Multi-Sem Fusion: Multimodal Semantic Fusion for 3D Object Detection

Shaoqing Xu, Fang Li, Ziying Song, Jin Fang, Sifen Wang, Zhi-Xin Yang, *Member, IEEE*

***Abstract*—LIDAR and camera fusion techniques are promising for achieving 3D object detection in autonomous driving. Most multi-modal 3D object detection frameworks integrate semantic knowledge from 2D images into 3D LiDAR point clouds to enhance detection accuracy. Nevertheless, the restricted resolution of 2D feature maps impedes accurate re-projection and often induces a pronounced boundary-blurring effect, which is primarily attributed to erroneous semantic segmentation. To address these limitations, we present the *Multi-Sem Fusion (MSF)* framework, a versatile multi-modal fusion approach that employs 2D/3D semantic segmentation methods to generate parsing results for both modalities. Subsequently, the 2D semantic information undergoes re-projection into 3D point clouds utilizing calibration parameters. To tackle misalignment challenges between the 2D and 3D parsing results, we introduce an Adaptive Attention-based Fusion (AAF) module to fuse them by learning an adaptive fusion score. Then the point cloud with the fused semantic label is sent to the following 3D object detectors. Furthermore, we propose a Deep Feature Fusion (DFF) module to aggregate deep features at different levels to boost the final detection performance. The effectiveness of the framework has been verified on two public large-scale 3D object detection benchmarks by comparing them with different baselines. And the experimental results show that the proposed fusion strategies can significantly improve the detection performance compared to the methods using only point clouds and the methods using only 2D semantic information. Moreover, our approach seamlessly integrates as a plug-in within any detection framework.**

***Index Terms*—3D Object Detection, Multimodal Fusion, Self-Attention**

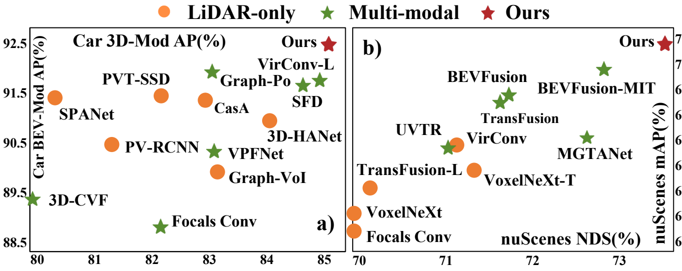

Fig. 1: The proposed *Multi-Sem Fusion (MSF)* is a general multi-modal fusion framework which can be employed for different 3D object detectors. a) is the result of our framework on the KITTI test split on both 3D detection and BEV detection mAP. b) gives the performance of our proposed framework on the public nuScenes benchmark on both NDS and mAP.

I. Introduction

VISION-BASED perception tasks, like **3D Object Detection**, semantic segmentation, and lane detection, have been extensively studied with the development of Autonomous Driving(AD) and intelligent transportation systems. Among them, LiDAR-based 3D Object detection is the mainstream of current research. The depth information can be easily obtained from LiDAR sensors to localize objects. However, the texture and color information has been totally lost due to the sparse scanning. Therefore, False Positive (FP) detection and wrong categories classification often arise for LiDAR-based object detection frameworks.

On the other hand, images can provide details texture and color information while the depth information has been totally lost. The fusion method with various sensor data is an encouraging way for boosting the perception performance of AD. Generally, Wang et al. [1] shows that multi-modal fusion approaches in object detection tasks can be divided into early fusion-based [2], [3], cascade fusion-based [4]–[9] and late fusion-based approaches [10]. Early fusion-based approaches aim to create a new type of data by directly combining the raw data before inputting it into the detection framework. Typically, such methods require pixel-level correspondence between different sensor data types. Different from the early fusion-based methods, late fusion-based approaches fuse the detection results at the bounding box level. While deep fusion-based methods usually extract the features with different types of deep neural networks first and then fuse them at the features level. Currently, most multi-modal 3D object detection frameworks leverage semantic information from 2D images to improve detection accuracy in 3D LiDAR point clouds. For instance, *PointPainting* [3] , a classic early fusion-based approach, takes both the point cloud and 2D image semantic predictions as input and outputs detection results, which can be utilized with any LiDAR-based 3D object detector implemented using either point cloud or voxel-based frameworks.

This work was funded in part by the Science and Technology Development Fund, Macau SAR (Grant no. 0018/2019/AKP, SKL-IOTSC(UM)-2021-2023 and 0059/2021/AFJ), in part by the Guangdong Science and Technology Department, China (Grant no. 2020B1515130001), in part by the University of Macau (Grant No.: MYRG2020-00253-FST and MYRG2022-00059-FST), and in part by the Zhuhai UM Research Institute (Grant No.: HF-011-2021). *(Corresponding author: Zhi-Xin Yang).*

Shaoqing Xu and Zhi-Xin Yang are with the State Key Laboratory of Internet of Things for Smart City and Department of Electromechanical Engineering, University of Macau, Macau 999078, China (e-mail: shaoqing.xu@connect.um.edu.mo, zxyang@um.edu.mo)

Fang Li is with School of Mechanical Engineering, Beijing Institute of Technology. (e-mail:a319457899@163.com)

Ziying Song is with School of Computer and Information Technology, Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing 100044, China (e-mail: 22110110@bjtu.edu.cn)

Sifen Wang is with School of Transportation Science and Engineering, Beihang University, Beijing 100083, China (e-mail: sfwang@buaa.edu.cn)

Fang Jin is with Inceptio, email: jin.fang@inceptio.ai

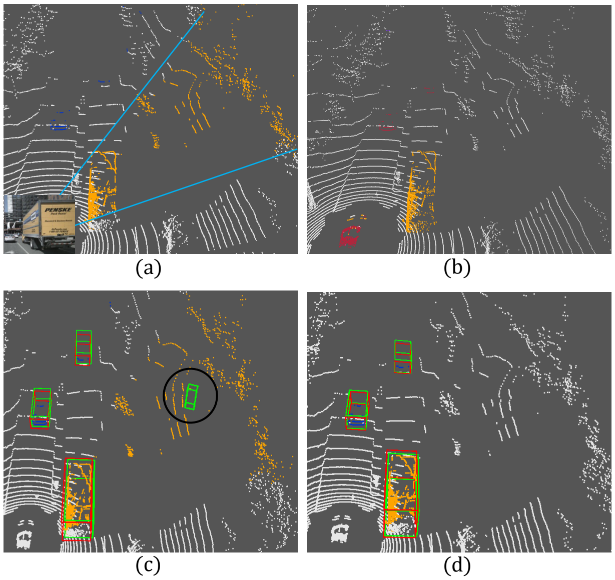

Fig. 2: (a) shows the point cloud with the 2D segmentation results, where the frustum within the blue line highlight indicates misclassified areas due to the blurring effect at the object's boundary; (b) displays the 3D segmentation results, where misclassified points are colored in red; (c) and (d) show the results based on 2D painted point cloud (with an obvious False Positive (FP) Detection) and the proposed *Multi-Sem Fusion (MSF)* framework respectively.

However, the blurring effect at object's boundary often happens inevitably in image-based semantic segmentation methods. This effect becomes much worse when reprojecting the 2D semantic projections into the 3D point clouds. An example of this effect has been shown in Fig. 2. Taking the big truck at the left bottom of the sub-fig, an instance of this issue is illustrated in Fig. 2-(a), the significant frustum area in the blue line highlight of the background (i.e., orange points) incorrectly classified as foreground due to inaccurate projection. Additionally, the re-projection of 3D points onto 2D image pixels is not a one-to-one process because of the digital quantization and many-to-one projection issues. Notably, segmentation results from the 3D point clouds as seen in Fig. 2-(b) outperform the 2D image at obstacle boundaries. However, the category classification from 3D point clouds often yields worse results (as demonstrated by points in red) when compared to the 2D image, primarily due to the loss of detailed texture information in point clouds.

The effectiveness of utilizing 2D image semantic information for 3D object detection has been demonstrated, despite the presence of some semantic errors. However, a natural question arises: can the final detection performance be further improved by fusing both the 2D and 3D semantic results in an effective way? To address this question, we introduce a general multimodal fusion framework *Multi-Seg Fusion* which fusing the multi-modal data at the semantic level to improve the final 3D object detection performance. First of all, we obtain the 2D/3D semantic information throughout 2D/3D parsing approaches by using images and raw point clouds. Then, each point has two types of semantic information after projecting point clouds onto 2D semantic images based on the intrinsic and extrinsic calibration parameters. However, the semantic results conflict usually happens for a certain point, rather than concatenating the two types of information directly, we propose an AAF strategy to fuse different types of semantic information in point or voxel-level adaptive manner. It's achieved by the learned context features with the self-attention mechanism. Specifically, attention scores are learned for each point or voxel to balance the importance of the two different semantic results.

Furthermore, in order to detect objects with different sizes in an efficient way, a DFF module is proposed here to fuse the features at multi-scale receptive fields and a channel attention network to gather related channel information in feature map. Then the fused features are passed for the following classification and regression heads to generate the final detection.

The results on both the KITTI and nuScenes dataset test splits are illustrated in Fig. 1. The figures distinctly showcase that our framework attains SOTA results, surpassing other recent SOTA methods in whether single-modality or multimodality 3D object detection methods. These extensive experiment results serve to underscore the effectiveness of our proposed framework.

Compared to the *FusionPainting* [11], We enhanced it with new relevant state-of-the-art methods, theoretical developments, and experimental results. This work significantly improves the 3D object detection accuracy hugely on the nuScenes 3D object detection benchmark. Moreover, we also evaluate the proposed framework on the KITTT 3D object detection dataset, and the experimental results on both public datasets demonstrate the superiority of our framework. In generally, this work can be characterized by the following contributions:

1) The proposed *Multi-Sem Fusion* framework offers a general approach for multi-modal fusion at semantic level, which improves 3D object detection performance by integrating multi-modal data.
2) Rather than combining different semantic results directly, an adaptive attention-based fusion module is proposed to fuse different kinds of semantic information at point or voxel-level by learning the fusion attention scores.
3) Furthermore, a deep feature fusion module is proposed to fuse deep features at different levels to better detect objects of various sizes. This is achieved by fusing deep features extracted from different layers of the network and channel attention technology.
4) The proposed fusion framework for 3D object detection is evaluated on two public benchmarks, demonstrating its superiority and achieving state-of-the-art (SOTA) results on the KITTI and nuScenes dataset. Taking the proposed framework as the baseline, we also won the champion in the fourth nuScenes object detection challenge at ICRA Workshop.

II. Related Work

A. Single-sensor 3D Object Detection

Typically, the classification of single-sensor 3D Object Detection methods can be categorized into two groups: LiDAR-based and image-based approache, which take LiDAR point cloud and image-captured data as inputs, respectively.

LiDAR-only. The existing LIDAR-based 3D object detection methods can be generally categorized into three main groups as projection-based [12]–[16], voxel-based [17]–[23] and point-based [24], [25]. SECOND [17] proposes a sparse convolution operation to replace heavy 3D convolutions for faster inference. CenterPoint [26] presents an anchor-free approach for 3D object detection and achieves state-of-the-art performance on the nuScenes benchmark. PointPillars [27] divides points into vertical columns and extracts features with PointNet [28], allowing for use in 2D object detection pipelines for 3D object detection. [25], [29]–[31] also improved discriminative feature learning.

Camera-only. In previous image-based 3D object detection methods, features are extracted by constructing a network from either single or multiple images for predicting 3D bounding boxes. Some monocular-based approaches [32], [33] attempt to regress and predict 3D boxes directly from a single image, while others [34], [35] suggest constructing intermediate-level representations and performing detection on top of them. Depth estimation [36], [37] has also been used to enhance 3D detection ability due to its necessity in the process. Another approaches [38], [39] for obtaining relatively accurate depth use stereo or multi-view images to create 3D geometry volumes for object detection. However, although depth estimated from multi-view images is better than that from a single image, it still lags behind the accuracy achieved with point clouds.

B. Multi-sensors Fusion-based 3D Object Detection

Multi-modal 3D object detection utilizes the advantages of both structure information in point cloud and textures information in image [40], [41],. F-PointNet [42], PointFusion [43] and [44] generate object proposal in the 2D image first and then fuse the features of image and point cloud for 3D BBox generation. CasA [23] achieves high detection performance based on three-stage refinement. VirConv [45] is a virtual-point-based multimodal 3D object detection which addressed the density and noise problems. HANet [46] presents a novel representation of point clouds (3-D heatmap), which obtains a Gaussian response to the Euclidean distance between points and achieves well performance. Multi-task fusion [47] has been proven to be effective in various tasks, such as jointing semantic segmentation with 3D object detection tasks as demonstrated in [24]. Using the frustum-based association method, CenterFusion [48] focuses on the fusion of radar and camera sensors. It associates the radar detections with objects in the image and creates radar-based feature maps to supplement the image features through a middle-fusion approach.

Transformers technology has achieved remarkable success in 2D detection. Building on this success, recent approaches such as TransFusion [6], AutoLignV2 [49], and DeepInteraction [50] have extended the application of Transformers to 3D object detection. These methods leverage Transformers for feature interaction between multi-modal data, marking a paradigm shift in the field. The core concept of such methods involves leveraging deformable attention to dynamically adjust feature aggregation to address the challenging projection problem and facilitate the integration of image semantic features into voxel features. This integration proves beneficial for the downstream detection task. 3D perception conducted in bird's eye-view (BEV) has attracted immense attention in recent years. Such interest is attributed to BEV's capacity to provide a physics-interpretable approach for fusing information from various views, modalities, time series, and agents. BEVFusion [7], [51] applies a lift-splat-shoot (LSS) operation to project image feature onto bird's eye view(BEV) space and concatenates it with LiDAR feature. UVTR [52] generates a unified representation in the 3D voxel space by deformable attention. How to align and integrate features from multi-modality input plays a vital role and thereby leaves extensive space to innovate.

III. Multi-Sem Fusion Framework

As shown in Fig. 3 provides an overview of the *Multi-Sem Fusion* framework which is designed to fully leverage the information obtained from a variety of sensors, we advocate fusing them at two different levels. First, the two types of information are early fused by painting the point cloud with both the 2D and 3D semantic parsing results. To handle the inaccurate segmentation results, an AAF module is proposed to learn an attention score for different sensors for the following fusion purpose. By taking the points together with fused semantic information as inputs, deep features can be extracted from the backbone network. By considering that different-sized object requires different levels features, a novel DFF module is proposed to enrich features from the backbone with different levels for exploring the global context information and the local spatial details in a deep way. Our framework comprises three primary models: a multi-modal semantic segmentation module, an Adaptive-attention fusion (AAF) module, and a deep feature fusion (DFF) module. To obtain the semantic segmentation from both RGB images and LiDAR point clouds, we can leverage any existing 2D and 3D scene parsing approaches. Then the 2D and 3D semantic information are fused by the AAF module. Bypassing the points together with fused semantic labels into the backbone network, the DFF module is used to further improve the results by aggregating the features information within various receptive fields and a channel attention module.

A. 2D/3D Semantic Parsing

2D Image Parsing. By incorporating 2D images into our approach, we can leverage their rich texture and color information to complement the analysis of 3D point clouds. For acquiring 2D semantic labels, a modern semantic segmentation method is employed here for generating pixel-wise segmentation results. More importantly, the proposed framework allows for the use of a wide range of state-of-the-art segmentation approaches without requiring modification. This agnostic approach enables researchers and practitioners to more easily leverage their preferred segmentation models and techniques (e.g., [53]–[55], etc). We employ Deeplabv3+ [53] for generating the semantic results here. The network takes

This article has been accepted for publication in IEEE Transactions on Geoscience and Remote Sensing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGRS.2024.3387732

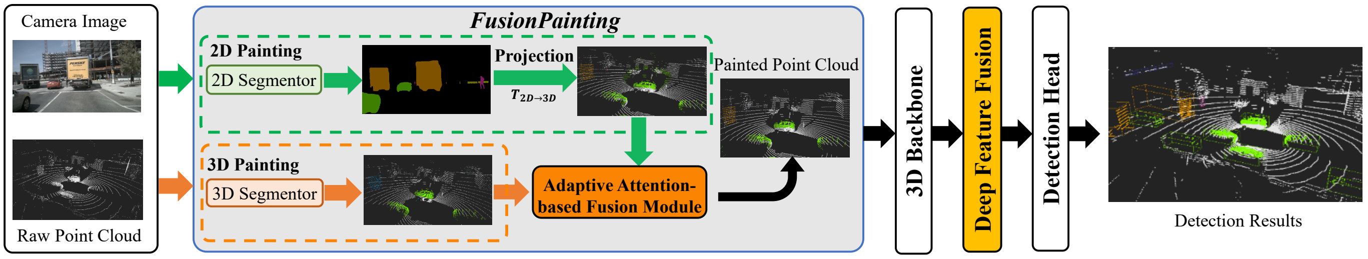

Fig. 3: Overview of the proposed *Multi-Sem Fusion* framework. We first obtain the semantic information from both input point clouds and 2D images with 2D and 3D parsing approaches. The semantic information from the two types of data is then fused at the semantic level using the proposed AAF module. Furthermore, a DFF module is also proposed to fuse the deep features at different spatial levels to boost the detection for accurate kinds of size object. Finally, fused features are sent to the detection heads for producing the final detection results.

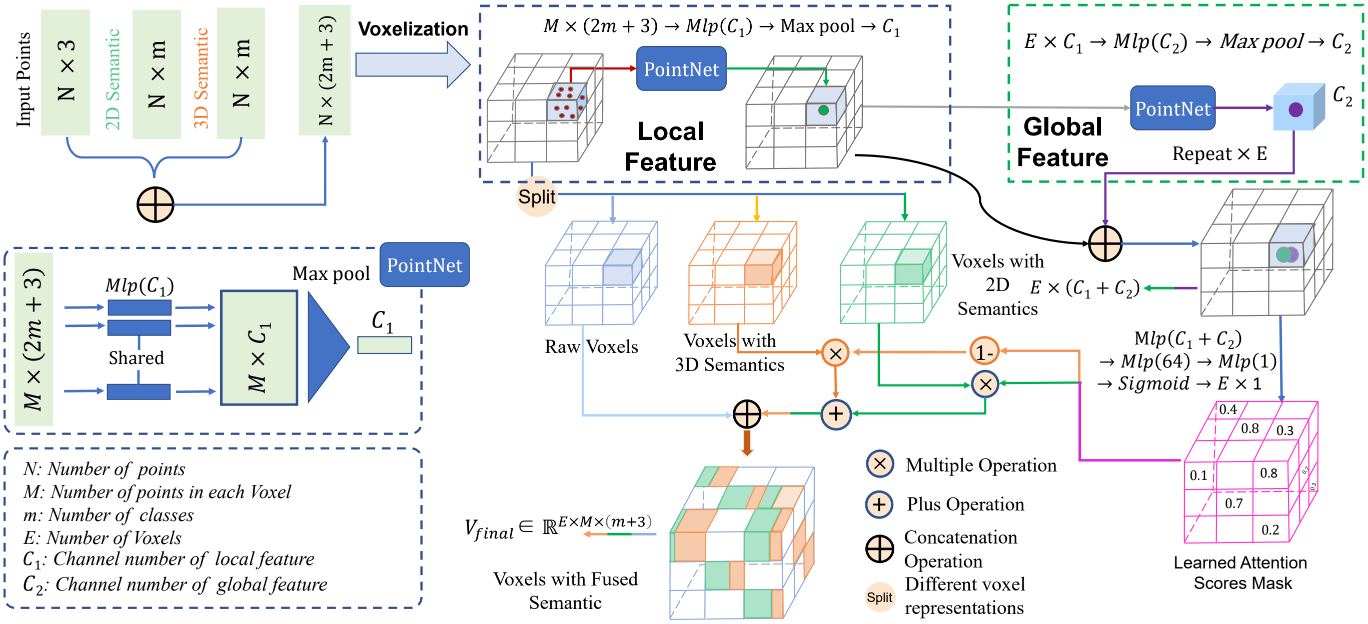

Fig. 4: The architecture of the proposed AAF module for 2D/3D semantic fusion. The proposed framework leverages the input points and 2D/3D semantic results to learn attention scores throughout an adaptive attention network, which are then used to paint the raw points or voxel with adaptive 2D/3D semantic labels

2D images as input and produces pixel-wise semantic classes scores for both the foreground and background categories. Assuming that the obtained semantic map is $S \in \mathbb{R}^{w\times h\times m}$, where (w, h) is the input image size and m is the number of category. By employing the intrinsic and extrinsic matrices, the 2D semantic information can be easily re-projected into the 3D point cloud. Specifically, by assuming that the parameter of the intrinsic matrix is $\mathbf{K} \in \mathbb{R}^{3\times 3}$, extrinsic matrix $\mathbf{M} \in \mathbb{R}^{3\times 4}$ and the original 3D points clouds is $\mathbf{P} \in \mathbb{R}^{N\times 3}$, we can derive the projection of the LiDAR 3D point cloud onto the camera as Eq.(1) shown:

$$\mathbf{P}^{'} = \mathrm{Proj}(\mathbf{K}, \mathbf{M}, \mathbf{P}), \tag{1}$$

where $\mathbf{P}^{'}$ represent the LIDAR point in camera coordinates and “Proj” denote the projection process. By this projection, we can assign the semantic segmentation results obtained from the 2D image to their corresponding 3D points which is denoted by $\mathbf{P}_{2D} \in \mathbb{R}^{N\times m}$.

3D Point Cloud Parsing. As we have mentioned above, parsing results from the point clouds can well overcome the boundary blur influence while keeping the distance information. Similar to the 2D image segmentation, any SOTA 3D parsing approach can be employed here [56]–[58]. We employ the Cylinder3D [59] for generating the semantic results because of its impressive performance on the AD scenario. More importantly, the ground truth point-wise semantic annotations can be generated from the 3D object bounding boxes roughly while any extra semantic annotations are not necessary. Specifically, for assigning semantic labels to the 3D points, we directly assign class labels to the points inside a 3D bounding box for foreground instances, while considering points outside all the 3D bounding boxes as the background. This approach enables our proposed framework to work directly on 3D detection benchmarks without requiring additional point-wise semantic annotations. After training the network, we will obtain the parsing results which is denoted by $\mathbf{P}_{3D} \in \mathbb{R}^{N\times m}$.

B. Adaptive Attention-based 2D/3D Semantic Fusion

As mentioned in previous work PointPainting [3], 2D semantic segmentation network have achieved impressive performance, however, the blurring effect at the shape boundary is also serious. Therefore, the point clouds with 2D semantic segmentation usually have misclassified regions around the objects' boundary as shown in the frustum region in Fig. 2 (a) The points behind the big truck has been totally misclassified. On the contrary, the parsing results from the 3D point clouds usually perform a clear and accurate object boundary without blurring effect e.g., Fig. 2(b). However, the disadvantages of the 3D segmentor are also obvious. One drawback is that without the color and texture information, the 3D segmentor is difficult to distinguish these categories with similar shapes from the point cloud-only. In order to boost advantages while suppressing disadvantages, an AAF module has been proposed to adaptively combine the 2D/3D semantic segmentation results. Then the Optimized semantic information is ready for the following 3D object detectors backbone to further extract the enhanced feature to improve the final object detection accuracy results.

AAF Module. The detailed architecture of the proposed AAF module is depicted in Fig. 4. The input point clouds are defined as a set of points $\{\mathbf{P}_i, i = 1, 2, 3...N\}$, with each point $\mathbf{P}_i$ containing (x, y, z) coordinates and other optional information such as intensity. In the following context, we'll focus solely on the coordinates that meaning only the coordinates (x, y, z) are considered as input data. Our objective is to develop an efficient strategy for integrating semantic information from both 2D images and 3D point clouds.. Here, we propose a novel approach that employ the adaptive attention fusion(AAF) module to learn each cell attention score in voxel level or point level to adaptively combine the two types of semantic results. Specifically, the module begins by concatenating the point clouds coordinate attributes (x, y, z) and 2D/3D semantic segmentation labels to obtain a fused point clouds with the shape of $N \times (2m + 3)$. In order to reduce memory consumption during the fusion process, we have implemented a voxel-level fusion approach instead of operating at the point level. To achieve this, the point clouds are first evenly divided into voxels, and the voxels are represented as $\{V_i, i = 1, 2, 3...E\}$, where E is the overall count of the voxels. Specially, each voxel cell $V_i = (\mathbf{P}_i, \mathbf{V}^i_{2D}, \mathbf{V}^i_{3D}) \in \mathbb{R}^{M\times(2m+3)}$ containing a fixed number of M points, and each point attributes consist of $\mathbf{P}_i$, $\mathbf{V}^i_{2D}$, $\mathbf{V}^i_{3D}$ which represent point coordinates, predicted 2D and 3D semantic segmentation results vector respectively. To ensure consistency across voxels, we utilize a sampling technique to maintain a fixed number of points in each voxel. Subsequently, we utilize local and global feature aggregation techniques to calculate attention weights for each voxel, which help to determine the relative importance of 2D and 3D semantic segmentation labels. In other words, the AFF module chooses which category of 2D/3D semantic information should be trusted in the voxel.

In order to get local features, we utilize a PointNet [28]-like module to extract voxel-wise information within each non-empty voxel. Specifically, for the i-th voxel, the corresponding local feature is represented as:

$$V_i = f(p_1, p_2, \cdots, p_M) = \max_{i=1,\ldots,M}\{\text{MLP}_l(p_{i'})\} \in \mathbb{R}^{C_1}, \tag{2}$$

where $\{p_{i'}, i^{'} = 1, 2, 3...M\}$ indicates the LIDAR points insize each voxel. $\text{MLP}_l(\cdot)$ and max represent the muti-layer perception (MLP) and max-pooling module, respectively. Specifically, $\text{MLP}_l(\cdot)$ is composed of a linear layer, a batch normalization layer, an activation layer. Throughout the network, we got outputs for each voxel local feature with C_1 channels. For global feature information, we aggregate information based on the E voxels where E is the total number of voxels. First, we employ a $\text{MLP}_g(\cdot)$ module to rich each voxel features from C_1 dimensions to C_2. Then, we apply another PointNet-like module on all the voxels as follows expression:

$$V_{global} = f(V_1, V_2, \cdots, V_E) = \max_{i=1,\ldots,E}\{\text{MLP}_g(V_i)\} \in \mathbb{R}^{C_2}. \tag{3}$$

To obtain the final fused local and global features, we expand the global feature vector V_{global} to the same size as the number of voxels and then concatenate it with each local feature V_i. This operation creates a combined feature representation for each voxel that captures both local and global information as $V_{gl} \in \mathbb{R}^{E\times(C_1+C_2)}$.

After getting fused features V_{gl} from the network, we can estimate an attention score of two kinds of semantic information results for each point in voxel throughout another MLP module $\text{MLP}_{att}(\cdot)$ on V_{gl} and a Sigmod activation function $\sigma(\cdot)$. Then, we multiply the resulting attention confidence score with corresponding one-hot semantic vectors for each voxel, as shown in Eq. (4), Eq. (5):

$$\mathbf{V}^i_{a.S} = \mathbf{V}^i_{2D} \times \sigma(\text{MLP}_{att}(V^i_{gl})), \tag{4}$$

$$\mathbf{V}^i_{a.T} = \mathbf{V}^i_{3D} \times (1 - \sigma(\text{MLP}_{att}(V^i_{gl}))) \tag{5}$$

where $\mathbf{V}^i_{2D}$ and $\mathbf{V}^i_{3D}$ are the point labels in each voxel from 2D and 3D semantic segmentation results which are encoded with one-hot format. The final semantic vector V^i_{final} of each voxel can be obtained by concatenating or element-wise addition of $\mathbf{V}^i_{a.T}$ and $\mathbf{V}^i_{a.S}$.

After undergoing the Sigmoid operation, the resulting values fall within the range of 0 to 1, The weighted semantic information, represented as $\mathbf{V}_{a.S}$ and $\mathbf{V}_{a.T}$ is obtained by element-wise multiplication with their respective semantic result vectors.

$$\mathbf{V}_{final} = \mathbf{V}_{a.T} + \mathbf{V}_{a.S} \tag{6}$$

The augmented fusion information is aggregated according to Eq. (6) shown. Following this, the resultant information is seamlessly integrated with the raw voxel features (x, y, z). Finally, the enhanced results denoted as $\mathbf{V}_{final} \in \mathbb{R}^{E\times M\times(m+3)}$ which intricately captures adaptively weighted insights derived from both 2D and 3D semantic labels.

For point-level operations, it is imperative to treat each point feature as a local feature and leverage neighboring points' features, rather than considering the entire point cloud feature as the global feature. These enhanced features are then fed into Eq. (4), Eq. (5) and Eq. (6) to obtain the final fused enhanced feature. However, it's important to note that this approach may result in increased GPU memory consumption.

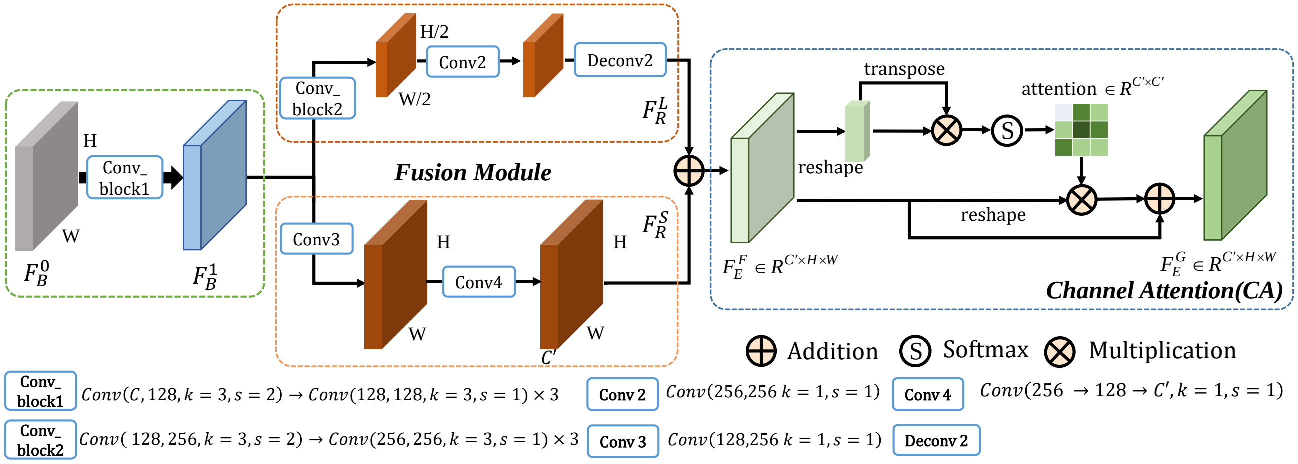
H/2
Conv2
Deconv2
Conv_ block2
W/2
F_R^L
Fusion Module
H
Conv_ block1
W
F_B^0
F_B^1
Conv3
H
Conv4
H
W
W
C'
F_R^S
transpose
attention $\in R^{C'\times C'}$
S
reshape
reshape
$F_E^F \in R^{C'\times H\times W}$
$F_E^G \in R^{C'\times H\times W}$
Channel Attention(CA)
Addition
Softmax
Multiplication
Conv_ block1 $Conv(C, 128, k = 3, s = 2) \rightarrow Conv(128, 128, k = 3, s = 1) \times 3$
Conv_ block2 $Conv(128, 256, k = 3, s = 2) \rightarrow Conv(256, 256, k = 3, s = 1) \times 3$
Conv 2 $Conv(256, 256\ k = 1, s = 1)$
Conv 3 $Conv(128, 256\ k = 1, s = 1)$
Conv 4 $Conv(256 \rightarrow 128 \rightarrow C', k = 1, s = 1)$
Deconv 2

Fig. 5: An illustration of the proposed Deep Feature Fusion (DFF) module which includes one *fusion* module and one *channel attention* module respectively. The *fusion* module includes two branches for producing features with different field-of-view.

C. Deep Feature Fusion Module

In AD scenarios, determining the identity and location of objects is crucial for the subsequent planning and control modules. Therefore, it is not only necessary to recognize what objects are present, but also where they are located. In typical object detection frameworks, they correspond to the classification and regression branches respectively. Empirically, global context information is important to recognize the specific class attributes. On the contrary, the object's attributes (e.g., dimension, orientation, and precise location, etc) regression branch prioritize the capture of detailed spatial information around the ROI (region of interest) in a relatively small range. For accurate kinds of size object detection, therefore various scales receptive fields are necessary. This issue has been considered in most object detection frameworks. However, how to use various fields of view fields in an efficient way is vitally important.

To handle this issue, a specific DFF module is proposed to aggregate both the low-level and high-level features with different receptive fields. The Fig. 5 shows the architecture of the DFF module. First, the backbone features F_B^0 from the feature extractor pass the *Conv_block1* with several convolution layers to obtain the F_B^1 as a basic feature. Here, *Conv_block1* has four Conv modules and the first $conv$ module takes C channels as input and outputs 128 channels, and the following three $convs$ share the same input channels and output channels. For each $conv$ module in the Fig. 5, it consists one $Con2d$, a batch normalization layer, and a Rectified Linear Unit (ReLU) activation layer. For easy understanding, we have given the stride and kernel size for each $conv$ operation at the bottom of Fig. 5. Then, the feature F_B^1 will pass two branches to obtain the features with different receptive fields. For one branch, the features are down-sampled into $1/2$ size with $Conv-block2$ first and then pass the $Conv2$ operation. Finally, the outputs are up-sampled into the feature map $F_R^L \in \mathbb{R}^{H\times W\times C'}$ with $Deconv2$. For the other branch, F_B^1 will pass $Covn3$ and $Conv4$ to obtain the features F_R^S consecutively. And the shape of the output F_R^L is the same as F_R^S. Furthermore, we use the addition operation for fusing different level perception field features to improve the feature representation.

After adding the high-level and low-level features element-wisely, a channel-attention (CA) module like [60] is employed to further fuse both of them. The architecture of the module can be found in Fig. 5, which is named **CA**. Usually, the channel feature from low-level to high-level throughout the backbone will take the loss of information. In order to minimize this influence, we utilize the CA module to selectively emphasize interdependent channel maps by integrating relative features among all channel maps. Specially, the enhanced feature $F_E^F \in \mathbb{R}^{C\times H\times W}$ is reshaped to $F_E^F \in \mathbb{R}^{C\times N}$, where N is the pixel numbers of each channel. The channel dependencies for feature F_E^F are captured using the similar self-attention mechanism, followed by updating the enhanced feature through a weighted sum of all channel maps. The process as shown in Eq.(7), Eq.(8):

$$x_{ji} = \frac{\exp\left(F_{E\,i}^{F}\cdot F_{E\,j}^{F}\right)}{\sum_{i=1}^{C}\exp\left(F_{E\,i}^{F}\cdot F_{E\,j}^{F}\right)} \qquad (7)$$

$$F_{E\,j}^{G} = \beta\sum_{i=1}^{C}\left(x_{ji}F_{E\,i}^{F}\right) + F_{E\,j}^{F} \qquad (8)$$

where x_{ji} measures the i^{th} channel's impact on the j^{th} channel. In addition, we perform a matrix multiplication between the transpose of x, which channel attention map $\mathbf{x} \in \mathbb{R}^{C\times C}$. Moreover, β is a scale parameter, which gradually learns a weight from 0. The Eq. (8) shows that the final feature of each channel is a weighted sum of the features of all channels and original features, which models the long-range semantic dependencies between feature maps. Finally, F_E^G is taken as the inputs for the following classification and regression heads.

D. 3D Object Detection Framework

The proposed AAF and DFF modules are detector independent and any off-the-shelf 3D object detectors can be directly employed as the baseline of our proposed framework. The 3D

Methods	**mAP** (Mod.)(%)	**Car** $AP(\%)$			**Pedestrian** $AP(\%)$			**Cyclist** $AP(\%)$		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
SECOND [17]	66.64	90.04	81.22	78.22	56.34	52.40	46.52	83.94	66.31	62.37
SECOND *	68.11	91.04	82.31	79.31	59.28	54.18	50.20	85.11	67.52	63.36
Improvement	+1.47	+1.00	+1.09	+1.09	+2.94	+1.78	+3.68	+1.17	+1.54	+0.99
Pointpillars [27]	62.90	87.59	78.17	75.23	53.58	47.58	44.04	82.21	62.95	58.66
PointPillars*	65.78	89.58	78.60	75.63	60.22	54.23	49.49	84.83	64.50	60.17
Improvement	+2.88	+1.99	+0.43	+0.4	+6.64	+6.65	+5.45	+2.62	+1.55	+1.51
PV-RCNN [29]	71.82	92.23	83.10	82.42	65.68	59.29	53.99	91.57	73.06	69.80
PV-RCNN*	73.95	91.85	84.59	82.66	69.12	61.61	55.96	94.90	75.65	71.03
Improvement	+2.13	-0.38	+1.49	+0.24	+3.44	+2.32	+1.97	+3.33	+2.59	+1.23
CasA+PV [23]	71.79	92.53	85.59	83.27	68.50	59.39	53.26	90.35	70.39	65.42
CasA+PV*	73.78	93.72	86.69	86.54	71.02	61.87	55.44	92.23	72.79	67.32
Improvement	+1.99	+1.19	+1.10	+2.67	+2.52	+2.48	+2.18	+1.88	+2.40	+1.90
VirConv [45]	75.28	93.48	88.59	87.69	73.78	65.13	58.32	89.48	72.13	66.32
VirConv*	76.67	93.91	88.95	88.41	75.79	66.75	60.18	93.35	74.32	69.36
Improvement	+1.39	+0.43	+0.36	+0.72	+2.01	+1.42	+1.86	+3.87	+2.19	+3.02

TABLE I: 3D object detection evaluation on KITTI "val" split into different baseline approaches, where * represents the boosted baseline by adding the proposed fusion modules. "Easy", "Mod." and "Hard" represent the three difficult levels defined by official benchmark and **mAP** (Mod.) represents the average **AP** of "Car", "Pedestrian" and "Cyclist" on "Mod." level. For easy understanding, we also highlight the improvements with different colors, where red represents an increase and green represents a decrease compared to the baseline method. This table is better to be viewed in color mode.

Methods	**Reference**	**Modality**	**Car 3D** $AP(\%)$			**Car BEV** $AP(\%)$		
			Easy	Mod.	Hard	Easy	Mod.	Hard
PointRCNN [25]	CVPR 2019	L	86.96	75.64	70.70	-	-	-
PV-RCNN [29]	CVPR 2020	L	90.25	81.43	76.82	94.98	90.65	86.14
VoxelRCNN [61]	AAAI 2021	L	90.90	81.62	77.06	-	-	-
CasA [23]	TGRS 2022	L	91.58	83.06	80.08	95.19	91.54	86.82
Graph-Po [62]	ECCV 2022	L	91.79	83.18	77.98	95.79	92.12	87.11
3D-HANet [46]	TGRS 2023	L	90.79	84.18	77.57	94.33	91.13	86.33
PVT-SSD [63]	CVPR 2023	L	90.6	82.29	76.85	95.23	91.63	86.43
3D-CVF [64]	ECCV 2020	L+C	89.20	80.05	73.11	93.52	89.56	82.45
Focals-Conv [65]	CVPR 2022	L+C	90.55	82.28	77.59	92.67	89.00	86.33
VPFNet [22]	TMM 2022	L+C	91.02	83.21	78.20	93.94	90.52	86.25-
Graph-VoI [62]	ECCV 2022	L+C	91.89	83.27	77.78	95.69	90.10	86.85
SFD [66]	CVPR 2022	L+C	91.73	84.76	77.92	95.64	91.85	86.83
VirConv-L [45]	CVPR 2023	L+C	91.41	85.05	80.22	95.53	91.95	87.07
Ours *	-	L+C	91.37	85.21	80.37	95.76	92.67	88.43

TABLE II: Evaluation of bird's-eye view object detection and 3D object detection results on KITTI test dataset with other SOTA methods. Notably, the results in $Ours$ is based on VirCon-L and outperform all the other methods in both 3D AP and BEV AP metrics.

detector receives the points or voxels produced by the AAF module as inputs and can keep backbone structures unchanged to obtain the backbone features. Then, the backbone features are boosted by passing the proposed DFF module. Finally, detection results are generated from the classification and regression heads.

IV. Experimental Results

To verify the effectiveness of our proposed framework, we evaluate it on two large-scale 3D object detection dataset in AD scenarios as KITTI [67] and nuScenes [68]. Furthermore, the proposed modules is also evaluated on different kinds of baselines for verifying its generalizability, including SECOND [17], PointPilars [27] and PVRCNN [29], CasA [23], VirConv [45], BEVFusion [7] etc.

A. Evaluation on KITTI Dataset

KITTI is one of the most popular benchmarks for 3D object detection in AD, which contains 7481 samples for training and 7518 samples for testing. The objects in each class are divided into three difficulty levels as "easy", "moderate", and "hard", according to the object size, the occlusion ratio, and the truncation level. Since the ground truth annotations of the test samples are not available and the access to the test server is limited, we follow the idea in [69] and split the training data into "train" and "val" where each set contains 3712 and 3769 samples respectively. In this dataset, both the LiDAR point clouds and the RGB images have been provided. In addition, both the intrinsic parameters and extrinsic parameters between different sensors have been given.

Evaluation Metrics. We follow the official metrics provided by the KITTI for evaluation. AP_{70} is used for "Car" category while AP_{50} is used for "Pedestrain" and "Cyclist". Specifically, before the publication of [70], the KITTI official benchmark used the 11 recall positions for comparison. After that, the official benchmark changes the evaluation criterion from 11-points to 40-points because the latter one is proved to be more stable than the former [70]. Therefore, we use the 40-points criterion for all the experiments here. In addition, similar to [3], the average AP (mAP) of three Classes for "Moderate" is also taken as an indicator for evaluating the average performance on all three classes.

Baselines. Five different baselines have been used for evaluation on KITTI:

1) *SECOND [17]* is the first to employ the sparse convolution on the voxel-based 3D object detection framework to accelerate the efficiency of LiDAR-based 3D object detection.
2) *PointPillars [27]* is proposed to further improve the detection efficiency by dividing the point cloud into vertical pillars rather than voxels. For each pillar, *Pillar Feature Net* is applied to extract the point-level feature.
3) *PV-RCNN [29]* is a hybrid point-voxel-based 3D object detector, which can utilize the advantages from both the point and voxel representations.
4) *CasA+PV [23]* is a method that aggregates object features from multiple stages refines region proposals by cascade attention structure. And $+pv$ means that conducted experiments on the *PV-RCNN* framework.
5) *VirConv [45]* is based on a new operator VirConv (Virtual Sparse Convolution), for virtual-point-based 3D object detection.

Implementation Details. DeeplabV3+ [71] and Cylinder3D [72] are employed for 2D and 3D scene parsing respectively. More details, the DeeplabV3+ is pre-trained on Cityscape [1], and the Cylinder3D is trained on KITTI point clouds by taking points in 3D ground trues bounding box as foreground annotation. For AAF module, $m = 4$, $C_1 = 64, C_2 = 128$, respectively. The voxel size for *PointPillars* and *SECOND, CasA+PV, VirConv* are $0.16m \times 0.16m \times 4m$ and $0.05m \times 0.05m \times 0.1m$, respectively. In addition, both of the two baselines use the same optimization (e.g., AdamW) and learning strategies (e.g., one cycle) for all the experiments. In the DFF module, we set $C = 256$ and $C' = 512$ for *SECOND*, *PV-RCNN*, *CasA+PV* and *VirConv* framework, $C = 64$ and $C' = 384$ for *PointPillars* network. The kernel and stride size are represented with k and s in Fig.5, respectively.

The proposed approach is implemented with PaddlePaddle [73] and all the methods are trained on NVIDIA Tesla V100 with 8 GPUs. The AdamW is taken as the optimizer and the one-cycle learning strategy is adopted for training the network. For *PointPillars*, the batch size is 4 per GPU and the maximum learning rate is 0.003 while the bath size is 2 per GPU and the maximum learning rate is 0.001 for *PV-RCNN, SECOND, CasA+PV and VirConv*.

Quantitative Evaluation. We verified our framework on five different baselines, including the recent SOTA method CasA+PV and VirConv. From the Tab. I, we can clearly see that remarkable improvements have been achieved on the five baselines across all the categories of the index of 3D mAP which highlighting its adaptability to different state-of-the-art approaches. Taking Pointpillars as an example, the proposed method has achieved 0.43%, 6.65%, 1.55% points improvements on "Car", "Pedestrian" and "Cyclist" respectively. Interestingly, compared to "Car", "Pedestrian" and "Cyclist" give much more improvements by the fusion painting modules. And we also got impressive results on KITTI test split dataset based on VirConv baseline. The result of the 3D object and Bird's-eye view on 'Car' category as shown as Tab. II.

B. Evaluation on the nuScenes Dataset

The nuScenes [68] is a recently released large-scale (with a total of 1,000 scenes) AD benchmark with different kinds of information including LiDAR point cloud, radar point, Camera images, and High Definition Maps, etc. For a fair comparison, the dataset has been divided into "train", "val", and "test" three subsets officially, which includes 700 scenes (28130 samples), 150 scenes (6019 samples), and 150 scenes (6008 samples) respectively. Objects are annotated in the LiDAR coordinate and projected into different sensors' coordinates with pre-calibrated intrinsic and extrinsic parameters. For the point clouds stream, only the keyframes (2fps) are annotated. With a 32 lines LiDAR scanner, each frame contains about 300,000 points for 360-degree viewpoint. For the object detection task, the obstacles have been categorized into 10 classes as "car", "truck", "bicycle" and "pedestrian" etc. Besides the point clouds, the corresponding RGB images are also provided for each keyframe, and for each keyframe, there are 6 cameras that can cover 360 fields of view.

Evaluation Metrics. The evaluation metric for nuScenes is totally different from KITTI and they propose to use mean Average Precision (mAP) and nuScenes detection score (NDS) as the main metrics. Different from the original mAP defined in [78], nuScenes consider the BEV center distance with thresholds of $\{0.5,\ 1,\ 2,\ 4\}$ meters, instead of the IoUs of bounding boxes. NDS is a weighted sum of mAP and other metric scores, such as average translation error (ATE) and average scale error (ASE). For more details about the evaluation metric please refer to [68].

Baselines. We integrated the proposed module into four state-of-the-art (SOTA) baselines to assess its effectiveness. Similar to the KITTI dataset, we utilized both *SECOND* and *PointPillars* and *CenterPoint* [26]. Additionally, we also extended our framework to the trending topic of multi-modal fusion, specifically on *BEVFusion* [7]. The experiments consistently demonstrate the efficacy of our method across various SOTA detection frameworks.

Implementation Details HTCNet [54] and *Cylinder3D* [72] are employed here for obtaining the 2D and 3D semantic segmentation results respectively. We used the *HTCNet* model trained on nuImages [2] dataset directly for generating the semantic labels. For *Cylinder3D*, we train it directly on the nuScenes 3D object detection dataset while the point cloud semantic label is produced by taking the points inside each bounding box. In AAF module, $m = 11$, $C_1 = 64$ and $C_2 = 128$ respectively. In DFF module, $C = 256$ and $C' = 512$ for *SECOND*, *CenterPoint* and *BEVFusion* while $C = 64$ and $C' = 384$ for *PointPillars*. The setting for kernel size k and stride s are given in Fig. 5 and the same for all

[1] https://www.cityscapes-dataset.com

[2] https://www.nuscenes.org/nuimages

Methods	NDS (%)	mAP (%)	AP (%)									
			Car	Truck	Bus	C.V.	Trailer	Pedestrian	Moto.	Bicycle	T.C.	Barrier
SECOND [17]	61.96	50.85	81.61	51.93	68.53	17.95	38.19	77.37	40.14	18.16	56.86	57.75
SECOND*	67.61	62.61	84.77	60.32	72.41	23.59	42.89	83.36	66.50	54.62	74.99	63.67
Improvement ↑	**+5.65**	**+11.76**	+3.16	+8.39	+3.88	+5.64	+4.70	+5.99	+26.36	+36.46	+18.13	+5.92
PointPillars [27]	57.50	43.46	80.67	49.35	62.01	11.64	34.86	70.80	26.74	5.27	44.00	49.23
PointPillars*	66.43	61.75	84.79	57.50	70.52	22.43	42.32	83.41	66.97	54.68	75.43	59.42
Improvement ↑	**+8.93**	**+18.29**	+4.12	+8.15	+8.51	+10.79	+7.46	+12.61	+40.23	+49.41	+31.43	+10.19
CenterPoint [26]	64.82	56.53	84.73	54.52	66.95	15.81	36.13	83.42	56.81	37.57	64.76	64.56
CenterPoint*	69.91	65.85	87.00	61.91	71.53	23.97	41.22	87.95	73.85	64.06	79.89	67.14
Improvement ↑	**+5.09**	**+9.32**	+2.77	+7.39	+4.58	+8.16	+5.09	+4.53	+17.04	+26.49	+15.13	+2.58
BEVFusion-MIT [7]	71.3	68.4	88.5	65.3	75.7	28.6	42.0	89.1	77.1	66.3	77.5	72.8
BEVFusion*	72.1	69.6	89.1	66.4	76.8	29.7	42.9	89.9	79.5	67.8	80.3	73.6
Improvement ↑	**+0.8**	**+1.2**	+0.6	+1.1	+1.2	+1.1	+0.9	+0.8	+2.4	+1.5	+2.8	+2.58

TABLE III: Evaluation results on nuScenes validation dataset. "NDS" and "mAP" mean nuScenes detection score and mean Average Precision. "T.C.", "Moto." and "C.V ." are short for "traffic cone", "motorcycle", and "construction vehicle" respectively. " * " denotes the improved baseline by adding the proposed fusion module. The red color represents an increase compared to the baseline. This table is also better to be viewed in color mode.

Methods	Modality	NDS(%)	mAP(%)	AP (%)									
				Car	Truck	Bus	Trailer	C.V.	Ped	Moto	Bicycle	T.C	Barrier
VoxelNeXt [74]	L	70.0	64.5	84.6	53.0	64.7	55.8	28.7	85.8	73.2	45.7	79.0	**74.6**
Focals Conv [65]	L	70.0	63.8	86.7	56.3	**67.7**	59.5	23.8	87.5	64.5	36.3	81.4	74.1
3DSSD [75]	L	56.4	46.2	81.2	47.2	61.4	30.5	12.6	70.2	36.0	8.6	31.1	47.9
TransFusion-L [6]	L	70.2	65.5	86.2	56.7	66.3	58.8	28.2	86.1	68.3	44.2	82.0	78.2
CenterPoint [26]	L	67.3	60.3	85.2	53.5	63.6	56.0	20.0	54.6	59.5	30.7	78.4	71.1
VirConv [45]	L	71.2	67.2	-	-	-	-	-	-	-	-	-	-
MGTANet [76]	L	72.7	67.5	88.5	59.8	67.2	61.5	30.6	87.3	75.8	52.5	85.5	66.3
PointPainting [3]	L & C	58.1	46.4	77.9	35.8	36.2	37.3	15.8	73.3	41.5	24.1	62.4	60.2
UVTR [52]	L & C	71.1	67.1	**87.5**	56.0	67.5	59.5	**33.8**	86.3	73.4	**54.8**	79.6	73.0
3DCVF [77]	L & C	62.3	52.7	83.0	45.0	48.8	49.6	15.9	74.2	51.2	30.4	62.9	65.9
BEVFusion [51]	L & C	71.8	69.2	88.1	60.9	39.9	62.1	34.4	89.2	72.2	52.2	85.2	78.2
TransFusion [6]	L & C	71.7	68.9	87.1	60.0	68.3	60.8	33.1	88.4	73.6	52.9	86.7	78.1
BEVFusion-MIT [7]	L & C	72.9	70.2	88.6	60.1	69.8	63.8	39.4	89.2	74.1	51.0	86.5	80.0
Our proposed	L & C	**73.6**	**71.2**	88.5	**60.8**	**71.3**	**64.6**	**39.1**	**89.7**	**78.3**	53.5	**87.1**	**79.5**

TABLE IV: Comparison with other SOTA methods on the nuScenes 3D object detection testing benchmark. "L" and "C" in the modality column represent LiDAR and Camera sensors respectively. The result of ours is based on BEVFusion-MIT. For easy understanding, the highest score in each column is shown in bold font. To be clear, only the results with publications are listed here.

four detectors. The voxel size for *PointPillars* and *SECOND* are $0.2m \times 0.2m \times 8m$, $0.1m \times 0.1m \times 0.2m$, respectively. And *CenterPoint* and *Transfusion* employed $0.075m \times 0.075m \times 0.075m$. We use AdamW [79] as the optimizer with the max learning rate is 0.001. Following [68], 10 previous LiDAR sweeps are stacked into the keyframe to make the point clouds more dense.

All the baselines are trained on NVIDIA Tesla V100 (8 GPUs) with a batch size of 4 per GPU for 20 epochs. AdamW is taken as the optimizer and the one-cycle learning strategy is adopted for training the network with the maximum learning rate is 0.001.

Quantitative Evaluation.

The proposed framework has been evaluated on nuScenes benchmark for both "val" and "test" splits. The results of the comparison with four baselines are given in Tab. III. From this table, we can see that significant improvements have been achieved on both the mAP and NDS across all four baselines. For the *PointPillars*, the proposed module gives **8.93** and **18.29** points improvements on NDS and mAP respectively. For *SECOND*, the two values are **5.65** and **11.76** respectively. Even for the *CenterPoint*, the proposed module can also give **5.09** and **9.32** points improvements. Event on the strong baseline *BEVFusion*, we still get more appreciable results with **0.8** NDS and **1.2** mAP profits. And results in Fig. 6.(c) can obviously find that the proposed modules can robustly boost the detection performance on different baselines.

The results in Fig. 6 (b) also shows the improvements on different categories and we can easily find that all the classes have been improved. Specially, small sizes such as "Traffic Cone", "Moto" and "Bicycle" have received more improvements compared to other categories. Taking "Bicycle" as an example, the mAP has been improved by **36.46%, 49.41%** and **26.49%** compared to *SECOND*, *PointPillar* and *Centerpoint* respectively. This phenomenon can be explained as these categories with small sizes are hard to be recognized in point clouds because of a few LiDAR points on them. In this case, the semantic information from the 2D/3D parsing results is extremely helpful for improving 3D object detection performance.

To compare the proposed framework with other SOTA methods, we evaluate our method(adding fusion modules on *BEVFusion*) on the nuScenes test split. The detailed results are given in Tab. IV. From this table, we can find that the proposed method achieves the best performance on both the mAP and NDS scores. Even compared to the latest novelty baseline *BEVFusion*, 0.8 NDS and 1.2 mAP points improvements have been achieved by adding our proposed fusion modules. For easy understanding, we have highlighted the best performances in bold in each column.

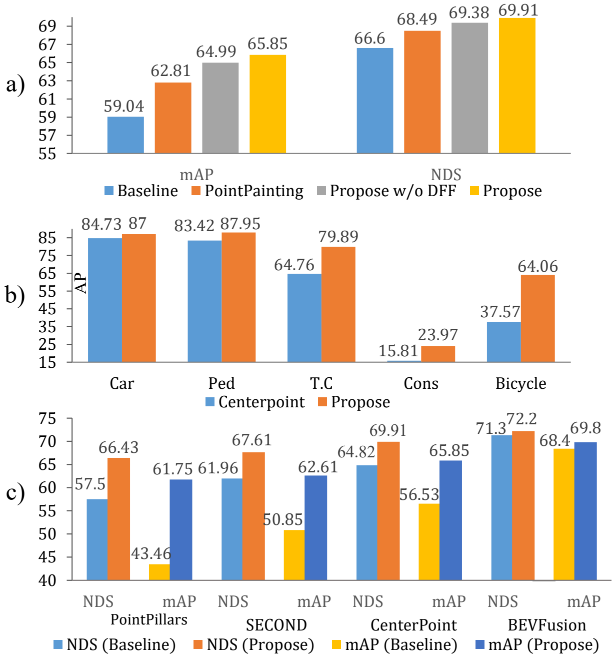

Fig. 6: The performance of *Multi-Sem Fusion (MSF)*. a) gives the performance of CenterPoint [80] with the proposed modules on the public nuScenes benchmark. b) gives the improvements on different categories respectively. c) illustrate the improvements on four different baselines. In addition, "w/o" represent "without" in short.

C. Ablation Studies

To verify the effectiveness of different modules, a series of ablation experiments have been designed to analyze the contribution of each component for the final detection results. Specially, three types of experiments are given as *different semantic representations* and *different fusion modules* and *effectiveness of channel attention*.

Representations	**mAP** (%)	**NDS** (%)
PointPillar [27]	43.46	57.50
Semantic ID	50.96 (+7.50)	60.69 (+3.19)
Onehot Vector	52.18 (+8.72)	61.59 (+4.09)
Semantic Score	53.10 (+9.64)	62.20 (+4.70)

TABLE V: Ablation studies on the nuScenes [67] dataset for fusing the semantic results with different representations.

Strategies	AP(%)			mAP(%)
	Car(Mod.)	Ped.(Mod.)	Cyc.(Mod.)	
SECOND [17]	88.99	56.21	70.65	71.95
3D Sem.	+ 0.81	+ 0.70	+ 0.63	+ 0.71
2D Sem.	+ 1.09	+ 1.35	+ 1.20	+ 1.41
AAF	+ 1.62	+ 1.78	+ 1.91	+ 1.77
AAF & DFF	+ 2.63	+ 3.45	+ 2.63	+ 2.90

TABLE VI: Evaluation of ablation studies on the public KITTI [67] dataset. Similar to PointPainting [3], we provide the 3D BEV detection here. To be clear, only the results of "Mod" have been given and **mAP** is the average mean of all the three categories.

Different Semantic Representations. First of all, we investigate the influence of the different semantic result representations on the final detection performance. Three different representations as "Semantic ID", "Onehot Vector" and "Semantic Score" are considered here. For "Semantic ID", the digital class ID is used directly and for the "Semantic Score", the predicted probability after the Softmax operation is used. In addition, to convert the semantic scores to a "Onehot Vector", we assign the class with the highest score as "1" and keep other classes as "0". Here, we just add the semantic feature to the original (x, y, z) coordinates with concatenation operation and *PointPillars* is taken as the baseline on nuScenes dataset due to its efficiency of model iteration. From the results in Tab. V, we can easily find that the 3D semantic information can significantly boost the final object detection performance regardless of different representations. More specifically, the "Semantic Score" achieves the best performance among the three which gives 9.64 and 4.70 points improvements for **mAP** and **NDS** respectively. We guess that the semantic score can provide more information than the other two representations because it not only provides the class ID but also the confidence for all the classes.

Different Fusion Modules. We also execute different experiments to analyze the performances of each fusion module on both KITTI and nuScene dataset. The *SECOND* is chosen as the baseline for KITTI while all the three detectors are verified on nuScene. The results are given in Tab. VI and Tab. VII for KITTI and nuScenes respectively. To be clear, the "3D Sem" and "2D Sem." represent the 2D and 3D parsing results. "AAF" represents the fusion of the 2D and 3D semantic information with the proposed adaptive attention module and the "AAF + DFF" represents the module with both the two fusion strategies.

Strategies	**SECOND**		**PointPillars**		**CenterPoint**	
	mAP(%)	**NDS**(%)	**mAP**(%)	**NDS**(%)	**mAP**(%)	**NDS**(%)
SECOND	50.85	61.96	43.46	57.50	56.53	64.82
3D Sem.	+ 3.60	+ 1.45	+ 8.72	+ 4.09	+ 4.59	+ 1.86
2D Sem.	+ 8.55	+ 4.07	+15.64	+ 7.55	+ 6.28	+ 3.67
AAF	+ 11.30	+ 5.45	+17.31	+ 8.54	+ 8.46	+ 4.56
AAF & DFF	+ 11.76	+ 5.65	+ 18.19	+ 8.93	+ 9.32	+ 5.09

TABLE VII: Ablation studies for different fusion strategies on the nuScenes benchmark. The first row is the performance of the baseline method and the following values are the gains obtained by adding different modules.

In Tab. VI, the first row is the performance of the baseline method and the following values are the gains obtained by adding different modules. From the table, we can obviously find that all the modules give positive effects on all three categories for the final detection results. For all the categories, the proposed modules give 2.9 points improvements averagely

Strategies	**AP(%)**			**mAP(%)**
	Car(Mod.)	Ped.(Mod.)	Cyc.(Mod.)	
SECOND	88.99	56.21	70.65	71.95
One scale CA	89.68 (+0.69)	57.77 (+1.56)	71.39 (+0.74)	73.70 (+1.75)
Multi-scale CA	90.22 (+1.23)	58.11 (+1.90)	71.77 (+1.12)	74.41 (+2.46)

TABLE VIII: Ablation with/without multi-scale channel attention (CA) on KITTI dataset. The meaning of *one-scale CA* is we just use F_R^L to the *CA* module. And multi-scale denote that we use both fused F_R^L and F_R^S feature to the next module.

and "Pedestrian" achieves the most gain which achieves 3.45 points. Furthermore, we find that deep fusion gives the most improvements compared to the other three modules in this dataset.

Tab. VII gives the results on the nuScenes dataset. To be clear, a boosted version of baseline is presented here than in [11] by fixing the sync-bn issue in the source code. From this table, we can see that both the 2D and 3D semantic information can significantly boost the performances while the 2D information provide more improvements than the 3D information. This phenomenon can be explained as that the 2D texture information can highly benefit the categories classification results which is very important for the final **mAP** computation. In addition, by giving the 2D information, the recall ability can be improved especially for objects in long distance with a few scanned points. In other words, the advantage for 3D semantic information is that the point clouds can well handle the occlusion among objects which are very common in AD scenarios which is hard to deal with in 2D. After the fusion process, all the detectors achieve much better performances than only one semantic information (2D or 3D). The Fig. 6 (a) also illustrates the contribution of each module, we can also be observed that the detection results consistently improved by adding more modules gradually on *CenterPoint* baseline. More details can be found in Tabel.VII

Furthermore, a deep fusion module is also proposed to aggregate the backbone features to further improve the performance. From the Tab. Tab. VI and Tab. VII, we find that the deep fusion module can slightly improve the results for all three baseline detectors. Interestingly, compared to the *SECOND* and *PointPillars*, *CenterPoint* gives much better performance by adding the deep fusion module. This can be explained that the large deep backbone network in *CenterPoint* gives much deeper features that are more suitable for the proposed deep feature fusion module.

Effectiveness of Multi-scale CA. Additionally, a small experiment was devised to assess the efficacy of multi-scale attention in the Deep Feature Fusion (DFF) module. The baseline for this experiment was established using SECOND and tested on the KITTI dataset. The ablation experiment results presented in Tab. VIII indicate that leveraging the one-scale features led to a notable improvement of 1.75 points in mAP compared to the baseline. Furthermore, the introduction of the multi-scale operation resulted in an additional 0.71 points improvement.

D. Computation Time

Besides the performance, the computation time is also very important. Here, we test the computation time of each module based on the *PointPillars* and *SECOND* on the KITTI dataset in Tab. IX. For a single Nvidia V100 GPU, the *PointPillars* takes 53.5 ms per frame. By adding the 2D and 3D semantic information, the inference time increases 3.0 ms and 3.4 ms respectively while the time increases about 9 ms by adding two types semantic information and the AFF & DFF modules. The inference time for *SECOND* is given at the right column of Tab. IX. Compared with the baseline, the 2D/3D semantic segmentation information gives nearly no extra computational burden. We explain this phenomenon as that in *SECOND* the simple *mean* operation is employed for extracting the features for each voxel and the computation time of this operation will not change too much with the increasing of the feature dimension. For *PointPillars*, the MPL is employed for feature extracting in each pillar, therefore the computation time will increase largely with the increasing of the feature dimension.

Types	**PointPillars**	**SECOND**
Baseline	53.5 (ms)	108.2 (ms)
+3D Sem	+3.0 (ms)	+0.5 (ms)
+2D Sem	+3.4 (ms)	+0.2 (ms)
The Proposed	62.5 (ms)	113.3 (ms)

TABLE IX: Inference time of different modules. Here, *+3D Sem* and *+2D Sem* denote the time of adding 3D/2D semantic information to input data. *The Proposed* denotes the proposed framework with the modules including two types of semantic information and *AAF* & *DFF* modules.

In addition, we also record the time used for obtaining the 2D/3D semantic results. For Deeplab V3+, the inference time is about 32 ms per frame while for Cylinder3D, it takes about 140 ms per frame. Furthermore, the re-projection of 2D image to 3D point clouds also takes about 3 ms for each frame. The almost time consuming here are 2D/3D point clouds segmentation operations. But in the practical using there just need a extra segmentation head after detection network backbone. In other words, multi-head is needed here for both detection and segmentation task. These just taking few milliseconds when using model inference acceleration operation, like C++ inference library TensorRT.

E. Qualitative Detection Results

We show some qualitative detection results on nuScenes and KITTI dataset in Fig. 7 in which Fig. 7 (a) is the ground truth, (b), (c), and (d) are the detect result of baseline (CenterPoint) without any extra information, with 2D and 3D semantic information respectively and (e) is final results with all the fusion modules. From these figures, we can easily find that there is some false positive detection caused by the frustum blurring effect in 2D painting, while the 3D semantic results give a relatively clear boundary of the object but provides some worse class classification. More importantly, the proposed framework which combines both the two complementary information from 2D and 3D segmentation can give much more accurate detection results.

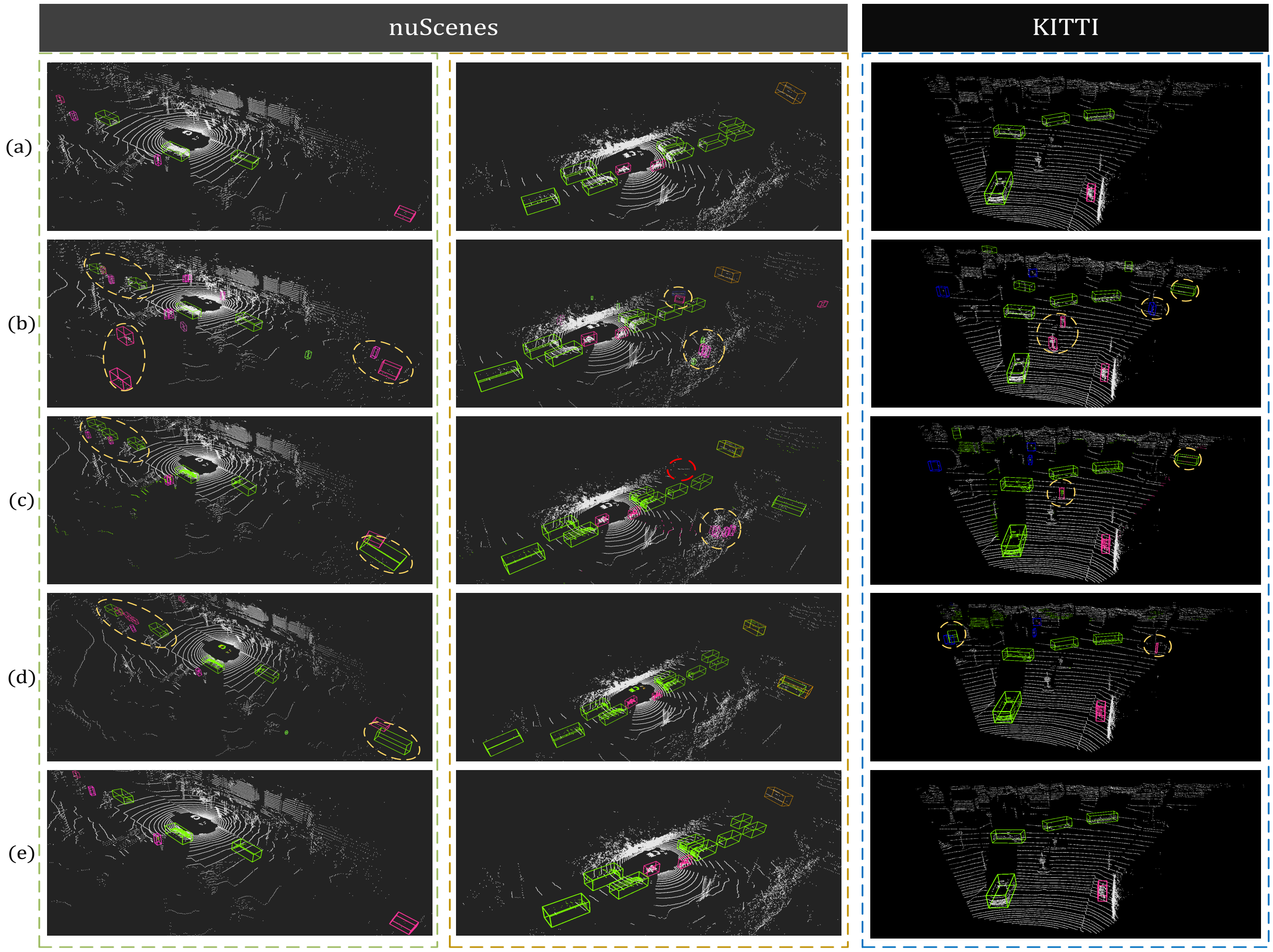

Fig. 7: Visualization of detection results with Open3d [81], where (a) is the ground-truth, (b) is the baseline method based on only point cloud, (c), (d) and (e) are the detection results based on the 2D semantic, 3D semantic and fusion semantic information, respectively. Especially, the yellow and red dash ellipses show some false positive and false negative detection. For nuScenes dataset, the baseline detector used here is CenterPoint, and for KITTI is SECOND.

V. Conclusion and Future Works

In this work, we proposed an effective framework *Multi-Sem Fusion* to fuse the RGB image and LiDAR point clouds in two different levels. For the first level, the proposed AAF module aggregates the semantic information from both the 2D image and 3D point clouds segmentation, resulting in adaptivity with learned weight scores. For the second level, a DFF module is proposed to fuse further the boosted feature maps with different receptive fields by channel attention module. Thus, the features can cover objects of different sizes. More importantly, the proposed modules are detector independent, which can be seamlessly employed in different frameworks. The effectiveness of the proposed framework has been evaluated on public benchmark and outperforms the state-of-the-art approaches. However, the limitation of the current framework is also obvious. Both the 2D and 3D parsing results are obtained by offline approaches, which prevent the application of our approach in real-time AD scenarios. An interesting research direction is sharing the backbone features for object detection and segmentation and taking the segmentation as an auxiliary task.

References

[1] Y. Wang, Q. Mao, H. Zhu, J. Deng, Y. Zhang, J. Ji, H. Li, and Y. Zhang, “Multi-modal 3d object detection in autonomous driving: a survey,” *International Journal of Computer Vision*, pp. 1–31, 2023. 1

[2] J. Dou, J. Xue, and J. Fang, “Seg-voxelnet for 3d vehicle detection from rgb and lidar data,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 4362–4368, IEEE, 2019. 1

[3] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, “Pointpainting: Sequential fusion for 3d object detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4604–4612, 2020. 1, 4, 8, 9, 10

[4] J. Yan, Y. Liu, J. Sun, F. Jia, S. Li, T. Wang, and X. Zhang, “Cross modal transformer: Towards fast and robust 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18268–18278, 2023. 1

[5] Y. Tian, K. Wang, Y. Wang, Y. Tian, Z. Wang, and F.-Y. Wang, “Adaptive and azimuth-aware fusion network of multimodal local features for 3d object detection,” *Neurocomputing*, vol. 411, pp. 32–44, 2020. 1

[6] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, “Transfusion: Robust lidar-camera fusion for 3d object detection with transformers,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1090–1099, 2022. 1, 3, 9

[7] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2774–2781, IEEE, 2023. 1, 3, 7, 8, 9

[8] Z. Song, C. Jia, L. Yang, H. Wei, and L. Liu, "Graphalign++: An accurate feature alignment by graph matching for multi-modal 3d object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 1

[9] Z. Song, H. Wei, L. Bai, L. Yang, and C. Jia, "Graphalign: Enhancing accurate feature alignment by graph matching for multi-modal 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3358–3369, 2023. 1

[10] S. Pang, D. Morris, and H. Radha, "Clocs: Camera-lidar object candidates fusion for 3d object detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020. 1

[11] S. Xu, D. Zhou, J. Fang, J. Yin, Z. Bin, and L. Zhang, "Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection," in *IEEE Intelligent Transportation Systems Conference*, 2021. 2, 11

[12] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, IEEE, 2018. 3

[13] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *IEEE conference on Computer Vision and Pattern Recognition*, pp. 3569–3577, 2018. 3

[14] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *IEEE conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, 2018. 3

[15] Z. Liang, Z. Zhang, M. Zhang, X. Zhao, and S. Pu, "Rangeioudet: Range image based real-time 3d object detector optimized by intersection over union," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7140–7149, 2021. 3

[16] P. Hu, J. Ziglar, D. Held, and D. Ramanan, "What you see is what you get: Exploiting visibility for 3d object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11001–11009, 2020. 3

[17] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018. 3, 7, 8, 9, 10

[18] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang, "Voxel-fpn: Multi-scale voxel feature aggregation for 3d object detection from lidar point clouds," *Sensors*, vol. 20, no. 3, p. 704, 2020. 3

[19] L. Wang, Z. Song, X. Zhang, C. Wang, G. Zhang, L. Zhu, J. Li, and H. Liu, "Sat-gcn: Self-attention graph convolutional network-based 3d object detection for autonomous driving," *Knowledge-Based Systems*, vol. 259, p. 110080, 2023. 3

[20] J. Yin, J. Shen, C. Guan, D. Zhou, and R. Yang, "Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11495–11504, 2020. 3

[21] D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, and R. Yang, "Iou loss for 2d/3d object detection," in *2019 International Conference on 3D Vision (3DV)*, pp. 85–94, IEEE, 2019. 3

[22] Z. Song, H. Wei, C. Jia, Y. Xia, X. Li, and C. Zhang, "Vp-net: Voxels as points for 3-d object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023. 3, 7

[23] H. Wu, J. Deng, C. Wen, X. Li, C. Wang, and J. Li, "Casa: A cascade attention network for 3-d object detection from lidar point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022. 3, 7, 8

[24] D. Zhou, J. Fang, X. Song, L. Liu, J. Yin, Y. Dai, H. Li, and R. Yang, "Joint 3d instance segmentation and object detection for autonomous driving," in *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1839–1849, 2020. 3

[25] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–779, 2019. 3, 7

[26] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 3, 8, 9

[27] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, 2019. 3, 7, 8, 9, 10

[28] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017. 3, 5

[29] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10529–10538, 2020. 3, 7, 8

[30] D. Zhang, Z. Zheng, H. Niu, X. Wang, and X. Liu, "Fully sparse transformer 3d detector for lidar point cloud," *IEEE Transactions on Geoscience and Remote Sensing*, 2023. 3

[31] D. Zhang, X. Wang, Z. Zheng, and X. Liu, "Unsupervised domain adaptive 3-d detection with data adaption from lidar point cloud," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022. 3

[32] A. Simonelli, S. Bulò, L. Porzi, M. Lopez-Antequera, and P. Kontschieder, "Disentangling monocular 3d object detection," *Cornell University - arXiv*, May 2019. 3

[33] T. Wang, X. Zhu, J. Pang, and D. Lin, "Fcos3d: Fully convolutional one-stage monocular 3d object detection," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Oct 2021. 3

[34] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 3

[35] Y. You, Y. Wang, W. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Weinberger, "Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving," *Cornell University - arXiv*, Jun 2019. 3

[36] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3d object detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021. 3

[37] X. Shi, Z. Chen, and T.-K. Kim, *Distance-Normalized Unified Representation for Monocular 3D Object Detection*, p. 91–107. Jan 2020. 3

[38] R. Chen, S. Han, J. Xu, and H. Su, "Point-based multi-view stereo network," 3

[39] Y. Chen, S. Liu, X. Shen, and J. Jia, "Dsgn: Deep stereo geometry network for 3d object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. 3

[40] L. Wang, X. Zhang, Z. Song, J. Bi, G. Zhang, H. Wei, L. Tang, L. Yang, J. Li, C. Jia, *et al.*, "Multi-modal 3d object detection in autonomous driving: A survey and taxonomy," *IEEE Transactions on Intelligent Vehicles*, 2023. 3

[41] Z. Song, G. Zhang, J. Xie, L. Liu, C. Jia, S. Xu, and Z. Wang, "Voxelnextfusion: A simple, unified, and effective voxel fusion framework for multimodal 3-d object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023. 3

[42] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918–927, 2018. 3

[43] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 244–253, 2018. 3

[44] X. Du, M. H. Ang, S. Karaman, and D. Rus, "A general pipeline for 3d detection of vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3194–3200, IEEE, 2018. 3

[45] H. Wu, C. Wen, S. Shi, X. Li, and C. Wang, "Virtual sparse convolution for multimodal 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21653–21662, 2023. 3, 7, 8, 9

[46] Q. Xia, Y. Chen, G. Cai, G. Chen, D. Xie, J. Su, and Z. Wang, "3-d hanet: A flexible 3-d heatmap auxiliary network for object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–13, 2023. 3, 7

[47] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3d object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7345–7353, 2019. 3

[48] R. Nabati and H. Qi, "Centerfusion: Center-based radar and camera fusion for 3d object detection," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1527–1536, 2021. 3

[49] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, and F. Zhao, "Autoalignv2: Deformable feature aggregation for dynamic multi-modal 3d object detection," *arXiv preprint arXiv:2207.10316*, 2022. 3

[50] Z. Yang, J. Chen, Z. Miao, W. Li, X. Zhu, and L. Zhang, "Deepinteraction: 3d object detection via modality interaction," *Advances in Neural Information Processing Systems*, vol. 35, pp. 1992–2005, 2022. 3

[51] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, and Z. Tang, "Bevfusion: A simple and robust lidar-camera fusion framework," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10421–10434, 2022. 3, 9

[52] Y. Li, Y. Chen, X. Qi, Z. Li, J. Sun, and J. Jia, "Unifying voxel-based representation with transformer for 3d object detection," 2022. 3, 9

[53] A. R. Choudhury, R. Vanguri, S. R. Jambawalikar, and P. Kumar, "Segmentation of brain tumors using deeplabv3+," in *International MICCAI Brainlesion Workshop*, pp. 154–167, Springer, 2018. 3

[54] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, *et al.*, "Hybrid task cascade for instance segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4974–4983, 2019. 3, 8

[55] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," in *The IEEE International Conference on Computer Vision (ICCV)*, pp. 1314–1324, October 2019. 3

[56] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh, "Polarnet: An improved grid representation for online lidar point clouds semantic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9601–9610, 2020. 4

[57] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, "2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network," *arXiv preprint arXiv:2102.04530*, 2021. 4

[58] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, "Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation," *arXiv preprint arXiv:2103.12978*, 2021. 4

[59] P. Cong, X. Zhu, and Y. Ma, "Input-output balanced framework for long-tailed lidar semantic segmentation," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2021. 4

[60] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3146–3154, 2019. 6

[61] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel r-cnn: Towards high performance voxel-based 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1201–1209, 2021. 7

[62] H. Yang, Z. Liu, X. Wu, W. Wang, W. Qian, X. He, and D. Cai, "Graph r-cnn: Towards accurate 3d object detection with semantic-decorated local graph," in *European Conference on Computer Vision*, pp. 662–679, Springer, 2022. 7

[63] H. Yang, W. Wang, M. Chen, B. Lin, T. He, H. Chen, X. He, and W. Ouyang, "Pvt-ssd: Single-stage 3d object detector with point-voxel transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13476–13487, 2023. 7

[64] J. H. Yoo, Y. Kim, J. Kim, and J. W. Choi, "3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pp. 720–736, Springer, 2020. 7

[65] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal sparse convolutional networks for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5428–5437, June 2022. 7, 9

[66] X. Wu, L. Peng, H. Yang, L. Xie, C. Huang, C. Deng, H. Liu, and D. Cai, "Sparse fuse dense: Towards high quality 3d detection with depth completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5418–5427, 2022. 7

[67] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, 2012. 7, 10

[68] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020. 7, 8, 9

[69] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017. 7

[70] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3d object detection," in *IEEE International Conference on Computer Vision*, pp. 1991–1999, 2019. 7, 8

[71] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018. 8

[72] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9939–9948, 2021. 8

[73] Y. Ma, D. Yu, T. Wu, and H. Wang, "Paddlepaddle: An open-source deep learning platform from industrial practice," *Frontiers of Data and Domputing*, vol. 1, no. 1, pp. 105–115, 2019. 8

[74] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, "Voxelnext: Fully sparse voxelnet for 3d object detection and tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21674–21683, June 2023. 9

[75] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," in *the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11040–11048, 2020. 9

[76] J. Koh, J. Lee, Y. Lee, J. Kim, and J. W. Choi, "Mgtanet: Encoding sequential lidar points using long short-term motion-guided temporal attention for 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 1179–1187, 2023. 9

[77] J. H. Yoo, Y. Kim, J. S. Kim, and J. W. Choi, "3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVII* (A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, eds.), vol. 12372 of *Lecture Notes in Computer Science*, pp. 720–736, Springer, 2020. 9

[78] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010. 8

[79] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. 9

[80] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," *arXiv preprint arXiv:2006.11275*, 2020. 10

[81] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018. 12

Shaoqing Xu received his M.S. degree in transportation engineering from the School of Transportation Science and Engineering in Beihang University. He is currently working toward the Ph.D. degree in electromechanical engineering with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao SAR, China. His research interests include intelligent transportation systems, Robotics and computer vision.

Fang Li received the B.S. degree from Harbin Institute of Technology, Weihai, China, in 2020. His research focuses on LiDAR-based 3D object detection during the graduate study and received M.S. degree in mechanical engineering in Beijing Institute of Technology, Beijing, China. His research interests include 3D Object Detection and applications in Autonomous Driving.

Ziying Song was born in Xingtai, Hebei Province, China, in 1997. He received his B.S. degree from Hebei Normal University of Science and Technology (China) in 2019. He received a master's degree from Hebei University of Science and Technology (China) in 2022. He is now a Ph.D. student majoring in Computer Science and Technology at Beijing Jiaotong University (China), with research focus on Computer Vision.

Jin Fang is currently a researcher with Inceptio. He obtained his B.E. degree from Huazhong University of Science and Technology, and received the M.E degree in Information Engineering from Peking University, in 2016. Currently, his research interests include LiDAR Simulation, 3D Object Detection, and applications in Autonomous Driving.

Sifen Wang received his M.S. degree in transportation engineering in 2020 from Beihang University. He is currently working toward the Ph.D. degree in the School of Transportation Science and Engineering, Beihang University, Beijing, China. His research interests include intelligent vehicle, deep reinforcement learning and computer vision.

Zhi-Xin Yang (Member, IEEE) received the B.Eng. degree in mechanical engineering from the Huazhong University of Science and Technology, and the Ph.D. degree in industrial engineering and engineering management from the Hong Kong University of Science and Technology, respectively. He is currently an Associate Professor with the University of Macau. His current research interests include robotics, machine vision, intelligent fault diagnosis and safety monitoring.