Computer Systems
Science and Engineering

Tech Science Press

Doi:10.32604/csse.2025.061655

ARTICLE

Point-Based Fusion for Multimodal 3D Detection in Autonomous Driving

Xinxin Liu and Bin Ye*

School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, 221116, China

*Corresponding Author: Bin Ye. Email: yebin@cumt.edu.cn

Received: 29 November 2024; Accepted: 09 January 2025; Published: 20 February 2025

ABSTRACT: In the broader field of mechanical technology, and particularly in the context of self-driving vehicles, cameras and Light Detection and Ranging (LiDAR) sensors provide complementary modalities that hold significant potential for sensor fusion. However, directly merging multi-sensor data through point projection often results in information loss due to quantization, and managing the differing data formats from multiple sensors remains a persistent challenge. To address these issues, we propose a new fusion method that leverages continuous convolution, point-pooling, and a learned Multilayer Perceptron (MLP) to achieve superior detection performance. Our approach integrates the segmentation mask with raw LiDAR points rather than relying on projected points, effectively avoiding quantization loss. Additionally, when retrieving corresponding semantic information from images through point cloud projection, we employ linear interpolation and upsample the image feature maps to mitigate quantization loss. We employ nearest-neighbor search and continuous convolution to seamlessly fuse data from different formats. Moreover, we integrate pooling and aggregation operations, which serve as conceptual extensions of convolution, and are specifically designed to reconcile the inherent disparities among these data representations. Our detection network operates in two stages: in the first stage, preliminary proposals and segmentation features are generated; in the second stage, we refine the fusion results together with the segmentation mask to yield the final prediction. Notably, in our approach, the image network is used solely to provide semantic information, serving to enhance the point cloud features. Extensive experiments on the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset demonstrate the effectiveness of our approach, which achieves both high precision and robust performance in 3D object detection tasks.

KEYWORDS: Autonomous driving; 3D object detection; multi-sensor fusion; deep learning

1 Introduction

Driven in part by the growing interest in self-driving vehicles, substantial research methods have focused on 3D object detection [1–5]. Ensuring the safe navigation of autonomous vehicles requires accurate perception of objects in three-dimensional space. While existing approaches leverage LiDAR points for 3D object detection [6–10], these data alone lack expressive semantic information. As a result, the model may be forced to contend with incomplete or ambiguous cues, making reliable interpretation more challenging [11,12]. Additionally, the inherent sparsity of point clouds, particularly for small objects at greater distances, further complicates the task for a single LIDAR sensor.

Meanwhile, other methods [13–17] attempt to infer 3D locations and dimensions directly from images. Compared with point clouds, images offer a more detailed and compact representation, offering more extensive semantic cues for distinguishing varied instances and complex backgrounds. However, the inherently

Copyright © 2025 The Authors. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

nature of 2D images leads to a loss of reliable depth information, making purely image-based 3D detection approaches more challenging.

To address these challenges, numerous studies have focused on fusing multiple sensor modalities to strengthen the semantic information of point clouds. Classical two-stage architectures such as AVOD [18] and MV3D [19] process data from each sensor through its own backbone network, then combine them at the object proposal stage before refining to a final prediction. Approaches like F-PointNet [20] and F-ConvNet [21] leverage image network to extract semantic information, thereby limiting the search range inside a 3D view frustum, and then apply point-based detection algorithm to generate the 3D proposals. Similarly, ContFuse [22] integrate data from multiple sensors using a learned MLP, bilinear interpolation and k-nearest-neighbor searches on LIDAR points, effectively merging diverse modalities. PointPainting [23] project LIDAR points onto the image plane using a transformation matrix, and then directly utilize the information extracted from both LIDAR data and images. PointFusion [24] utilizes PointNet [6] to obtain point-wise feature and Faster RCNN [25] to achieve ROI, and the later fusion operation is conducted under the cropped ROI.

However, although classical two-stage architecture [18,19] offer end-to-end optimization, their reliance on plane-based fusion rather than raw LiDAR points leads to coarse and computationally slow predictions. Object identification based on the 3D view cone method [20,21] suffers from the limitations of image segmentation. It is challenging to address occlusion, which results in a limited number of foreground points. While Liang et al. [22] attempt to utilize continuous convolution [26] and deep fusion strategy to address the issue of different sensor's data formats, their combination is still based on BEV map that inevitably introduces precision loss. PointPainting [23] incorporates image-based semantic segmentation to enhance point features; however, its fusion approach relies solely on a transformation matrix without further refinement, leading to reduced precision due to this coarse operation. PointFusion [24] fuses data directly from raw LiDAR points to avoid quantification loss found in AVOD [18] and MV3D [19], but its image network merely performs object detection without providing the semantic information needed to enrich point-level features.

To overcome these drawbacks, we present a novel fusion module. Although AVOD [18] and MV3D [19] adopt deep fusion approaches to merge intermediate features and achieve improved fusion performance, their reliance on BEV or front-view maps—projections of raw LiDAR points—leads to significant precision loss. Notably, LIDAR data is sparse and continuous, whereas image data is inherently discrete. In contrast, our fusion method employs continuous convolution and nearest-neighbor search to directly handle the disparate formats of multi-sensor data, maintaining high-quality fusion without resorting to BEV maps. Different from Contfuse [22], our fusion module is performed directly on raw point data to avoid quantization loss. Furthermore, we integrate pooling and aggregation operations, which serve as conceptual extensions of convolution, and are specifically designed to reconcile the inherent disparities among these data representations. We argue that PointPainting [23] introduces quantification loss fusion approach by relying solely on a transformation matrix; to mitigate this issue, we apply linear interpolation and upsample image feature maps to mitigate this issue. Additionally, PointPainting [23] does not address the inherent data format discrepancies, and we solve this problem by applying the same approach used to resolve these discrepancies in AVOD [18] and MV3D [19]. Finally, while PointFusion [24] lacks semantic information, our method incorporates rich semantics, thereby achieving superior performance.

Our algorithm consists of two sub-algorithms and a dedicated fusion module. Specifically, we employ a classical two-stage object detection network for image segmentation, which processes color images to produce semantic information. The detection sub-algorithm is a 3D object detection framework that takes

raw LIDAR points as input and outputs preliminary predictions. The fusion module integrates these image-derived semantics with the LiDAR-based features, effectively bridging the segmentation and detection networks. By incorporating semantic cues into the LiDAR point data, our approach aims to enhance the accuracy of 3D object detection. Experiments on the KITTI dataset [27] demonstrate the effectiveness of our method.

Our contributions can be summarized into three crucial components:

1. We employ nearest-neighbor search and continuous convolution to seamlessly fuse data from different formats, and incorporate pooling and aggregation operations—conceptual extensions of convolution—specifically designed to address the inherent disparities among these data representations.
2. We integrate segmentation masks directly with raw LiDAR points instead of relying on projected points, thereby avoiding quantization loss. Additionally, when retrieving semantic information from images through point cloud projection, we apply linear interpolation and upsample the image feature maps to further mitigate quantization loss.
3. We conduct extensive experiments on the KITTI dataset [27], validating both the effectiveness and the efficiency of our approach.

2 Related Works

2.1 Camera-Based 3D Object Detection

With the rapid development of 2D image object detection, it's natural to consider using images to process 3D object detection. Mousavian et al. [13] and Li et al. [14] utilize 2D bounding boxes and surrounding image pixels to estimate the dimension and orientation of 3D objects. Chen et al. [15] projecting a 3D bounding box on the ground plane and leveraging features including semantic and instance segmentation, contextual information, shape, and spatial position to obtain proposals. Wang et al. [16] try to convert the image to a point cloud depth map and perform 3D object detection via LIDAR-based approaches. However, although cameras capture fine texture information, they cannot directly acquire depth information. Camera-based approaches [13–17] estimate depth at a per-pixel level, resulting in limited information for distant objects, and the loss of depth precision is unavoidable.

2.2 LIDAR-Based 3D Object Detection

Compared with images, point clouds directly obtain depth information whether in front-view or bird's eye view and the object depth is invariant. PointNet [6,7] leads the way to directly extract features from raw point clouds. Building on this, VoxelNet [9] emerged as a groundbreaking algorithm that downsamples point clouds into voxels. PointRCNN [10] generates 3D proposals from the point cloud and refines proposals in the second stage. LIDAR R-CNN [11] exploits a series of solutions based on the box to supplement the challenge of object scale loss. CenterPoint [12] learns to estimate objects with a keypoint detector [28] which extracts object properties from the object center. However, using a single LiDAR sensor results in coarse detection of distant and small objects due to sparse point clouds. Therefore, a common solution is to fuse image data with LiDAR points to achieve higher detection performance.

2.3 LIDAR-Camera 3D Object Detection

Laser scanners offer precise depth measurements, while cameras capture detailed semantics, making their combination appealing for multi-modal fusion. MV3D [19] adopts a multi-view fusion approach by incorporating a bird's eye view, a front view, and a corresponding RGB-image as input. Their two-stage network is composed of a 3D proposal sub-algorithm and a region-based fusion sub-algorithm.

ContFuse [22] conducts k-nearest points search and continuous convolution on the BEV map to improve their fusion precision. PointPainting [23] projects LIDAR points onto the output of image semantic feature using a calibration matrix to append LIDAR point segmentation information. The painted points can then be employed in LIDAR-based object detection [8–10]. PointFusion [24] utilizes PointNet [6] to obtain point-wise feature and Faster RCNN [25] to achieve ROI, and the later fusion operation is conducted under the cropped ROI.

We argue that direct fusion approaches [19,23] are flawed, as they inevitably introduce quantization loss. While Liang et al. [22] attempt to utilize continuous convolution [26] and deep fusion strategy to address the issue of different sensor's data formats, their combination is still based on BEV map that inevitably introduces precision loss. Although Vora et al. [23] attempt to enhance the points feature by incorporating image semantic segmentation, their fusion method relies solely on a transformation matrix without further refinement, resulting in precision loss due to its coarse operation. Meanwhile, PointFusion [24] fuses data directly from raw LiDAR points to avoid quantification loss found in AVOD [18] and MV3D [19], but its image network merely performs object detection without providing the semantic information needed to enrich point-level features. Inspired by Xie et al. [29] and Liang et al. [22], we learn to achieve direct fusion by merging image and point features using continuous convolution, point-pooling and a learned MLP.

3 Method

In this section, we introduce our method, which comprises a novel fusion module and two sub-algorithms. As shown in Fig. 1, the main components of our algorithm consist of a point cloud object detection algorithm and an image object detection algorithm, with the fusion module serving as a bridge that integrates data from both. The overall framework operates in two stages. For stage one, the detection sub-algorithm takes raw LIDAR points as input and generates preliminary predictions, including coordinates derived directly from the raw points, point-wise features obtained from the extractor, masks and 3D ROIs. Simultaneously, the segmentation sub-algorithm processes images to extract semantic features. These first-stage outputs are then passed into the fusion module, which fuse the data from LIDAR and images. In the second stage, the fused data is received to produce the final predictions. During this stage, only the point cloud data is processed for object detection, as our primary goal is to enhance LiDAR-based 3D detection using image-derived semantics. Note that we transform point coordinates into a canonical form, while keeping the other features unchanged. Local and global features are then combined and fed into the second stage. To address the perspectival differences between LiDAR and camera data, we apply convolution-based operations. We further incorporate pooling and a learned MLP to bolster the performance of continuous convolution, enhancements empirically shown to improve fusion quality. We integrate segmentation masks directly with raw LiDAR points instead of relying on projected points, thereby avoiding quantization loss. Additionally, when retrieving semantic information from images through point cloud projection, we apply linear interpolation and upsample the image feature maps to further mitigate quantization loss. We use image segmentation instead of a classifier, since relying solely on 2D bounding boxes for 3D detection provides only a coarse ROI and makes it difficult for the subsequent network to handle distant, small-scale objects. By leveraging semantic segmentation, we supply richer information that facilitates more accurate and robust 3D object detection.

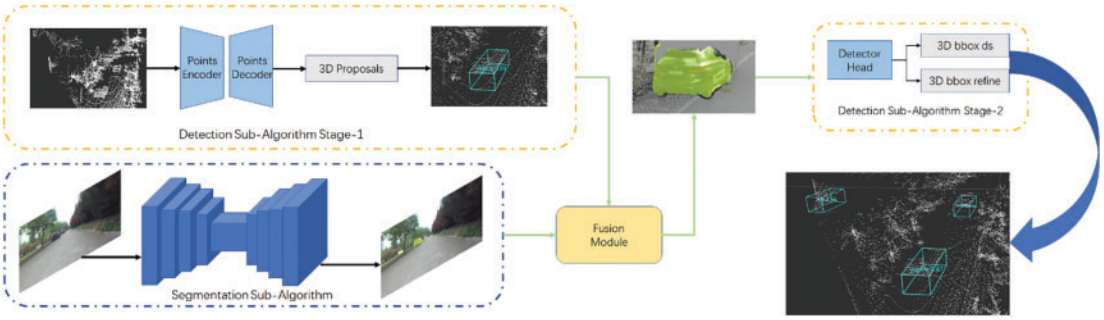

Figure 1: The main architecture of our proposed framework

We argue that early fusion is sensitive to data format differences and the model is inflexible, making subsequent operations challenging. On the other hand, late fusion discards intermediate features and demands substantial computational resources. Therefore, we employ the middle fusion strategy that blends feature-level and decision-level fusion. At intermediate layers, we integrate features from multiple modalities. Drawing on the deep fusion approach introduced in MV3D [19], we make key improvements. Specifically, in our fusion module, we apply pooling operations and continuous convolutions directly, and then incorporate a learned MLP trained on the output of consecutive convolutions. Finally, we concatenated the processed data from the aforementioned three parts to complete the fusion operation. This approach enables the algorithm to effectively leverage distinct modalities with unique feature representations and varying levels of complexity [3].

3.1 Fusion Module

In this section, we present our new fusion method. The fusion module in our system performs fusion operations directly on the raw LIDAR points, eliminating the reliance on bird's-eye-view projections and thereby avoiding quantization loss. When retrieving corresponding semantic features from the images for the point clouds, we employ interpolation and upsampling of the feature maps to reduce quantization loss. To address the inherent differences in data formats, we employ continuous convolution and k-nearest neighbor search, and we enhance this process by incorporating pooling operations and a learned MLP. The overall architecture of the fusion module is shown in Fig. 2.

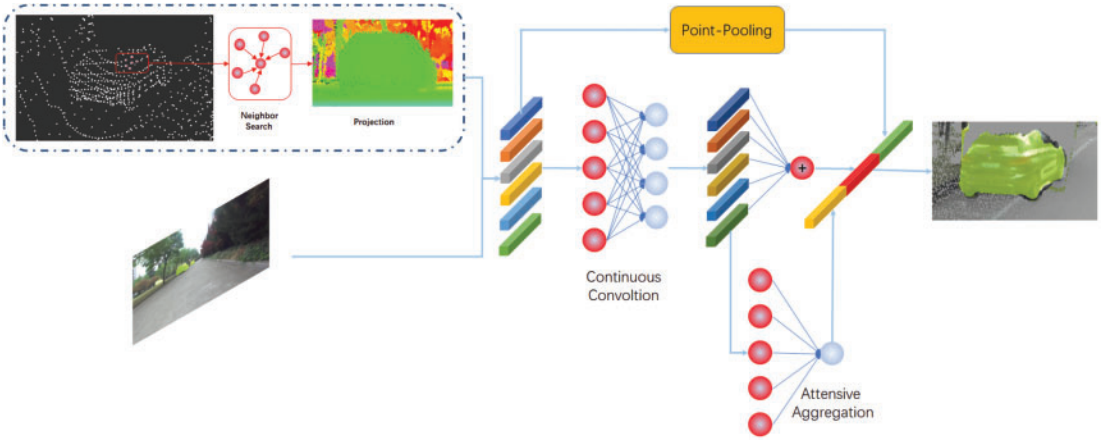

Figure 2: The illustration of our proposed fusion module

We argue that different perspectives and data formats across modalities are primary sources of precision loss. While images capture fine textures, they are limited to 2D projections of the real world. In contrast, LiDAR points provide rich 3D structures of the scenes. Liang et al. [22] introduce k-nearest points search and bilinear interpolation to improve point cloud object detection. However, their reliance on the BEV format quantizes the 3D point clouds, causing precision loss and leading to inaccuracies in neighbor searches and feature combination on the BEV map. Although Liang et al. later introduced MMF [30] to establish a closer connection between images and BEV using multi-task learning, they still refrained from applying continuous convolution directly on the 3D points. Vora et al. [23] project LIDAR points onto the image plane using a transformation matrix, then directly utilize the information extracted from both LIDAR data and images. However, due to the different perspectives offered by BEV maps and images, directly merging their data is too coarse to yield accurate and informative features. PointFusion [24] utilizes PointNet [6] to obtain point-wise features and Faster RCNN [25] to achieve ROIs, performing subsequent fusion within these cropped regions. Although their fusion is based on raw LIDAR points and proposed a dense fusion method, it operates within a limited spatial scope. As a result, their approach struggles with small or distant objects because the image network only performs object detection rather than providing the semantic information needed to enrich point-level features.

To address the quantization loss stemming from BEV usage in Liang et al. [22,30], we perform feature fusion directly on the raw points to avoid such degradation. Furthermore, when projecting point clouds onto image feature maps, we enlarge the feature maps to reduce quantization loss during this process. In addition, we incorporate pooling and a multilayer perceptron to further enhance the performance of continuous convolution. Specifically, these components are employed to address the issue of feature sparsity. While PointFusion [24] suffers from poor detection accuracy due to the absence of image semantic information, our method enriches point clouds with these semantic cues. In contrast to PointPainting [23], which fuses multi-sensor features through direct projection and thus results in a coarse integration, we utilize continuous convolution and nearest-neighbor search to effectively address the differences in data formats across various sensors. This approach enables a more precise and nuanced fusion of data, enhancing the overall accuracy and robustness of the detection process.

In particular, the fusion module is separated by five phases:

1. We conduct KNN search for each source LIDAR point.
2. We project the neighboring points of the target point onto the image plane using the matrix supplied by KITTI [27].
3. We retrieve the corresponding semantic feature from image, and compute the geometric offset between neighboring points and target point. We subsequently concatenate them, and the concatenation can be defined as:

$$f_k^{'} = \text{CONCAT}\left(f_k, x_k - x_i\right) \tag{1}$$

where x_i is the coordinate of target point p_i, and the target point has several neighbor points, so x_k is their coordinates. Since we conduct KNN search for each source point, the range of i is from 1 to N, where N is the number of points. k is the number of neighbor points, which range from 1 to K, where K is the number of neighbor points. Therefore, $x_k - x_i$ represents the geometric offset, f_k is the corresponding image semantic feature of point p_k, CONCAT $(.)$ is the operation of concatenation. Then, we concatenate them and $f_k^{'}$ is the final output.

4. We utilize continuous convolution to fuse the concatenated features from step 2. The convolution can be defined as:

$$y_{cc}^{i} = \sum y_{cc,k}^{i}, y_{cc,k}^{i} = \mathrm{MLP}_{cc}\left(f_{k}^{'}\right) \tag{2}$$

where $f_{k}^{'}$ is the output from step 3, $\mathrm{MLP}_{cc}(.)$ represent the process of continuous convolution, and $y_{cc,k}^{i}$ is the processed feature of neighbor point p_k. Then, we concatenate all the neighbor points and y_{cc}^{i} is the final output.

5. To achieve fine fusion performance, we add a pooling operation and a learned MLP to enhance convolution operation. The pooling operation is conducted before convolution and the learned MLP is conducted based on convolution.
The pooling operation can be defined as:

$$y_{\mathrm{pool}}^{i} = \mathrm{POOL}(F'), F' = \left[f_{1}^{'T}, f_{2}^{'T}, \ldots, f_{K}^{'T}\right]^{T} \tag{3}$$

where $f_{k}^{'}$ is the output from stage 3, we combine them into F' and perform pooling operations uniformly. $\mathrm{POOL}(.)$ is the operation of pooling. y_{pool}^{i} is the pooled feature of all nearest neighbors of point p_i.
The aggregation can be defined as:

$$y_{a}^{i} = \mathrm{MLP}_{aggr}\left(y_{cc}^{i}\right) \tag{4}$$

where y_{cc}^{i} is the features outputted from step 4, $\mathrm{MLP}_{aggr}(.)$ represents the operation of aggregation and y_{a}^{i} is the final output.
The fusion module produces the final output by combining the three pieces mentioned above through concatenation:

$$y_{o}^{i} = \mathrm{CONCAT}\left(y_{cc}^{i}, y_{a}^{i}, y_{\mathrm{pool}}^{i}\right) \tag{5}$$

3.2 Image Segmentation Sub-Algorithm

Note that features derived from a classifier [25] are sufficient for 2D object detection, as this task merely requires a 2D bounding box and associated confidence scores. However, 3D object detection demands finer-grained correspondence between the LiDAR and image data, necessitating more meticulous information. Although PointFusion [24] aimed to achieve better detection performance by adding images and implementing a dense fusion strategy, its results remain unsatisfactory. We believe that the lack of semantic information in image detection is a key factor underlying these limitations.

We assert that image segmentation is highly effective for enhancing fusion performance. It provides pixel-level segmentation, accurately distinguishing foreground objects from the background, which facilitates the integration of color images and LIDAR points. Additionally, segmentation suppresses background information, thereby reducing computational load and improving fusion efficiency. Pixel-level features enable precise correspondence between images and points, ensuring accurate data alignment. The semantic information extracted from images is utilized to enhance the features of LIDAR points, thereby improving the performance of detection. In our framework, we employ Mask RCNN [31] to extract semantic information from images, serving as the segmentation sub-algorithm. It is important to note that alternative segmentation algorithms can be substituted as needed, allowing for flexibility in adapting to different requirements or advancements in segmentation technology.

3.3 3D Detection Sub-Algorithm

We contend that point-based fusion exhibits greater fusion performance compared to BEV-based fusion. Implementing point-wise fusion necessitates a 3D detection algorithm that processes raw 3D points. Therefore, we utilize PointNet++ [7], the improved version of PointNet [6] for our detection algorithm stage one. In this stage, we extract point-wise features, which are then utilized to perform 3D bounding box estimation and point-based segmentation, thereby generating preliminary 3D proposals. The second stage is consistent with PointRCNN [10]. Initially, bounding boxes are expanded to obtain informative contextual information. Subsequently, the information of local spatial points is achieved by rotation and translation and concentrated with global features. The concentrated features are subsequently feed to the encoder to conduct 3D box refinement. Importantly, the second stage refines fusion features that contains semantic information from images, meaning that it does not rely solely on point cloud data as in the original research. This integration of image semantics enhances the point cloud features, thereby improving the overall detection performance.

3.4 Loss

Note that the core of our framework is 3D point object detection, and our loss function is centered around the point cloud object detection. We first present the overall loss:

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{seg}} \tag{6}$$

The detection sub-algorithm loss can be defined as:

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{refine}} \tag{7}$$

where $\mathcal{L}_{\text{reg}}$ is stage one loss and $\mathcal{L}_{\text{refine}}$ is stage two loss, here we give more fine explanation, for $\mathcal{L}_{\text{reg}}$:

$$\begin{aligned}
\mathcal{L}_{\text{reg}} &= \frac{1}{N_{\text{pos}}} \sum \left(\mathcal{L}_{\text{bin}}^{(p)} + \mathcal{L}_{\text{res}}^{(p)} \right), \\
\mathcal{L}_{\text{bin}}^{(p)} &= \sum_{u \in \{x,z,\theta\}} \left(\mathcal{F}_{\text{cls}} \left(\widehat{\text{bin}}_u^{(p)}, \text{bin}_u^{(p)} \right) + \mathcal{F}_{\text{reg}} \left(\widehat{\text{res}}_u^{(p)}, \text{res}_u^{(p)} \right) \right), \\
\mathcal{L}_{\text{res}}^{(p)} &= \sum_{v \in \{y,h,w,l\}} \mathcal{F}_{\text{reg}} \left(\widehat{\text{res}}_v^{(p)}, \text{res}_v^{(p)} \right)
\end{aligned} \tag{8}$$

where $\widehat{\text{bin}}_u^{(p)}$ is the predicted bin assignments and $\text{bin}_u^{(p)}$ is the ground truth, $\widehat{\text{res}}_v^{(p)}$ is the predicted residuals and $\text{res}_u^{(p)}$ is the ground truth. Note that we mainly consider foreground point p, so N_{pos} is the number of p. $\mathcal{F}_{\text{cls}}$ and $\mathcal{F}_{\text{reg}}$ is the classification loss and regression loss, respectively. We use $(x, y, z, h, w, l, \theta)$ to represent and refine the 3D proposal, where (x, y, z) is location, (h, w, l) is size and θ represents orientation.

While for $\mathcal{L}_{\text{refine}}$:

$$\mathcal{L}_{\text{refine}} = \frac{1}{\|\mathcal{B}\|} \sum_{i \in \mathcal{B}} \mathcal{F}_{\text{cls}} \left(\text{prob}_i, \text{ label}_i \right) + \frac{1}{\|\mathcal{B}_{\text{pos}}\|} \sum_{i \in \mathcal{B}_{\text{pos}}} \left(\tilde{\mathcal{L}}_{\text{bin}}^{(i)} + \tilde{\mathcal{L}}_{\text{res}}^{(i)} \right) \tag{9}$$

where $\mathcal{B}$ is the output of stage one, and $\mathcal{B}_{\text{pos}}$ is the proposal of the foreground points. $\tilde{\mathcal{L}}_{\text{bin}}^{(i)}$ and $\tilde{\mathcal{L}}_{\text{res}}^{(i)}$ are similar to $\mathcal{L}_{\text{bin}}^{(p)}$ and $\mathcal{L}_{\text{res}}^{(p)}$ in (8). prob_i is the estimated confidence of $\tilde{b}_i$, where $\tilde{b}_i$ denotes the bounding boxes.

For the segmentation sub-algorithm, we extract semantic information and project the labeled points onto the image to establish correspondences. Given that foreground points are typically fewer than background points, we address this imbalance by employing focal loss [32]:

$$\mathcal{L}_{\text{seg}}\ (p_t) = -\alpha_t\,(1-p_t)^{\gamma}\log(p_t)$$
$$\text{where } p_t = \begin{cases} p & \text{forground} \\ 1-p & \text{background} \end{cases} \tag{10}$$

and we set $\alpha_t = 0.25, \gamma = 2$ as default.

4 Results and Discussion

In this section, we introduce the implementation details of our algorithm and compare it with other methods for 3D detection of the KITTI dataset [27]. Meanwhile, we conduct extensive ablation studies to investigate each component of our algorithm to validate our design.

4.1 Experimental Setup

4.1.1 Network Architecture

We utilize the architecture of Mask RCNN [31], a classical two-stage image segmentation network, as our segmentation sub-algorithm. Mask RCNN [31] was chosen over other segmentation methods because it performs both object detection and instance segmentation, making it ideal for our task. U-Net [33], though efficient for medical images, lacks the ability to detect and segment objects simultaneously. While Deeplab [34] excels at semantic segmentation, it doesn't distinguish between different instances of the same object. In contrast, Mask RCNN [31] combines object proposal generation and precise mask prediction, making it more suitable for handling complex scenes with multiple objects. Although different segmentation networks can be used, our focus is on leveraging semantic features to enhance 3D object detection performance. Therefore, we do not emphasize the detailed framework of the segmentation algorithm, and we employ the same experimental setup of original research across all experiments.

We utilize PointRCNN [10], a point-based 3D detection method as our 3D detection sub-algorithm. Note that our fusion is conducted at the end of the detection network's first stage due to our fusion strategy. For input, we select a subset of 16,384 points from every frame as PointRCNN [10]. Through empirical testing, we found that this number strikes a balance between capturing sufficient detail and maintaining computational efficiency. Increasing the number of points beyond 16,384 offered only marginal improvements, while fewer points led to a noticeable drop in performance. If a scene contains fewer than 16,384 points, we randomly duplicate some points to meet the required number. We would like to point out that although randomly duplicating some points may have an impact on the experiment, such as overfitting and reduced model generalization, the effect is minimal and can be considered negligible. The fusion module receives point-wise features obtained from the extractor, the raw points' coordinate, mask and ROIs as input. The fusion strategy is modified from deep fusion and achieve positive effects.

4.1.2 Implementation and Training Details

For the detection sub-algorithm, we utilize raw 3D point clouds rather than employing BEV format data. The input configuration for the 3D point clouds follows the methodology established in PointRCNN [10]. The designated area of interest for LIDAR points is defines as $[0, 70.4] \times [-40, 40] \times [-1, 3]$ in the LIDAR coordinate system. In the stage-1 sub-algorithm, we categorize all points located inside the ground-truth boxes as foreground points, while considering all other points as background points. For the preliminary

proposal, we set the search range $S = 3$, bin size $\delta = 0.5$ m and orientation bin number $n = 12$, considering the actual size of vehicle. For the box proposal refinement sub-algorithm, we set the search range $S = 1.5$, bin size $\delta = 0.5$ m and orientation bin size $\omega = 10°$, the bounding box enlarged parameter η is set as 1. During our testing, we discovered that sampling the LIDAR points, similar to how training is conducted, yields better results compared to employing all the data points. As a result, we have consistently applied this subsampling strategy across all our models. Since subsampling is random, it inevitably results in the loss of some point features, and certain point features are critical. While this approach may introduce some level of randomness in the evaluation outcomes, we have observed that the results remain stable for a particular model, in some cases, even show improvement. We abandon the operation of GT-AUG because the correspondence between points and pixels will be disrupted. This is because GT-AUG inevitably requires placing ground truth boxes and the points inside them from other scenes into the same positions in another scene, even though this operation is random and not repeated [10].

We trained the model using a batch size of 2 on 2 GPUs. The learning rate was initialized at 0.01, and was decreased by a factor of 0.1 at the 35th and 45th epochs. The training concluded after 80 epochs with 1 warm-up epoch set at the beginning.

4.2 Results on KITTI Dataset

The KITTI dataset's [27] 3D object detection benchmark comprises 7481 training samples. Following the approach outlined in PointRCNN [10], we split these samples into 3712 for training and 3769 for validation. KITTI [27] categorizes the labels into three difficulty levels: easy, moderate, and hard, based on bounding box heights, occlusion, and truncation levels. The leaderboard ranks all entries by Average Precision (AP) in the moderate subset. We train our model on the training set and evaluate its performance on the validation set, reporting the results accordingly.

4.2.1 Evaluation Metric

To ensure a fair comparison, all results are evaluated using average precision with an IoU threshold of 0.7 for cars, calculated on the validation set of the KITTI dataset [27]. The IoU threshold of 0.7 is widely regarded as a standard in the 3D object detection community, particularly for the KITTI dataset [27], as it strikes a good balance between precision and recall. While other thresholds could offer additional insights, 0.7 remains the most commonly used standard, providing a consistent and reliable evaluation of detection performance in line with previous studies.

4.2.2 Comparison with Other Methods

Our approach is trained end-to-end. During training, the ground truth segmentation mask is derived from the first-stage point cloud object detection. Specifically, the point cloud is projected onto the corresponding image to generate the ground truth mask. As 3D object annotations provide supervision only for binary classes [10], and our primary focus is on cars, we exclusively select the car class for our training and evaluation process.

For PointPainting [23], we conducted its painted PointPillar [8] in default and surprisingly found that it not only failed to achieve better performance but actually performed worse, despite its high performance in BEV domain. This indicates that in 3D point cloud object detection, simply projecting the point cloud onto a plane without further processing result in a loss of accuracy. At the same time, our algorithm clearly outperforms PointPainting [23], which we believe is due to our use of linear interpolation and apply operations on the feature maps to reduce quantization loss. Additionally, we employ nearest-neighbor search

and continuous convolution to seamlessly fuse data from different formats. This highlights the significant precision loss when directly fusing image and point cloud data. PointFusion [24] attempts to fuse image data to achieve better detection performance. However, although they propose a dense fusion and the fusion operation is based on raw LIDAR points, the supplied image information is ROI without semantic segmentation. It insufficient for detection network to achieve fine performance. In contrast, our algorithm incorporates semantic information from the image, which is the key reason for our superior performance. ContFuse [22] employs KNN and continuous convolution to conduct object detection. However, their approach is limited to 2D plane and thus falls short in achieving fine object detection. In contrast, by directly fusing on the raw point cloud, our method achieves better results, further demonstrating its effectiveness. PointRCNN [10] is conducted without GT-AUG, while other settings kept at their default values. Their detection accuracy is limited due to the lack of supplementary image data. MV3D [19] uses multi-view to operate object identification, but suffers from severe issues with perspective and data format of multi-modal. These are the main drawbacks of their algorithm and the reason for its low accuracy. In contrast, our algorithm overcomes these challenges by employing nearest-neighbor search and continuous convolution to address data format discrepancies, and by directly fusing data on the raw point cloud to reduce quantization loss. Additionally, we use interpolation and apply operations on feature maps to further minimize quantization loss, resulting in improved detection performance. The comparison with other methods is shown in Table 1.

Table 1: Performance comparison of 3D AP with previous methods on KITTI val split

Method	Modality	3D AP (Car)		
		Easy	Moderate	Hard
MV3D [19]	Image + LIDAR	71.29	62.68	56.56
PointFusion [24]	Image + LIDAR	77.92	63.00	53.27
ContFuse [22]	Image + LIDAR	82.54	66.22	64.04
PointPainting [23]	Image + LIDAR	87.08	78.43	75.58
PointPillar [8]	LIDAR	87.82	78.55	75.73
PointRCNN [10]	LIDAR	88.45	77.67	76.30
Ours	Image + LIDAR	92.82	87.39	82.73

4.3 Ablation Study

We conduct ablation studies on the fusion module to analyze the effects of our method. The experiments are trained on train split and evaluated on val split of the KITTI dataset [27].

Fusion module

Here, we conduct ablation experiments on the fusion module. Note that when we directly conduct 3D object detection, the 3D object detection is generated by a single PointRCNN [10] without leveraging image semantic information. Our framework still works rather than breaking when the fusion module fails, thanks to our fusion strategy that conducts feature combination in the middle layer. Point-pooling and attentive aggregation are augmentation of continuous convolution and can be conducted independently. Therefore, we consider the condition where only one operation is active, and it is not feasible to perform point-pooling and attentive aggregation without the convolution operation. We achieve better results when we add the convolution module, while incorporating point-pooling and attentive aggregation has further enhance object identification performance. The ablation experiments are shown in Table 2. The ablation

study about hyperparametric K is shown in Table 3. To maintain simplicity, we set hyperparametric $d = +\infty$ as ContFuse [22] mentioned, since the model might ignore distant neighbors. We observe that $K = 5$ is significantly inferior $K = 3$. The explanation for this may be that a larger value of K includes points that are far away, which introduces noises into the target point features. The results obtained with $K = 1$ were unsatisfactory, which we believe is due to insufficient information being captured.

Table 2: Ablation study about the effects of fusion module on KITTI val split

Cont conv	Point-pooling	Att aggr	3D AP (Car)		
			Easy	Moderate	Hard
×	×	×	88.45	77.67	76.30
√	×	×	89.53	80.39	77.79
√	√	×	91.33	85.53	80.52
√	×	√	90.98	86.31	79.26
√	√	√	92.82	87.39	82.73

Table 3: Ablation study about the K

K	3D AP (Car)		
	Easy	Moderate	Hard
1	91.43	86.54	80.51
3	92.82	87.39	82.73
5	91.64	86.33	81.26

5 Conclusion

In this paper, we present a novel fusion method and a multi-sensor object detection algorithm that integrates image segmentation with 3D point cloud detection. Our fusion approach leverages the geometric offsets of points and retrieves corresponding semantic information from images to facilitate effective feature combination. It is worth mentioning that, when retrieving the corresponding image features, we use linear interpolation and apply operations on the feature maps to reduce quantization loss. We then apply convolution, pooling operations, and a learned MLP to achieve high-quality fusion results. A key advantage of our method is that we perform feature combinations directly on raw points rather than on the BEV plane, thereby avoiding precision loss. Additionally, the incorporation of pooling operations and a learned MLP further enhances the performance of continuous convolution. We conduct extensive experiments on the KITTI dataset, demonstrating that our proposed method achieves precise and robust object detection results. We propose an algorithm for autonomous driving. To address the complex road conditions in autonomous driving environments, we need to verify the generalizability of our algorithm. Therefore, in our future work, we plan to validate it on additional datasets. Additionally, the current algorithm only considers vehicles; we also need to incorporate more objects, such as cyclists and pedestrians, which are all highly valuable areas of research.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Bin Ye; data collection: Xinxin Liu; analysis and interpretation of results: Xinxin Liu; draft manuscript preparation: Xinxin Liu and Bin Ye. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Bin Ye, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Wang X, Li K, Chehri A. Multi-sensor fusion technology for 3D object detection in autonomous driving: a review. IEEE Trans Intell Transp. 2024;25(2):1148–65. doi:10.1109/TITS.2023.3317372.
2. Feng D, Harakeh A, Waslander SL, Dietmayer K. A review and comparative study on probabilistic object detection in autonomous driving. IEEE Trans Intell Transp. 2022;23(8):9961–80. doi:10.1109/TITS.2021.3096854.
3. Feng D, Haase-Schütz C, Rosenbaum L, Hertlein H, Gläser C, Timm F, et al. Deep multi-modal object detection and semantic segmentation for autonomous driving: datasets, methods, and challenges. IEEE Trans Intell Transp. 2021;22(3):1341–60. doi:10.1109/TITS.2020.2972974.
4. Jiao Y, Jie Z, Chen S, Chen J, Ma L, Jiang Y-G. MSMDFusion: fusing LiDAR and camera at multiple scales with multi-depth seeds for 3D object detection. Paper presented at: 2023 IEEE Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada.
5. Wang K, Zhou M, Lin Q, Niu G, Zhang X. Geometry-guided point generation for 3D object detection. IEEE Signal Process Lett. 2025;32:136–40. doi:10.1109/LSP.2024.3503359.
6. Charles RQ, Su H, Kaichun M, Guibas LJ. PointNet: deep learning on point sets for 3D classification and segmentation. Paper presented at: 2017 IEEE Conference on Computer Vision and Pattern Recognition; 2017 Jul 21–26; Honolulu, HI, USA.
7. Qi CR, Yi L, Su H, Guibas LJ. PointNet++: deep hierarchical feature learning on point sets in a metric space. Paper presented at: 31st Proceeding Advances in Neural Information Processing Systems; 2017 Dec 4–9; Long Beach, CA, USA.
8. Lang AH, Vora S, Caesar H, Zhou L, Yang J, Beijbom O. PointPillars: fast encoders for object detection from point clouds. Paper presented at: 2019 IEEE Conference on Computer Vision and Pattern Recognition; 2019 Jun 15–50; Long Beach, CA, USA.
9. Zhou Y, Tuzel O. VoxelNet: end-to-end learning for point cloud based 3D object detection. Paper presented at: 2018 IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA.
10. Shi S, Wang X, Li H. PointRCNN: 3D object proposal generation and detection from point cloud. Paper presented at: 2019 IEEE Conference on Computer Vision and Pattern Recognition; 2019 Jun 15–20; Long Beach, CA, USA.
11. Li Z, Wang F, Wang N. LiDAR R-CNN: an efficient and universal 3D object detector. Paper presented at: 2021 IEEE Conference on Computer Vision and Pattern Recognition; 2021 Jun 20–25; Nashville, TN, USA.
12. Yin T, Zhou X, Krähenbühl P. Center-based 3D object detection and tracking. Paper presented at: 2021 IEEE Conference on Computer Vision and Pattern Recognition; 2021 Jun 20–25; Nashville, TN, USA.
13. Mousavian A, Anguelov D, Flynn J, Košecká J. 3D bounding box estimation using deep learning and geometry. Paper presented at: 2017 IEEE Conference on Computer Vision and Pattern Recognition; 2017 Jul 21–26; Honolulu, HI, USA.
14. Li B, Ouyang W, Sheng L, Zeng X, Wang X. GS3D: an efficient 3D object detection framework for autonomous driving. Paper presented at: 2019 IEEE Conference on Computer Vision and Pattern Recognition; 2019 Jun 15–20; Long Beach, CA, USA.
15. Chen X, Kundu K, Zhang Z, Ma H, Fidler S, Urtasun R. Monocular 3D object detection for autonomous driving. Paper presented at: 2016 IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 27–30; Las Vegas, NV, USA.

16. Wang Y, Chao W-L, Garg D, Hariharan B, Campbell M, Weinberger KQ. Pseudo-LiDAR from visual depth estimation: bridging the gap in 3D object detection for autonomous driving. Paper presented at: 2019 IEEE Conference on Computer Vision and Pattern Recognition; 2019 Jun 15–20; Long Beach, CA, USA.
17. Xu B, Chen Z. Multi-level fusion based 3D object detection from monocular images. Paper presented at: 2018 IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA.
18. Ku J, Mozifian M, Lee J, Harakeh A, Waslander SL. Joint 3D proposal generation and object detection from view aggregation. Paper presented at: 2018 IEEE International Conference on Intelligent Robots and Systems; 2018 Oct 1–5; Madrid, Spain.
19. Chen X, Ma H, Wan J, Li B, Xia T. Multi-view 3D object detection network for autonomous driving. Paper presented at: 2017 IEEE Conference on Computer Vision and Pattern Recognition; 2017 Jul 21–26; Honolulu, HI, USA.
20. Qi CR, Liu W, Wu C, Su H, Guibas LJ. Frustum PointNets for 3D object detection from RGB-D data. Paper presented at: 2018 IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA.
21. Wang Z, Jia K. Frustum ConvNet: sliding frustums to aggregate local point-wise features for amodal 3D object detection. Paper presented at: 2018 IEEE International Conference on Intelligent Robots and Systems; 2019 Nov 3–8; Macau, China.
22. Liang M, Yang B, Wang S, Urtasun R. Deep continuous fusion for multi-sensor 3D object detection. Paper presented at: 2018 European Conference Computer Vision; 2018 Sep 8–14; Cham, Switzerland.
23. Vora S, Lang AH, Helou B, Beijbom O. PointPainting: sequential fusion for 3D object detection. Paper presented at: 2019 IEEE Conference on Computer Vision and Pattern Recognition; 2020 Jun 13–19; Seattle, WA, USA.
24. Xu D, Anguelov D, Jain A. PointFusion: deep sensor fusion for 3D bounding box estimation. Paper presented at: 2012 IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA.
25. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2017;39(6):1137–49. doi:10.1109/TPAMI.2016.2577031.
26. Wang S, Suo S, Ma W-C, Pokrovsky A, Urtasun R. Deep parametric continuous convolutional neural networks. Paper presented at: 2018 IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA.
27. Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite. Paper presented at: 2012 IEEE Conference on Computer Vision and Pattern Recognition; 2012 Jun 16–21; Providence, RI, USA.
28. Zhou X, Wang D, Krähenbühl P. Objects as points. arXiv:1904.07850. 2019.
29. Xie L, Xiang C, Yu Z, Xu G, Yang Z, Cai D, et al. PI-RCNN: an efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module. Paper presented at: 2020 AAAI Conference Artifical Intelligence; 2020 Apr 15–20; New York, NY, USA.
30. Liang M, Yang B, Chen Y, Hu R, Urtasun R. Multi-task multi-sensor fusion for 3D object detection. Paper presented at: 2019 IEEE Conference on Computer Vision and Pattern Recognition; 2019 Jun 15–20; Long Beach, CA, USA.
31. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. Paper presented at: 2017 IEEE International Conference on Computer Vision; 2017 Oct 22–29; Venice, Italy.
32. Lin T-Y, Goyal P, Girshick R, He K, Dollar P. Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell. 2020;42(2):318–27. doi:10.1109/TPAMI.2018.2858826.
33. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015: 18th International Conference; 2015 Oct 5–9; Munich, Germany.
34. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell. 2018;40(4):834–48. doi:10.1109/TPAMI.2017.2699184.