MSeg3D: Multi-modal 3D Semantic Segmentation for Autonomous Driving

Jiale Li[1] Hang Dai[2*] Hao Han[3] Yong Ding[3*]
[1]College of Information Science and Electronic Engineering, Zhejiang University
[2]School of Computing Science, University of Glasgow
[3]School of Micro-Nano Electronics, Zhejiang University
[*]Corresponding authors{Hang.Dai@glasgow.ac.uk, dingyong09@zju.edu.cn}.

Abstract

LiDAR and camera are two modalities available for 3D semantic segmentation in autonomous driving. The popular LiDAR-only methods severely suffer from inferior segmentation on small and distant objects due to insufficient laser points, while the robust multi-modal solution is under-explored, where we investigate three crucial inherent difficulties: modality heterogeneity, limited sensor field of view intersection, and multi-modal data augmentation. We propose a multi-modal 3D semantic segmentation model (MSeg3D) with joint intra-modal feature extraction and inter-modal feature fusion to mitigate the modality heterogeneity. The multi-modal fusion in MSeg3D consists of geometry-based feature fusion GF-Phase, cross-modal feature completion, and semantic-based feature fusion SF-Phase on all visible points. The multi-modal data augmentation is reinvigorated by applying asymmetric transformations on LiDAR point cloud and multi-camera images individually, which benefits the model training with diversified augmentation transformations. MSeg3D achieves state-of-the-art results on nuScenes, Waymo, and SemanticKITTI datasets. Under the malfunctioning multi-camera input and the multi-frame point clouds input, MSeg3D still shows robustness and improves the LiDAR-only baseline. Our code is publicly available at https://github.com/jialeli1/lidarseg3d.

1. Introduction

Scene understanding for safe autonomous driving can be achieved through semantic segmentation using camera 2D images and LiDAR 3D point clouds, which densely classifies each smallest sensing unit of the modality. The image-based 2D semantic segmentation has been developed with massive solid studies [12, 34, 61, 63]. The camera image has rich appearance information about the object but severely suffers from illumination, varying object scales, and indirect applications in the 3D world. Another modality, LiDAR point cloud, drives 3D semantic segmentation with laser points [1, 3, 11, 37]. Unfortunately, irregular laser points are too sparse to capture the details of objects. The inaccurate segmentation appears especially on small and distant objects. The other under-explored direction is using multi-modal data to increase both the robustness and accuracy in 3D semantic segmentation [67].

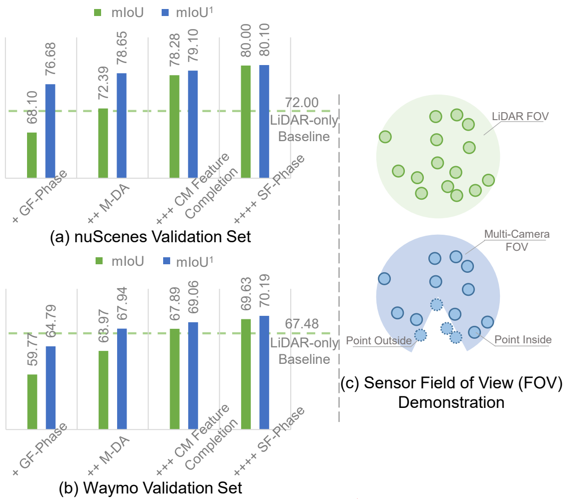
Despite the conceptual superiority, the development of multi-modal segmentation model is still nontrivial, lagging behind the single-modal methods [27]. We rationally attribute the difficulties to the three following aspects. **i**) Heterogeneity between modalities. Due to sparse points and dense pixels, point cloud feature extractors [14, 31] and image feature extractors [15, 44, 47] are developed distinctly. Separate intra-modal feature extractors are used to address the heterogeneity [13, 20, 25, 42, 51], but the lack of joint optimization leads to suboptimal features from irrelevant network parameters. **ii**) Limited intersection on the field of view (FOV) between sensors. Only the points falling into the intersected FOV are geometrically associated with multi-modal data, while simply considering the intersected multi-modal data is not sufficient to be practically applicable. Performing fusion solely in the limited FOV intersection like [20,67] results in unsatisfactory overall segmentation performance as shown in Fig. 1. **iii**) Multi-modal data augmentation. For example, PMF [67] uses only several 2D augmentations for spatially aligned point cloud projection image and camera RGB image. Under the constraint of modal alignment or 2D representation of point cloud, the multi-modal segmentation works [20,25,67] discard many useful and critical point cloud augmentation transformations with sacrificed perception performance [36,48].

Accordingly, we propose a top-performing multi-modal 3D semantic segmentation method termed MSeg3D, inherently motivated by addressing the aforementioned three technical difficulties. **i**) Unlike separately extracting modal features in existing methods [13, 25, 42, 51], we jointly optimize intra-modal feature extraction and inter-modal feature fusion to drive maximum correlation and complementarity between heterogeneous modalities. **ii**) To overcome the disregarded multi-modal fusion outside FOV intersection [25, 67], we propose a cross-modal feature completion and a semantic-based feature fusion phase SF-Phase to collaborate with the geometry-based feature fusion phase GF-Phase. For points outside the FOV intersection, the former completes the missing camera features using predicted pseudo-camera features, under the explicit guidance of cross-modal supervision. For all the points outside and inside the FOV intersection, the later SF-Phase leverages the multi-head attention [41] to model the semantic relation between point and interested categories so that we can attentively fuse the semantic embeddings aggregated from all the visible fields to each point. **iii**) The challenging multi-modal data augmentation is reinvigorated by being decomposed as the asymmetric transformations in the LiDAR, camera worlds, and local cameras, which enables flexible permutation to enrich training samples.

As the proposed components accumulated in Fig. 1, mIoU and mIoU[1] are significantly increased while the gaps between mIoU and mIoU[1] are gradually decreased. Our contributions are four-fold: **i**) We propose a multi-modal segmentation model MSeg3D with joint intra-modal feature extraction and inter-modal feature fusion, achieving state-of-the-art 3D segmentation performance on the competitive nuScenes [3], Waymo [37], and SemanticKITTI [1] datasets for autonomous driving. The proposed framework won 2nd place in the Waymo 3D semantic segmentation challenge at CVPR 2022. **ii**) We propose a cross-modal feature completion and a semantic-based feature fusion phase. To our best knowledge, it is the first time to address the overlooked and inapplicable multi-modal fusion outside the sensor FOV intersection. **iii**) By applying augmentation transformations asymmetrically on point cloud and images, the proposed asymmetrical multi-modal data augmentation significantly increases the diversity of multi-modal samples for training model with robust improvements. **iv**) Extensive experimental analyses on the improvement and robustness of our method clearly investigate our designs.

2. Related Work

LiDAR-only 3D Semantic Segmentation is promoted by SemanticKITTI [1], nuScenes [3], and Waymo [37] datasets. The methods follow the U-Net [33] architecture, but progress from three point cloud representations. **i**) **Point**. The point methods derived from PointNet++ [31] cost heavy computation on sampling and gathering disordered neighbors, especially on a large-scale LiDAR point cloud. The major point methods [17, 32, 39, 53] perform well on small synthetic point cloud [56] rather than sparse LiDAR point cloud. **ii**) **2D images**. PolarNet-series [62, 65] and others [26, 40, 46, 49] project 3D point cloud as 2D images in bird's-eye-view and range-view, achieving efficient LiDAR segmentation with 2D CNNs. But the 3D-to-2D projection damages 3D information and performance. **iii**) **3D Voxel**. The state-of-the-art SPVNAS [38], Cylinder3D [66] and SDSeg3D [21] explicitly explore 3D structure information in different 3D coordinate systems, which efficiently perform 3D sparse convolutions on non-empty voxels [8, 54]. The point and image representations also potentially facilitate voxel features in RPVNet [50] and AF2-S3Net [7]. We also perform effective and efficient 3D voxel feature learning.



(a) nuScenes Validation Set

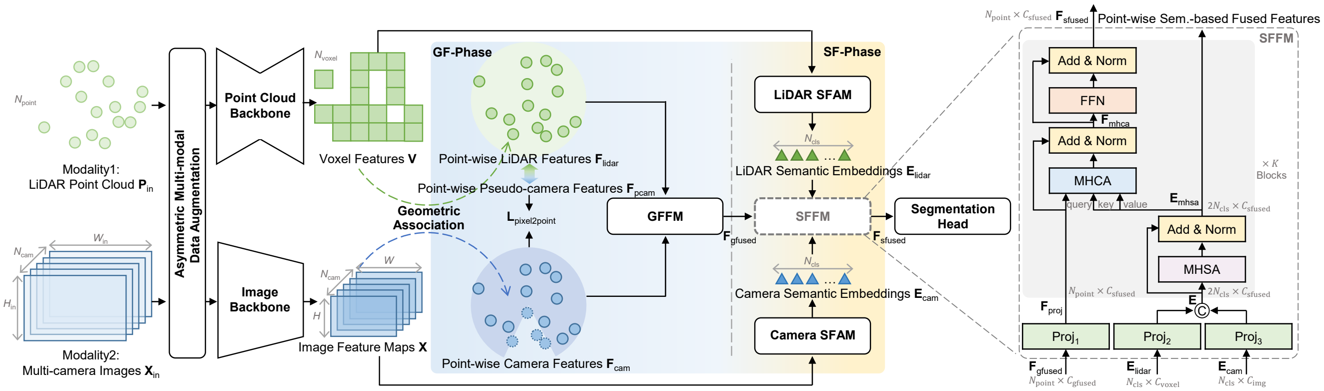
(b) Waymo Validation Set

(c) Sensor Field of View (FOV) Demonstration

Figure 1. Performance (mIoU and mIoU[1]) evaluation on nuScenes [3] (a) and Waymo [37] (b) validation sets. All points are distinguished as points inside the sensor FOV intersection (points inside) and points outside the sensor FOV intersection (points outside). We cumulatively add ("+") the proposed components.

Multi-modal 3D Semantic Segmentation is investigated in [13, 20, 25, 67] using LiDAR and camera with unsatisfactory performance. The suboptimal feature sources of RGB value or separately learned CNN feature are warped into point cloud range image as additional input features [20, 25]. Recently, PMF [67] projects point cloud onto the image plane for fusion with image data, which is obviously limited by camera FOV. They discard the fusion and segmentation over points outside. Despite the improved segmentation performance within the FOV, another LiDAR model is required to take responsibility for the remaining area [43, 67]. Such inequity and inconvenience have prevented researchers from focusing on multi-modal 3D semantic segmentation. The image format of point cloud in the aforementioned works limits them to do only image augmentation such as 2D flipping rather than 3D point cloud augmentation. Instead, we point out the necessity of point cloud augmentation for our 3D segmentation network. Unlike another multi-modal 3D object detector PointAugmenting [43] that preserves the point cloud augmentation while keeping the image unchanged, our method starts from both modalities with

[1] In the following text, mIoU[1] denotes the segmentation performance evaluated on only the points inside FOV intersection by excluding the points outside like PMF [67] and other methods [20, 25].



N_{point}
Modality1:
LiDAR Point Cloud $\mathbf{P}_{\text{in}}$
N_{cam}
W_{in}
H_{in}
Modality2:
Multi-camera Images $\mathbf{X}_{\text{in}}$
Asymmetric Multi-modal Data Augmentation
Point Cloud Backbone
Image Backbone
N_{voxel}
Voxel Features **V**
N_{cam}
W
H
Image Feature Maps **X**
GF-Phase
Point-wise LiDAR Features $\mathbf{F}_{\text{lidar}}$
Point-wise Pseudo-camera Features $\mathbf{F}_{\text{pcam}}$
Geometric Association
$\mathbf{L}_{\text{pixel2point}}$
Point-wise Camera Features $\mathbf{F}_{\text{cam}}$
GFFM
$\mathbf{F}_{\text{gfused}}$
SF-Phase
LiDAR SFAM
N_{cls}
LiDAR Semantic Embeddings $\mathbf{E}_{\text{lidar}}$
SFFM
$\mathbf{F}_{\text{sfused}}$
Segmentation Head
N_{cls}
Camera Semantic Embeddings $\mathbf{E}_{\text{cam}}$
Camera SFAM
$N_{\text{point}} \times C_{\text{sfused}}$ $\mathbf{F}_{\text{sfused}}$ Point-wise Sem.-based Fused Features
SFFM
Add & Norm
FFN
$\mathbf{F}_{\text{mhca}}$
Add & Norm
MHCA
query key value
$\mathbf{E}_{\text{mhsa}}$
$2N_{\text{cls}} \times C_{\text{sfused}}$
× K Blocks
Add & Norm
MHSA
$N_{\text{point}} \times C_{\text{sfused}}$
$2N_{\text{cls}} \times C_{\text{sfused}}$
$\mathbf{E}$
©
$\mathbf{F}_{\text{proj}}$
Proj$_1$
Proj$_2$
Proj$_3$
$\mathbf{F}_{\text{gfused}}$
$N_{\text{point}} \times C_{\text{gfused}}$
$\mathbf{E}_{\text{lidar}}$
$N_{\text{cls}} \times C_{\text{voxel}}$
$\mathbf{E}_{\text{cam}}$
$N_{\text{cls}} \times C_{\text{img}}$

Figure 2. Overview of our multi-modal 3D semantic segmentation model (MSeg3D). For multi-modal feature fusion, GF-Phase mainly includes the Geometry-based Feature Fusion Module (GFFM), while SF-Phase consists of LiDAR Semantic Feature Aggregation Module (SFAM), camera SFAM, and Semantic-based Feature Fusion Module (SFFM).

asymmetric transformations. Besides, 2D3DNet [13] aims to train a 3D model using reduced 3D annotations but more unlabeled data and additional manual 2D annotations to train a 2D model for obtaining 2D segmentation in advance. The 2D segmentation is also painted onto the points as the additional input of 3D model [42]. Such a phased training approach lacks joint 2D-3D optimization. Our motivation, technique, and results are distinguished from them.

3. Method

3.1. Preliminary

Problem Setup. Let $\{\mathbf{P}_{\text{in}}, \mathbf{X}_{\text{in}}\}$ be a multi-modal sample, where $\mathbf{P}_{\text{in}} \in \mathbb{R}^{N_{\text{point}} \times C_{\text{in}}}$ denotes LiDAR point cloud of N_{point} points with C_{in}-dimensional input features (e.g., 3D coordinates and reflectance) and $\mathbf{X}_{\text{in}} \in \mathbb{R}^{N_{\text{cam}} \times 3 \times H_{\text{in}} \times W_{\text{in}}}$ denotes multi-camera RGB images of N_{cam} cameras. Based on sensor calibration, the point with 3D coordinates (x, y, z) can be projected onto the c-th local camera image with the pixel coordinates (c, u, v). N_{cls} semantic categories for each point in the 3D point cloud.

LiDAR Feature Extraction. Given the input $\{\mathbf{P}_{\text{in}}, \mathbf{X}_{\text{in}}\}$ in Fig. 2, we first conduct intra-modal feature extraction with parallel backbones for addressing the heterogeneity between point cloud and images. Due to the superiority of 3D voxel, we perform voxel-based LiDAR feature extraction by a general sparse 3D U-Net [9, 35] in Supplementary Material (SM). The input point cloud is first divided by a quantization step d, then grouped into non-empty voxels by averaging the point-wise initial features of the local points inside a voxel. These non-empty voxels $\mathbf{V}_{\text{in}} \in \mathbb{R}^{N_{\text{voxel}} \times C_{\text{in}}}$ constitute the sparse tensor input to point cloud backbone for learning the expressive voxel features $\mathbf{V} \in \mathbb{R}^{N_{\text{voxel}} \times C_{\text{voxel}}}$.

Camera Feature Extraction. To enhance the LiDAR features with jointly-optimized camera features, we use a trainable image backbone (HRNet-w48 [44] by default) to non-linearly project $\mathbf{X}_{\text{in}}$ as $\mathbf{X} \in \mathbb{R}^{N_{\text{cam}} \times C_{\text{img}} \times H \times W}$, with down-sampled shape $H \times W$ and increased channels C_{img}.

The point cloud and image backbones can be flexibly selected from various mature networks. Moreover, we jointly optimize both backbones in the overall 3D segmentation model to learn relevant semantic feature representations from different modalities. The expressive voxel features $\mathbf{V}$ and image features $\mathbf{X}$ guarantee the inter-modal feature fusion, which consists of GF-Phase, cross-modal feature completion, and SF-Phase in the right half of Fig. 2.

3.2. GF-Phase: Geometry-based Feature Fusion

The point-centric segmentation motivates us to warp voxel features $\mathbf{V}$ and image feature maps $\mathbf{X}$ to points in $\mathbf{P}_{\text{in}}$ as point-wise LiDAR features $\mathbf{F}_{\text{lidar}} \in \mathbb{R}^{N_{\text{point}} \times C_{\text{voxel}}}$ and point-wise camera features $\mathbf{F}_{\text{cam}} \in \mathbb{R}^{N_{\text{point}} \times C_{\text{img}}}$ by using the geometric association of LiDAR and multi-camera.

Learning Point Feature From Voxels and Pixels. We devoxelize voxel features $\mathbf{V}$ as point-wise LiDAR features $\mathbf{F}_{\text{lidar}} = [f_{\text{lidar},i}]_{i=1}^{i=N_{\text{point}}} \in \mathbb{R}^{N_{\text{point}} \times C_{\text{voxel}}}$ for augmenting the point-wise fusion and segmentation. Given a point, we interpolate its point feature from the three nearest neighboring voxels [22]. For the i-th point (x_i, y_i, z_i) with LiDAR feature $f_{\text{lidar},i}$ in $\mathbf{F}_{\text{lidar}}$, we use the known point-wise image coordinates (c_i, u_i, v_i) to perform bilinear interpolation $\mathcal{I}$ on the image feature maps $\mathbf{X}[c_i] \in \mathbb{R}^{C_{\text{img}} \times H \times W}$ within the c_i-th local camera as Eq. 1. Zeros are temporarily padded to those points outside the camera FOV, which allows each point to be decorated with the C_{img}-dimensional feature $f_{\text{cam},i}$. To distinguish points outside, we set a binary mask $\mathbf{B}$ of N_{point} elements with 0 for them and 1 otherwise.

$$f_{\text{cam},i} = \mathcal{I}(\mathbf{X}[c_i], \frac{H}{H_{\text{in}}} u_i, \frac{W}{W_{\text{in}}} v_i), \tag{1}$$

$$\mathbf{F}_{\text{cam}} = [f_{\text{cam},i}]_{i=1}^{i=N_{\text{point}}} \in \mathbb{R}^{N_{\text{point}} \times C_{\text{img}}}. \tag{2}$$

Geometry-based Feature Fusion Module (GFFM) first projects $f_{\text{lidar},i}$ and $f_{\text{cam},i}$ as C_{int}-dimensional by using fully connected layers $\mathcal{F}_{\text{lidar}}$ and $\mathcal{F}_{\text{cam}}$, and concatenates ("©") them together for subsequent learnable fusion using another MLP with C_{gfused} output channels as $f_{\text{gfused},i} = MLP(\mathcal{F}_{\text{lidar}}(f_{\text{lidar},i}) \copyright \mathcal{F}_{\text{cam}}(f_{\text{cam},i}))$ and $\mathbf{F}_{\text{gfused}} = [f_{\text{gfused},i}]_{i=1}^{i=N_{\text{point}}} \in \mathbb{R}^{N_{\text{point}} \times C_{\text{gfused}}}$. The GF-Phase combines the multi-modal features as $\mathbf{F}_{\text{gfused}}$ mainly with geometry clues, so-called the geometry-based feature fusion phase.

3.3. SF-Phase: Semantic-based Feature Fusion

Though the simple yet effective GF-Phase yields appreciable performance gains inside FOV intersection through unbiased considerations of both modalities, three drawbacks remain to be addressed. **i**) The points outside still cannot retrieve their realistic camera features. **ii**) The relative importance between modalities varies in different areas [5, 58], which is overlooked in GF-Phase. **iii**) Explicit relational modeling for category-wise semantic representations [18, 19] is not available. As shown in Fig. 2, we further propose the SF-Phase after GF-Phase. Beyond the geometric association in euclidean space, we aggregate LiDAR features and camera features into category-wise semantic embeddings as $\mathbf{E}_{\text{lidar}}$ and $\mathbf{E}_{\text{cam}}$ for performing multi-modal fusion in a hidden semantic space.

LiDAR Semantic Feature Aggregation Module (LiDAR SFAM) starts from voxel features $\mathbf{V}$ and ends with aggregated N_{cls} LiDAR semantic embeddings in $\mathbf{E}_{\text{lidar}}$. For N_{cls} categories, we assume $\mathbf{E}_{\text{lidar}} \in \mathbb{R}^{N_{\text{cls}} \times C_{\text{voxel}}}$ and a distribution matrix $\mathbf{D}_{\text{lidar}} \in (0,1)^{N_{\text{cls}} \times N_{\text{voxel}}}$. The $\mathbf{D}[j,i]$ closer to 1 indicates that the i-th voxel is more likely to belong to the j-th category than other voxels, so that it contributes more to describing this category in $\mathbf{E}_{\text{lidar}}[j]$. Otherwise, the i-th voxel should contributes less to $\mathbf{E}_{\text{lidar}}[j]$. Now we can compute $\mathbf{E}_{\text{lidar}}$ by matrix product as $\mathbf{E}_{\text{lidar}} = \mathbf{D}_{\text{lidar}}\mathbf{V}$. Two more steps (Eqs. 3∼4) are required to compute $\mathbf{D}_{\text{lidar}}$. We first predict an intermediate segmentation $\mathbf{D}'_{\text{lidar}} \in \mathbb{R}^{N_{\text{voxel}} \times N_{\text{cls}}}$ on the voxel features $\mathbf{V}$ using an MLP-based auxiliary voxel segmentation head $\mathcal{H}_{\text{voxel}}$. We then perform a spatial softmax among voxels to normalize the values into $(0,1)$ for each category.

$$\mathbf{D}'_{\text{lidar}} = \mathcal{H}_{\text{voxel}}(\mathbf{V}), \tag{3}$$

$$\mathbf{D}^{T}_{\text{lidar}} = [Softmax(\mathbf{D}'_{\text{lidar}}[:,j])]_{j=1}^{j=N_{\text{cls}}}. \tag{4}$$

For more accurate $\mathbf{D}_{\text{lidar}}$, we explicitly guide $\mathbf{D}'_{\text{lidar}}$ by using the voxel segmentation supervision in Eq. 15.

Camera Semantic Feature Aggregation Module (Camera SFAM) similarly aggregates $\mathbf{E}_{\text{cam}} \in \mathbb{R}^{N_{\text{cls}} \times C_{\text{img}}}$ from image feature maps $\mathbf{X} \in \mathbb{R}^{N_{\text{cam}} \times C_{\text{img}} \times H \times W}$ using another distribution matrix $\mathbf{D}_{\text{cam}} \in (0,1)^{N_{\text{cls}} \times N_{\text{pixel}}}$ and another auxiliary image segmentation head $\mathcal{H}_{\text{img}}$ implemented with the simple FCN head [34]. The difference lies in that the camera SFAM is designed with all the N_{pixel} pixels of image feature maps across N_{cam} local cameras, where N_{pixel} is $N_{\text{cam}} \times H \times W$.

The $\mathcal{H}_{\text{img}}$ first predicts the intermediate segmentation $\mathbf{D}'_{\text{img}} \in \mathbb{R}^{N_{\text{cam}} \times N_{\text{cls}} \times H \times W}$ by $\mathcal{H}_{\text{img}}(\mathbf{X})$. Note that $\mathbf{D}'_{\text{img}}$ and $\mathbf{X}$ are then rearranged into $\mathbf{D}'_{\text{cam}} \in \mathbb{R}^{N_{\text{cls}} \times N_{\text{pixel}}}$ and $\mathbf{X}' \in \mathbb{R}^{N_{\text{pixel}} \times C_{\text{img}}}$ by transposing and reshaping operations. The final camera semantic embddings $\mathbf{E}_{\text{cam}}$ in Eq. 7 is the matrix product of $\mathbf{D}_{\text{cam}}$ and $\mathbf{X}'$, where the normalized $\mathbf{D}_{\text{cam}}$ is the result of applying spatial softmax on $\mathbf{D}'_{\text{cam}}$ as Eq. 6.

$$\mathbf{D}'_{\text{img}} = \mathcal{H}_{\text{img}}(\mathbf{X}), \tag{5}$$

$$\mathbf{D}_{\text{cam}} = [Softmax(\mathbf{D}'_{\text{cam}}[j,:])]_{j=1}^{j=N_{\text{cls}}}, \tag{6}$$

$$\mathbf{E}_{\text{cam}} = \mathbf{D}_{\text{cam}}\mathbf{X}'. \tag{7}$$

Note that the image segmentation annotations are unavailable for guiding the $\mathbf{D}'_{\text{img}}$ in such 3D semantic segmentation datasets [3, 37]. Nevertheless, we also address this by the cross-modal semantic supervision scheme, which is detailed in Eq. 16 and Eq. 17.

Semantic-based Feature Fusion Module (SFFM). As shown in the right dashed box in Fig. 2, given the input of point-wise geometry-based fused features $\mathbf{F}_{\text{gfused}}$, category-wise semantic embeddings $\mathbf{E}_{\text{lidar}}$ and $\mathbf{E}_{\text{cam}}$, the SFFM first projects them into a C_{sfused}-dimensional space as $\mathbf{F}_{\text{proj}} = Proj_1(\mathbf{F}_{\text{gfused}})$, and $\mathbf{E} = Proj_2(\mathbf{E}_{\text{lidar}}) \copyright Proj_3(\mathbf{E}_{\text{cam}})$, respectively. Before stacking K gray blocks in SFFM, we concatenate the projected multi-modal semantic embeddings together as $\mathbf{E} \in \mathbb{R}^{2N_{\text{cls}} \times C_{\text{sfused}}}$, which can be regarded as a dictionary that describes the typical characteristics for each category from the perspectives of LiDAR and cameras.

In each gray block, we model the semantic relations among all the $2N_{\text{cls}}$ category-wise semantic embeddings using the Multi-Head Self-Attention ($MHSA$) [41] as Eq. 8, where $Norm$ indicates the LayerNorm operation. Let C_{shsa} be C_{sfused}/N_H, the $MHSA$ can be decomposed into N_{H} Single-Head Self-Attention ($SHSA$) operations with $\mathbf{E}_h \in \mathbb{R}^{2N_{\text{cls}} \times C_{\text{shsa}}}$ as

$$\mathbf{E}_{\text{mhsa}} = Norm(\mathbf{E} + MHSA(\mathbf{E})), \tag{8}$$

$$MHSA(\mathbf{E}) = [SHSA(\mathbf{E}_h)]_{h=1}^{h=N_{\text{H}}}, \tag{9}$$

$$SHSA(\mathbf{E}_h) = Softmax(\frac{\mathbf{Q}_h\mathbf{K}_h^T}{\sqrt{C_{\text{shsa}}}})\mathbf{V}_h. \tag{10}$$

Since $\mathbf{E}$ includes $\mathbf{E}_{\text{lidar}}$ and $\mathbf{E}_{\text{cam}}$, the attention matrix $Softmax(\frac{\mathbf{Q}_h\mathbf{K}_h^T}{\sqrt{C_{\text{shsa}}}}) \in \mathbb{R}^{2N_{\text{cls}} \times 2N_{\text{cls}}}$ intuitively includes four parts of relation modeling on $\mathbf{E}_{\text{lidar}} \rightarrow \mathbf{E}_{\text{lidar}}$, $\mathbf{E}_{\text{lidar}} \rightarrow \mathbf{E}_{\text{cam}}$, $\mathbf{E}_{\text{cam}} \rightarrow \mathbf{E}_{\text{lidar}}$, and $\mathbf{E}_{\text{cam}} \rightarrow \mathbf{E}_{\text{cam}}$.

After updating $\mathbf{E}$ as $\mathbf{E}_{\text{mhsa}}$, we perform the further fusion between the point-wise projected features $\mathbf{F}_{\text{proj}}$ and the semantic embeddings $\mathbf{E}_{\text{mhsa}}$, using the Multi-Head Cross-Attention ($MHCA$) for semantic relation modeling in Eq 12 and the Feed-Forward Network (FFN) for feature embedding in Eq. 11, respectively.

$$\mathbf{F}_{\text{sfused}} = Norm(\mathbf{F}_{\text{mhca}} + FFN(\mathbf{F}_{\text{mhca}})), \tag{11}$$

$$\mathbf{F}_{\text{mhca}} = Norm(\mathbf{F}_{\text{proj}} + MHCA(\mathbf{F}_{\text{proj}}, \mathbf{E}_{\text{mhsa}}, \mathbf{E}_{\text{mhsa}})). \tag{12}$$

Unlike Eq. 8, the $MHCA$ in Eq. 12 sets the query $\mathbf{Q}$ from point-wise feature $\mathbf{F}_{\text{proj}}$, and sets the key $\mathbf{K}$ and value $\mathbf{V}$ from semantic embeddings $\mathbf{E}_{\text{mhsa}}$. Thus, an attention matrix with shape $N_{\text{point}} \times 2N_{\text{cls}}$ is computed to attentively aggregate the more important semantic embeddings respective to individual points, beyond the paired geometric association in GF-Phase. Following the common usages of multi-head attention [4], K (i.e., 6) gray blocks are stacked.

Discussion on SFFM. As a useful and general technique [4,28,47], multi-head attention [41], although not proposed in this paper, is effectively tailored to multi-modal feature fusion with different motivations and designs by us. **i**) In Eq. 12, the $MHCA$ enables the point-wise feature to attend to multi-modal semantic embeddings, so that both points inside and outside can be consistently supported by expressive multi-modal semantic embeddings. **ii**) The $MHCA$ attention matrix computes the relative importance of two modalities to each point, improving the unbiased considerations of modalities in GF-Phase. **iii**) The $MHSA$ in Eq. 8 explicitly models the category-wise intra-modal and inter-modal semantic relations, deriving the commonality learning. **iv**) The LiDAR SFAM and Camera SFAM aggregate the long sequences of voxel features $\mathbf{V}$ and image features $\mathbf{X}'$ into the short sequences of $\mathbf{E}_{\text{lidar}}$ and $\mathbf{E}_{\text{cam}}$ with $N_{\text{voxel}}/N_{\text{cls}}$ and $N_{\text{pixel}}/N_{\text{cls}}$ times, which enables efficient computation of the multi-head attention.

3.4. Cross-modal Feature Completion

As shown in Fig. 2, a cross-modal feature completion module with pixel-to-point loss $\mathbf{L}_{\text{pixel2point}}$ (Eq. 13) is set, where the point-wise pseudo-camera feature $\mathbf{F}_{\text{pcam}}$ is mapped from the point-wise LiDAR features $\mathbf{F}_{\text{lidar}}$ by another MLP-based $\mathcal{H}_{\text{pcam}}$ as $\mathcal{H}_{\text{pcam}}(\mathbf{F}_{\text{lidar}})$. In practice, we compute the mean square error loss $\mathcal{L}_{\text{mse}}$ between the $\mathbf{F}_{\text{pcam}}$ and $\mathbf{F}_{\text{cam}}$ of the points inside for learning from the correctly paired cross-modal features relationship.

$$\mathbf{L}_{\text{pixel2point}} = \mathcal{L}_{\text{mse}}(\mathbf{B}\mathbf{F}_{\text{pcam}}, \mathbf{B}Detach(\mathbf{F}_{\text{cam}})). \tag{13}$$

$$\mathbf{F}_{\text{cam}}[i,:] = \mathbf{F}_{\text{pcam}}[i,:] \ |_{i \in \{j | \mathbf{B}[j]=0\}} \ . \tag{14}$$

Note that the points outside are ignored by the binary mask $\mathbf{B}$, which is mentioned in GF-Phase. The gradients of $\mathbf{F}_{\text{cam}}$ are also detached for optimizing the learning of $\mathbf{F}_{\text{pcam}}$.

We employ such a cross-modal feature completion module due to two motivations: **i**) Optimizing $\mathbf{L}_{\text{pixel2point}}$ forces LiDAR features $\mathbf{F}_{\text{lidar}}$ to imitate camera features $\mathbf{F}_{\text{cam}}$ with $\mathcal{H}_{\text{pcam}}$, then the learned pseudo-image features $\mathbf{F}_{\text{pcam}}$ can be switched to replace the padded zeros in $\mathbf{F}_{\text{cam}}$ as Eq. 14 in the inference stage, which serves as the cross-modal feature completion to enhance the feature learning. Thus, we further reduce feature gaps between points outside and inside. **ii**) $\mathcal{H}_{\text{pcam}}$ transfers rich appearance priors from the camera branch to the LiDAR branch with an effective consistency constraint for enhancing the intra-modal feature learning in the training stage.

3.5. Cross-modal Semantic Supervision

Point Supervision. Let $\mathbf{Y}$ of N_{point} elements be the point-wise 3D semantic segmentation labels. The value of $\mathbf{Y}[i]$ is in $[0, N_{\text{cls}} - 1]$, where 0 denotes the ignored category. An MLP based point segmentation head $\mathcal{H}_{\text{point}}$ is built on $\mathbf{F}_{\text{sfused}}$ for the 3D segmentation prediction $\hat{\mathbf{Y}} = \mathcal{H}_{\text{point}}(\mathbf{F}_{\text{sfused}})$. Following [6, 21, 66], we adopt point loss $\mathbf{L}_{\text{point}}$ as a combination of cross-entropy loss $\mathcal{L}_{\text{ce}}$ and lovasz-softmax loss $\mathcal{L}_{\text{lovasz}}$ [2] as $\mathcal{L}_{\text{ce}}(\hat{\mathbf{Y}}, \mathbf{Y}) + \mathcal{L}_{\text{lovasz}}(\hat{\mathbf{Y}}, \mathbf{Y})$.

Point-to-voxel Supervision. For guiding the $\mathbf{D}'_{\text{lidar}}$ in Eq. 3, the voxel loss $\mathbf{L}_{\text{p2v}}$ is defined as:

$$\mathbf{L}_{\text{p2v}} = \mathcal{L}_{\text{ce}}(\mathbf{D}'_{\text{lidar}}, \mathbf{Y}_{\text{p2v}}) + \mathcal{L}_{\text{lovasz}}(\mathbf{D}'_{\text{lidar}}, \mathbf{Y}_{\text{p2v}}), \tag{15}$$

where the voxel labels $\mathbf{Y}_{\text{p2v}}$ of N_{voxel} elements can be determined from the labels of points in the voxel. To avoid ambiguity in $\mathbf{L}_{\text{p2v}}$, the label of the voxel containing points of multiple categories is set to zero as the ignored category.

Point-to-pixel Supervision. The problem of how to use only point labels to correctly guide $\mathbf{D}'_{\text{img}}$ (Eq. 5) is still unsolved. For $\mathbf{D}'_{\text{img}}$ of each training sample, we initialize a $N_{\text{cam}} \times H \times W$ map $\mathbf{Y}_{\text{point2pixel}}$ of all zeros as the image segmentation label. We then project each point onto the down-sampled image planes and retrieve the point label $\mathbf{Y}[i]$ for the corresponding nearest ($\langle\cdot\rangle$) pixel as Eq. 16. Although the generated image label $\mathbf{Y}_{\text{point2pixel}}$ is sparse, it provides sufficient supervision. In SM, Fig. S2 and Fig. S3 provide visualizations of $\mathbf{D}'_{\text{img}}$ to show that these sparse point-to-pixel supervisions can propagate to other regions with similar appearance, which guides $\mathbf{D}'_{\text{img}}$ correctly. Since $\mathbf{Y}_{\text{point2pixel}}$ is sparse, we only compute the basic cross-entropy loss $\mathcal{L}_{\text{ce}}$ in Eq. 17.

$$\mathbf{Y}_{\text{point2pixel}}[c_i, \langle \frac{H}{H_{\text{in}}} u_i \rangle, \langle \frac{W}{W_{\text{in}}} u_i \rangle] = \mathbf{Y}[i], \tag{16}$$

$$\mathbf{L}_{\text{point2pixel}} = \mathcal{L}_{\text{ce}}(\mathbf{D}'_{\text{img}}, \mathbf{Y}_{\text{point2pixel}}). \tag{17}$$

Total Loss Function. To jointly optimize all the modules in the proposed segmentation model, the total loss combines loss terms across different modalities into $\mathbf{L}$ as $\alpha_1 \mathbf{L}_{\text{point}} + \alpha_2 \mathbf{L}_{\text{p2v}} + \alpha_3 \mathbf{L}_{\text{point2pixel}} + \alpha_4 \mathbf{L}_{\text{pixel2point}}$, where $\alpha_1 \sim \alpha_4$ are set as 1.0, 1.0, 0.5, 1.0 to balance loss terms.

3.6. Asymmetric Multi-modal Data Augmentation

The segmentation output and our multi-modal fusion mechanism are point-centered. In Eq. 1 and Eq. 16, the point and the pixel always can be bridged by the coordinate pair of (x_i, y_i, z_i) and (c_i, u_i, v_i), as long as we pre-compute the coordinate pairs and keep the order of coordinate pairs among all the points in synchronization. With this ordering constraint, we have the flexibility

Table 1. Data augmentation transformations on LiDAR (L) point cloud and camera (C) images. We only treat random flipping as a naive symmetric transformation that can be simply applied to both modalities simultaneously.

L-only	• Global rotation around the Z axis with a random angle in $[-\frac{\pi}{4}, +\frac{\pi}{4}]$. • Global translation with a random vector $(\Delta x, \Delta y, \Delta z)$ sampled from a Gaussian distribution with mean zero and the standard deviation 0.5. • Global scaling with a random scaling factor in $[0.95, 1.05]$. ...
Symmetric	• Random flipping along the X, Y axis with probability 0.5.
C-only	• Scaling with a random ratio in $[1.0, 1.5]$. • Horizontal rotation with a random angle in $[-1^\circ, 1^\circ]$. • Random cropping with a size $(H_{\text{in}}, W_{\text{in}})$. • Color jitter from torchvision [30] with random parameters of 0.3 for brightness, 0.3 for contrast, 0.3 for saturation, 0.1 for hue. • JPEG with a random compression ratio in $[30, 70]$ and probability 0.5 [64]. ...

to decouple multi-modal data augmentation: **i**) We can preserve all the LiDAR transformations in Tab. 1 with the images and the coordinates (c, u, v) unchanged. **ii**) Once coordinates (c, u, v) are synchronously transformed with images, we can further apply the camera-only transformations in Tab. 1 to images, which are asymmetric to the LiDAR transformations. **iii**) We can apply independent transformations between the images of local cameras since the image view is naturally sliced by local cameras.

The asymmetry lies in not only the LiDAR and camera world but also the local cameras. Thus, we significantly diversify the multi-modal data augmentation for 3D semantic segmentation using the introduced camera transformations with the preserved LiDAR transformations. More potential transformations can be exploited beyond Tab. 1.

4. Experiment

4.1. Experimental Setup

nuScenes Dataset splits 28,130 training samples, 6,019 validation samples, and 6,008 testing samples [3, 11]. Each nuScenes sample contains relatively sparser point cloud of 32 beams and RGB images captured by 6 cameras: front, front-left, front-right, back, back-left, and back-right. Points outside are projected below the image bottom due to different vertical FOVs of LiDAR and cameras. Following official protocol, the N_{cls} is 17. Semantic segmentation annotations are only on point clouds and not on images.

Waymo Dataset for 3D semantic segmentation includes 23,691 training samples, 5,976 validation samples, and 2,982 testing samples [37]. Each sample contains the point cloud of 64 beams and the RGB images captured by 5 cameras: front, front-left, front-right, side-left, and side-right. There is no rear camera on Waymo ego-vehicle, hence more points outside unfairly increase the difficulty of multi-modal segmentation. The N_{cls} is 23. Similarly, semantic segmentation annotations are only on point clouds.

SemanticKITTI Dataset is collected by a LiDAR with 64 beams [1]. Following [10, 50, 66, 67], we use sequences 00 to 10 (excluding 08) with 19,130 training samples and sequence 08 with 4,071 validation samples. It provides only the images of the front-view camera. The N_{cls} is 20.

Evaluation Metric is the mean Intersection-over-Union (mIoU) defined as $\frac{1}{C}\sum_{c=1}^{N_{\text{cls}}-1}\frac{TP_c}{TP_c+FP_c+FN_c}$, where TP_c, FP_c, FN_c denote the number of true positive, false positive, and false negative predictions for the c-th category out of $N_{\text{cls}}-1$ valid categories. Besides, nuScenes computes another metric fwIoU by weighting category-wise IoU with the point frequency of its category.

Settings. Although lightweight LiDRA-only baseline models can be trained with larger batchsize and more epochs to obtain relatively better results, we still train all the models under the same schedule: a batch of 32 random samples distributed on 16 Tesla V100 GPUs with 24 epochs. More implementation details on point cloud voxelization, network architecture, model training, and inference are all included in SM.

4.2. Comparison with State-of-the-art Methods

Results on nuScenes. From Tab. 2, multi-modal methods counterintuitively lag far behind LiDAR-only methods. In the top-20 submissions on nuScenes benchmark [27], there are only four multi-modal methods. Details of Cylinder3D++, SPVCNN++, LIFusion, and CPFusion are not available till submission. Our framework achieves the best mIoU and fwIoU with multi-modal fusion mechanism and asymmetric multi-modal data augmentation. Especially, our method achieves superior performance on challenging small objects such as pedestrians and traffic cones, where the laser points are typically insufficient for LiDAR-only methods. Our method also significantly outperforms all public and private multi-modal submissions in Tab. 2.

Results on Waymo. We perform evaluation on Waymo benchmark [45] in Tab. 3, where the LiDAR sensor provides denser laser points and the multi-camera excludes the rearview. Such denser point cloud and incomplete camera FOV unfairly weaken the advantages of multiple modalities. Although, our method still achieves state-of-the-art performance using a general point cloud backbone network [9, 35]. We believe that the performance can be improved by incorporating stronger point cloud backbone networks from other state-of-the-art LiDAR-only methods.

Results on SemanticKITTI. While comparisons on the highly competitive nuScenes and Waymo datasets validate the effectiveness of our method, Tab. 4 shows the experimental comparison on SemanticKITTI [1]. Since SemanticKITTI provides only front-view images, PMF [67] evaluates the multi-modal segmentation performance on points inside the FOV intersection. We follow PMF to report mIoU[1] on points inside in Tab. 4. Our method not only shows improvements to the LiDAR-only methods but also outperforms other multi-modal methods.

Table 2. Performance comparison on nuScenes testing set [27]. The modalities available on nuScenes include LiDAR (L), Camera (C), and Radar (R). Top-1 results are in bold. *Submission entries without a published paper by the CVPR 2023 deadline of Nov 11, 2022.

Method	Modality	**mIoU**	**fwIoU**	barrier	bicycle	bus	car	cons. vehicle	motorcycle	pedestrian	traffic-cone	trailer	truck	driveable	other	sidewalk	terrain	manmade	vegetation
PolarNet [62]	L	69.42	87.38	72.16	16.81	77.01	86.53	51.14	69.65	64.80	54.11	69.70	63.53	96.64	67.14	77.70	72.13	87.13	84.47
JS3C-Net [52]	L	73.60	88.06	80.14	26.15	87.79	84.54	55.17	72.56	71.28	66.26	76.79	71.11	96.80	64.47	76.86	74.09	87.48	86.10
Cylinder3D [66]	L	77.16	89.92	82.76	29.75	84.34	89.41	63.03	79.29	77.21	73.40	84.55	69.17	97.66	70.24	80.29	75.51	90.41	87.55
AMVNet [24]	L	77.27	90.08	80.64	31.96	81.73	88.93	67.07	84.33	76.11	73.48	84.87	67.30	97.37	67.37	79.41	75.45	91.45	88.69
SPVNAS [38]	L	77.35	89.68	80.00	29.98	91.92	90.81	64.68	78.99	75.62	70.94	81.01	74.64	97.44	69.23	79.95	76.10	89.28	87.06
Cylinder3D++ [66]	L	77.86	89.93	82.76	33.89	84.34	89.41	69.63	79.42	77.26	73.40	84.55	69.41	97.66	70.24	80.29	75.51	90.42	87.55
AF2S3Net [7]	L	78.34	88.51	78.87	52.21	89.93	84.17	**77.42**	74,30	77.32	71.95	83.88	73.78	97.13	66.47	77.51	74.01	87.69	86.80
SPVCNN++ [38]	L	81.12	90.97	**86.35**	43.13	91.90	**92.18**	75.90	75.72	83.44	77.31	86.82	**77.36**	97.69	**71.22**	81.08	77.19	91.67	88.98
LIFusion* [27]	LC	75.74	89.32	58.13	36.30	86.67	84.28	59.96	79.69	80.30	77.77	83.23	68.74	97.18	68.19	77.04	74.45	91.03	88.95
PMF [67]	LC	77.03	89.34	82.11	40.33	80.94	86.42	63.72	79.22	79.75	75.86	81.17	67.05	97.28	67.69	78.05	74.48	89.94	88.46
CPFusion* [27]	LCR	77.72	89.19	83.67	37.03	89.02	86.24	70.08	77.47	78.07	74.53	82.78	67.94	96.64	68.24	79.53	74.91	90.47	86.95
2D3DNet [13]	LC	79.96	90.08	83.01	**59.35**	87.99	85.09	63.70	**84.39**	81.95	75.96	84.79	71.93	96.88	67.35	79.81	75.96	92.05	89.18
MSeg3D	LC	**81.14**	**91.35**	83.11	42.46	**94.92**	92.01	67.10	78.58	**85.66**	**80.47**	**87.53**	77.32	**97.74**	69.82	**81.22**	**77.83**	**92.35**	**90.07**

Table 3. Performance comparison on Waymo testing set [45]. *Submission entries without a published paper by the CVPR 2023 deadline of Nov 11, 2022.

Method	**mIoU**
LiDARMultiNet* [55]	71.13
HorizonSegExpert* [45]	69.44
SPVCNN++ [38]	67.70
PolarFuse* [45]	67.28
SalsaNext [40]	55.85
MSeg3D	70.51

Table 4. Performance comparison on SemanticKITTI [1] validation set following PMF [67]. The results of the other methods are from PMF paper.

Method	Modality	**mIoU**[1]
SalsaNext [10]	L	59.4
SPVNAS [38]	L	62.3
Cylinder3D [66]	L	64.9
PointPainting [42]	LC	54.5
RGBAL [25]	LC	56.2
PMF [67]	LC	63.9
L-Baseline	L	64.8
MSeg3D	LC	66.7

Table 5. Analysis on the performance gap on all points and points inside, and the data augmentation (DA). Only GF-Phase is simply used for multi-modal fusion (M-Fusion) here. DA includes LiDAR DA (L-DA) and Multi-modal DA (M-DA).

LiDAR	M-Fusion	DA	nuScenes		Waymo	
			mIoU	mIoU[1]	mIoU	mIoU[1]
√	×	L-DA	72.00	70.76	67.48	67.41
√	GF-Phase	×	68.10	76.68	59.77	64.79
√	GF-Phase	L-DA	71.35	77.48	60.34	65.70
√	GF-Phase	M-DA	72.39	78.65	63.97	67.94

Table 6. Analysis on multi-modal feature fusion of GF-Phase, cross-modal (CM) feature completion, and SF-Phase. The supervised CM feature completion is decomposed as $\mathbf{L}_{\text{pixel2point}}$ (Eq. 13) and "Comp." (Eq. 14). The gap can be formulated as mIoU - mIoU[1].

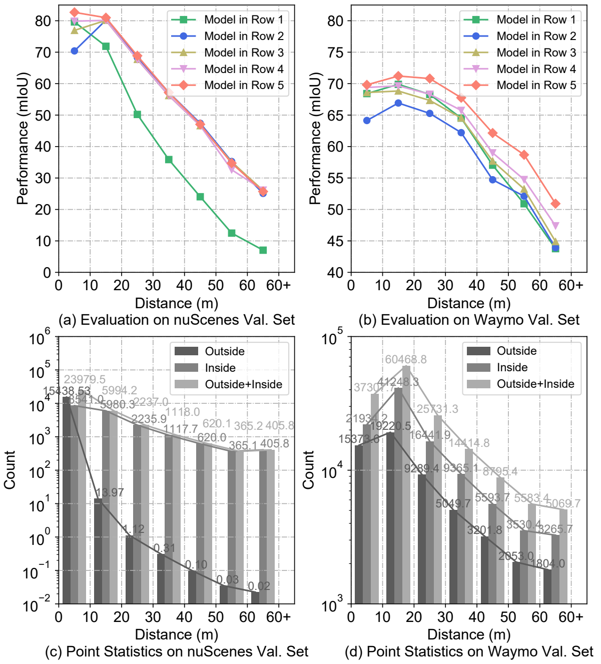
LiDAR	M-Fusion		CM Feature Completion		nuScenes			Waymo		
	GF-Phase	SF-Phase	$\mathbf{L}_{\text{pixel2point}}$	Comp.	mIoU	mIoU[1]	Gap	mIoU	mIoU[1]	Gap
√	×	×	×	×	72.00	70.76	+1.24	67.48	67.41	+0.07
√	√	×	×	×	72.39	78.65	-6.26	63.97	67.94	-3.97
√	√	×	√	×	76.44	79.10	-2.66	67.13	69.06	-1.93
√	√	×	√	√	78.28	79.10	-0.82	67.89	69.06	-1.17
√	√	√	√	√	80.00	80.10	-0.10	69.63	70.19	-0.56

4.3. Ablation Study

Difficulties Arising from Multi-modality are investigated in Tab. 5 from two perspectives of the performance gap between all points and points inside, as well as the multi-modal data augmentation. The experiments in Tab. 5 start with a vanilla variant of the multi-modal model using the basic GF-Phase fusion without the multi-modal data augmentation. The mIoU is relatively worse than the mIoU[1], since the geometry-based feature fusion ideally ignores the inevitable points outside with the missing camera features [20, 25, 67], which also motivates us to close the performance gaps between mIoU and mIoU[1] by using the cross-modal feature completion and semantic-based SF-Phase. Besides, in rows 2 and 1, the multi-modal segmentation performance will be worsened if we deprecated the augmentation transformations that cannot be directly applied to both modalities. However, the proposed asymmetric multi-modal data augmentation allows the accumulation of the transformations on point cloud and image, achieving the best performance on both datasets.

Multi-modal Feature Fusion Modules. Tab. 6 shows that our supervised cross-modal feature completion and SF-Phase benefit both mIoU and mIoU[1] with gradually narrowed gaps. In row 3, $\mathbf{L}_{\text{pixel2point}}$ can also bring improvements, which indicates that transferring appearance information from dense image to sparse point cloud is applicable and beneficial for joint feature learning. In row 4, completing camera features with pseudo-camera features facilitates the feature gap reduction between points inside and outside. Eventually, the SF-Phase further narrows the gaps between mIoU and mIoU[1] to the smallest values of 0.10 and 0.56, addressing the limitation of geometric associations by semantic-based feature fusion. Thus, we set the model in the last row as our final framework.

mIoU Breakdown Over Distance. Fig. 3 presents the distance-based evaluation corresponding to the models in Tab. 6. The LiDAR-only model degrades at long distances due to more sparse points. Instead, the multi-modal model in row 5 effectively alleviates the performance



(a) Evaluation on nuScenes Val. Set

(b) Evaluation on Waymo Val. Set

(c) Point Statistics on nuScenes Val. Set

(d) Point Statistics on Waymo Val. Set

Figure 3. Distance-based mIoU evaluation on nuScenes and Waymo validation (Val.) sets using the models in Tab. 6.

Table 7. Robustness analysis on MSeg3D by removing some cameras as malfunction. "#Camera" denotes the number of available cameras. "×" denotes LiDAR-only baseline.

#Camera	6	5	4	3	2	1	0	×
nuScenes	80.00	78.69	77.62	76.69	76.01	75.44	74.47	72.00
Waymo	-	69.63	69.18	68.86	68.13	68.09	68.04	67.48

Table 8. Robustness analysis on MSeg3D with multi-frame point clouds input. "#L-Frame" is the frame number for LiDAR.

nuScenes	#L-Frame	1	10	20	25	30	40	25
	Camera	×	×	×	×	×	×	√
	mIoU	72.00	74.66	75.37	75.77	75.28	75.15	81.12
Waymo	#L-Frame	1	5	10	15	20	30	10
	Camera	×	×	×	×	×	×	√
	mIoU	67.48	69.18	69.45	68.88	68.78	68.64	70.20

degradation. In Fig. 3 (a) and (b), improvements between the multi-modal model in row 5 and the LiDAR-only model in row 1 are also increased along the distance due to increased sparsity. The reasons why the improvement of the multi-modal model on Waymo is not as similar as that on nuScenes are further analyzed in Fig. 3 (c) and (d). The Waymo point cloud has denser points, which reduces the dependence on the image. More points outside are distributed along different distances due to no rear cameras on Waymo. Although more points outside hinder the applicability of multi-modal fusion, our final multi-modal model still yields a low-performance gap of 0.56 mIoU on Waymo in row 5.

Robustness Against Camera Malfunction. In Tab. 7, our MSeg3D performs properly under the unfavorable condition of camera malfunction, and it can easily deal with the practical situation without switching the model.

Table 9. Scalability analysis by barely varying the backbone settings on nuScenes validation set.

Image Backbone	mIoU	#Params(M)	Latency(s)
×	72.00	21.28	0.083
SegFormer-B0 [47]	78.14	25.33	0.204
SegFormer-B5 [47]	78.89	103.48	0.479
HRNet-w18 [44]	79.21	31.56	0.265
ResNet101 [15]	79.36	64.60	0.567
HRNet-w48 [44]	80.00	87.34	0.445

Note that even with all cameras removed, MSeg3D still outperforms the LiDAR-only baseline, where our cross-modal feature completion supervision provides effective cross-modal information transfer in training.

Robustness Against Multi-frame Point Clouds Input. Since the densified points alleviate the sparsity, collapsing laser points from neighboring frames usually boosts LiDAR-only 3D perception [16, 52]. In Tab. 8, we follow [57] collapsing multiple previous frames to the current frame from the provided ego-vehicle motion information. For LiDAR-only model, the improvements are saturated with 25 and 10 frames on nuScenes and Waymo. Under such conditions, our MSeg3D still achieves further improvements in the last column, which shows that multi-frame point clouds input can be an optional extension.

Complexity Scalability. From Tab. 9, a lightweight image backbone also delivers significant performance gains due to multi-modal input. The best-performing HRNet-w48 is our default image backbone. However, the multi-camera image input makes the efficiency bottleneck lie in the image backbone, which deserves in-depth study for autonomous driving [23]. We believe that real-time image segmentation networks such as BiSeNet [59, 60] and Fast-SCNN [29] can accelerate our approach. Overall, our MSeg3D can adapt to complexity scalability among various backbones, which flexibly trades off performance and efficiency.

5. Conclusion

We propose a novel multi-modal 3D semantic segmentation method termed MSeg3D for autonomous driving, based on LiDAR and multi-camera sensors. Our cross-modal feature completion and semantic-based feature fusion solve the overlooked problem, where multi-modal fusion can occur only in the sensor FOV intersection. The proposed asymmetric multi-modal data augmentation enables the multi-modal segmentation model to be effectively trained with reliable performance. Our method achieves state-of-the-art performance on nuScenes, Waymo, and SemanticKITTI. The comprehensive experiments validate improvements and robustness. We hope that our work can inspire further investigation into multi-modal fusion for 3D semantic segmentation in autonomous driving.

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

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jürgen Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *ICCV*, pages 9296–9306, 2019. 1, 2, 6, 7

[2] 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 *CVPR*, pages 4413–4421, 2018. 5

[3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11618–11628, 2020. 1, 2, 4, 6

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 5

[5] Bowen Cheng, Ross B. Girshick, Piotr Dollár, Alexander C. Berg, and Alexander Kirillov. Boundary IoU: Improving object-centric image segmentation evaluation. In *CVPR*, pages 15334–15342, 2021. 4

[6] Hui-Xian Cheng, Xian-Feng Han, and Guo-Qiang Xiao. CENet: Toward concise and efficient LiDAR semantic segmentation for autonomous driving. In *ICME*, pages 1–6. IEEE, 2022. 5

[7] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. (AF)2-S3Net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *CVPR*, pages 12547–12556, 2021. 2, 7

[8] Christopher B. Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019. 2

[9] OpenPCDet Contributors. OpenPCDet: An open-source toolbox for 3D object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2023. 3, 6

[10] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. SalsaNext: fast, uncertainty-aware semantic segmentation of LiDAR point clouds for autonomous driving. *arXiv preprint arXiv:2003.03653*, 2020. 6, 7

[11] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic NuScenes: A large-scale benchmark for LiDAR panoptic segmentation and tracking. *IEEE Robotics Autom. Lett.*, 7(2):3795–3802, 2022. 1, 6

[12] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019. 1

[13] Kyle Genova, Xiaoqi Yin, Abhijit Kundu, Caroline Pantofaru, Forrester Cole, Avneesh Sud, Brian Brewington, Brian Shucker, and Thomas A. Funkhouser. Learning 3D semantic segmentation with only 2D image supervision. In *3DV*, pages 361–372. IEEE, 2021. 1, 2, 3, 7

[14] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 8

[16] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3D object detection. In *CVPR*, pages 10998–11006, 2020. 8

[17] 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 *CVPR*, pages 11105–11114, 2020. 2

[18] Zhenchao Jin, Tao Gong, Dongdong Yu, Q. Chu, Jian Wang, Changhu Wang, and Jie Shao. Mining contextual information beyond image for semantic segmentation. In *ICCV*, pages 7211–7221, 2021. 4

[19] Zhenchao Jin, Bin Liu, Qi Chu, and Nenghai Yu. ISNet: Integrate image-level and semantic-level context for semantic segmentation. In *ICCV*, pages 7169–7178, 2021. 4

[20] Georg Krispel, Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. FuseSeg: LiDAR point cloud segmentation fusing multi-modal data. In *WACV*, pages 1863–1872, 2020. 1, 2, 7

[21] Jiale Li, Hang Dai, and Yong Ding. Self-distillation for robust LiDAR semantic segmentation in autonomous driving. In *ECCV*, volume 13688, pages 659–676, 2022. 2, 5

[22] Jiale Li, Hang Dai, Ling Shao, and Yong Ding. From voxel to point: IoU-guided 3D object detection for point cloud with voxel-to-point decoder. In *ACM MM*, pages 4622–4631, 2021. 3

[23] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 8

[24] Venice Erin Liong, Thi Ngoc Tho Nguyen, Sergi Widjaja, Dhananjai Sharma, and Zhuang Jie Chong. AMVNet: Assertion-based multi-view fusion network for LiDAR semantic segmentation. *CoRR*, abs/2012.04934, 2020. 7

[25] Khaled El Madawi, Hazem Rashed, Ahmad El Sallab, Omar Nasr, Hanan Kamel, and Senthil Kumar Yogamani. RGB and LiDAR fusion based 3D semantic segmentation for autonomous driving. In *ITSC*, pages 7–12, 2019. 1, 2, 7

[26] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. RangeNet++: Fast and accurate LiDAR semantic segmentation. In *IROS*, pages 4213–4220, 2019. 2

[27] nuScenes. nuScenes leaderboard of lidar segmentation task. https://www.nuscenes.org/lidar-segmentation?externalData=all&mapData=all&modalities=Any, 2023. 1, 6, 7

[28] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3D object detection with pointformer. In *CVPR*, pages 7463–7472, 2021. 5

[29] Rudra P. K. Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-SCNN: Fast semantic segmentation network. In *BMVC*, page 289, 2019. 8

[30] Pytorch. Pytorch document of color jitter. https://pytorch.org/vision/stable/generated/torchvision.transforms.ColorJitter.html?highlight=colorjitter, 2023. 6

[31] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017. 1, 2

[32] Shi Qiu, Saeed Anwar, and Nick Barnes. Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In *CVPR*, pages 1757–1767, 2021. 2

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi, editors, *MICCAI*, volume 9351, pages 234–241, 2015. 2

[34] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE TPAMI*, 39(4):640–651, 2017. 1, 4

[35] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. *IEEE TPAMI*, 43(8):2647–2664, 2021. 3, 6

[36] Vishwanath A. Sindagi, Yin Zhou, and Oncel Tuzel. MVX-Net: Multimodal VoxelNet for 3D object detection. In *ICRA*, pages 7276–7282, 2019. 1

[37] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2443–2451, 2020. 1, 2, 4, 6

[38] 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 *ECCV*, volume 12373, pages 685–702, 2020. 2, 7

[39] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and deformable convolution for point clouds. In *ICCV*, pages 6410–6419, 2019. 2

[40] Cortinhal Tiago, Tzelepis George, and Erdal Aksoy Eren. SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds. In *Advances in Visual Computing*, pages 207–222, 2020. 2, 7

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 2, 4, 5

[42] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. PointPainting: Sequential fusion for 3D object detection. In *CVPR*, pages 4603–4611, 2020. 1, 3, 7

[43] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. PointAugmenting: Cross-modal augmentation for 3D object detection. In *CVPR*, pages 11794–11803, 2021. 2

[44] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE TPAMI*, 43(10):3349–3364, 2021. 1, 3, 8

[45] Waymo. Waymo Open Dataset leaderboard of 3D semantic segmentation challenge. https://waymo.com/open/challenges/2022/3d-semantic-segmentation/, 2023. 6, 7

[46] 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 *ICRA*, pages 4376–4382, 2019. 2

[47] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, pages 12077–12090, 2021. 1, 5, 8

[48] Liang Xie, Chao Xiang, Zhengxu Yu, Guodong Xu, Zheng Yang, Deng Cai, and Xiaofei He. PI-RCNN: an efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module. In *AAAI*, pages 12460–12467, 2020. 1

[49] 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 *ECCV*, pages 1–19, 2020. 2

[50] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. RPVNet: A deep and efficient range-point-voxel fusion network for LiDAR point cloud segmentation. In *ICCV*, pages 16004–16013, 2021. 2, 6

[51] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Bin Zhou, and Liangjun Zhang. FusionPainting: Multimodal fusion with adaptive attention for 3D object detection. In *ITSC*, pages 3047–3054, 2021. 1

[52] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep LiDAR point cloud segmentation via learning contextual shape priors from scene completion. In *AAAI*, pages 3101–3109, 2021. 7, 8

[53] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *CVPR*, pages 5588–5597, 2020. 2

[54] Yan Yan, Yuxing Mao, and Bo Li. SECOND: sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2

[55] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. LidarMultiNet: Towards a unified multi-task network for LiDAR perception. *CoRR*, abs/2209.09385, 2022. 7

[56] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas J. Guibas. A scalable active framework for region annotation in 3D shape collections. *ACM TOG*, 35(6):210:1–210:12, 2016. 2

[57] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D object detection and tracking. In *CVPR*, pages 11784–11793, 2021. 8

[58] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal virtual point 3D detection. In *NeurIPS*, pages 16494–16507, 2021. 4

[59] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. BiSeNet V2: bilateral network with guided aggregation for real-time semantic segmentation. *IJCV*, 129(11):3051–3068, 2021. 8

[60] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, volume 11217, pages 334–349, 2018. 8

[61] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, volume 12351, pages 173–190, 2020. 1

[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 *CVPR*, pages 9598–9607, 2020. 2, 7

[63] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 6230–6239, 2017. 1

[64] Stephan Zheng, Yang Song, Thomas Leung, and Ian J. Goodfellow. Improving the robustness of deep neural networks via stability training. In *CVPR*, pages 4480–4488, 2016. 6

[65] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-PolarNet: Proposal-free LiDAR point cloud panoptic segmentation. In *CVPR*, pages 13194–13203, 2021. 2

[66] 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 *CVPR*, pages 9939–9948, 2021. 2, 5, 6, 7

[67] Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-aware multi-sensor fusion for 3D LiDAR semantic segmentation. In *ICCV*, pages 16260–16270, 2021. 1, 2, 6, 7