LIF-Seg: LiDAR and Camera Image Fusion for 3D LiDAR Semantic Segmentation

Lin Zhao, Hui Zhou, Xinge Zhu, Xiao Song, Hongsheng Li, Member, IEEE, and Wenbing Tao, Member, IEEE

Abstract—Camera and 3D LiDAR sensors have become indispensable devices in modern autonomous driving vehicles. Camera provides fine-grained texture and color information in 2D space, while LiDAR captures more precise and farther-away distance measurements of the surrounding environments. The complementary information from these two sensors makes the fusion of two modalities a desired option. However, two primary challenges in the fusion of camera and LiDAR hinder its performance, i.e., how to effectively fuse the information from these two modalities and how to precisely align them (suffering from the weak spatiotemporal synchronization problem). This article proposes a coarse-to-fine LiDAR and camera fusion-based network, named LIF-Seg, for LiDAR segmentation. For the first challenge, unlike these previous works fusing the point cloud and image information in a one-to-one manner, the proposed method introduces a simple but effective early-fusion strategy to fully utilize the contextual information of images. Second, to tackle the weak spatiotemporal synchronization problem, an offset rectification approach is designed to align the features of the two modalities. The cooperation of these two components leads to the success of the effective camera-LiDAR fusion. Experimental results on the nuScenes dataset show the superiority of LIF-Seg over existing methods by a large margin. Ablation studies and analyses further illustrate that the LIF-Seg can effectively address the weak spatiotemporal synchronization problem.

Index Terms—LiDAR and Camera, LiDAR segmentation, contextual information, weak spatiotemporal synchronization.

I. INTRODUCTION

With the rapid development of autonomous driving, 3D scene perception has received more and more attention in recent years, especially in computer vision and deep learning. LiDAR has become an indispensable 3D sensor in autonomous driving. Point clouds acquired by LiDAR, compared with data from other sensors (e.g., cameras and radars), can provide rich geometric and scale information, accurate distance measurements, and fine semantic descriptions, which are quite helpful in understanding 3D scenes for autonomous driving planning and execution.

LiDAR point cloud semantic segmentation aims to assign a special semantic label to each 3D point, which is an essential component of autonomous driving. LiDAR semantic segmentation enables the perception system to recognize and locate dynamic objects and drivable surfaces, making it a crucial task in autonomous driving. Although relatively mature solutions [1], [2] have been developed in 3D object detection to support real-world autonomous driving, it struggles with recognizing and locating drivable surfaces. In general, LiDAR point clouds are sparse, with sparsity increasing as the reflection distance increasing. This sparsity makes it difficult for the semantic segmentation model to accurately segment small objects in the distance, as illustrated in the left of Fig. 1.

As previously mentioned, LiDAR points can provide accurate distance measurements and capture object structures. However, they are often sparse, unordered, and unevenly distributed. Recent methods [4], [5], [6], [7] that rely solely on LiDAR point clouds have shown significantly improvements in 3D semantic segmentation. However, these methods are still limited due...
to the lack of dense and rich information on the objects, such as color and texture, as depicted in the right of Fig. 1. Compared with point clouds, camera images have more regular and dense pixels and contain richer semantic information (e.g., color and texture), enabling the distinction of different semantic categories. However, they lack depth and scale information. Therefore, the complementary information from LiDAR and camera makes the fusion of these two modalities a desired option. However, the challenge is how to effectively fuse these two modalities to leverage the strengths of both sensors such that more reliable and accurate semantic segmentation results can be generated.

Recently, the emergence of autonomous driving datasets containing both LiDAR point clouds and images, such as KITTI [8] and nuScenes [3], provides the possibility of combining the strengths of both modalities for improved semantic segmentation. These datasets have also played an important role in advancing point cloud semantic segmentation in academia and industry. However, as illustrated in Fig. 2, there exists a weak spatiotemporal synchronization problem between the LiDAR and cameras. When the LiDAR points are projected onto the corresponding camera image, the weak spatiotemporal synchronization problem causes some foreground points to fall on the background area and some background points to appear in the foreground area. Several factors contribute to this issue, including errors in sensor installation, calibration matrix, relative motion, and differences in capture frequency between sensors. Some strategies can alleviate this issue. For example, the KITTI and nuScenes realign the point clouds and images with time-stamped sensor metadata, but there remains a certain level of deviation. The weak spatiotemporal synchronization problem also limits the effectiveness of camera-LiDAR fusion.

Motivated by the above findings, we propose a novel coarse-to-fine framework, named LIF-Seg, for fusing the LiDAR and camera image information to improve the performance of LiDAR semantic segmentation. Firstly, unlike previous works fusing the point cloud and image information in a one-to-one manner, in the coarse stage, LiDAR points are projected onto each camera image, and the $3 \times 3$ contextual information of each pixel is concatenated to the intensity measurement of the LiDAR points. The concatenated LiDAR points are fed into a LiDAR semantic segmentation sub-network (e.g., Cylinder3D [5]) to obtain coarse LiDAR point features. To address the weak spatiotemporal synchronization problem, an offset rectification approach is designed to align the coarse features and image semantic features. Specifically, an image semantic segmentation sub-network (e.g., DeepLabv3+ [9]) is used to extract image semantic features. The coarse features are projected onto each image, and further fused with image semantic features to predict an offset between each projected point and its corresponding image semantic pixel. The predicted offset compensates and aligns two-modality features, and the aligned image semantic features are then fused with the coarse features. In the refinement stage, the fused features are fed into a sub-network to generate more accurate predictions. The LIF-Seg not only effectively fuses the LiDAR point features and image features at different levels, but also addresses the weak spatiotemporal synchronization problem between the LiDAR and cameras.

The main contributions of this work are as follows: (1) We fully utilize the raw image contextual information and introduce a simple but effective early-fusion strategy. (2) We propose an offset rectification method to address the weak spatiotemporal synchronization problem between LiDAR and cameras. (3) We construct a coarse-to-fine LiDAR and camera fusion-based network LIF-Seg for LiDAR semantic segmentation. Experimental results on the nuScenes dataset demonstrate the effectiveness of our method.

II. RELATED WORK

In this section, we will briefly review existing works related to our method: deep learning for 3D point clouds, LiDAR point cloud semantic segmentation, LiDAR and camera fusion methods, and image semantic segmentation. Especially, we mainly focus on the LiDAR-only and fusion-based methods.

A. Deep Learning for 3D Point Clouds

Different from 2D image processing methods, point cloud processing is a challenging task because of its irregular and unordered properties. PointNet [10] is one of the first works of directly learning the point features from raw point clouds through a shared Multi-Layer Perceptron (MLP) and max-pooling. Subsequent works [11], [12], [13], [14], [15], [16], [17], [18], [19] build upon these methods (e.g., PointNet, PointNet++) and further improve the semantic segmentation performance by promoting the effectiveness of sampling, grouping and ordering. Other methods [20], [21], [22] introduce graph networks to extract hierarchical point features. Although these methods have achieved promising segmentation results on indoor point clouds, most of them cannot be readily applicable to large-scale outdoor LiDAR point clouds due to variations in density and scene range. Moreover, a large number of points result in these methods requiring expensive computational and memory consumption when adapting to outdoor scenes.

B. LiDAR Point Cloud Semantic Segmentation

As the availability of public datasets [3], [23] increases, the research on LiDAR point cloud semantic segmentation is developing. Currently, these methods can be grouped into three main categories: projection-based, voxel-based and multi-view fusion-based methods.
Projection-based methods aim to convert 3D point clouds into a 2D image that is regular and dense, which can be processed using 2D CNNs. Approaches such as SqueezeSeg [24], SqueezeSegv2 [25], RangeNet++ [26], SalsaNext [27] and KPRNet [4] utilize the spherical projection mechanism to transform the point clouds into range image and adopt an encoder-decoder network to obtain semantic information. For instance, KPRNet [4] presents an improved architecture and achieves promising results by using a strong ResNeXt-101 backbone with an Atrous Spatial Pyramid Pooling (ASPP) block. Additionally, it also applies KPConv [28] as the segmentation head, replacing the inefficient KNN postprocessing. PolarNet [29], on the other hand, utilizes a polar Birds-Eye-View (BEV) instead of the cartesian grid-based BEV projections. However, these projection-based methods inevitably lose and alter the original topology, which may result in the failure of geometric information modeling. Voxel-based methods rasterize point clouds into voxels and then apply vanilla 2D or 3D convolutions to obtain segmentation results. More recently, some works [30], [31] are proposed to accelerate the 3D convolution and improve the performance with less computational and memory consumption. Following the previous works [30], [31], [32], JS3C-Net [33] and S3Net [34] achieve promising semantic segmentation results on outdoor scenarios. Among them, JS3C-Net [33] proposes a new LiDAR semantic segmentation network assisted by learned contextual shape priors. Furthermore, Cylinder3D [5] utilizes cylindrical partition and an asymmetrical residual block to further reduce computation.

Multi-view fusion-based methods aim to address the limitations of voxel-based and projection-based methods for LiDAR semantic segmentation. These methods combine various operations, such as voxel-based, projection-based and point-wise operations, to extract more semantic information. Recent methods [6], [7], [35], [36], [37], [38], [39], [40], [41] focus on blending two or more different views to achieve better segmentation results. For instance, [38], [39] combine point-wise information from BEV and range-image in the early stage and then feed it to the subsequent network. AMVNet [37] utilizes the uncertainty of different view outputs for late-fusion. PVCNN [35], FusionNet [40] and $(AF)^2$-S3Net [6] use a point-voxel fusion scheme to achieve better segmentation results. RPVNet [7] proposes a deep fusion network that fuses range-point-voxel three views by a gated fusion mechanism. However, the performance of these methods is also limited due to the LiDAR point clouds lacking rich color and texture information.

C. LiDAR and Camera Fusion Methods

Several methods [42], [43], [44], [45], [46], [47], [48], [49], [50] have been proposed for the fusion of camera and LiDAR to leverage their respective strengths, especially in the 3D object detection task. One example is PI-RCNN [47], which fuses camera and LiDAR features by applying point-wise convolution on 3D points and point-pooling with an aggregation operation. Another method, CLOCs [48], operates on the combined output candidates before non-maximum suppression of any 2D and any 3D detector. 3D-CVF [49] uses a cross-view spatial feature fusion strategy to combine camera and LiDAR features for better detection performance. EPNet [50] proposes a LiDAR-guided Image Fusion module to enhance the LiDAR point features with corresponding image semantic features at multiple scales. PointPainting [46] projects LiDAR points into the output of an image-only semantic segmentation network, appends class scores to each point, and feeds the resulting data to a LiDAR detector. Although these methods have achieved promising performance in 3D object detection, few previous works have focused on 3D semantic segmentation by combining the advantages of camera and LiDAR sensors or addressing the weak spatiotemporal synchronization problem between them.

D. Image Semantic Segmentation

Image semantic segmentation is a fundamental task in computer vision that has made significant strides in recent years. FCN [51] is the pioneering work that directly employs fully convolutional layers to generate image semantic segmentation results. The DeepLab [9] family of methods utilizes atrous convolution and ASPP modules to capture the contextual information in the image. While STDC2 [52] reduces the inference time-consuming by using a detailed guidance module to encode low-level spatial information, its performance is relatively low. To balance the efficiency and performance, we adopt the DeepLabv3+ [9] as the image segmentation submodel in this work.

III. PROPOSED METHOD

Exploiting the complementary advantages of LiDAR and camera sensors is very important for achieving accurate LiDAR semantic segmentation. However, existing fusion methods often fail to fully utilizing the rich contextual information from camera images and addressing the weak spatiotemporal synchronization problem between the LiDAR and cameras, which limits the ability of the fusion model to recognize fine-grained patterns. In this article, we propose LIF-Seg, a coarse-to-fine framework that addresses these limitations and improves the performance of LiDAR segmentation from two aspects, including raw image contextual information fusion in early-fusion and aligned high-level image semantic information fusion in mid-fusion. The LIF-Seg takes both LiDAR points and camera images as input, and predicts the semantic label of each point. It consists of three main stages: coarse feature extraction stage, offset learning stage, and refinement stage. We provide detailed descriptions of each stage in the following subsections.

A. Coarse Feature Extraction Stage

LiDAR points can provide accurate distance measurements and capture the structural information of objects, whereas camera images contain more regular and dense pixels and richer semantic information. Some methods [46], [47], [48] attempt to blend LiDAR and camera views together in different stages (e.g., early-fusion, mid-fusion, and late-fusion) for 3D object detection. Most of these methods only fuse the raw or high-level
Algorithm 1: LIF-Seg($L, I, T, K$).

Input:
LiDAR points $L \in \mathbb{R}^{N \times D}$ with $N$ points and $D \geq 3$.
Images $I = \{I_i\} = \{1, 2, \ldots, n\}$ with $n$ cameras.
Transformation matrices $T = \{T_i\} = \{1, 2, \ldots, n\}$.
Camera matrices $K = \{K_i\} = \{1, 2, \ldots, n\}$.

Output:
Segmentation scores $S \in \mathbb{R}^{N \times C}$ with $C$ classes.

1: # Coarse Feature Extraction Stage
2: Let $Idx = \text{List}()$, $Mask = \text{List}()$
3: Let $P = \text{Zeros}([N, 3 \times w^2])$ with $w \times w$ context.
4: for $i = 1$ to $n$ do
5: \hspace{1em} $[p_{ix}, p_{iy}, 1]^T = \text{PROJECT}(K_i, T_i, L^h_{xyz})$
6: \hspace{1em} $idx = \text{Stack}([p_{ix}, p_{iy}, \text{axis} = 1], \# idx \in \mathbb{R}^{N \times 2}$
7: \hspace{1em} $mask = (0 < idx[:, 0] < H \text{ and } 0 < idx[:, 1] < W)$
8: \hspace{1em} $idx = \text{idx[mask, :]}$, $\# idx \in \mathbb{R}^{N_i \times 2}$, $N_i \leq N$
9: \hspace{1em} $p = \text{Context}(I_i, idx, w)$ \hspace{1em} $p \in \mathbb{R}^{N_i \times w \times w \times 3}$
10: \hspace{1em} $P[mask, :] = \text{Reshape}(p, [N_i, 3 \times w^2])$
11: \hspace{1em} $Idx.append(idx)$, $Mask.append(mask)$
end for
12: $L' = \text{Concatenate}([L, P], \text{axis} = 1)$
13: $F_{\text{coarse}} = \text{LiDAR.Seg}(L')$, $\# F_{\text{coarse}} \in \mathbb{R}^{N \times C_0}$
14: # Offset Learning Stage
15: $F_{\text{image}} = \text{Seg.Net}(I)$ \hspace{1em} $\# F_{\text{image}} \in \mathbb{R}^{N \times H \times W \times C_1}$
16: \hspace{1em} $F_{\text{points}} = \text{Zeros}([N, H, W, C_0])$
17: \hspace{1em} for $i = 1$ to $n$ do
18: \hspace{2em} $idx = \text{Idx}[i]$, $mask = \text{Mask}[i]$
19: \hspace{2em} $F_{\text{points}}[i, \text{idx}[:, 0], \text{idx}[:, 1], :] = F_{\text{coarse}}[\text{mask, :}]$
20: end for
21: $F_{\text{offset}} = \text{Concatenate}([F_{\text{image}}, F_{\text{points}}], \text{axis} = 3)$
22: $F_{\text{offset}} = \text{Convs}(F_{\text{offset}})$ \hspace{1em} $\# F_{\text{offset}} \in \mathbb{R}^{N \times H \times W \times 2}$
23: \hspace{1em} $L' = \text{Zeros}([N, C_1])$
24: \hspace{1em} $O = \text{Zeros}([N, 2])$ \hspace{1em} # Point-wise offset
25: \hspace{1em} for $i = 1$ to $n$ do
26: \hspace{2em} $idx = \text{Idx}[i]$, $mask = \text{Mask}[i]$
27: \hspace{2em} $o = \text{Offset}(i, \text{idx}[:, 0], \text{idx}[:, 1])$
28: \hspace{2em} $O[mask, :] = o$
29: \hspace{2em} $idx = \text{idx} + o$ \hspace{1em} # Updating the index of points
30: \hspace{2em} $F_{\text{image}}[\text{mask, :}] = F_{\text{image}}[i, \text{idx}[:, 0], \text{idx}[:, 1], :]$\hspace{1em} end for
31: # Refinement Stage
32: $F = \text{Concatenate}([F_{\text{coarse}}, F_{\text{image}}], \text{axis} = 1)$
33: $S = \text{LiDAR.Seg}(F)$, $\# F \in \mathbb{R}^{N \times (C_0 + C_1)}$, $S \in \mathbb{R}^{N \times C}$

end for

image information in a one-to-one manner. However, the contextual information of the image is also important when fusing the views from LiDAR and the camera. In the coarse stage, we fuse the LiDAR points and raw image contextual information to obtain the coarse features.

As depicted in Fig. 3 and outlined in Algorithm 1, each point in LiDAR points $L$ has a spatial location $(x, y, z)$ and reflectance $r$. The LiDAR points are transformed onto each camera image by a homogenous transformation and a projection. This process can be formulated as follows:

$$[p_{ix}, p_{iy}, 1]^T = K_i T_i L^h_{xyz},$$

where $K_i$ and $T_i$ denote the intrinsic matrix and homogenous transformation matrix associated with the camera image $I_i$, respectively. $L^h_{xyz}$ represents homogenous coordinates of LiDAR points $L$, $p_{ix} \in \mathbb{R}^N$ and $p_{iy} \in \mathbb{R}^N$ represent the indexes (pixel coordinates) of LiDAR points $L$ on camera image $I_i$, with $N$ representing the total number of LiDAR points. The general transformation is given by $T_{\text{camera-lidar}}$. For nuScenes dataset, the complete transformation for each camera is:

$$T_i = T_{\text{camera-ego}} T_{\text{ego-cam}} T_{\text{ego-ego}}^T T_{\text{ego-lidar}},$$

with transforms: LiDAR frame to the ego-vehicle frame for the timestamp of the sweep $T_{\text{ego-vehicle}}$; ego frame to the global frame $T_{\text{ego-ego}}$; global frame to the ego-vehicle frame for the timestamp of the image $T_{\text{ego-ego}}$; and ego frame to the camera $T_{\text{camera-ego}}$. Once the LiDAR points have been transformed to the camera coordinate, the corresponding camera matrix $K_i$ is used to project the points onto the image $I_i$. Afterwards, the $w \times w$ (e.g., $3 \times 3$) image context information for each projected point position is sampled, reshaped and concatenated with the corresponding LiDAR point. The resulting concatenated points are then fed into a LiDAR semantic segmentation sub-network (e.g., Cylinder3D [5]) to obtain the coarse features $F_{\text{coarse}}$.

B. Offset Learning Stage

Although having achieved promising results in benchmark datasets, the early-fusion and mid-fusion methods suffer from performance limitations caused by the weak spatiotemporal synchronization problem between the LiDAR and cameras. To address the problem mentioned above, our proposed LIF-Seg predicts an offset between the projected LiDAR point and the corresponding pixel. The predicted offset is used to compensate and update the position of projected point features. Following this, the aligned image semantic features are fused with the coarse features to enhance segmentation performance.

In this stage, as illustrated in Fig. 4 and outlined in Algorithm 1, we first utilize an image semantic segmentation sub-network to obtain high-level image semantic features $F_{\text{image}}$ (without

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
The former maximizes the point accuracy, while the latter maximizes the intersection-over-union score. For the offset prediction in the subsection III-B, taking the nuScenes [3] dataset as an example, there is no directly available supervision information for the offset learning since the corresponding camera images lack pixel-level semantic or instance annotation. Therefore, an auxiliary loss $L_{aux}$ is employed to supervise offset learning. Specifically, for points belonging to the foreground categories, their point-wise offset $O \in \mathbb{R}^{N \times 2}$ are constrained by an $L_1$ regression loss $L_{reg}$:

$$L_{reg} = \frac{1}{N} \sum_{i} \sum_{j} ||o_i - (\hat{c}_i - p_i)|| \cdot m_i,$$  

where $m = \{m_1, \ldots, m_N\}$ is a binary mask. $m_i = 1$ if point $p_i$ is in a 2D bounding box on image plane and $m_i = 0$ otherwise. $\hat{c}_i$ is the centroid of the 2D bounding box to which point $p_i$ belongs. Thus, the $\hat{c}_i$ can be formulated as follows:

$$\hat{c}_i = \frac{1}{N_{g(i)}} \sum_{j \in B_{g(i)}} p_j,$$

where $g(i)$ is the index of the 2D bounding box that contains point $p_i$, $N_{g(i)}$ is the number of points in 2D bounding box $B_{g(i)}$. To ensure that these points move towards their corresponding centroid in the horizontal direction, we utilize a direction loss $L_{dir}$ to constrain the direction of predicted point-wise offset $O$. Following [54], the $L_{dir}$ is formulated as the average of minus cosine similarities:

$$L_{dir} = -\frac{1}{N} \sum_{i} \sum_{j} o_i \cdot \frac{\hat{c}_i - p_i}{\|\hat{c}_i - p_i\|_2} \cdot m_i.$$

Thus, the auxiliary loss can be formulated as $L_{aux} = L_{reg} + L_{dir}$. The training objective of our network is:

$$L = L_{sem} + \alpha L_{aux},$$

where $\alpha$ is the weight of auxiliary segmentation loss and set to 0.01 in our experiments.

IV. EXPERIMENTS

This section presents an evaluation of the proposed LIF-Seg on the nuScenes [3] dataset, aiming to demonstrate its effectiveness. In the following, subsection IV-A provides a brief introduction to the dataset and evaluation metric. Subsection IV-B details the implementation of our approach, while subsection IV-C exhibits the detailed experiment results for LiDAR-camera fusion and comparisons with state-of-the-art methods on the nuScenes dataset. Subsection IV-D provides ablation studies to validate the effectiveness of offset learning. Finally, the model complexity is presented in subsection IV-E.

A. Dataset and Evaluation Metric

The newly released nuScenes [3] dataset is a large-scale multi-modal dataset for LiDAR semantic segmentation, with more than 1000 scenes collected from different areas of Boston and Singapore. The scenes are split into 28,130 training frames, 6,019 validation frames, and 6,008 testing frames. The annotated dataset provides up to 32 classes. After merging similar
classes and removing rare classes, only 16 classes are retained for the LiDAR semantic segmentation. The dataset is collected by using a Velodyne HDL-32E sensor, cameras and radars with complete 360° coverage. In this work, we use the LiDAR point clouds and RGB images from all 6 cameras. Furthermore, this dataset has an imbalance challenge in different categories. In particular, classes like cars and pedestrians are most frequent, while bicycles and construction vehicles have relatively limited training data. Additionally, the nuScenes dataset poses further challenges as it is collected from different locations and diverse weather conditions. Furthermore, the nuScenes point clouds are sparser compared to other datasets since the sensor has fewer beams and lower horizontal angular resolution.

To evaluate the LiDAR semantic segmentation performance of our proposed approach, the mean intersection-over-union (mIoU) [55] over all classes is taken as the evaluation metric. The mIoU can be formulated as:

\[
mIoU = \frac{1}{C} \sum_{i=1}^{C} IoU_i, \tag{7}
\]

\[
IoU_i = \frac{p_{ii}}{p_{ii} + \sum_{j \neq i} p_{ij} + \sum_{k \neq i} p_{ki}}, \tag{8}
\]

where \( C \) is the number of classes, and \( p_{ij} \) denotes the number of points from class \( i \) predicted as class \( j \).

B. Implementation Details

**Image Semantic Network Details:** The image semantic segmentation sub-network used in this work is DeepLabV3+ [9], which takes all camera images as input and generates full-resolution semantic features \( F_{image} \in \mathbb{R}^{n \times H \times W \times C_1} \) (without normalization using softmax or sigmoid), where \( n = 6 \) is the number of cameras and \( C_1 = 16 \) is the dimension of features. However, due to the absence of any public segmentation pretrain model on the nuScenes dataset, we train the DeepLabV3+ using the nuImages dataset. The nuImages dataset consists of 100 \( k \) images annotated with semantic segmentation labels, and all classes of nuImages dataset are part of nuScenes dataset. It should be noted that the images of nuImages are hardly present in the image set corresponding to LiDAR point clouds of the nuScenes samples.

**LiDAR Network Details:** The LiDAR point clouds segmentation sub-network in coarse and refinement stages employs Cylinder3D [5] as its architecture. For the nuScenes dataset, we partition the LiDAR point clouds cylindrically to generate a 3D representation with the size 480 × 360 × 32, where three dimensions indicate the radius, angle and height, respectively. The resolutions in these dimensions are set to 0.1, 1 and 0.25 respectively. Besides, the feature dimension \( C_0 \) of coarse features \( F_{coarse} \) is set to \( C_0 = C \), where \( C \) is the number of semantic categories. Finally, the window size \( w \) of image contextual information is set to 3.

C. Performance Results and Analyses

In this sub-section, we first conduct extensive experiments on the validation set of the nuScenes [3] dataset to validate the effectiveness of different LiDAR-camera fusion strategies, including early-fusion of LiDAR information with different contextual information from the camera image, as well as mid-fusion of the LiDAR point features with image semantic features. Afterwards, we exhibit the comparisons with state-of-the-art methods on the nuScenes dataset. For all experiments, we adopt the retrained DeepLabV3+ [9] to extract image features and the Cylinder3D [5] to take as the LiDAR segmentation baseline. For a fairer and clearer comparison, we retrain the baseline network Cylinder3D using the code released by the author on GitHub. If there are no extra notes, we use the same fusion strategy to fuse LiDAR and camera image information for all models.

**Early-fusion and Mid-fusion:** For the early-fusion, LiDAR points are projected onto camera images by using transformation matrices and camera matrices. According to the position of projected points, we can query the contextual information of the image using a window size \( w \times w \), such as \( 1 \times 1, 3 \times 3 \) and \( 5 \times 5 \). The \( w \times w \) contextual information is reshaped to a vector and concatenated to the corresponding LiDAR point. The concatenated points are fed into the baseline network Cylinder3D to obtain the segmentation results, and the models of different contextual information fusion are denoted as \( C+1 \times 1, C+3 \times 3 \) and \( C+5 \times 5 \), respectively. Additionally, image semantic features obtained by DeepLabV3+ are appended to each LiDAR point to enhance the point features (denoted as \( C+\text{Sem}. \)). Moreover, we also fuse the \( 3 \times 3 \) image contextual information and image semantic features in early-fusion (denoted as \( C+3 \times 3+\text{Sem}. \)). Based on the fusion setup \( C+3 \times 3+\text{Sem}. \), the output features are further processed by using a refinement sub-network (denoted as \( C+3 \times 3+\text{Sem.}+\text{Ref}. \)). For the mid-fusion, image semantic features are concatenated with LiDAR point features obtained by the baseline network (denoted as \( C+\text{Mid}. \)). The fused features are applied to two convolutional layers to generate segmentation results. Besides, we also fuse the \( 3 \times 3 \) image contextual information in the early-stage based on the mid-fusion method \( C+\text{Mid.} \) (denoted as \( C+3 \times 3+\text{Mid.} \)). Finally, Cylinder3D is also taken as a refinement sub-network to replace the two convolutional layers in \( C+3 \times 3+\text{Mid.} \) (denoted as \( C+3 \times 3+\text{Mid.}+\text{Ref}. \)).

The LiDAR semantic segmentation results of different LiDAR-camera fusion strategies are reported in Table 1. Compared with the baseline method Cylinder3D and the \( C+1 \times 1 \), it can be observed that the direct fusion of the LiDAR and image information can improve the performance of LiDAR semantic segmentation. Compared with the early-fusion methods \( C+1 \times 1, C+3 \times 3 \) and \( C+5 \times 5 \), the setup \( C+3 \times 3 \) achieves the best mIoU score because of the fusion of image contextual information. The \( C+1 \times 1 \) fusion setup, lacking contextual information, exhibits limited recognition capacity for fine-grained patterns. The \( C+5 \times 5 \) fusion setup, with a context window size that is too large, contains redundant information, which impairs recognition of the semantic category of the central point.

1[Online]. Available: https://github.com/VainF/DeepLabV3Plus-Pytorch
2[Online]. Available: https://www.nuscenes.org/images
3[Online]. Available: https://github.com/xinge008/Cylinder3D
Similar to the 3D detector PointPainting [46], the early-fusion method C+Sem. can also improve the performance of LiDAR segmentation. Besides, the C+3×3+Sem. indicates fusing the LiDAR points, image contextual information and semantic features can effectively improve the performance of semantic segmentation. Comparing the early-fusion setup C+3×3+Sem. and C+3×3+Sem.+Ref., the gain achieved by the refinement sub-network is smaller. This is because C+3×3+Sem. belongs to the early-fusion, and its output features have been processed by a segmentation sub-network. Therefore, its output features do not need to be refined by using a refinement sub-network. The fusion methods C+Mid. and C+3×3+Mid. are only slightly better than the baseline because of lacking the well-designed mid-fusion module. The experiment results of C+3×3+Mid.+Ref. indicates that a well-designed mid-fusion module can effectively improve the segmentation performance. These experimental results show that the image context information and image semantic features are beneficial to LiDAR segmentation. In this work, LiDAR points and image contextual information are fused in the coarse stage, while point features and aligned image semantic features are fused in the refinement stage.

**Comparison with the SOTA Methods:** Following [5], we conduct experiments on the nuScenes [3] dataset to evaluate the effectiveness of our method. Table II presents the LiDAR semantic segmentation results on the nuScenes validation set. RangeNet++ [26] and Salsanext [27] perform the post-processing. From Table II, we can see that our proposed method achieves better performance than other methods and shows dominance in many categories. Specifically, the proposed method outperforms Cylinder3D [5] by 2.1 mIoU. Moreover, compared with the state-of-the-art projection-based methods (e.g., RangeNet++ and Salsanext), the LIF-Seg achieves about 6% ~ 12% performance gain. Note that the points of nuScenes are very sparse (35 k points/frame), especially for objects such as bicycles, motorcycles, traffic-cones, and pedestrians, etc. Therefore, the LiDAR segmentation task is more challenging. From Table II, we can see that our method significantly outperforms other approaches in these sparse categories, because the LIF-Seg effectively fuses the LiDAR points, the camera image contextual information and image semantic features through a coarse-to-fine framework. Qualitative results of LiDAR segmentation are presented in Fig. 6. Besides, we also submit our segmentation results to the nuScenes evaluation server. Table III reports the performance of our LIF-Seg and other state-of-the-art methods on the testing set of nuScenes [3] dataset. Compared to PolarNet [29], our LIF-Seg achieves better segmentation results. Compared to the concurrent methods AMVNet [37] and (AF)²-S3Net [6], our method also achieves comparable results. These results demonstrate the effectiveness of our proposed method.

### D. Ablation Studies

In this sub-section, we conduct ablation experiments on the validation set of the nuScenes [3] dataset to validate the effectiveness of offset learning. For a fairer and clearer comparison,
Fig. 6. Comparison results of Cylinder3D and our method in LiDAR semantic segmentation tasks on nuScenes dataset validation set. Best viewed in color.

| Methods           | mIoU |
|-------------------|------|
| PolarNet* [29]    | 69.4 |
| AMVNet [37]       | 76.1 |
| \( (AF)^2 \) S3Net [6] | 78.3 |
| LIF-Seg (Ours)    | 77.4 |

* indicates that its results are reproduced by avmnet [37]. Best and second best results are bolded and underlined.
we use the same configuration and sequential fusion strategy for all models, unless otherwise noted. Detailed ablation results are presented in Table IV. The offset learning stage is removed from the full pipeline of LIF-Seg, resulting in a decrease in the LiDAR segmentation performance from 78.2 to 77.6 mIoU. The offset prediction results are visually depicted in Fig. 7. From Fig. 7, we can see that the projected points move towards their corresponding centroid in the horizontal direction, which makes these points fall on the instance object as much as possible. In Fig. 8, we also visualize the background points that are near to object when projecting them onto the camera image. From Fig. 8, we can see that the background points were not wrongly moved towards object centroids, because the auxiliary loss is only valid for foreground points. These results demonstrate the effectiveness of our method.

### E. Model Complexity

In this work, our proposed LIF-Seg is primarily comprised of two components: image segmentation and LiDAR segmentation. Therefore, the model size, GPU memory and inference time consumption of LIF-Seg are primarily determined by these two components. Specifically, the model size of LIF-Seg is 651 MB, which contains 225 MB for image segmentation and 428 MB for LiDAR segmentation. As for the inference time, it is calculated as the average iteration time in parallel. The inference time of LIF-Seg is 0.37 s, which contains 0.13 s for image segmentation. The total GPU memory consumption for LIF-Seg is 18 GB. Both the inference time and GPU memory consumption are measured on the V100 GPU.

### V. Conclusion

In this article, we propose a coarse-to-fine framework, termed as LIF-Seg, to improve the 3D semantic segmentation performance from two aspects, including raw image contextual information fusion in the early-stage, and aligned high-level image semantic information fusion by tackling the weak spatiotemporal synchronization problem between the LiDAR and camera. The LIF-Seg consists of three main stages: coarse stage, offset learning stage and refinement stage. In the coarse stage, the LiDAR points and raw image contextual information are fused and fed into a LiDAR semantic segmentation sub-network to generate coarse features. Subsequently, the coarse features and image semantic features obtained by an image segmentation sub-network are fused to predict an offset between each projected LiDAR point and image pixel. The predicted offsets are used to align the coarse features and image semantic features. In the refinement stage, the coarse features and aligned image semantic features are fused and fed into a LiDAR semantic segmentation sub-network to obtain more accurate semantic segmentation results. Extensive experimental results on the nuScenes dataset demonstrate the effectiveness of our method. In the future, unsupervised learning methods can be incorporated into the proposed framework to predict a transformation matrix between the LiDAR and camera to address the weak spatiotemporal synchronization problem completely and further improve the performance of LiDAR segmentation.

![Table IV](image)

| Method                  | mIoU |
|-------------------------|------|
| Remove the offset learning stage | 77.6 |
| LIF-Seg (Ours)          | 78.2 |

---

**Fig. 7.** Example scenes from the nuScenes [3] dataset. The red point is the position of the original LiDAR point projected onto the camera image, and the cyan point is the updated position by using the predicted offset.

**Fig. 8.** Example scenes from the nuScenes [3] dataset. In first and third rows, the red point is the position of the original LiDAR point projected onto the camera image. In second and fourth rows, the cyan point is the updated projected position by using the predicted offset.

---

**References**

[1] A. H. Lang et al., “Pointpillars: Fast encoders for object detection from point clouds,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 12697–12705.

[2] S. Shi et al., “PV-RCNN: Point-voxel feature set abstraction for 3D object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10529–10538.

[3] H. Caesar et al., “nuScenes: A multimodal dataset for autonomous driving,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 11621–11631.

[4] D. Kochanov, F. K. Nejadi, and O. Booij, “KPRNet: Improving projection-based LiDAR semantic segmentation,” in Proc. Eur. Conf. Comput. Vis. Workshops, Aug. 2020, pp. 1–4.

[5] Z. Xinge et al., “Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 9939–9948.

[6] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, “AFP2-S3Net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 12542–12551.

[7] J. Xu et al., “RPVNet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 16004–16013.
A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2012, pp. 3354–3361.

L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 801–818.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3-D classification and segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 652–660.

C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet: Deep hierarchical feature learning on point sets in a metric space,” in Adv. Neural Inf. Process. Syst., 2017, pp. 505–514.

W. Wu, Z. Qi, and L. Fuxin, “PointConv: Deep convolutional networks on 3D point clouds,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9613–9622.

L. Zhao and W. Tao, “JSNet: Joint instance and semantic segmentation of 3D point clouds,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 12951–12958.

H. Liu, Y. Guo, Y. Ma, Y. Lei, and G. Wen, “Semantic context encoding for accurate 3D point cloud segmentation,” IEEE Trans. Multimedia, vol. 23, pp. 1–1, 2020.

C. Chen, S. Qian, Q. Fang, and C. Xu, “HAPGN: Hierarchical attentive pooling network for point cloud segmentation,” IEEE Trans. Multimedia, vol. 22, pp. 2045–2055, 2021.

M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, “PointHop: An explainable machine learning method for point cloud classification,” IEEE Trans. Multimedia, vol. 22, pp. 1744–1755, Jul. 2020.

J. Zhang, C. Zhu, L. Zheng, and K. Xu, “Fusion-aware point convolution for online semantic 3D scene segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 4534–4543.

S. Qiu, S. Anwar, and N. Barnes, “Geometric back-projection network for point cloud classification,” IEEE Trans. Multimedia, vol. 24, pp. 1943–1955, 2022.

L. Zhao and W. Tao, “JSNet++: Dynamic filters and pointwise correlation for 3D point cloud instance and semantic segmentation,” IEEE Trans. Circuits Systems for Video Technol., vol. 33, no. 4, pp. 1854–1867, Apr. 2023.

L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, “Graph attention convolution for point cloud semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 10296–10305.

Y. Wang et al., “Dynamic graph CNN for learning on point clouds,” ACM Trans. On Graph., vol. 38, no. 5, p. 1–12, 2019.

Z. Fu and W. Hu, “Dynamic point cloud inpainting via spatial-temporal graph learning,” IEEE Trans. Multimedia, vol. 23, pp. 3022–3034, 2021.

J. Behley et al., “SemanticKITTI: A dataset for semantic scene understanding of lidar sequences,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 9297–9307.

B. Wu, A. Wan, X. Yue, and K. Keutzer, “SqueezeSeg: Convolutional neural networks with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud,” in Proc. IEEE Int. Conf. Robot. Autom., 2018, pp. 1887–1893.

B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, “SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud,” in Proc. Int. Conf. Robot. Autom., 2019.

A. Milioti, I. Vizzo, J. Behley, and C. Stachniss, “RangeNet++: Fast and accurate LiDAR semantic segmentation,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2019, pp. 4213–4220.

T. Cortinhal, G. Tzelepis, and E. F. Aksoy, “Salsanext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds,” in Proc. Adv. Vis. Comput.: 15th Int. Symp., Oct. 2020, pp. 207–222.

H. Thomas et al., “kpConv: Flexible and deformable convolution for point clouds,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 6411–6420.

Y. Zhang et al., “PolarNet: An improved grid representation for online lidar point clouds semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 9601–9610.

B. Graham, M. Engelcke, and L. van der Maaten, “3D semantic segmentation with submanifold sparse convolutional networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 9224–9232.

H. Tang et al., “Searching efficient 3D architectures with sparse point-voxel convolution,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 685–702.

L. Zhao, S. Xu, L. Liu, D. Ming, and W. Tao, “SVASeg: Sparse voxel-based attention for 3D LiDAR point cloud semantic segmentation,” Remote Sens., vol. 14, no. 1082, 2022.

X. Yan et al., “Sparse single sweep LiDAR point cloud segmentation via learning contextual shape priors from scene completion,” in Proc. AAAI Conf. Artif. Intell., 2021, pp. 3101–3109.
Hui Zhou received the bachelor’s and master’s degrees from the University of Science and Electronic Technology of China, Chengdu, China, in 2015 and 2018, respectively. He is currently a Research Scientist for autonomous driving in SenseTime Research. His research interests include computer vision and machine learning.

Xinge Zhu received the B.Eng. degree in computer science from Shandong University, Jinan, China, in 2015. He is currently working toward the Ph.D. degree with The Chinese University of Hong Kong, Hong Kong, under the supervision of Professor Dahua Lin. His interest focuses on the computer vision and machine learning. Particularly, he is interested in 3D perception for autonomous vehicles, including 3D detection and 3D segmentation.

Xiao Song received the B.Eng. degree from the Department of Automation, Southeast University, Nanjing, China in 2016, and the M.Eng. degree from the Department of Automation, Shanghai Jiao Tong University, Shanghai, China in 2019. He is currently a Researcher of leading LiDAR object detection research and development with the Autonomous Driving Group, SenseTime. His research interests include computer vision, deep learning, stereo matching, and 3D object detection.

Hongsheng Li (Member, IEEE) received the bachelor’s degree in automation from the East China University of Science and Technology, Shanghai, China, in 2006, and the Ph.D. degree in computer science from Lehigh University, Bethlehem, PA, USA, in 2012. He is currently an Assistant Professor with the Department of Electronic Engineering, Chinese University of Hong Kong, Hong Kong. From 2013 to 2015, he was an Associate Professor with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China. He has authored or co-authored more than 70 papers in premier conferences on computer vision and machine learning, including CVPR, ICCV, ECCV, NeurIPS, ICLR, and AAAI. He was the recipient of 2020 IEEE CAS Society Outstanding Young Author Award. He won the first place in Object Detection from Videos (VID) track of ImageNet challenge 2016 as the team leader and 2015 as a team co-leader. He is an Associate Editor for the *Neurocomputing* and the Area Chair of NeurIPS 2021. His research interests include computer vision, machine learning, and medical image analysis.

Wenbing Tao (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2004. From 2008 to 2009, he was a Research Fellow with the Division of Mathematical Sciences, Nanyang Technology University, Singapore. He is currently a Full Professor with the School of Artificial Intelligence and Automation, HUST. He has authored or co-authored numerous papers and conference papers in the area of computer vision and 3D reconstruction. His research interests include computer vision, image segmentation, point cloud segmentation, object recognition, and 3D reconstruction.