the perceptual features of both images and point clouds.

that extracts point cloud features by multi-layer perception. A subsequent extension, *i.e.*, PointNet++ [45], further aggregates a multi-scale sampling mechanism to aggregate global and local features. However, these methods do not consider the varying sparsity of point clouds in outdoor scenes. Cylinder3D [64] addresses this issue by using 3D cylindrical partitions and asymmetrical 3D convolutional networks. However, direct methods have a high computational complexity, which limits their applicability in auto-driving. Projection-based methods are more efficient because they convert 3D point clouds to a 2D grid. In projection-based methods, researchers focus on exploiting effective projection methods, such as spherical projection [40, 54] and bird's-eye projection [62]. Such 2D representations allow researchers to investigate efficient network architectures based on existing 2D convolutional networks [1, 13, 21]. In addition to projection-based methods, one can easily improve the efficiency of networks by existing neural architecture search [8, 22, 42] and model compression techniques [23, 33, 57].

2.3. Multi-Sensor Fusion Methods

To leverage the benefits of both camera and LiDAR, recent work has attempted to fuse information from two complementary sensors to improve the accuracy and robustness of the 3D semantic segmentation algorithm [30, 37, 39, 52]. RGBAL [37] converts RGB images to a polar-grid mapping representation and designs early and mid-level fusion strategies. PointPainting [52] obtains the segmentation results of images and projects them to the LiDAR space by using bird's-eye projection [62] or spherical projection [40]. The projected segmentation scores are concatenated with the original point cloud features to improve the performance of LiDAR networks. Unlike existing methods that perform

Algorithm 1 General Scheme of PMF
***
Require: Training data $\{\mathbf{P}, \mathbf{X}, \mathbf{y}\}$, TSNet with submodels $M, \widetilde{M}$, hyperparameters τ, λ, γ.
***
1: **while** *not convergent* **do**
2:     Project the point clouds $\mathbf{P}$ by using perspective projection to obtain $\widetilde{\mathbf{X}}$.
3:     Use $\{\widetilde{\mathbf{X}}, \mathbf{X}\}$ as the inputs of TSNet and compute the output probabilities $\{\widetilde{\mathbf{O}}, \mathbf{O}\}$ with Eq. (2).
4:     Compute the perceptual confidence $\widetilde{\mathbf{C}}$ and $\mathbf{C}$.
5:     Construct perception-aware losses to measure the perceptual difference with Eqs. (7) and (10).
6:     Update $\widetilde{M}$ and M by minimizing the objective in Eqs. (8) and (11).
7: **end while**
***

feature fusion in the LiDAR domain, PMF exploits a collaborative fusion of multimodal data in camera coordinates.

3. Proposed Method

In this work, we propose a perception-aware multi-sensor fusion (PMF) scheme to perform effective fusion of the perceptual information from both RGB images and point clouds. Specifically, as shown in Figure 3, PMF contains three components: (1) perspective projection; (2) a two-stream network (TSNet) with residual-based fusion modules; (3) perception-aware losses. The general scheme of PMF is shown in Algorithm 1. We first project the point clouds to the camera coordinate system by using perspective projection. Then, we use a two-stream network that contains a camera stream and a LiDAR stream to extract perceptual features from the two modalities, separately. The features from the

camera stream are fused into the LiDAR stream by residual-based fusion modules. Finally, we introduce perception-aware losses into the optimization of the network.

3.1. Formulation of Perspective Projection

Existing methods [37, 52] mainly project images to the LiDAR coordinate system using spherical projection. However, due to the sparse nature of point clouds, most of the appearance information from the images is lost with spherical projection (see Figure 1). To address this issue, we propose perspective projection to project the sparse point clouds to the camera coordinate system.

Let $\{\mathbf{P}, \mathbf{X}, \mathbf{y}\}$ be one of the training samples from a given data set, where $\mathbf{P} \in \mathbb{R}^{4 \times N}$ indicates a point cloud from LiDAR and N denotes the number of points. Each point $\mathbf{P}_i$ in point cloud $\mathbf{P}$ consists of 3D coordinates (x, y, z) and a reflectance value (r). Let $\mathbf{X} \in \mathbb{R}^{3 \times H \times W}$ be an image from an RGB camera, where H and W represent the height and width of the image, respectively. $\mathbf{y} \in \mathbb{R}^N$ is the set of semantic labels for point cloud $\mathbf{P}$.

In perspective projection, we aim to project the point cloud $\mathbf{P}$ from LiDAR coordinate to the camera coordinate to obtain the 2D LiDAR features $\widetilde{\mathbf{X}} \in \mathbb{R}^{C \times H \times W}$. Here, C indicates the number of channels w.r.t. the projected point cloud. Following [17], we obtain $\mathbf{P}_i = (x, y, z, 1)^\top$ by appending a fourth column to $\mathbf{P}_i$ and compute the projected point $\widetilde{\mathbf{P}}_i = (\widetilde{x}, \widetilde{y}, \widetilde{z})^\top$ in the camera coordinates by

$$\widetilde{\mathbf{P}}_i = \mathbf{T}\mathbf{R}\mathbf{P}_i, \tag{1}$$

where $\mathbf{T} \in \mathbb{R}^{3 \times 4}$ is the projection matrix from LiDAR coordinates to camera coordinates. $\mathbf{R} \in \mathbb{R}^{4 \times 4}$ is expanded from the rectifying rotation matrix $\mathbf{R}^{(0)} \in \mathbb{R}^{3 \times 3}$ by appending a fourth zero row and column and setting $\mathbf{R}(4, 4) = 1$. The calibration parameters $\mathbf{T}$ and $\mathbf{R}^{(0)}$ can be obtained by the approach in [19]. Subsequently, the corresponding pixel (h, w) in the projected image $\widetilde{\mathbf{X}}$ w.r.t. the point $\mathbf{P}_i$ is computed by $h = \widetilde{x}/\widetilde{z}$ and $w = \widetilde{y}/\widetilde{z}$.

Because the point cloud is very sparse, each pixel in the projected $\widetilde{\mathbf{X}}$ may not have a corresponding point $\mathbf{p}$. Therefore, we first initialize all pixels in $\widetilde{\mathbf{X}}$ to 0. Following [13], we then compute 5-channel LiDAR features, *i.e.*, (d, x, y, z, r), for each pixel (h, w) in the projected 2D image $\widetilde{\mathbf{X}}$, where $d = \sqrt{x^2 + y^2 + z^2}$ represents the range value of each point.

3.2. Architecture Design of PMF

As images and point clouds are different-modality data, it is difficult to handle both types of information from the two modalities by using a single network [30]. Motivated by [15, 49], we propose a two-stream network (TSNet) that contains a camera stream and a LiDAR stream to process the features from camera and LiDAR, separately, as illustrated
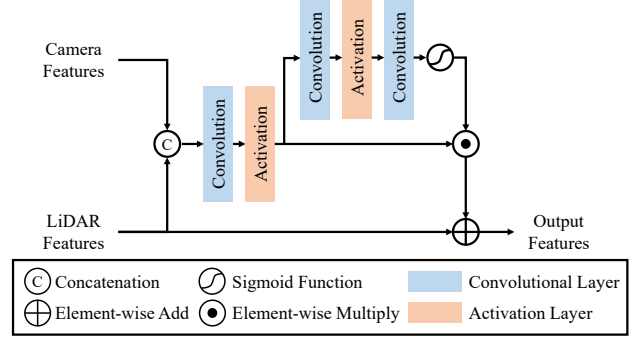

Figure 4. Illustration of the residual-based fusion (RF) module. RF fuses features from both the camera and LiDAR to generate the complementary information of the original LiDAR features.

in Figure 3. In this way, we can use the network architectures designed for images and point clouds as the backbones of each stream in TSNet.

Let $\widetilde{M}$ and M be the LiDAR stream and the camera stream in TSNet, respectively. Let $\widetilde{\mathbf{O}} \in \mathbb{R}^{S \times H \times W}$ and $\mathbf{O} \in \mathbb{R}^{S \times H \times W}$ be the output probabilities w.r.t. each network, where S indicates the number of semantic classes. The outputs of TSNet are computed by

$$\begin{cases} \mathbf{O} = M(\mathbf{X}), \\ \widetilde{\mathbf{O}} = \widetilde{M}(\widetilde{\mathbf{X}}). \end{cases} \tag{2}$$

Since the features of images contain many details of objects, we then introduce a residual-based fusion module, as illustrated in Figure 4, to fuse the image features to the LiDAR stream. Let $\{\mathbf{F}_l \in \mathbb{R}^{C_l \times H_l \times W_l}\}_{l=1}^L$ be a set of image features from the camera stream, where l indicates the layer in which we obtain the features. C_l indicates the number of channels of the l-th layer in the camera stream. H_l and W_l indicate the height and width of the feature maps from the l-th layer, respectively. Let $\{\widetilde{\mathbf{F}}_l \in \mathbb{R}^{\widetilde{C}_l \times H_l \times W_l}\}_{l=1}^L$ be the features from the LiDAR stream, where $\widetilde{C}_l$ indicates the number of channels of the l-th layer in the LiDAR stream. To obtain the fused features, we first concatenate the features from each network and use a convolutional layer to reduce the number of channels of the fused features. The fused features $\mathbf{F}_l^{fuse} \in \mathbb{R}^{\widetilde{C}_l \times H_l \times W_l}$ are computed by

$$\mathbf{F}_l^{fuse} = f_l([\widetilde{\mathbf{F}}_l; \mathbf{F}_l]), \tag{3}$$

where $[\cdot; \cdot]$ indicates the concatenation operation. $f_l(\cdot)$ is the convolution operation w.r.t. the l-th fusion module.

Considering that the camera is easily affected by different lighting and weather conditions, the information from RGB images is not reliable in an outdoor environment. We use the fused features as the complement of the original LiDAR features and design the fusion module based on the residual structure [24]. Incorporating with the attention module [5],

the output features $\mathbf{F}_l^{out} \in \mathbb{R}^{\widetilde{C}_l \times H_l \times W_l}$ of the fusion module are computed by

$$\mathbf{F}_l^{out} = \widetilde{\mathbf{F}}_l + \sigma(g_l(\mathbf{F}_l^{fuse})) \odot \mathbf{F}_l^{fuse}, \qquad (4)$$

where $\sigma(x) = 1/(1 + e^{-x})$ indicates the sigmoid function. $g_l(\cdot)$ indicates the convolution operation in the attention module w.r.t. the l-th fusion module. $\odot$ indicates the element-wise multiplication operation.

3.3. Construction of Perception-Aware Loss

The construction of perception-aware loss is very important in our method. As demonstrated in Figure 2, because the point clouds are very sparse, the LiDAR stream network learns only the local features of points while ignoring the shape of objects. In contrast, the camera stream can easily capture the shape and texture of objects from dense images. In other words, the perceptual features captured by the camera stream and LiDAR stream are different. With this intuition, we introduce a perception-aware loss to make the fusion network focus on the perceptual features from the camera and LiDAR.

To measure the perceptual confidence of the predictions w.r.t. the LiDAR stream, we first compute the entropy map $\widetilde{\mathbf{E}} \in \mathbb{R}^{H \times W}$ by

$$\widetilde{\mathbf{E}}_{h,w} = -\frac{1}{\log S} \sum_{s=1}^{S} \widetilde{\mathbf{O}}_{s,h,w} \log(\widetilde{\mathbf{O}}_{s,h,w}). \qquad (5)$$

Following [46], we use $\log S$ to normalize the entropy to $(0, 1]$. Then, the perceptual confidence map $\widetilde{\mathbf{C}}$ w.r.t. the LiDAR stream is computed by $\widetilde{\mathbf{C}} = \mathbf{1} - \widetilde{\mathbf{E}}$. For the camera stream, the confidence map is computed by $\mathbf{C} = \mathbf{1} - \mathbf{E}$.

Note that not all information from the camera stream is useful. For example, the camera stream is confident inside objects but may make mistakes at the edge. In addition, the predictions with lower confidence scores are more likely to be wrong. Incorporating with a confidence threshold, we measure the importance of perceptual information from the camera stream by

$$\widetilde{\mathbf{\Omega}}_{h,w} = \begin{cases} \max(\widetilde{\mathbf{C}}_{h,w} - \mathbf{C}_{h,w}, 0), & \text{if } \widetilde{\mathbf{C}}_{h,w} > \tau, \\ 0, & \text{otherwise.} \end{cases} \qquad (6)$$

Here τ indicates the confidence threshold.

Inspired by [25, 28, 61], to learn the perceptual information from the camera stream, we construct the perception-aware loss w.r.t. the LiDAR stream by

$$\widetilde{\mathcal{L}}_{per} = \frac{1}{Q} \sum_{h=1}^{H} \sum_{w=1}^{W} \widetilde{\mathbf{\Omega}}_{h,w} D_{KL}(\widetilde{\mathbf{O}}_{:,h,w} || \mathbf{O}_{:,h,w}), \qquad (7)$$

where $Q = H \cdot W$ and $D_{KL}(\cdot || \cdot)$ indicates the Kullback-Leibler divergence [25].

In addition to the perception-aware loss, we also use multi-class focal loss [32] and Lovász-softmax loss [4], which are commonly used in existing segmentation work [13, 64], to train the LiDAR stream.[2]

The objective w.r.t. the LiDAR stream is defined by

$$\widetilde{\mathcal{L}} = \widetilde{\mathcal{L}}_{foc} + \lambda \widetilde{\mathcal{L}}_{lov} + \gamma \widetilde{\mathcal{L}}_{per}, \qquad (8)$$

where $\widetilde{\mathcal{L}}_{foc}$ and $\widetilde{\mathcal{L}}_{lov}$ indicate the multi-class focal loss and Lovász-softmax loss, respectively. λ and γ are the hyperparameters that balance different losses.

Similar to the LiDAR stream, we construct the objective for the optimization of the camera stream. Following Eq. (6), the importance of the information from the LiDAR stream is computed by

$$\mathbf{\Omega}_{h,w} = \begin{cases} \max(\mathbf{C}_{h,w} - \widetilde{\mathbf{C}}_{h,w}, 0), & \text{if } \mathbf{C}_{h,w} > \tau, \\ 0, & \text{otherwise.} \end{cases} \qquad (9)$$

The perception-aware loss w.r.t. the camera stream is

$$\mathcal{L}_{per} = \frac{1}{Q} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{\Omega}_{h,w} D_{KL}(\mathbf{O}_{:,h,w} || \widetilde{\mathbf{O}}_{:,h,w}). \qquad (10)$$

Then the objective w.r.t. the camera stream is defined by

$$\mathcal{L} = \mathcal{L}_{foc} + \lambda \mathcal{L}_{lov} + \gamma \mathcal{L}_{per}. \qquad (11)$$

4. Experiments

In this section, we empirically evaluate the performance of PMF on the benchmark data sets, including SemanticKITTI [3] and nuScenes [7]. SemanticKITTI is a large-scale data set based on the KITTI Odometry Benchmark [18], providing 43,000 scans with pointwise semantic annotation, where 21,000 scans (sequence 00-10) are available for training and validation. The data set has 19 semantic classes for the evaluation of semantic benchmarks. nuScenes contains 1,000 driving scenes with different weather and light conditions. The scenes are split into 28,130 training frames and 6,019 validation frames. Unlike SemanticKITTI, which provides only the images of the front-view camera, nuScenes has 6 cameras for different views of LiDAR.

4.1. Implementation Details

We implement the proposed method in PyTorch [43], and use ResNet-34 [24] and SalsaNext [13] as the backbones of the camera stream and LiDAR stream, respectively. Because we process the point clouds in the camera coordinates, we incorporate ASPP [9] into the LiDAR stream network to adjust the receptive field adaptively. To leverage the benefits

---

[2]The details of the multi-class focal loss and Lovász-softmax loss can be found in the supplementary material.

Table 1. Comparisons on SemanticKITTI validation set. **L** indicates LiDAR-only methods. **L+C** indicates fusion-based methods. * indicates the results based on our implementation. The **bold** numbers indicate the best results, and the blue numbers indicate the second best results.

Method	Input	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mIoU (%)
#Points (k)	-	6384	44	52	101	471	127	129	5	21434	974	8149	67	6304	1691	20391	882	8125	317	64	-
RandLANet [26]	L	92.0	8.0	12.8	74.8	46.7	52.3	46.0	0.0	93.4	32.7	73.4	0.1	84.0	43.5	83.7	57.3	73.1	48.0	27.3	50.0
RangeNet++ [40]	L	89.4	26.5	48.4	33.9	26.7	54.8	69.4	0.0	92.9	37.0	69.9	0.0	83.4	51.0	83.3	54.0	68.1	49.8	34.0	51.2
SequeezeSegV2 [55]	L	82.7	15.1	22.7	25.6	26.9	22.9	44.5	0.0	92.7	39.7	70.7	0.1	71.6	37.0	74.6	35.8	68.1	21.8	22.2	40.8
SequeezeSegV3 [56]	L	87.1	34.3	48.6	47.5	47.1	58.1	53.8	0.0	95.3	43.1	78.2	0.3	78.9	53.2	82.3	55.5	70.4	46.3	33.2	53.3
SalsaNext [13]	L	90.5	44.6	49.6	**86.3**	54.6	74.0	81.4	0.0	93.4	40.6	69.1	0.0	84.6	53.0	83.6	64.3	64.2	54.4	39.8	59.4
MinkowskiNet [11]	L	95.0	23.9	50.4	55.3	45.9	65.6	82.2	0.0	94.3	43.7	76.4	0.0	87.9	57.6	87.4	67.7	71.5	63.5	**43.6**	58.5
SPVNAS [51]	L	**96.5**	44.8	63.1	59.9	64.3	72.0	**86.0**	0.0	93.9	42.4	75.9	0.0	**88.8**	59.1	**88.0**	67.5	73.0	63.5	44.3	62.3
Cylinder3D [64]	L	96.4	**61.5**	**78.2**	66.3	69.8	**80.8**	**93.3**	0.0	94.9	41.5	78.0	**1.4**	87.5	50.0	86.7	**72.2**	68.8	63.0	42.1	**64.9**
PointPainting* [52]	L+C	94.7	17.7	35.0	28.8	55.0	59.4	63.6	0.0	95.3	39.9	77.6	0.4	87.5	55.1	87.7	67.0	72.9	61.8	36.5	54.5
RGBAL* [37]	L+C	87.3	36.1	26.4	64.6	54.6	58.1	72.7	0.0	95.1	**45.6**	77.5	0.8	78.9	53.4	84.3	61.7	72.9	56.1	41.5	56.2
PMF (Ours)	L+C	95.4	47.8	62.9	68.4	**75.2**	78.9	71.6	0.0	**96.4**	43.5	**80.5**	0.1	88.7	**60.1**	**88.6**	72.7	**75.3**	**65.5**	43.0	63.9

Table 2. Comparisons on the nuScenes validation set. The **bold** numbers indicate the best results.

Method	barrier	bicycle	bus	car	construction	motorcycle	pedestrian	traffic-cone	trailer	truck	driveable	other-flat	sidewalk	terrain	manmade	vegetation	mIoU (%)
#Points (k)	1629	21	851	6130	194	81	417	112	370	2560	56048	1972	12631	13620	31667	21948	-
RangeNet++ [40]	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	65.5
PolarNet [62]	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	71.0
Salsanext [13]	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	72.2
Cylinder3D [64]	**76.4**	40.3	**91.3**	**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	76.1
PMF (Ours)	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**	**76.9**

of existing image classification models, we initialize the parameters of ResNet-34 with the pretrained ImageNet models from [43]. We also adopt hybrid optimization methods [60] to train the networks w.r.t. different modalities, *i.e.*, SGD with Nesterov [41] for the camera stream and Adam [29] for the LiDAR stream. We train the networks for 50 epochs on both the benchmark data sets. The learning rate starts at 0.001 and decays to 0 with a cosine policy [36]. We set the batch size to 8 on SemanticKITTI and 24 on nuScenes. We set τ, γ, λ to 0.7, 0.5, and 1.0, respectively.[3] To prevent overfitting, a series of data augmentation strategies are used, including random horizontal flipping, color jitter, 2D random rotation, and random cropping. Our source code is available at https://github.com/ICEORY/PMF.

4.2. Results on SemanticKITTI

To evaluate our method on SemanticKITTI, we compare PMF with several state-of-the-art LiDAR-only methods including SalsaNext [13], Cylinder3D [64], etc. Since SemanticKITTI provides only the images of the front-view camera, we project the point clouds to a perspective view and keep only the available points on the images to build a subset of SemanticKITTI. Following [13, 28, 64], we use sequence 08 for validation. The remaining sequences (00-07 and 09-10) are used as the training set. We evaluate the

---

[3]We investigate the effect of τ, γ, λ in the supplementary material.
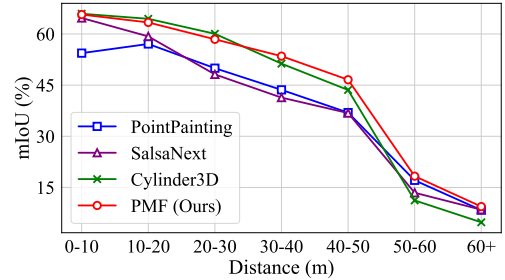

Figure 5. Distance-based evaluation on SemanticKITTI. As the distance increases, the point cloud become sparse.

release models of the state-of-the-art LiDAR-only methods on our data set. Because SPVNAS [51] did not release its best model, we report the result of the best-released model (with 65G MACs). In addition, we reimplement two state-of-the-art fusion-based methods, *i.e.*, RGBAL [37] and PointPainting [52], on our data set.

From Table 1, PMF achieves the best performance among projection-based methods. For example, PMF outperforms SalsaNext by **4.5%** in mIoU. However, PMF performs worse than the state-of-the-art 3D convolutional method, *i.e.*, Cylinder3D, by 1.0% in mIoU. As long-distance perception is also critical to the safety of autonomous cars, we also conduct a distance-based evaluation on SemanticKITTI. From Figure 5, because the point clouds becomes sparse when the distance increases, LiDAR-only methods suffer from great perfor-

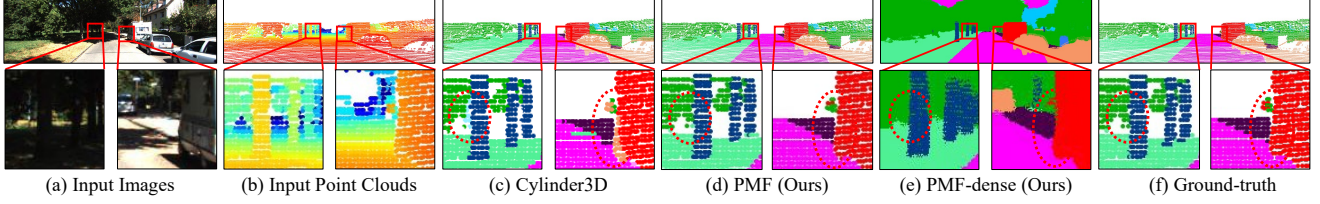(a) Input Images   (b) Input Point Clouds   (c) Cylinder3D   (d) PMF (Ours)   (e) PMF-dense (Ours)   (f) Ground-truth

Figure 6. Qualitative results on SemanticKITTI. The red dashed circle indicates the difference between the results of PMF and the baseline.
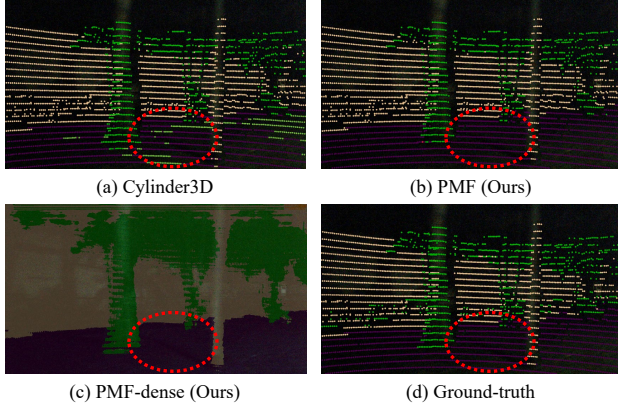

(a) Cylinder3D   (b) PMF (Ours)

(c) PMF-dense (Ours)   (d) Ground-truth

Figure 7. Qualitative results on nuScenes. We use the corresponding images (night) as the background of both the predictions and labels. We highlight the difference between the results of PMF and the baseline with the red dashed circle.

mance degradation at long distances. In contrast, since the images provide more information for distant objects, fusion-based methods outperform LiDAR-only methods at large distances. Specifically, PMF achieves the best performance when the distance is larger than 30 meters. This suggests that our method is more suitable to address segmentation with sparse point clouds. This ability originates from our fusion strategy, which effectively incorporates RGB images.

4.3. Results on nuScenes

Following [64], to evaluate our method on more complex scenes, we compare PMF with the state-of-the-art methods on the nuScenes LiDAR-seg validation set. The experimental results are shown in Table 2. Note that the point clouds of nuScenes are sparser than those of SemanticKITTI (35k points/frame vs. 125k points/frame). Thus, it is more challenging for 3D segmentation tasks. In this case, PMF achieves the best performance compared with the LiDAR-only methods. Specifically, PMF outperforms Cylinder3D by **0.8%** in mIoU. Moreover, compared with the state-of-the-art 2D convolutional method, *i.e.*, SalsaNext, PMF achieves a **4.7%** improvement in mIoU. These results are consistent with our expectation. Since PMF incorporates RGB images, our fusion strategy is capable of addressing such challenging segmentation under sparse point clouds.

(a) Input Images with Noise   (b) Input Point Clouds

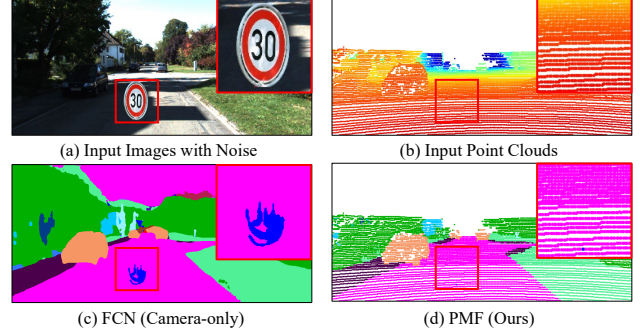(c) FCN (Camera-only)   (d) PMF (Ours)

Figure 8. Comparisons of PMF and camera-only methods on adversarial samples. The camera-only methods use only RGB images as inputs, while PMF uses both images and point clouds as inputs. We highlight the inserted traffic sign with red box.

4.4. Qualitative Evaluation

To better understand the benefits of PMF, we visualize the predictions of PMF on the benchmark data sets.[4] From Figure 6, compared with Cylinder3D, PMF achieves better performance at the edges of objects. For example, as shown in Figure 6 (d), the truck segmented by PMF has a more complete shape. More critically, PMF is robust to different lighting conditions. Specifically, as illustrated in Figure 7, PMF outperforms the baselines on more challenging scenes (*e.g.*, night). In addition, as demonstrated in Figure 6 (e) and Figure 7 (c), PMF generates dense segmentation results that combine the benefits of both the camera and LiDAR, which is significantly different from existing LiDAR-only and fusion-based methods.

4.5. Adversarial Analysis

To investigate the robustness of PMF on adversarial samples, we first insert extra objects (*e.g.*, a traffic sign) to the images and keeping the point clouds unchanged.[5] In addition, we implement a camera-only method, *i.e.*, FCN [35], on SemanticKITTI as the baseline. Note that we do not use any adversarial training technique during training. As demonstrated in Figure 8, the camera-only methods are easily affected by changes in the input images. In contrast,

---

[4]More visualization results on SemanticKITTI and nuScenes are shown in the supplementary material.

[5]More adversarial samples are shown in the supplementary material.

Table 3. Inference time of different methods on GeForce RTX 3090 using TensorRT. "-" indicates that the results are not available. For fair comparison, Cylinder3D is accelerated by sparse convolution.

Method	#FLOPs	#Params.	Inference time	mIoU	
				nuScenes	SemanticKITTI
PointPainting [52]	51.0 G	28.1 M	2.3 ms	-	54.5%
RGBAL [37]	55.0 G	13.2 M	2.7 ms	-	56.2%
SalsaNext [13]	31.4 G	6.7 M	1.6 ms	72.2%	59.4%
Cylinder3D [64]	-	55.9 M	62.5 ms	76.1%	**64.9%**
PMF (Ours)	854.7 G	36.3 M	22.3 ms	**76.9%**	63.9%

Table 4. Ablation study for the network components on the SemanticKITTI validation set. **PP** denotes perspective projection. **RF** denotes the residual-based fusion module. **PL** denotes perception-aware loss. The **bold** number is the best result.

	Baseline	PP	ASPP	RF	PL	mIoU (%)
1	✓					57.2
2	✓	✓				57.6
3	✓	✓	✓			59.7
4	✓		✓	✓		55.8
5	✓	✓	✓	✓		61.7
6	✓	✓	✓	✓	✓	**63.9**

because PMF integrates reliable point cloud information, the noise in the images is reduced during feature fusion and imposes only a slight effect on the model performance.

4.6. Efficiency Analysis

In this section, we evaluate the efficiency of PMF on GeForce RTX 3090. Note that we consider the efficiency of PMF in two aspects. First, since predictions of the camera stream are fused into the LiDAR stream, we remove the decoder of the camera stream to speed up the inference. Second, our PMF is built on 2D convolutions and can be easily optimized by existing inference toolkits, *e.g.*, TensorRT. In contrast, Cylinder3D is built on 3D sparse convolutions [20] and is difficult to be accelerated by TensorRT. We report the inference time of different models optimized by TensorRT in Table 3. From the results, our PMF achieves the best performance on nuScenes and is $2.8\times$ faster than Cylinder3D (22.3 ms vs. 62.5 ms) with fewer parameters.

5. Ablation Study

5.1. Effect of Network Components

We study the effect of the network components of PMF, *i.e.*, perspective projection, ASPP, residual-based fusion modules, and perception-aware loss. The experimental results are shown in Table 4. Since we use only the front-view point clouds of SemanticKITTI, we train SalsaNext as the baseline on our data set using the officially released code. Comparing the first and second lines in Table 4, perspective projection achieves only a 0.4% mIoU improvement over spherical projection with LiDAR-only input. In contrast, comparing the fourth and fifth lines, perspective projection

(a) Predictions of Car without PL   (b) Predictions of Car with PL
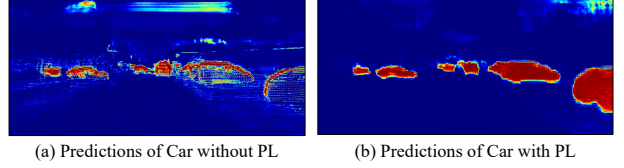
Figure 9. Comparisons of the predictions w.r.t. the networks trained with and without perception-aware loss. **PL** denotes the perception-aware loss. Red indicates predictions with higher confidence scores. We only show the predictions of Car for the sake of clarity.

brings a 5.9% mIoU improvement over spherical projection with multimodal data inputs. From the third and fifth lines, our fusion modules bring 2.0% mIoU improvement to the fusion network. Moreover, comparing the fifth and sixth lines, the perception-aware losses improve the performance of the network by 2.2% in mIoU.

5.2. Effect of Perception-Aware Loss

To investigate the effect of perception-aware loss, we visualize the predictions of the LiDAR stream networks with and without perception-aware loss in Figure 9. From the results, perception-aware loss helps the LiDAR stream capture the perceptual information from the images. For example, the model trained with perception-aware loss learns the complete shape of cars, while the baseline model focuses only on the local features of points. As the perception-aware loss introduces the perceptual difference between the RGB images and the point clouds, it enables an effective fusion of the perceptual information from the data of both modalities. As a result, our PMF generates dense predictions that combine the benefits of both the images and point clouds.

6. Conclusion

In this work, we have proposed a perception-aware multi-sensor fusion scheme for 3D LiDAR semantic segmentation. Unlike existing methods that conduct feature fusion in the LiDAR coordinate system, we project the point clouds to the camera coordinate system to enable a collaborative fusion of the perceptual features from the two modalities. Moreover, by fusing complementary information from both cameras and LiDAR, PMF is robust to complex outdoor scene. The experimental results on two benchmarks show the superiority of our method. In the future, we will extend PMF to other challenging tasks in auto-driving, *e.g.*, object detection.

References

[1] Eren Erdal Aksoy, Saimir Baci, and Selcuk Cavdar. Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving. *IEEE Intelligent Vehicles Symposium*, pages 926–932, 2020. 1, 3

[2] Vijay Badrinarayanan, Alex Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2481–2495, 2017. 1

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *IEEE International Conference on Computer Vision*, pages 9297–9307, 2019. 1, 5

[4] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018. 5

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 4

[6] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision*, pages 44–57. Springer, 2008. 1

[7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 2, 5

[8] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019. 3

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017. 1, 2, 5

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1, 2

[11] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 6

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 1

[13] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast semantic segmentation of lidar point clouds for autonomous driving. *arXiv preprint arXiv:2003.03653*, 2020. 1, 2, 3, 4, 5, 6, 8

[14] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34, 2012. 1

[15] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016. 4

[16] Chuang Gan, Hang Zhao, Peihao Chen, David Cox, and Antonio Torralba. Self-supervised moving vehicle tracking with stereo sound. In *IEEE International Conference on Computer Vision*, pages 7053–7062, 2019. 1

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 4

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 1, 5

[19] Andreas Geiger, Frank Moosmann, Oemer Car, and Bernhard Schuster. A toolbox for automatic calibration of range and camera sensors using a single shot. In *International Conference on Robotics and Automation*, 2012. 4

[20] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, 2018. 8

[21] Yong Guo, Yin Zheng, Mingkui Tan, Qi Chen, Jian Chen, Peilin Zhao, and Junzhou Huang. Nat: Neural architecture transformer for accurate and compact architectures. In *Advances in Neural Information Processing Systems*, 2019. 3

[22] Yong Guo, Yin Zheng, Mingkui Tan, Qi Chen, Zhipeng Li, Jian Chen, Peilin Zhao, and Junzhou Huang. Towards accurate and compact architectures via neural architecture transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

[23] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2015. 3

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4, 5

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 5

[26] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 1, 2, 6

[27] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 603–612, 2019. 2

[28] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12605–12614, 2020. 5, 6

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 6

[30] Georg Krispel, Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Fuseseg: Lidar point cloud segmentation fusing multi-modal data. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1874–1883, 2020. 1, 3, 4

[31] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2016. 2

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 5

[33] Jing Liu, Bohan Zhuang, Zhuangwei Zhuang, Yong Guo, Junzhou Huang, Jinhui Zhu, and Mingkui Tan. Discrimination-aware network pruning for deep model compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

[34] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, P. Yin, Wen Chen, Hesheng Wang, Haoang Li, and Y. Liu. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. pages 2831–2840, 2019. 1

[35] J. Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1, 2, 7

[36] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 6

[37] Khaled El Madawy, H. Rashed, Ahmad El Sallab, O. Nasr, H. Kamel, and S. Yogamani. Rgb and lidar fusion based 3d semantic segmentation for autonomous driving. *IEEE Intelligent Transportation Systems Conference*, pages 7–12, 2019. 1, 2, 3, 4, 6, 8

[38] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *European Conference on Computer Vision*, pages 552–568, 2018. 2

[39] Gregory P Meyer, Jake Charland, Darshan Hegde, Ankit Laddha, and Carlos Vallespi-Gonzalez. Sensor fusion for joint 3d object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 3

[40] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4213–4220. IEEE, 2019. 1, 2, 3, 6

[41] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Proceedings of the USSR Academy of Sciences*, volume 269, pages 543–547, 1983. 6

[42] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yong Guo, Peilin Zhao, Junzhou Huang, and Mingkui Tan. Disturbance-immune weight sharing for neural architecture search. *arXiv preprint arXiv:2003.13089*, 2020. 3

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037, 2019. 5, 6

[44] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 2

[45] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 2, 3

[46] Alfréd Rényi et al. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961. 5

[47] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008. 1

[48] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4758–4765. IEEE, 2018. 1

[49] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014. 4

[50] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. Darts: Deceiving autonomous cars with toxic signs. *arXiv preprint arXiv:1802.06430*, 2018. 1

[51] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision*, pages 685–702. Springer, 2020. 6

[52] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4604–4612, 2020. 1, 2, 3, 4, 6, 8

[53] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1451–1460. IEEE, 2018. 2

[54] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *IEEE International Conference on Robotics and Automation*, pages 1887–1893. IEEE, 2018. 1, 2, 3

[55] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *International Conference on Robotics and Automation*, pages 4376–4382. IEEE, 2019. 2, 6

[56] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *European Conference on Computer Vision*, pages 1–19. Springer, 2020. 2, 6

[57] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhang Cao, Chuangrun Liang, and Mingkui Tan. Generative low-bitwidth data free quantization. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020. 3

[58] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 1, 2

[59] Feihu Zhang, Jin Fang, Benjamin Wah, and Philip Torr. Deep fusionnet for point cloud semantic segmentation. In *European Conference on Computer Vision*, volume 2, page 6, 2020. 1

[60] Wenwei Zhang, Zhe Wang, and Chen Change Loy. Multi-modality cut and paste for 3d object detection. *arXiv preprint arXiv:2012.12741*, 2020. 6

[61] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018. 5

[62] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020. 1, 3, 6

[63] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017. 2

[64] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9939–9948, 2021. 2, 3, 5, 6, 7, 8