Transportation Research Part C 161 (2024) 104555

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Geometric information constraint 3D object detection from LiDAR point cloud for autonomous vehicles under adverse weather

Yuanfan Qi [a], Chun Liu [a,*], Marco Scaioni [b], Yanyi Li [a], Yihong Qiao [a], Xiaolong Ma [a], Hangbin Wu [a], Keke Zhang [c], Dazhi Wang [c]

[a] College of Surveying and Geo-informatics, Tongji University, Shanghai 200092, PR China
[b] Department of Architecture, Built Environment and Construction Engineering, Politecnico di Milano, Via Ponzio 31, Milano 20133, Italy
[c] Autonomous Driving Center, SAIC MOTOR R&D Innovation Headquarters, Shanghai 201804, PR China

ARTICLE INFO

Keywords:
3D object detection
Autonomous vehicles
LiDAR point cloud
Deep learning
Adverse weather

ABSTRACT

3D object detection, as the core of the autonomous vehicle perception module, is essential for efficient transportation and comfortable experiences. However, the challenge of 3D object detection under adverse weather conditions hinders the advancement of autonomous vehicles to higher levels. Hence, achieving accurate 3D object detection under adverse weather conditions is increasingly crucial as it forms the foundation for trajectory planning and driving strategy making in autonomous vehicles, thereby revolutionizing transportation modes for both goods and passengers. Advances in Light Detection and Ranging (LiDAR) technology have facilitated the development of 3D object detection in the past few years. Adverse weather, which inevitably occurs in real-world driving scenarios, could degrade measurement accuracy and point density of LiDAR and lead to particle interference. Detecting accurate 3D bounding boxes from sparse, incomplete point clouds with particle interference is difficult. Therefore, this research presents a novel geometric information constraint network for 3D object detection tasks from LiDAR point clouds under adverse weather (GIC-Net). In this study, we focus on how to incorporate geometric location information and line geometric feature information into the network against adverse weather. Further, we propose a geometric location constrained backbone module (GLC) to reduce rain and snow particle interference and ensure sufficient receptive fields. Then, we design a line geometric feature constraint module (LGFC) to add line constraints of 3D bounding boxes into the training process. Finally, a line loss function is designed, and features from the GLC and LGFC modules are fed into the multi-task detection head for accurate 3D bounding box prediction. Experiments on the Canadian Adverse Driving Conditions (CADC) autonomous vehicle dataset demonstrate the superiority of our method over six other state-of-the-art methods under adverse weather, which is at least 13.32 %, 4.67 %, and 10.44 % mAP higher than the other compared methods in the car, truck, and pedestrian classes respectively. Also, we further verify the better generalization ability of our network compared to other methods.

1. Introduction

In recent years, autonomous vehicles have emerged as a transformative mode of transportation with the potential to yield

* Corresponding author at: College of Surveying and Geo-informatics, Tongji University, No. 1239, Siping Road, Shanghai, PR China.
E-mail addresses: 2011478@tongji.edu.cn (Y. Qi), liuchun@tongji.edu.cn (C. Liu).

https://doi.org/10.1016/j.trc.2024.104555
Received 8 September 2023; Received in revised form 10 January 2024; Accepted 3 March 2024
Available online 16 March 2024

numerous societal benefits (Zhang et al., 2023a) including the mitigation of traffic accidents (Zhang et al., 2023b), reduction in fuel emissions (Liu et al., 2019), congestion alleviation (Yang et al., 2023a,b), and connectivity of neighborhoods (Gandomani et al., 2022), among other benefits. To operate successfully and avoid accidents in multiple driving scenarios, the 3D object detection task, as the core of autonomous vehicles perception module (Ma et al., 2022), has gained great attention due to the scene understanding capability it provides (Wu et al., 2022). Reliable and robust 3D object detection is the basis for downstream tasks of autonomous vehicle systems (e.g., object tracking (Yang et al., 2023a) and trajectory planning (Zhang et al., 2023a). However, incidents with autonomous vehicles under adverse weather are still rising (Chaturvedi et al., 2022), adverse weather-related traffic accidents account for at least 1.5 million incidents annually in the United States, resulting in up to 7,000 fatalities and approximately injuring around 800,000 individuals (Chaturvedi et al., 2022). Despite the considerable progress achieved through deep learning and computer advancements (Li et al., 2021a), the effectiveness of 3D object detection under adverse weather conditions remains challenging. Therefore, improving the performance of 3D object detection under adverse weather is a desirable goal for safe driving and efficient transportation, according to recommendations by the European Commission (Commission, 2021) and US Department of Transportation (Transportation, 2019).

Many sensors, including LiDAR, camera, and radar, are utilized by autonomous vehicles to obtain road information. Particularly noteworthy is LiDAR, which functions as an active sensor unaffected by lighting conditions while providing accurate point cloud data encompassing 3D coordinates along with distances between objects; azimuth angles; surface normal vectors information; etc.(Li et al., 2021b; Zhao et al., 2019), hereby stimulating many 3D object detection algorithms to choose LiDAR point cloud. Furthermore, LiDAR offers more comprehensive geometric information compared to RGB or radar images (Yu et al., 2022). Consequently, 3D object detection from LiDAR has been increasingly applied to 3D object detection tasks for autonomous vehicles over the years.

In this article, we define adverse weather as rainy weather and snowy weather. There are two main reasons for this. One, both types of weather have the probability of occurring in cities. Snowfall is observed in 77 % of countries globally (Zhang et al., 2023b), while rain occurs approximately 11 % of the time on a global scale (Zhang and Trenberth, 2018). Second, the presence of rain and snow poses significant hazards to driving safety, with studies indicating a doubling of traffic accidents in densely populated cities during rainy weather (Changnon, 1996) and at least 30,000 crashes occurring during snowfall in the United States (Administration, 2021). The perception module typically comprises hardware components like LiDAR along with software algorithms for tasks such as 3D object detection. However, actively pulsed hardware like LiDAR is susceptible to measurement accuracy and point density disturbances under adverse weather conditions such as rain and snow (Hahner et al., 2022), consequently impacting the precision of the associated software algorithms (Zhang et al., 2023b). Therefore, it is highly advisable to develop an accurate and robust method for detecting objects from LiDAR point clouds under adverse weather conditions to compensate for potential limitations arising from hardware deficiencies within the software algorithm side.

In recent years, deep learning-based methods have achieved great performance in multiple fields due to their ability to learn data representations with multiple levels of abstraction (LeCun et al., 2015). Notably, the advent of point cloud-based deep learning approaches, such as PointPillars (Lang et al., 2019), SECOND (Yan et al., 2018), PointRCNN (Shi et al., 2019), CenterPoints (Yin et al., 2021), Voxel-RCNN (Deng et al., 2021) and CG-SSD (Ma et al., 2022), has presented significant opportunities for addressing the challenges in 3D object detection under adverse weather conditions. Despite substantial progress achieved by numerous deep learning-based methods on publicly available datasets primarily focused on clear weather scenarios, their inadequate performance under adverse weather conditions hinders direct applicability in such scenarios (Wang et al., 2022). The key challenges associated with this issue primarily revolve around two aspects: firstly, the presence of a considerable number of rain and snow particles within LiDAR point clouds during adverse weather; secondly, abundant rain and snow lead to sparser point cloud density, making it challenging to devise a network architecture that accurately extracts 3D bounding boxes from sparse and incomplete surface point clouds.

To address the aforementioned challenges, we propose GIC-Net, an innovative deep learning network designed for efficient 3D object detection from LiDAR point clouds in adverse weather conditions. Our contributions can be summarized as follows:

(1) We introduce a geometric location constrained backbone module that effectively extracts feature representations by considering the geometric location information through ellipsoidal constraints, even in environments with rain and snow particle interference.

(2) We devise a linear adaptive down-sampling method to ensure sufficient receptive fields, allowing for adaptive adjustment of the number of down-sampling layers based on sparse features.

(3) We construct a line geometric feature constraint module using line features (length, slope, and intercept), which enhances object detection capabilities from sparse and incomplete point clouds by incorporating geometric features such as slope, intercept, and length of lines within 3D bounding boxes.

2. Related works

2.1. Influences of adverse weather on LiDAR

Based on basic publications (Gunn and Marshall, 1958; Uijlenhoet and Stricker, 1999) with meteorological knowledge, many researchers (Bijelic et al., 2018; Hahner et al., 2022) have introduced disturbance patterns under adverse weather and field experiments to reason about the impact of adverse weather on LiDAR. It has been established that light rain exerts negligible influence on LiDAR measurements (Fersch et al., 2016). Nevertheless, during intensified precipitation events, raindrops assume a dense smoke-like appearance leading to a significant reduction in the reflection intensity of LiDAR signals as demonstrated by Hasirlioglu's experiments (Hasirlioglu et al., 2016). Field-collected LiDAR data from rainy days also substantiate that rainfall introduces noise interference into

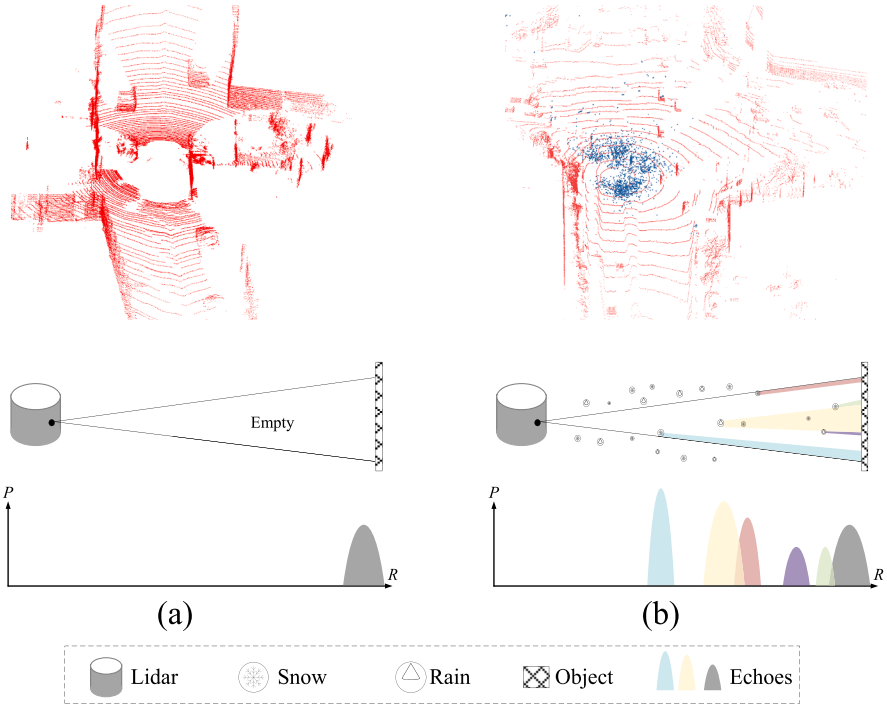

Fig. 1. One LiDAR beam under clear weather (a) and under adverse weather (b).

the point cloud resulting in compromised integrity (Filgueira et al., 2017; Lambert et al., 2020). Additionally, snowfall is accompanied by low temperatures, which can severely impede the sensitivity of LiDAR components, causing latency issues (Jokela et al., 2019). Furthermore, solid snowflakes generate substantial noise both ahead of and behind the LiDAR system while exhibiting voids due to swirling effects at the front and rear of vehicles. As depicted in Fig. 1, received power by the LiDAR sensor represents a superposition of multiple echoes returned upon contact with objects under adverse weather conditions. If the peak value associated with rain or snow particles surpasses that of target objects, genuine object echoes are lost while rain or snow particles lead to a fake peak value within this context (Hahner et al., 2022). A comprehensive framework encompassing adverse weather effects on LiDAR including rain and snow has been investigated in Zhang's study (Zhang et al., 2023b). In summary, adverse weather conditions result in the distribution of rain and snow particles that can be detected by LiDAR as clutter, obscuring the effective target object or attaching to the LiDAR window, thereby reducing the accuracy and density of the point cloud.

Algorithmically, previous studies have considered rain or snow as noise and conducted de-raining and de-snowing investigations. Traditional denoising methods include statistical outlier removal (SOR) (Rusu and Cousins, 2011), dynamic statistical outlier removal (DSOR) (Charron et al., 2018), and low-intensity outlier removal (LIOR) (Park et al., 2020). These methods design filtering models by analyzing the intensity and distance information from the point cloud. Deep learning-based approaches, such as WeatherNet (Heinzler et al., 2020), learn denoising models from large labeled datasets. In addition to denoising algorithms, some studies cope with the problem of sparse points by improving the point cloud quality. Xu et al. (Xu et al., 2021) propose SPG, which generates semantic points to recover the shape of 3D objects. Additionally, considering the incomplete point cloud problem caused by signal missing, Li et al. (Li et al., 2023a) supplement the point cloud with pseudo point clouds generated by images. However, traditional methods lack sufficient adaptability to road scenarios, while point cloud quality improvement algorithms will add additional point clouds and consume significant computing resources, both of which restrict real-time operation for autonomous vehicles. Therefore, utilizing denoising algorithms or point cloud quality improvement algorithms as a pre-step for object detection under adverse weather conditions in autonomous driving is inappropriate.

2.2. Point cloud object detection

With the rapid advancement of LiDAR-based 3D scene understanding technology, deep learning methods have been widely employed by researchers for object detection tasks. Point cloud-based approaches can be broadly categorized into three subcategories.

2.2.1. Point-based method

The point-based method directly utilizes raw point clouds as inputs and extracts features by aggregating local salient information from PointNet (Charles et al., 2017). To address the issue of a large number of 3D anchor boxes in the 3D space, Shi et al. propose PointRCNN (Shi et al., 2019), which generates 3D proposals by segmenting the entire scenario's point cloud and subsequently refining

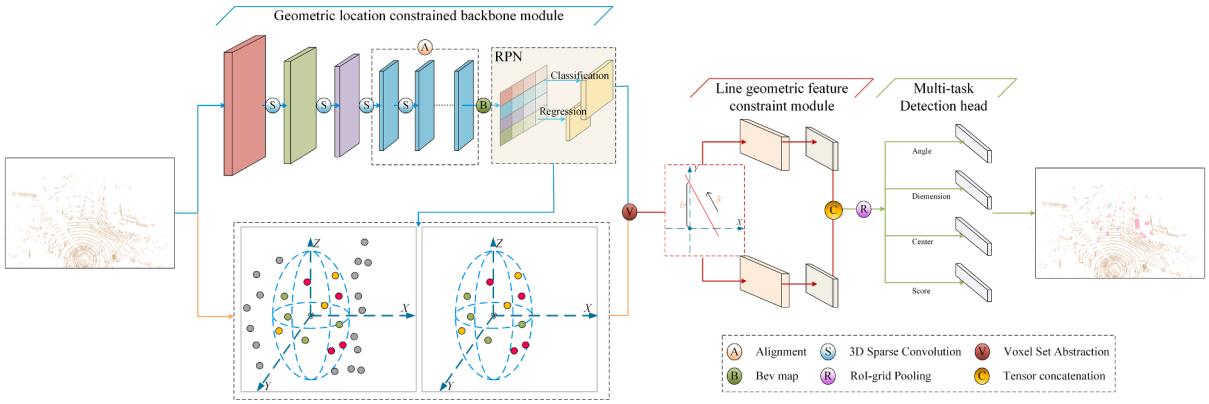

Fig. 2. Full pipeline of GIC-Net.

them. Additionally, to simultaneously consider local instance and global shape information of objects, Zhou et al. develop a spatial embedding module (Zhou et al., 2020) that efficiently assembles all foreground points into their respective object centers and predicts instance segmentation. Some methods leverage rich texture information provided by images and fuse it with the geometric information from the point cloud to enhance performance. ImVoteNet (Qi et al., 2020) extracts both geometric and semantic features from 2D images to assist in LiDAR-based 3D object detection. PointPainting (Vora et al., 2020) enriches the point cloud with image semantics by projecting LiDAR points onto an image-only semantic segmentation network output. Furthermore, SDTM framework (Gong et al., 2020) combines visual data with LiDAR-SLAM results to estimate the location and shape of objects on a 3D probability map projection level; however, these methods often introduce additional image inputs that may become ineffective due to blockage or distortion caused by adverse weather conditions (Zhang et al., 2023b). Moreover, due to the necessity of directly handling large-scale point clouds on roads, point-based methods are typically computationally expensive.

2.2.2. *Voxel-based method*

Various voxel-based approaches represent the irregular 3D real world using regular voxels in order to preserve shape information. To fully leverage 3D convolutional layers, VoxelNet (Zhou and Tuzel, 2018) introduces an end-to-end framework that learns high-level features from 3D voxels. In order to achieve improved orientation estimation and convergence speed, SECOND (Yan et al., 2018) devises a novel angle loss regression and utilizes spatial relationships between points to expedite 3D convolution. Differing from VoxelNet and SECOND, PointPillars (Lang et al., 2019) learns features from stacked pillars by PointNet (Charles et al., 2017), which inevitably results in a loss of information in the Z-axis direction. Li et al. propose PillarNeXt (Li et al., 2023b) to improve the pillar-based method by enlarging the receptive field. Additionally, as voxelization may result in the loss of spatial information within the raw point cloud data, Voxel-FPN (Kuang et al., 2020) effectively extracts and merges multi-scale voxel information by aggregating spatial details across multiple resolutions of voxels. Some methods capture bird's eye view (BEV) features by projecting voxels onto a two-dimensional BEV plane. CenterPoints (Yin et al., 2021) directly estimates object centers through keypoint detection and subsequently regresses other bounding box attributes. In road scenarios, target occlusion can cause the vehicle point cloud surface incomplete. To address the problem, CG-SSD (Ma et al., 2022) innovatively proposes a corner guided auxiliary module and inserts it into the network. To get rid of the dependency of hand-crafted proxies, VoxelNeXt (Chen et al., 2023) directly detects 3D objects based on sparse voxel features. Virtual points generated from an image can also be used for 3D object detection, but their dense nature and noise often seriously affect the detection accuracy. To solve this problem, Wu et al. present VirConv (Wu et al., 2023), which addresses the density and noise problems by discarding redundant voxels and encoding voxel features. While voxel-based methods can achieve a balance between accuracy and speed, they inevitably overlook the location information of point clouds in road scenarios (Yu et al., 2022), leading to the loss of geometric information during voxel feature extraction.

2.2.3. *Point-voxel method*

Point-voxel method: Instead of solely relying on voxels or points as inputs, point-voxel methods leverage both raw point cloud data and regular voxels to overcome the limitations mentioned above. However, directly employing point-based methods for disorderly and irregular point clouds may result in significant computational costs. To tackle this issue, STD (Yang et al., 2019) introduces a PointsPool layer for compact representation that combines advantages from both point- and voxel-based approaches. In Fast Point R-CNN method (Chen et al., 2019), authors employ a voxel-based network for initial predictions followed by a fusion of interior points' features to refine prediction results. More recently, PV-RCNN (Shi et al., 2020) and PV-RCNN++ (Shi et al., 2022) extract both point-based features and voxel-based features from convolution neural network (CNN) and refine bounding box by Roi-grid pooling. Nevertheless, most existing point-voxel methods have been evaluated under clear weather conditions without considering adverse weather interference on LiDAR point clouds.

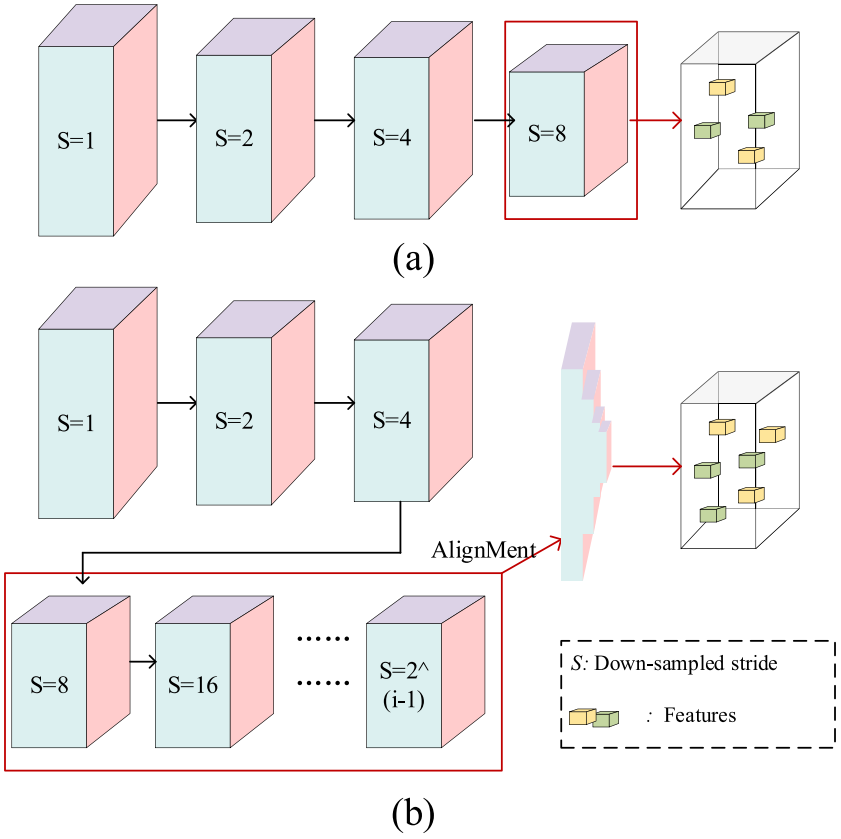

Fig. 3. Comparison between the other down-sampling layers (a) and ours (b).

2.3. Summary

The above analysis reveals that adverse weather conditions have a significant impact on the accuracy and integrity of LiDAR point cloud data, making it challenging to directly apply traditional denoising-based methods to the 3D object detection task in autonomous vehicles due to computational limitations and model maladaptation. In recent years, deep learning approaches applied to 3D point clouds have shown promising results in object detection tasks, indicating a vast potential for leveraging deep learning techniques in LiDAR-based 3D object detection (Huang et al., 2022). However, most existing methods are primarily designed and evaluated under clear weather conditions, leading to suboptimal performance when dealing with adverse weather scenarios.

3. The proposed network

The architecture of the proposed GIC-Net, depicted in Fig. 2, aims to predict the 3D bounding box from single frame LiDAR point cloud. GIC-Net comprises a geometric location constrained (GLC) backbone module, a line geometric feature constraint (LGFC) module, and a multi-task detection head. In the GLC module, grid voxels derived from the original point clouds are initially down-sampled using an adaptive method to ensure sufficient receptive fields. Subsequently, a sampling strategy inspired by ellipsoidal constraints and leveraging geometric location information is introduced to filter out invalid background points and aggregate multi-scale features into keypoints. Prior to the detection head, we incorporate the LGFC module, which performs supervised learning on geometric features of lines obtained under adverse weather conditions by LiDAR sensors. By jointly utilizing both GLC and LGFC modules, we aim to fully exploit the geometric information present in original point clouds. Finally, a multi-task detection head is employed for predicting accurate 3D bounding boxes.

3.1. Geometric location constrained backbone module (GLC)

The GLC module focuses a limited number of keypoints onto the surface of the 3D object, effectively mitigating interference from rain and snow point clouds while simultaneously expanding the receptive field. To achieve this objective, we propose a linear adaptive down-sampling method (LADS) and an sampling strategy (ECS), which incorporate additional down-sampling layers and leverage geometric location information.

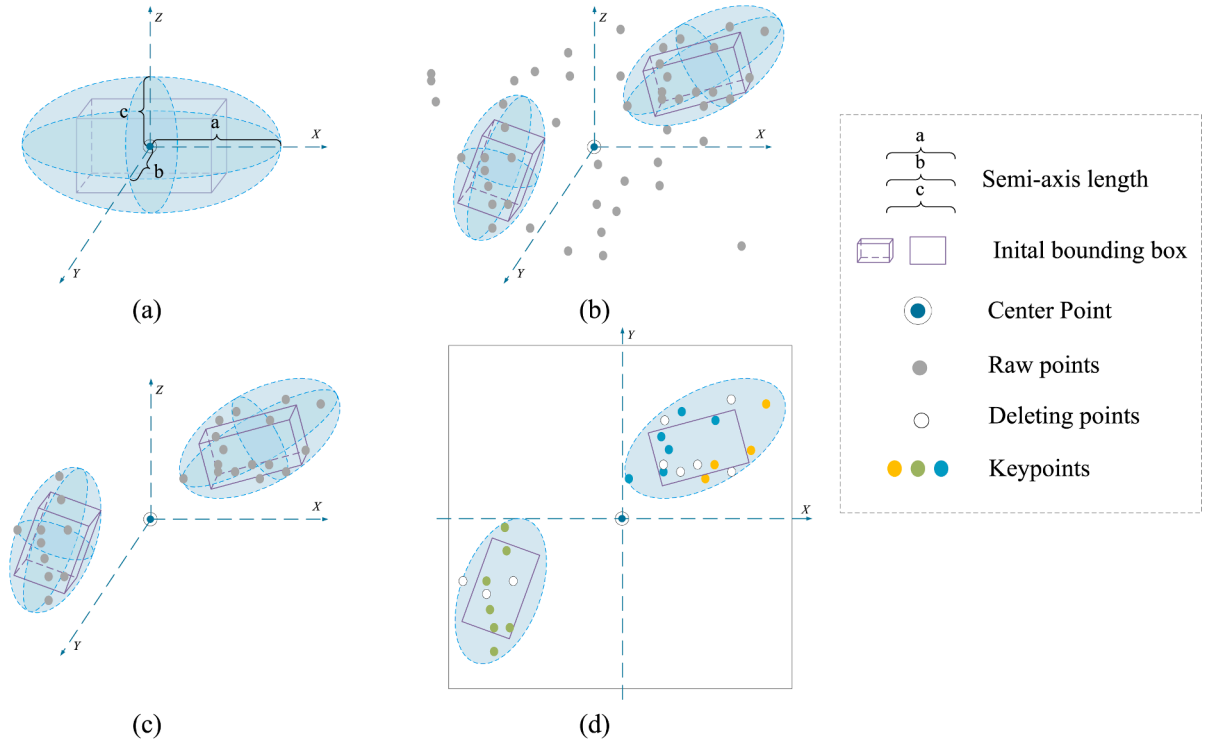

Fig. 4. Illustration of ECS strategy. (a) shows the ellipsoid in the same geometric location and coincident with the initial 3D bounding box. (b) and (c) show keypoints distribution before and after filtering by ECS, respectively. (d) shows the process of dividing into four slices for parallel computation.

3.1.1. Linear adaptive down-sampling

In deep learning-based object detection methods, strong receptive fields (Chen et al., 2023; Liu et al., 2023) are vital for accurate object detection. Although the commonly used multi-level down-sampling strides are widely applied in many 3D object detection methods. However, due to submanifold sparse convolution, the traditional CNN backbone network has proved its limitation and weakness (Chen et al., 2022) in sparse point clouds. To deal with this, we propose a linear adaptive down-sampling (LADS) method.

The sparse and irregular point clouds are first turned into voxels of the same voxel size $[V_x, V_y, V_z]$, where non-empty voxels are turned into three-dimensional tensors. Since 3D sparse convolution can achieve rich features and reduce GPU consumption (Dai et al., 2020; Ma et al., 2022), we apply 3D sparse convolution as the 3D backbone to obtain voxel-based features from the above tensors.

Plain 3D sparse CNN backbone networks, such as SECOND (Yan et al., 2018), have four down-sampling layers, which we name as $[L_1, L_2, L_3, L_4]$ (Fig. 3(a)). The four down-sampling layers output sparse features are $[F_1, F_2, F_3, F_4]$. $[P_1, P_2, P_3, P_4]$ denotes the position coordinates in 3D space of the corresponding features. Their down-sampled feature strides are $[1, 2, 4, 8]$. However, for large objects, such as trucks, its predictive power is inadequate. Hence, we set up several additional down-sampling layers here, and each layer is twice the feature stride of the previous one. Strides of additional down-sampling and positions of features starting from layer five are calculated as follows:

$$S_i = 2^{i-1} \tag{1}$$

$$P_i^{'} = \left\{ \left(2^{i-4} \times X_p, 2^{i-4} \times X_p, 2^{i-4} \times X_p\right) | p \in P_i \right\} (i \geq 5) \tag{2}$$

where i denotes the layer number, P_i denotes the positions in the 3D space of the corresponding features in different sampling layers, $P_i^{'}$ denotes their positions after enlarging the coordinates.

As the number of strides increases one by one, the feature dimensions of each stride gradually decrease by about half. When the number of feature dimensions of one stride can no longer be reduced (i.e. reduced to single digits, such as 2), this layer is regarded as the last down-sampling layer, and the features from F_4 to F_n are connected as follows:

$$\begin{cases} P_{all} = P_4 \cup (P_5^{'} \cup \left(P_6^{'} \cdots \left(P_{n-1}^{'} \cup P_n^{'}\right)\right) \\ F_{all} = F_4 \cup (F_5 \cup (F_6 \cdots (F_{n-1} \cup F_n)) \end{cases} \tag{3}$$

where F_{all} denotes the features from L_4 to L_n, and P_{all} denotes the positions of F_{all}. Noting that by equation (2), all their spatial

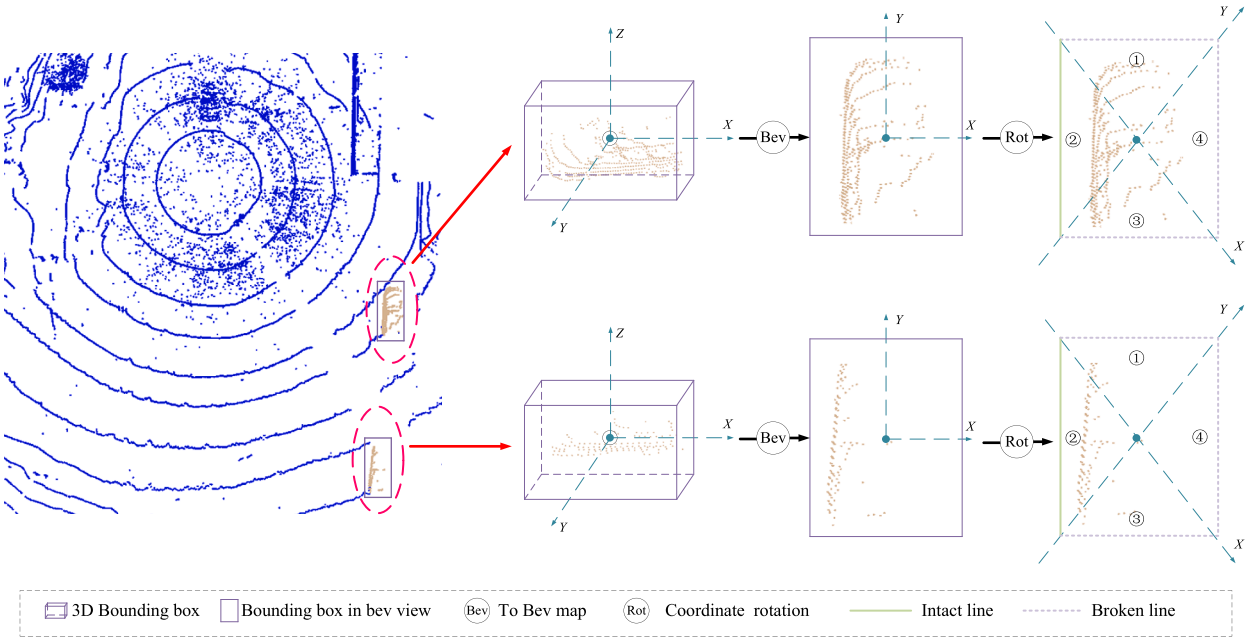

Fig. 5. The visibility of two types of lines in the bounding box.

resolutions are aligned to F_4 and can be directly connected. The feature at the higher layer is selected for multiple features sharing the same location.

After obtaining 3D voxel features, we project 3D feature F_{all} to bird's eye view (BEV) map in the vertical direction. Finally, after obtaining the BEV feature map, a 2D backbone network followed by the Region Proposal Network (RPN) is applied, here we adopt the same RPN strategy as other state-of-art models (Shi et al., 2020; Deng et al., 2021) for 3D proposal generation. To be specific, 2D features in the BEV map are naturally combined with the 2D detection heads composed of Conv2D-BatchNorm2D-ReLU (CBR) and ConvTranspose2D-BatchNorm2D-ReLU (CTBR) blocks. Using the 2D detection heads, the high-quality 3D proposal generation (Shi et al., 2020; Deng et al., 2021), here referred to as the initial 3D bounding box, can be obtained for the keypoints selection in the next section 3.1.2. Noting that through our LADS method, our network can learn multi-scale features with a negligible overhead (adding about 6 % extra time).

3.1.2. *Ellipsoidal constraint sampling strategy*

Considering the multi-scale of road scenario, to better extract multi-scale 3D features, the entire scenario is sampled into keypoints to combine with the voxel features from LADS for refinement. However, due to the influence of floating rain and snow particles, a large number of rain and snow point clouds could be detected by LiDAR, and these foreground point clouds are meaningless for feature representation and object detection (Transportation, 2019). Thus, we propose an ellipsoidal constraint keypoints sampling strategy with constraints on 3D geometric location information to generate keypoints around the 3D objects.

In ECS strategy, given an original frame point cloud in the training dataset as $P = \{P_1, P_2 \cdots P_n\} \in R^3$, we donate all initial 3D bounding boxes from section 3.1.1 as $B = \{C_i, l_i, w_i, h_i\}$, where $C_i \in R^3$ and C_i denotes the 3D center coordinate, where l_i, w_i, h_i denote length, width, and height of the initial 3D bounding box, respectively. Next, we fit the 3D neighbourhood space of 3D object through an ellipsoid E (Fig. 4(a)) as follows:

$$E = \{C_i, a_i, b_i, c_i\} \in R^3 \tag{4}$$

$$[a_i, b_i, c_i] = [l_i, w_i, h_i] \times 0.5 + R^e \tag{5}$$

where a_i, b_i, c_i denote three semi-axes of an ellipsoid, and R^e denotes a parameter representing the extended radius of the ellipsoid. In this paper, R^e is set to 1 according to Section 4.4.4. Since the 3D space location of the ellipsoid and the initial 3D bounding box coincide, all the keypoints P^{key} located in the ellipsoid are selected directly from the original point cloud P through one simple judgment formula as follows:

$$P^{key} = \left\{ p_i \left| \frac{x_i^2}{a_j^2} + \frac{y_i^2}{c_j^2} + \frac{z_i^2}{c_j^2} < 1 \right. \right\} \tag{6}$$

where $[a_j, b_j, c_j] \in E$, $[x_i, y_i, z_i] \in P_i$. Through the ellipsoid we construct (Fig. 4(a)) with the geometric location information of the

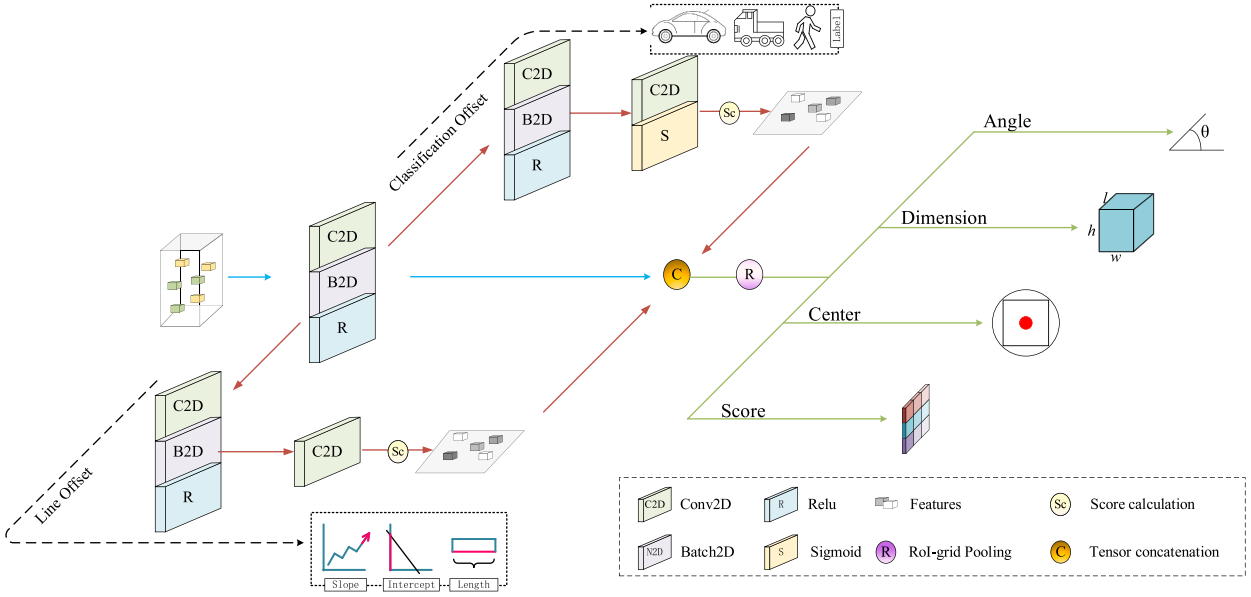

Fig. 6. The visibility of line geometric feature constraint module.

object, the disordered raw point clouds (Fig. 4(b)) are sampled around the 3D object (Fig. 4(c)). Finally, to improve computational efficiency, we evenly divide all point clouds into four pieces according to the X-Y plane. The Farthest Point Sampling (FPS) algorithm is applied to the ellipsoid of each piece to further sub-sample the keypoints (Fig. 4(d)), which makes full use of the parallel computing power of the GPU for acceleration.

Therefore, due to the constraints of geometric location information, the ECS strategy excludes the rain and snow interfering point clouds and focuses background keypoints on the 3D object surface, which can be combined with voxels features in Section 3.1.1 via voxel set abstraction (Shi et al., 2020). Through the GLC module, our network aggregates the multi-scale features from voxels to the keypoints, which can be used for high-quality 3D object detection tasks of different sizes.

3.2. *Line geometric feature constraint module (LGFC)*

By utilizing the GLC module, fused features are obtained. However, the reflection and refraction of laser light by rain and snow particles lead to incomplete and sparse objects represented by point clouds. It is challenging for a network to perceive objects from an incomplete and sparse structure. Thus, we proposed a line geometric feature constraint module. Using the line geometric features as constraints, the LGFC module enables the network to better capture lines that make up 3D bounding boxes.

3.2.1. *Line geometric features extraction*

As shown in Fig. 5, the bounding box in the BEV view is a rectangle consisting of four lines, and the point cloud of the object tends to be distributed inside the four lines of the rectangle. However, due to the interference of rain or snow and the angle of LiDAR measurement, the four lines surrounding the point cloud near the object cannot be all intact. Therefore, embedding broken line information as constraints in the network is necessary. Here we divide the bounding box lines from BEV view into two kinds: one intact line with a large number of point clouds around and three broken lines with a sparse number of point clouds around. Obviously, only three lines are needed to constrain and define a unique rectangle, so we only embed the geometric information of the three broken lines as constraints in the LGFC module.

To obtain the geometric features of the three broken lines during network training, we first need to compute the locations of four lines. First, based on the angle of rotation around the Y-axis θ and the location of the center point in the global coordinate system provided in the training set like KITTI (Geiger et al., 2013) and CADC (Pitropov et al., 2021), we turn the LiDAR global coordinate system into the local coordinate system of each respective bounding box as follows:

$$\begin{pmatrix} X \\ Y \end{pmatrix}_{local} = \begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix} R \tag{7}$$

where R denotes the rotation matrix defined by rotation angle θ. $\Delta X = X - X_{center}, \Delta Y = Y - Y_{center}$, where X_{center}, Y_{center} denote the 2D coordinate in the local coordinate. Next, four sectors are constructed by interconnecting the center point of the plane bounding box with the four corner points in the local coordinate system. The number of point clouds in each sector is counted, where the sector with the highest number of point clouds corresponds to one intact line, and the remaining three sectors correspond to three broken lines. In

this paper, we use length and slope-intercept form to represent the geometric features of the broken line according to the experiments provided in Section 4.4.3.

Since two points in the plane determine a line, finally, the geometric features corresponding to each broken line are extracted by the two endpoints of each line segment as follows:

$$F^{\text{slope}} = \frac{Y_1 - Y_2}{X_1 - X_2} \tag{8}$$

$$F^{\text{intercept}} = Y_1 - F^{\text{slope}} \times X_1 \tag{9}$$

$$F^{\text{length}} = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \tag{10}$$

where (X_1, Y_1),(X_2, Y_2) denote coordinates of the two ends of the broken line in the LiDAR coordinate system.

3.2.2. Constraint module

The constraint module establishes broken line feature constraints by calculating offsets. As shown in Fig. 6, the LGFC module derives two branching routes corresponding to the classification offsets and line feature offsets during the training process. Besides, the LGFC module consists of many CBR blocks, which learn the line features for more accurate rotation angle, length, and location information.

First, for every broken line, only one set of features extracted in Section 3.2.1 are regarded as truth features. All other features are negative. To constrain the unbalanced distribution of positive and negative feature samples, we penalize negative broken lines near positive broken lines with a widely-used unnormalized 2D Gaussian (Law and Deng, 2019). Note that we do not directly penalize constraining all negative broken lines because even negative lines may be connected to form a 3D bounding box similar to the ground truth box.

Next, this output is the constraint-score map $H \times W \times C$, where H and W denote the two sizes of the constraint-score map, and C denotes the offset value. Due to the rectangular nature of the bounding box in the BEV map, the line connecting the midpoint of any line segment (broken line, intact line) to the center point must be perpendicular to that segment. Therefore, we define the offset value as a 2D vector of line segment midpoints and centroids as follows:

$$\begin{vmatrix} D_x \\ D_y \end{vmatrix} = \begin{vmatrix} C_x \\ C_y \end{vmatrix} - 0.5 \times \begin{vmatrix} X_1 + X_2 \\ Y_1 + Y_2 \end{vmatrix} \tag{11}$$

Where C_x,C_y denote the center point value in the LiDAR coordinate system, where (X_1, Y_1),(X_2, Y_2) denote coordinates of the two ends of the broken line in the LiDAR coordinate system and D_x, D_y denote the object deviation value.

Finally, based on the 3D proposals from RPN, features concatenated from the GLC and LGFC modules are passed through a Roi-grid pooling operation (Shi et al., 2020) as input for the detection header.

3.3. Multi-task detection head

After GLC and LGFC modules, the entire 3D object detection network with multi-task detection heads is applied for accurate 3D bounding boxes. As shown in Fig. 6, our multi-task detection head consists of orientation angle, three dimensions (width, length, height), center point, and score.

Our GIC-Net aims to generate classification scores from the predicted 3D bounding box. It consists of three parts: classification loss of the detector (L_{cls}), regression loss (L_{reg}), and line deviation loss (L_{line}). To balance the positive and negative samples in our network, a variant of focal loss (Yan et al., 2018; Yu et al., 2022) is applied for the loss of the detector. Next, following the method (Lang et al., 2019; Yan et al., 2018), the regression loss function L_{reg} is calculated by smooth L1 loss.

In the LGFC module, we incorporate several line features to impose constraints on the lines forming the bounding box. It is evident that in a 3D space, as two lines become more similar, their start and end points tend to be closer. Hence, the line deviation loss can be perceived as the cumulative sum of geometric distances between the 12 lines of the predicted 3D bounding box and the ground truth bounding box. We compute Euclidean distances for both corresponding start and end points in both prediction and ground-truth boxes, subsequently adding them to approximately represent the geometric distance between these two lines. Notably, to mitigate excessive line discrepancies arising from incorrect angle estimation, we calculate both the angle prediction direction positive and completion opposite and take the smallest of them as the line loss as follows:

$$L_{line} = \sum_{i=1}^{PR} \sum_{j=1}^{GT} \delta_{ij} \min\left\{ \sum_{k=1}^{12} \| L_k^{ij} - L_k^{*} \|, \sum_{k=1}^{12} \| L_k^{ij} - L_k^{**} \| \right\} \tag{12}$$

$$L_k = P_s + P_e \tag{13}$$

where PR,GT denote the prediction box and the ground-truth box, P_s, P_e denote the start and end points of line segments. Finally, jointly applying the multi-task detection head, the total loss function of our GIC-Net is shown as:

$$L_{total} = L_{cls} + L_{reg} + W_l L_{line} \tag{14}$$

Table 1
Publicly available autonomous vehicle datasets with annotated LiDAR scans.

Dataset	Labeling	Almost all in adverse weather
SemanticKITTI (Behley et al., 2019)	point-wise	×
ApolloScape (Huang et al., 2020)	point-wise	×
WADS (Kurup and Bos, 2021)	point-wise	✓
NuScenes-LiDARseg (Caesar et al., 2020)	point-wise	×
RADIATE (Sheeny et al., 2021)	2D bounding boxes	✓
Zendar (Mostajabi et al., 2020)	2D bounding boxes	✓
Astyx (Meyer and Kuschk, 2019)	3D bounding boxes	×
Waymo (Sun et al., 2020)	3D bounding boxes	×
Once (Mao et al., 2021)	3D bounding boxes	×
VOD (Palffy et al., 2022)	3D bounding boxes	×
CADC (Pitropov et al., 2021)	**3D bounding boxes**	✓

Table 2
The details of datasets used in our experiments.

Dataset	Publish Year	Scenes	Classes	Area	Most weather	Goal
CADC	2020	7000 frames	Car, truck, pedestrian, animal	Canada	Rain, Snow	Performance under adverse weather
KITTI	2011	7481 frames	Car, truck, pedestrian, van	Germany	Sunny	Generalization ability under clear weather

Fig. 7. Examples of CADC dataset.

where W_l denotes the line loss, which is set to 0.25 based on experiments in Section 4.4.4.

4. Experimental results and analysis

4.1. Experimental datasets

4.1.1. CADC dataset

To validate the efficacy of GIC-Net in adverse weather conditions, our experimental dataset must fulfill two simultaneous requirements: 3D bounding box labeling and comprehensive coverage of adverse weather scenarios. In recent years, several annotated autonomous vehicle datasets incorporating LiDAR scans have been made available for perception tasks (Table 1). However, only the CADC autonomous vehicle dataset meets our requirements. Consequently, we evaluate GIC-Net using the CADC dataset (Table 2), which was collected during winter in Waterloo Region, Canada, encompassing continuous snowfall or rainfall following snowfall throughout data collection (Fig. 7). At the same time, the rain and snow on this dataset are not serious enough to make driving difficult and are typical routine rain and snow weather that can be driven. This dataset comprises 7000 frames with diverse object categories including "Car," "Truck," and "Pedestrian." The testing set constitutes approximately 20 % of the CADC dataset, while the remaining 80 % is allocated as training data, which is further divided into a training set and a validation set at a ratio of 5:1 following setting in previous works (Liu et al., 2020; Yu et al., 2022). All LiDAR data is captured with a 32-beam LiDAR. Based on KITTI metric guidelines (Geiger et al., 2013) considering object size, occlusion state, and truncation level, three levels are defined (L1, L2, and L3 corresponding to easy, moderate, and hard respectively, in the KITTI benchmark). Following official evaluation metrics from the KITTI benchmark, the detection performance is assessed by computing the average precision (AP) for each category in terms of its 3D representation. Given that the LiDAR detection range can reach up to 200 m within CADC (Pitropov et al., 2021), all the networks use the detection range to [(0, 192), (-192, 192), (-9.6, 9.6)] meters.

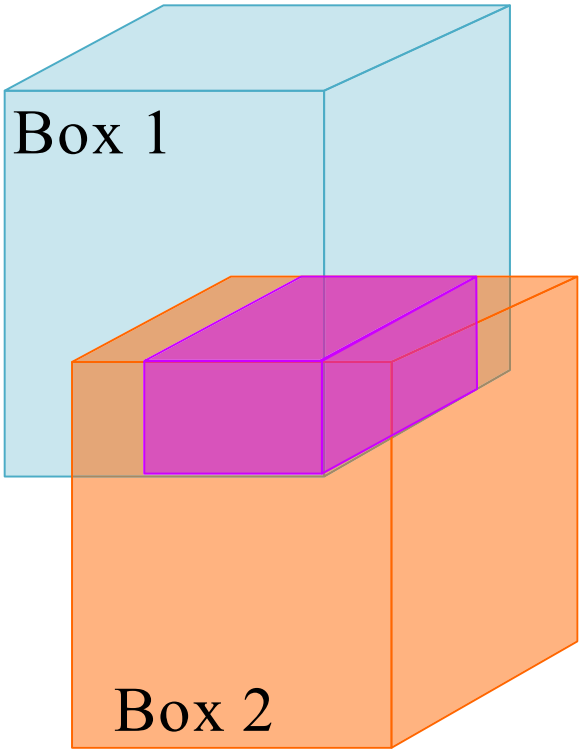

Fig. 8. Illustration of IOU. Box 1: ground-truth box; Box 2: predicted box.

4.1.2. KITTI dataset

To test the generalization ability of our network, we also evaluated it on the KITTI dataset (Geiger et al., 2013). KITTI is a dataset for autonomous vehicles, which contains 3712 training frames and 3769 validation frames for cars, pedestrians, and cyclists. All LiDAR data is acquired with a 64-beam LiDAR in Karlsruhe, Germany. Different from the CADC dataset, which annotates point clouds in both driving and parking areas, the KITTI dataset only focuses on driving areas. Due to the differences in LiDAR beam count, detection range, and weather, the point cloud density in the KITTI data set is higher than that in the CADC dataset. In order to evaluate the pre-trained GIC-Net effectively, we employed the whole KITTI validation dataset. For this evaluation, we set the detection range to [(0, 70.4), (-40, 40), (-3, 1)] meters.

4.2. Experimental details

4.2.1. Data augmentation

Following the standard of SECOND (Yan et al., 2018), all point clouds are augmented for better performance through flipping, rotating, and scaling. Every frame point cloud is randomly flipped along the x-axis, and the rotation angle is a random number of $\left(-\frac{\pi}{4}, \frac{\pi}{4}\right)$, and scaling is between $(0.95, 1.05)$. Note that in transfer learning (Section 4.5) we keep data augmentation consistent.

4.2.2. Evaluation metrics

3D average precision (AP) based on Intersection over Union (IOU) is utilized to evaluate the detection performance of the three different levels. As shown in Fig. 8, the blue, orange, and purple colors represent the ground-truth box B_g, the predicted box B_p, and the overlap box B_o between the ground-truth and the predicted result, respectively. Thus, IOU is calculated as follows:

$$IOU = \frac{Area(B_o)}{Area\left(B_g + B_p - B_o\right)} \tag{15}$$

Since the CADC dataset does not provide official IOU metrics, we follow the IOU threshold setting in the ONCE dataset (Mao et al., 2021) (0.7, 0.7, 0.3 for car, truck, and pedestrian) to determine whether the predicted box is positive. AP with 40 recall positions has been widely used as an evaluation metric (Wu et al., 2022; Xia et al., 2023), and 3D detection performances of different methods are evaluated and compared by the 3D AP with 40 recall positions. Finally, following the evaluation metric in SIEV-Net (Yu et al., 2022), the mean values of AP (mAP) at three levels are also utilized to evaluate the overall performance.

Table 3
Performance comparison on the CADC validation set.

Model	Car(R40)(%)				Truck(R40)(%)				Pedestrian(R40)(%)			
	L1	L2	L3	mAP	L1	L2	L3	mAP	L1	L2	L3	mAP
PointPillars	20.91	20.49	22.33	21.24	34.01	26.72	33.38	31.37	20.93	14.35	4.89	13.39
SECOND	26.39	21.97	23.73	24.03	35.97	27.39	25.68	29.68	17.63	9.47	4.46	10.52
PointRCNN	50.13	46.17	48.29	48.20	46.81	38.7	41.33	42.28	31.53	23.53	12.17	22.42
CenterPoints	27.31	24.27	26.69	26.09	47.98	40.45	39.27	42.57	16.62	12.04	6.85	11.84
Voxel-RCNN	51.95	50.11	51.15	51.07	39.92	35.06	37.93	37.64	17.7	13	7.78	12.83
CG-SSD	22.51	21.16	25.09	22.92	51.19	40.79	42.12	44.70	18.45	12.6	6.83	12.63
GIC-Net(Ours)	**71.54**	**65.2**	**65.29**	**67.34**	**62.3**	**54.02**	**56.91**	**57.74**	**50.18**	**38.74**	**18.33**	**35.75**

Table 4
Performance comparison on the CADC testing set.

Model	Car(R40)(%)				Truck(R40)(%)				Pedestrian(R40)(%)				Speed(ms)
	L1	L2	L3	mAP	L1	L2	L3	mAP	L1	L2	L3	mAP	
PointPillars	27.16	23.88	24.03	25.02	45.12	35.26	45.7	42.03	8.27	7.8	10.49	8.85	**35**
SECOND	31.04	22.9	24.94	26.29	30.03	27.82	31.45	29.77	16.55	13.47	17.11	15.71	42
PointRCNN	60.62	52.68	54.27	55.86	57.97	47.63	48.84	51.48	28.83	23.54	24.58	25.65	101
CenterPoints	37.86	28.8	30.85	32.50	44.09	37.85	45.81	42.58	14.88	12.4	16.33	14.54	50
Voxel-RCNN	55.53	50.43	51.35	52.44	43.5	39.94	46.23	43.22	16.72	13.24	17.11	15.69	56
CG-SSD	33.06	26.09	28.02	29.06	55.51	40.74	47.78	48.01	19.33	12.55	14.64	15.51	66
GIC-Net (Ours)	**75.36**	**66.27**	**65.91**	**69.18**	**60.44**	**51.55**	**56.45**	**56.15**	**39.98**	**33.26**	**35.03**	**36.09**	75

4.2.3. Experimental environment and optimization

To train GIC-Net for the 3D object detection, all experiments are conducted on a computer equipped with one Intel Core i9-12900 K central processing unit (CPU) @3.2 GHz, 32 GB of random-access memory (RAM) and one NVIDIA RTX3090 GPU with 24 GB of graphics memory. The environments include Python 3.8.16, CUDA 11.1.105, and cuDNN 8.7.0.

The optimization process is implemented by adopting adaptive moment estimation (Adam) (Loshchilov and Hutter, 2019). All methods are trained for 80 epochs with a batch size of two. During training, we set the momentum range from 0.95 to 0.85, the learning rate to 0.01, and weight decay to 0.001.

4.3. Experimental results and analysis

4.3.1. Validation set

To show the effectiveness of the GIC-Net, we compare it with several state-of-the-art methods, including PointPillars (Lang et al., 2019), SECOND (Yan et al., 2018), PointRCNN (Shi et al., 2019), CenterPoints (Yin et al., 2021), Voxel-RCNN (Deng et al., 2021) and CG-SSD (Ma et al., 2022). To be fair, we keep the parameters consistent with our network during the training stage with OpenPCDet (OpenPCDet-Development-Team, 2020) and take the best performance of each method. Table 3 shows the 3D object detection performances of GIC-Net and comparison methods on the validation set.

Our network achieves the mAP of 67.34 %, 57.74 %, and 35.75 % for the “Car”, “Truck”, and “Pedestrian” categories, respectively. Compared to PointPillars, SECOND, PointRCNN, CenterPoints, Voxel-RCNN, and CG-SSD models, GIC-Net exhibits significant improvements in terms of mAP for the “Car” category with enhancements ranging from 16.27 % to as high as 46.1 %. This improvement can be attributed to our GLC module, which incorporates location information as a constraint by utilizing down-sampled point clouds near objects as input to the network.

For truck and pedestrian detection, the AP of these two categories is lower than that of cars. This is because cars in the training set have ten times as many samples as trucks and pedestrians. And the number of point clouds on the surface of pedestrians is much smaller than that of trucks and cars, which makes it difficult for the network to learn the shape and orientation from a few points. However, compared with other methods, our network still achieves 13.04 %∼28.06 % and 13.33 %∼25.23 % improvements in 3D mAP on the “Truck” and “Pedestrian” categories, respectively. This is mainly because the LGFC module extracts the geometric feature information of the lines that make up the 3D bounding box and fuses them into our network.

4.3.2. Testing set

Table 4 presents the comparative results of the testing set, demonstrating that our network outperforms all other state-of-the-art methods using a single-frame LiDAR point cloud. Our network achieves superior performance on the testing set with 3D mAP scores of 69.18 %, 56.15 %, and 36.09 % for “Car”, “Truck” and “Pedestrian” classes, respectively. It surpasses all models by at least 13.32 %, 4.67 %, and 10.44 % in terms of mAP for car, truck, and pedestrian categories. Additionally, our network achieves an inference speed of approximately 13FPS (75 ms), which is considered the moderate level of computation time among the compared methods.

Fig. 9 visually illustrates the comparative performances of predicted bounding boxes and ground-truth using different methods,

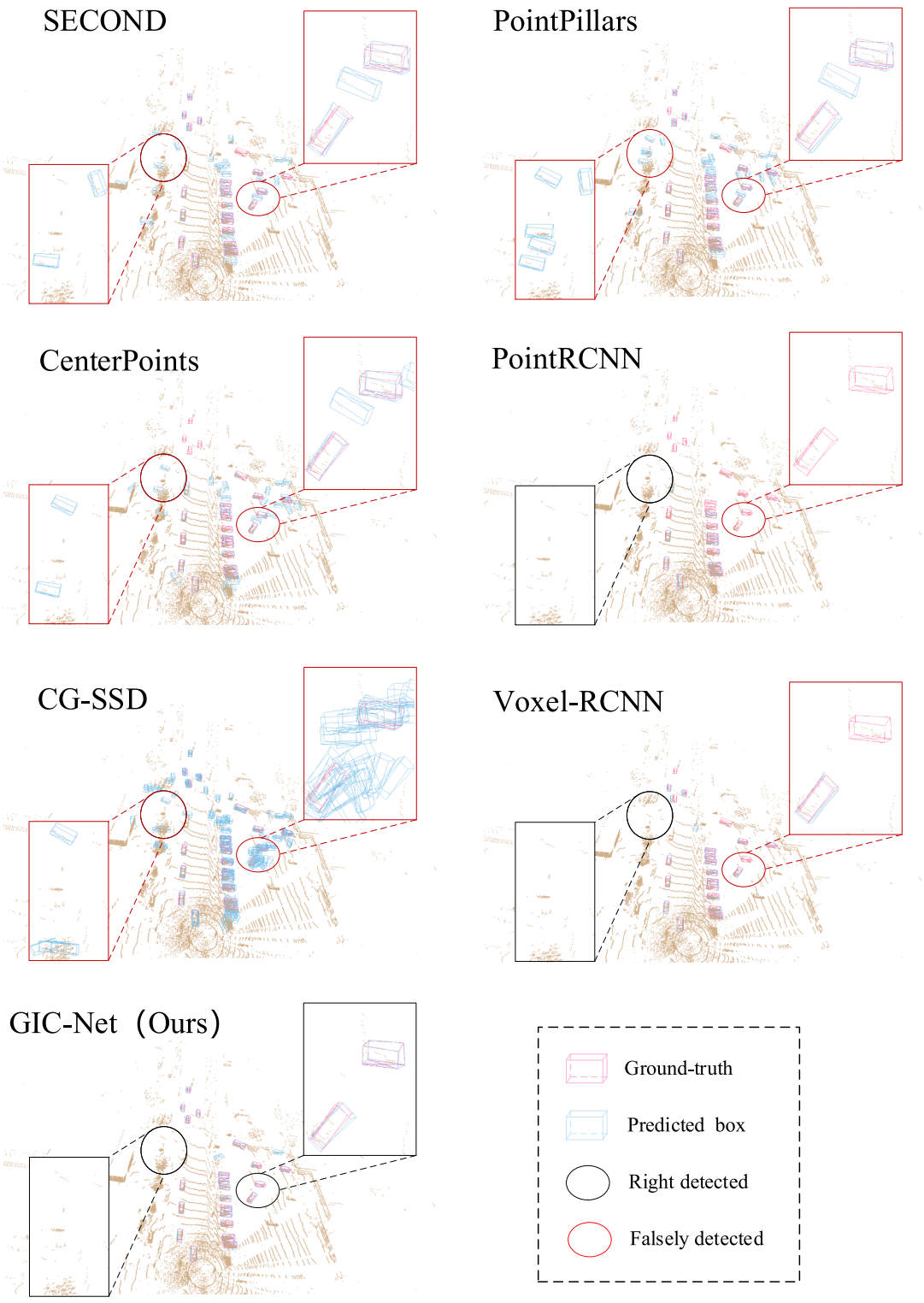

Fig. 9. Visualization of object detection. The ground-truth and predicted boxes are in red and blue, the red ovals indicate detection errors, and the black ovals indicate no error. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

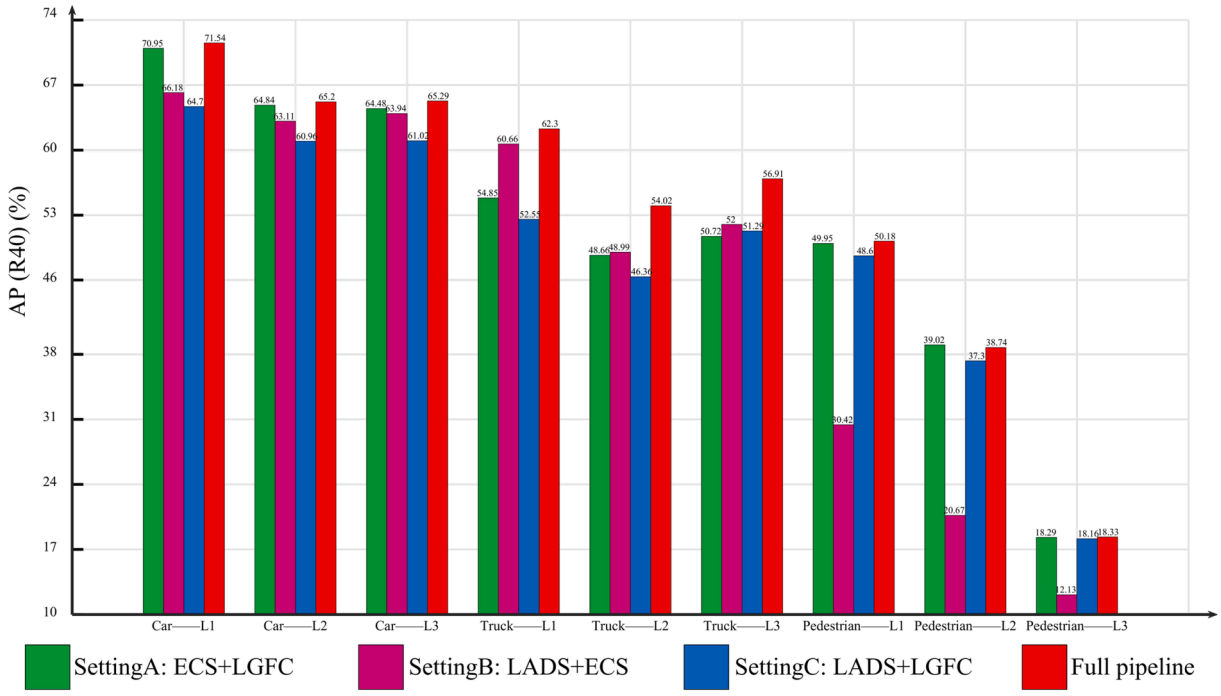

Fig. 10. Effects of different modules on the validation set.

Table 5
Extension to some BEV map-based models with LCFC module on the CADC validation set calculated by 40 recall positions.

Model	Paradigm	Car(mAP)(%)	Truck(mAP)(%)	Pedestrian(mAP)(%)
SECOND	Voxel-based	24.03	29.68	10.52
SECOND + LGFC		**31.39(+7.36)**	**30.01(+0.33)**	**19.28(+8.76)**
PointPillars	Pillar-based	21.24	31.37	13.39
PointPillars + LGFC		**21.33(+0.09)**	**36.82(+5.45)**	**23.47(+10.08)**

where GIC-Net demonstrates superior results in terms of 3D object detection compared to other networks. As depicted in Fig. 9, PointPillars, SECOND, CenterPoints, and CG-SSD exhibit false positive (FP) cases where non-car objects are incorrectly detected as cars. PointRCNN and Voxel-RCNN result in false negative (FN) cases where many cars are not detected as cars. This can be attributed to two main factors. Firstly, the sparsity and incompleteness of point clouds on object surfaces under adverse weather conditions lead to incorrect estimation of the direction and dimensions of 3D objects (Ma et al., 2022). Secondly, the presence of raindrops or snow particles makes feature extraction more challenging compared to clear weather conditions, and these aforementioned networks are primarily designed for clear weather scenarios. Benefiting from its geometric information constraint regarding location and broken lines, in both driving zones on the left side and parking zones on the right side in Fig. 9, our network exhibits significantly better performance than all other methods.

4.4. Ablation studies

4.4.1. Module effectiveness

To validate the superiority of the proposed module, we conduct ablation experiments on different representation modules. Our network comprises GLC module and LGFC module, where GLC module consists of two main components: the LADS method and the ECS strategy. In setting A, we replace the LADS method with normal down-sampling layers in SECOND (Yan et al., 2018). In setting B, we directly delete the LGFC module. In setting C, we replace the ECS strategy with FPS (Shi et al., 2020) method. Fig. 10 shows the quantitative results of the above experiments.

Fig. 10 presents quantitative results from these experiments. As depicted in Fig. 10, compared to settings A, B, and C, our full pipeline demonstrates significantly improved performance. Removing the LADS module leads to a drop in AP for trucks due to their requirement for large receptive fields, which is also the function of our LADS module. Eliminating the LGFC module results in a dramatic decrease in 3D object detection performance, particularly for small objects like pedestrians; this highlights how line feature constraints enhance detection ability across all categories within our network. Lastly, removing the ESC module causes a reduction in AP for all categories primarily because direct down-sampling from raw point clouds inevitably includes numerous foreground points

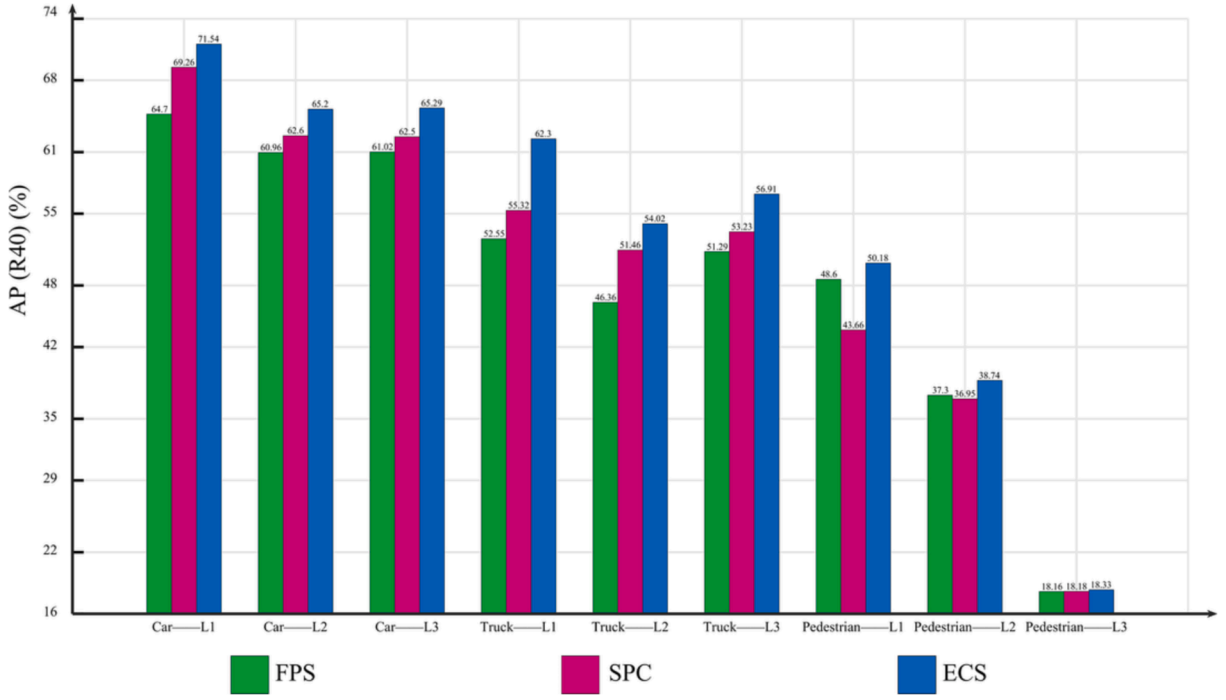

Fig. 11. Effects of different keypoints down-sampling methods on the validation set.

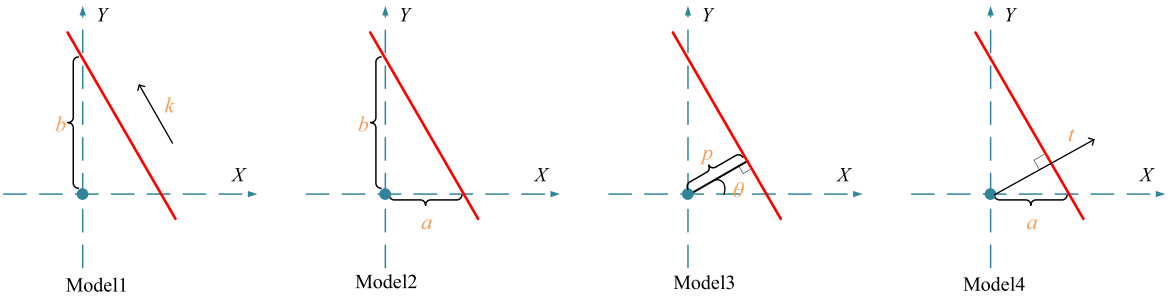

Fig. 12. Four different line models.

such as rain and snow particles; however, incorporating geometric location information effectively focuses background keypoints around objects, thereby enhancing effective feature aggregation capability within our network.

Extension of LGFC module as the plug-in to other models. In this paper, we propose an LGFC module, which consists of 2D convolutional modules and can be applied to some BEV-based 3D object detection models as a plug-in. Table 5 demonstrates the increase in performance for some state-of-the-art baselines, including two BEV-based methods: a Voxel-based model (i.e., SECOND) and a Pillar-based model (i.e., PointPillars). The experimental results show that our proposed LGFC module can bring + 0.09 % ~+10.08 % mAP to the original model as a plug-in. Especially, after plugging the LGFC module, the detection ability of the models for small objects (i.e., Pedestrians) is greatly improved. Thus, it can be plugged into some BEV-based models and has good generalizability.

4.4.2. Keypoints down-sampling

Experiments are conducted with different down-sampling methods to show the effectiveness of our proposal ECS strategy. Consequently, the following three ablation studies are conducted: setting A using FPS (Qi et al., 2017; Shi et al., 2020); setting B using a sectorized proposal-centric (SPC) sampling method (Transportation, 2019); setting C keeping our ECS strategy.

Fig. 11 demonstrates that our ECS strategy outperforms other strategies in terms of AP performance across all categories, thereby validating our assertion that our ECS strategy effectively generates more representative keypoints by concentrating a smaller number of keypoints within object-adjacent regions through location constraints. Moreover, employing direct down-sampling techniques such as FPS fails to enhance the performance significantly due to the inevitable introduction of numerous invalid point clouds.

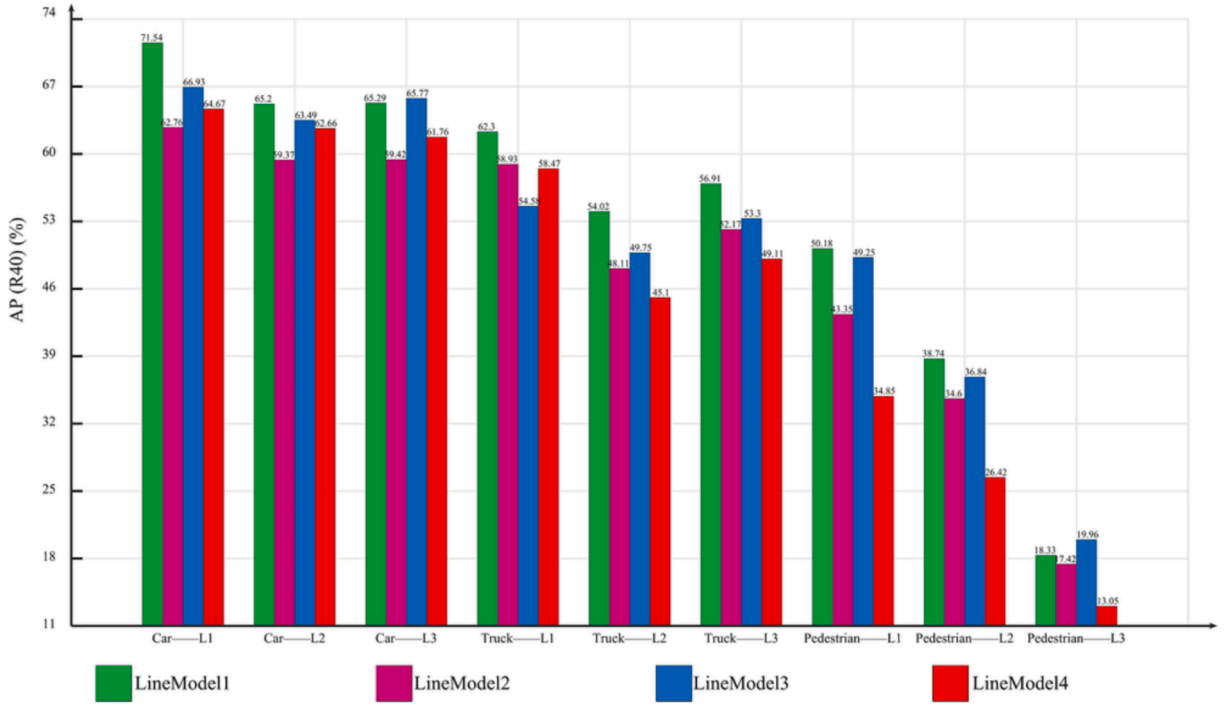

Fig. 13. Effects of different line models on the validation set.

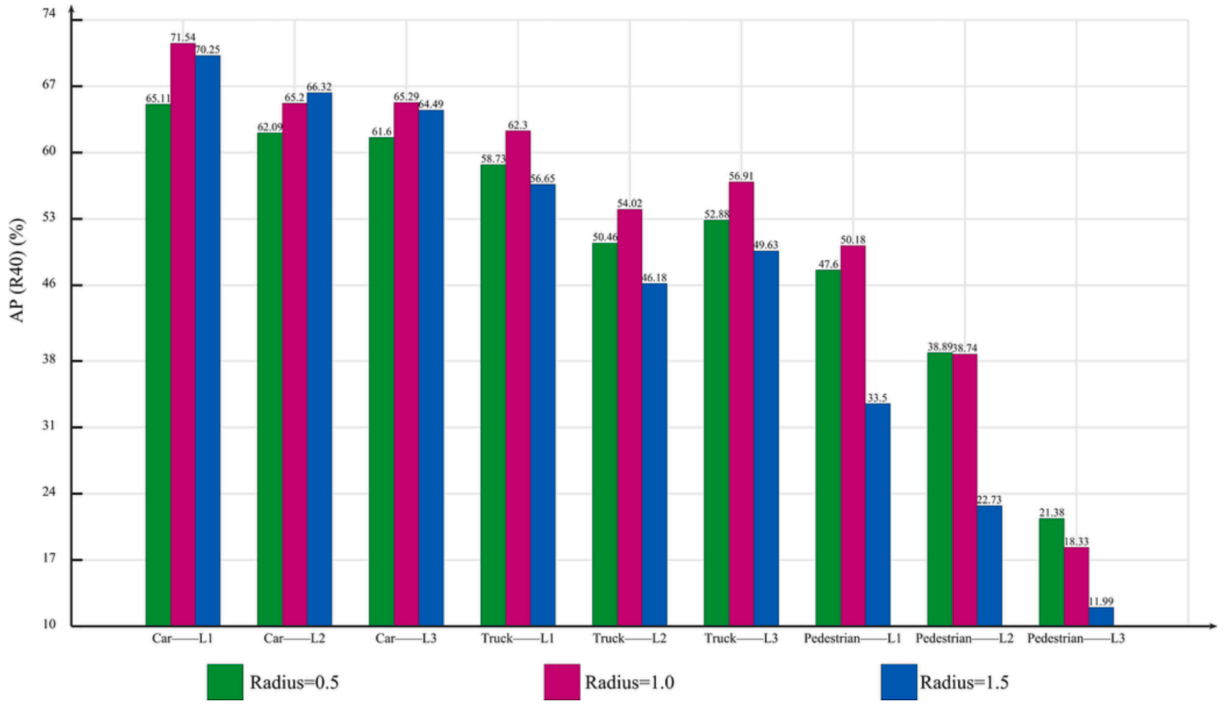

Fig. 14. Effects of different extended radii on the validation set.

4.4.3. Line features

We have introduced line features in Section 3.2; we further investigate the effects of different forms of line models in the LGFC. Specifically, we conduct the following ablation experiments for line features with four different line models (Fig. 12) as follows:

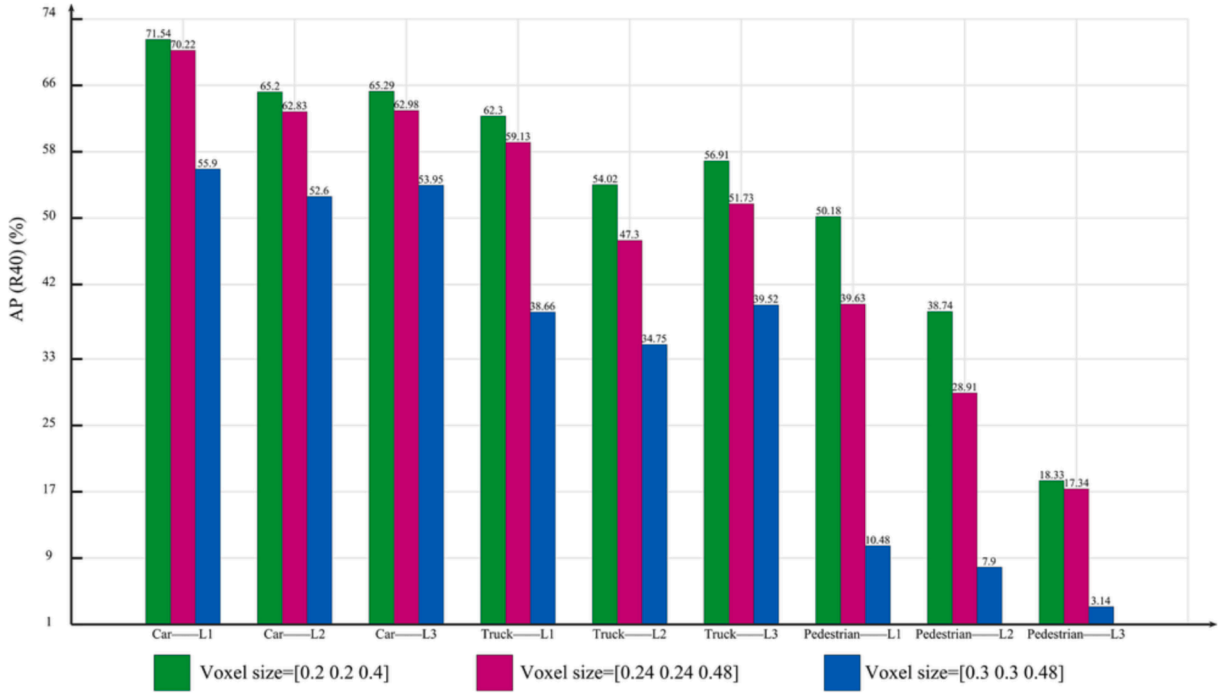

Fig. 15. Effects of different voxel sizes on the validation set.

$$model1 : y = kx + b \tag{16}$$

where k denotes the slope of the line, b denotes the intercept of the line in the y-axis.

$$model2 : \frac{x}{a} + \frac{y}{b} = 1 \tag{17}$$

where a, b denote the intercept of the line in the x-axis and y-axis, respectively.

$$model3 : x \cdot cos\theta + y \cdot sin\theta - p = 0 \tag{18}$$

where p denotes the shortest distance from the original point to the line, θ denotes the angle between the x-axis and the perpendicular from the origin to the line.

$$model4 : x = ty + a \tag{19}$$

where a denotes the intercept of the line in the x-axis, $t = -\frac{1}{k}$.

Fig. 13 demonstrates that line model 1 exhibits superior performance. The conducted ablation study provides evidence supporting the notion that incorporating the slope-intercept form of the line as constraint enhances the network's capacity to learn more precise bounding boxes for 3D objects. This improvement can be attributed to the implicit representation of both line length and angle between the object and road direction by parameters k and b.

4.4.4. Key hyperparameters

To evaluate the performance of GIC-Net, we conduct experiments by adjusting various key parameters on the validation set. In order to ensure fairness, only one parameter is modified at a time while keeping all other parameters constant.

A. Extended radius in GLC module

We initially investigate the impact of extended radius in the GLC module and observe that setting it to 1 results in improved performance (Fig. 14). When the radius is smaller (e.g., 0.5), our network has fewer points available for feature aggregation from voxels to points, whereas larger radius (e.g., 1.5) introduces redundant point clouds that interfere with feature extraction, thereby reducing overall performance.

B. Voxel size in GLC module

Voxel size ensures that the sizes of feature maps can be correctly down-sampled (Yan et al., 2018). Our findings indicate that better performance is achieved when using a voxel size of [0.2, 0.2, 0.4] (Fig. 15). Other voxel sizes such as [0.24, 0.24, 0.48] could lead to potential loss of local feature information and consequently decrease network performance.

C. Line weight of LGFC module in the loss function

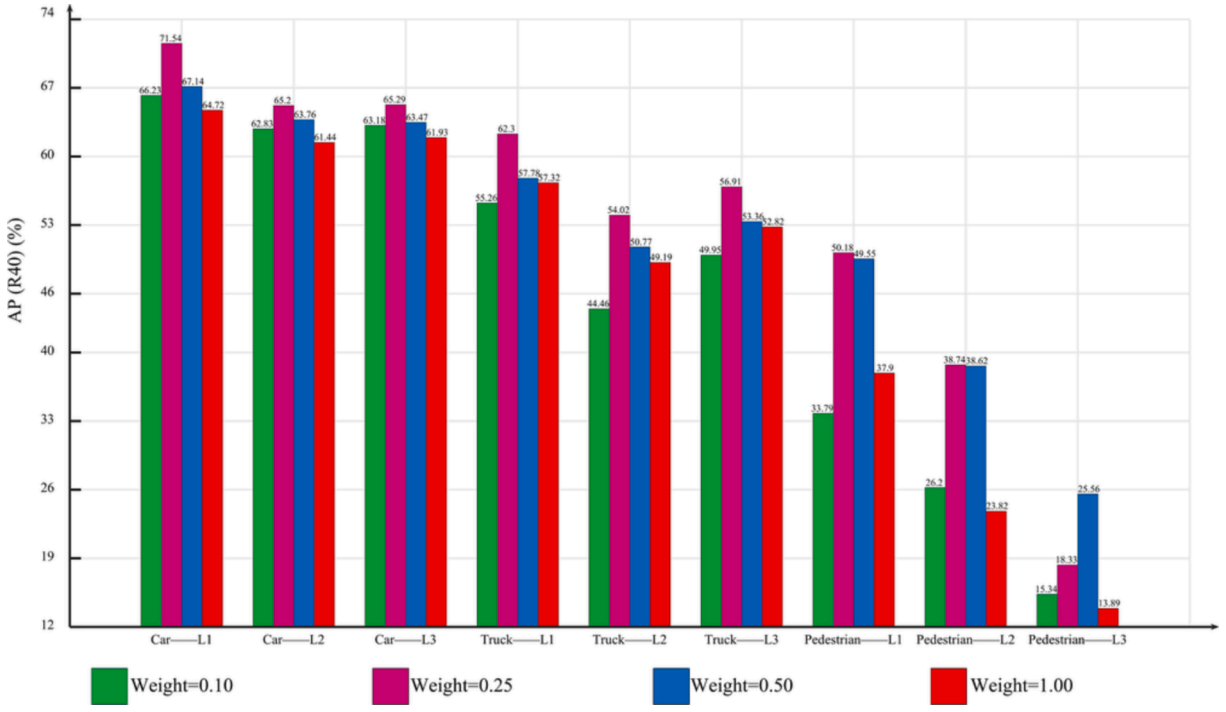

Fig. 16. Effects of different line weights on the validation set.

Table 6
Generalization ability comparison results with training on the CADC dataset and testing on the KITTI dataset.

Model	Car(R40) (%) 3D				Car(R40) (%) BEV			
	L1	L2	L3	3D mAP	L1	L2	L3	BEV mAP
PointPillars	0	0	0	0.00	0	0	0	0.00
SECOND	0	0	0	0.00	0	0	0	0.00
PointRCNN	3.55	3.14	1.36	2.68	3.75	3.71	1.46	2.97
CenterPoints	0.01	0.03	0.03	0.02	0.03	0.07	0.07	0.06
Voxel-RCNN	0	0	0	0.00	0	0	0	0.00
CG-SSD	9.95	13	13.58	12.18	36.42	39.54	40.71	38.89
GIC-Net (Ours)	56.21	59.96	58.7	58.29	63.7	67.64	67.43	66.26

The loss weight of line weight represents the weight and contribution of the LGFC module in the overall network. To determine an appropriate balance for this contribution weight, we examine its effects on the validation set. Our results demonstrate superior performance when setting the line weight equal to 0.25 (Fig. 16).

4.5. Generalization ability on clear weather

Since clear weather (e.g., sunny days) is often more common than adverse weather in cities, in this section, we assess the cross-dataset generalization capacity of our network compared to other methods, from adverse weather to clear weather. The aim is to explore the generalization ability of a network trained on a specific dataset (adverse weather) to generalize data of common types. Therefore, we adopt the validation set of the KITTI dataset collected under clear weather in Germany to validate the model trained in rain and snow scenarios in Canada. Only the car detection task is tested when validating the pre-trained GIC-Net on a different dataset. This is because cars are objects that appear in large numbers on almost all urban roads and are often of different shapes and sizes; thus, they are more challenging and can reflect the performance of object detection. Results with IOU larger than 50 % are regarded to be detected correctly.

Table 6 presents generalization ability comparison results obtained by training on the CADC dataset and testing on the KITTI dataset. For all methods, we train the model on the CADC dataset and perform the testing on the CADC dataset without fine-tuning. In addition to evaluating 3D AP, following previous works (Hoang Duy and Kim, 2022; Ma et al., 2022). we also report BEV AP results on the validation set using 40 recall thresholds. It is important to note that no retraining is performed on the KITTI dataset. As anticipated, performance is comparatively lower than when the network is evaluated on the CADC validation set. Due to the differences in road scenarios, weather conditions, detection range and LiDAR beams between KITTI and CADC datasets, the distribution of object point

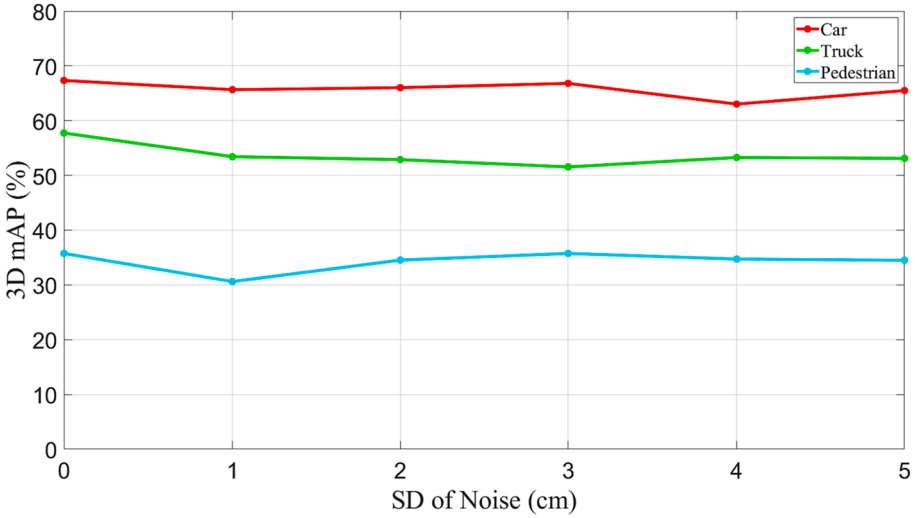

Fig. 17. Robustness test of detection performance with different noise levels.

clouds in the two datasets is quite different, which makes it difficult to accurately predict car dimensions and orientation angles for new scenarios. However, our GIC-Net outperforms current state-of-the-art methods with an improvement of at least 46.11 % in 3D mAP and 27.37 % in BEV mAP. Except for our model, almost all other deep-learning-based methods (i.e. PointPillars, SECOND, PointRCNN) fail to predict the bounding boxes in totally different driving scenarios. This is because all these networks learn features from points or voxels or pillars in the original dataset, and these features may result in overfitting of the model to the training data and lead to degradation of generalization ability in a new environment. In contrast, some features of CG-SSD are based on corners from bounding boxes, which has little correlation with the changes in point cloud distribution caused by scenario and weather transformations. In other words, training with features from bounding boxes avoids the degradation of model generalization ability, which is better illustrated by our model. Leveraging the advantages our LGFC module offers for line features from 3D bounding boxes, our network consistently achieves superior results compared to other approaches.

In addition, to demonstrate the effectiveness of our model in clear weather (i.e., trained in clear weather and validated in clear weather) we also test the performance on the KITTI val set with AP calculated by recall 11 positions. Our model achieves the 3D AP of 89.20 %, 79.18 %, and 78.62 % on “Easy”, “Moderate”, and “Hard” subsets for the “Car” category, which shows good performance in clear weather conditions.

4.6. Robustness test against noise

As mentioned previously, the point cloud data obtained under extremely severe rain and snow weather by a LiDAR scanner contains noise. To verify the robustness and generalization of our model to point clouds with different levels of noise under extreme weather, we add random noise in the point cloud scene. Following a similar study (Zhu et al., 2022), random noise with a standard deviation (SD) from 0.01 m to 0.05 m is added to each point in the raw point cloud to simulate extreme rain and snow interference. As demonstrated in Fig. 17, with the increase in the level of noise, the detection performance of the network declines slightly but still maintains a high level of accuracy.

4.7. Discussion

Ablation studies have demonstrated the efficacy of our network’s geometric location information and geometric line feature information (Section 4.4.1). Additionally, in adverse weather conditions, there is a substantial presence of rain and snow particles around objects, which may introduce redundant data. Our model exhibits improved performance when utilizing the geometric location information as a constraint to filter out invalid point clouds (Section 4.4.2). Adverse weather significantly reduces the LiDAR point cloud on the object’s surface, thereby increasing the network’s challenge in accurately regressing 3D bounding boxes. However, considering that an object comprises multiple lines in both 3D and BEV views, the slopes and intercepts of these lines can implicitly convey the orientation and length details of the bounding box (Section 4.4.3). Consequently, incorporating geometric feature information from broken lines into our network yields superior results.

However, it is important to be aware of certain findings. As previously explained, we employ the LGFC module to acquire information about broken lines in order to reconstruct a complete 3D bounding box from an incomplete object surface (e.g., the edge of a car with only a few dozen point clouds). Nevertheless, on the road, due to factors such as rain, snow, and vehicle occlusions, some object surfaces may exhibit similar morphology to that of a damaged vehicle surface, for instance, containers positioned alongside the road or balconies with linear shapes. In such cases, false positive cases can occur with the LGFC module mistakenly detecting objects outside

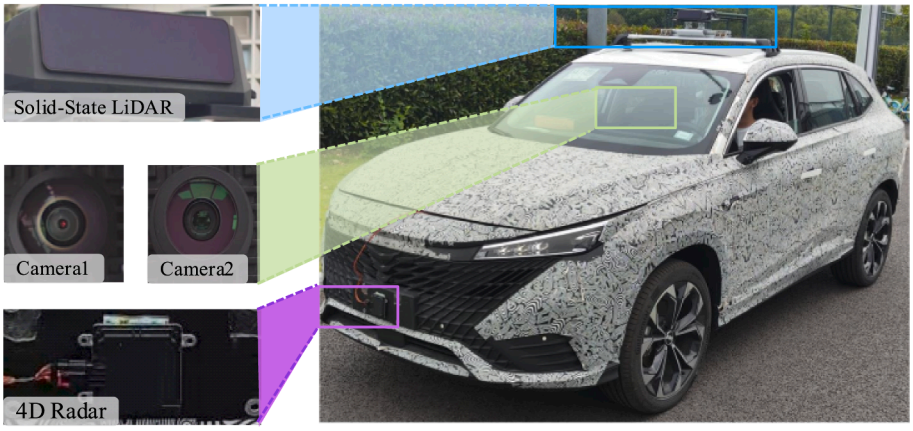

Fig. 18. The platform we are building.

the driving zone as vehicles. To address this issue effectively and enhance performance further, incorporating semantic prior knowledge specific to road scenarios (e.g., HD maps) becomes imperative. By leveraging semantic information pertaining to the surrounding environment of roads, both accuracy and robustness in 3D object detection can be significantly improved (Huang et al., 2023). However, given that structured HD maps are more difficult to obtain than conventional data sources like images and IMU data, integrating online 3D mapping techniques (Cong et al., 2023) could potentially serve as a viable solution to overcome this limitation in the future.

5. Conclusion and future work

Due to the evident limitations of LiDAR performance under adverse weather, the point cloud obtained by LiDAR presents challenges such as sparsity, incompleteness, and particle interference. These challenges pose significant difficulties in accurately inferring 3D bounding boxes from incomplete and noisy point clouds. In this paper, we propose a novel GIC-Net that effectively leverages geometric information as the constraint for 3D object detection from single frame LiDAR point cloud in adverse weather conditions. The network is trained and evaluated on the CADC autonomous vehicle dataset, achieving state-of-the-art results. Specifically, our proposed GLC module utilizes location information constraints to filter out rain and snow particles, enabling the integration of more representative keypoints while preserving accurate location information. Additionally, within the 3D backbone of the GLC module, we employ a linear adaptive down-sampling method along with additional down-sampling layers to ensure sufficient receptive fields. To further enhance accuracy in detecting 3D bounding boxes from sparse and incomplete point clouds, we introduce the LGFC module, which constructs line feature constraints. Experimental results demonstrate that GIC-Net exhibits strong 3D object detection capabilities under adverse weather conditions compared to current state-of-the-art methods on the CADC autonomous vehicle dataset with high mAP scores of 69.18 %, 56.15 %, and 36.09 % for car, truck, and pedestrian classes, respectively. Moreover, our proposed network demonstrates superior generalization abilities when compared to other existing methods. Furthermore, future work could consider several improvements. Considering that the majority of datasets of autonomous vehicles do not contain much adverse weather (Zhang et al., 2023b), we are currently developing a platform equipped with multiple sensors (Fig. 18) to facilitate data collection under diverse weather conditions. In addition, since this paper attempts to evaluate generalization ability through cross-dataset, we will take transfer learning tasks into account through fine-tuning in our future research.

CRediT authorship contribution statement

Yuanfan Qi: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Chun Liu:** Funding acquisition, Methodology, Validation. **Marco Scaioni:** Validation, Supervision. **Yanyi Li:** Formal analysis. **Yihong Qiao:** Visualization. **Xiaolong Ma:** Visualization. **Hangbin Wu:** Validation. **Keke Zhang:** Resources. **Dazhi Wang:** Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Major Program of the National Natural Science Foundation of China (Grant No. 42130106) and

Industry-University-Research Project of Shanghai Automotive Industry Science and Technology Development Foundation (No. 2202). We thank all anonymous reviewers for their constructive suggestions. We are also grateful to Akram Akbar and Shoujun Jia for their suggestions in paper writing. We are thankful for Yijun Liu for his assistance in the operation of the computer.

References

Administration, N.O.a.A., 2021. Getting traction: Tips for traveling in winter weather.
Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J., 2019. Semantickitti: A dataset for semantic scene understanding of lidar sequences, Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9297–9307.
Bijelic, M., Gruber, T., Ritter, W., 2018. A benchmark for LiDAR sensors in fog: Is detection breaking down? IEEE Intell. Veh. Sympos. (IV) 760–767.
Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O., 2020. nuScenes: A multimodal dataset for autonomous driving. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11618–11628.
Changnon, S.A., 1996. Effects of summer precipitation on urban transportation. Clim. Change 32, 481–494. https://doi.org/10.1007/BF00140357.
Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77-85.
Charron, N., Phillips, S., Waslander, S.L., 2018. De-noising of Lidar Point Clouds Corrupted by Snowfall, 2018 15th Conference on Computer and Robot Vision (CRV), pp. 254-261.
Chaturvedi, S.S., Zhang, L., Yuan, X., 2022. Pay "Attention" to Adverse Weather: Weather-aware Attention-based Object Detection, 2022 26th International Conference on Pattern Recognition (ICPR), pp. 4573-4579.
Chen, Y., Liu, S., Shen, X., Jia, J., 2019. Fast Point R-CNN, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9774-9783.
Chen, Y., Li, Y., Zhang, X., Sun, J., Jia, J., 2022. Focal sparse convolutional networks for 3D object detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5418–5427.
Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J., 2023. VoxelNeXt: fully sparse VoxelNet for 3D object detection and tracking. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 21674–21683.
Commission, E., 2021. Road safety in the European Union.10.2832/060333.
Cong, Y., Chen, C., Yang, B., Liang, F., Ma, R., Zhang, F., 2023. CAOM: Change-aware online 3D mapping with heterogeneous multi-beam and push-broom LiDAR point clouds. ISPRS J. Photogramm. Remote Sens. 195, 204–219. https://doi.org/10.1016/j.isprsjprs.2022.11.017.
Dai, A., Diller, C., Niessner, M., 2020. SG-NN: Sparse generative neural networks for self-supervised scene completion of RGB-D scans. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 846–855.
Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H., 2021. Voxel R-CNN: Towards high performance voxel-based 3D object detection. AAAI Conf. Artif. Intell. 1201–1209.
Filgueira, A., González-Jorge, H., Lagüela, S., Díaz-Vilariño, L., Arias, P., 2017. Quantifying the influence of rain in LiDAR performance. Measurement 95, 143–148. https://doi.org/10.1016/j.measurement.2016.10.009.
Gandomani, R., Mohamed, M., Amiri, A., Razavi, S., 2022. System optimization of shared mobility in suburban contexts. Sustainability 14. https://doi.org/10.3390/su14020876.
Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The KITTI dataset. Int. J. Robot. Res. 32, 1231–1237. https://doi.org/10.1177/0278364913491297.
Gong, Z., Lin, H., Zhang, D., Luo, Z., Zelek, J., Chen, Y., Nurunnabi, A., Wang, C., Li, J., 2020. A frustum-based probabilistic framework for 3D object detection by fusion of LiDAR and camera data. ISPRS J. Photogramm. Remote Sens. 159, 90–100. https://doi.org/10.1016/j.isprsjprs.2019.10.015.
Gunn, K., Marshall, J., 1958. The distribution with size of aggregate snowflakes. Atmos. Sci. 15, 452–461.
Hahner, M., Sakaridis, C., Bijelic, M., Heide, F., Yu, F., Dai, D., Van Gool, L., 2022. LiDAR snowfall simulation for robust 3D object detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16343–16353.
Hasirlioglu, S., Doric, I., Lauerer, C., Brandmeier, T., 2016. Modeling and simulation of rain for the test of automotive sensor systems. In: 2016 IEEE Intelligent Vehicles Symposium (IV), pp. 286–291.
Heinzler, R., Piewak, F., Schindler, P., Stork, W., 2020. CNN-based lidar point cloud de-noising in adverse weather. IEEE Rob. Autom. Lett. 5, 2514–2521. https://doi.org/10.1109/lra.2020.2972865.
Hoang Duy, L., Kim, G.-W., 2022. AEC3D: An efficient and compact single stage 3D multiobject detector for autonomous driving. IEEE Trans. Intell. Transp. Syst. 23, 23422–23432. https://doi.org/10.1109/tits.2022.3195633.
Huang, S., Cai, G., Wang, Z., Xia, Q., Wang, R., 2022. SSA3D: Semantic segmentation assisted one-stage three-dimensional vehicle object detection. IEEE Trans. Intell. Transp. Syst. 23, 14764–14778. https://doi.org/10.1109/tits.2021.3133476.
Huang, X., Wang, P., Cheng, X., Zhou, D., Geng, Q., Yang, R., 2020. The ApolloScape open dataset for autonomous driving and its application. IEEE Trans Pattern Anal Mach Intell 42, 2702–2719. https://doi.org/10.1109/TPAMI.2019.2926463.
Huang, Y., Zhou, J., Li, X., Dong, Z., Xiao, J., Wang, S., Zhang, H., 2023. MENet: Map-enhanced 3D object detection in bird's-eye view for LiDAR point clouds. Int. J. Appl. Earth Obs. Geoinf. 120 https://doi.org/10.1016/j.jag.2023.103337.
Jokela, M., Kutila, M., Pyykönen, P., 2019. Testing and validation of automotive point-cloud sensors in adverse weather conditions. Appl. Sci. 9 https://doi.org/10.3390/app9112341.
Kuang, H., Wang, B., An, J., Zhang, M., Zhang, Z., 2020. Voxel-FPN: multi-scale voxel feature aggregation for 3D object detection from LIDAR point clouds. Sensors (basel) 20. https://doi.org/10.3390/s20030704.
Kurup, A., Bos, J., 2021. DSOR: A Scalable Statistical Filter for Removing Falling Snow from LiDAR Point Clouds in Severe Winter Weather.10.48550/arXiv.2109.07078.
Lambert, J., Carballo, A., Cano, A.M., Narksri, P., Wong, D., Takeuchi, E., Takeda, K., 2020. Performance analysis of 10 models of 3D LiDARs for automated driving. IEEE Access 8, 131699–131722. https://doi.org/10.1109/access.2020.3009680.
Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O., 2019. PointPillars: Fast Encoders for Object Detection From Point Clouds, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12689-12697.
Law, H., Deng, J., 2019. CornerNet: Detecting objects as paired keypoints. Int. J. Comput. Vis. 128, 642–656. https://doi.org/10.1007/s11263-019-01204-1.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539.
Li, Y., Allu, K.R., Sun, Z., Tok, A.Y.C., Feng, G., Ritchie, S.G., 2021b. Truck body type classification using a deep representation learning ensemble on 3D point sets. Transp. Res. Part C: Emerg. Technol. 133 https://doi.org/10.1016/j.trc.2021.103461.
Li, J., Luo, C., Yang, X., 2023. PillarNeXt: Rethinking Network Designs for 3D Object Detection in LiDAR Point Clouds, 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17567-17576.
Li, G., Li, J., Wang, C., Peng, Q., Zhang, C., Gao, F., Tang, X., Guo, G., 2023a. Key supplement: Improving 3-D car detection with pseudo point cloud. IEEE Sens. J. 23, 18856–18866. https://doi.org/10.1109/jsen.2023.3292137.
Li, J., Xu, Z., Fu, L., Zhou, X., Yu, H., 2021a. Domain adaptation from daytime to nighttime: A situation-sensitive vehicle detection and traffic flow parameter estimation framework. Transp. Res. Part C: Emerg. Technol. 124 https://doi.org/10.1016/j.trc.2020.102946.
Liu, Z., Zhao, X., Huang, T., Hu, R., Zhou, Y., Bai, X., 2020. TANet: Robust 3D object detection from point clouds with triple attention. AAAI Conf. Artif. Intell.
Liu, Z., Gan, M., Xiong, L., Mao, X., Que, Y., 2023. Multilevel receptive field expansion network for small object detection. IET Image Proc. 17, 2385–2398. https://doi.org/10.1049/ipr2.12799.

Liu, F., Zhao, F., Liu, Z., Hao, H., 2019. Can autonomous vehicle reduce greenhouse gas emissions? A country-level evaluation. Energy Policy 132, 462–473. https://doi.org/10.1016/j.enpol.2019.06.013.
Loshchilov, I., Hutter, F., 2019. Decoupled Weight Decay Regularization. arXiv 1711.05101.
Ma, R., Chen, C., Yang, B., Li, D., Wang, H., Cong, Y., Hu, Z., 2022. CG-SSD: Corner guided single stage 3D object detection from LiDAR point cloud. ISPRS J. Photogramm. Remote Sens. 191, 33–48. https://doi.org/10.1016/j.isprsjprs.2022.07.006.
Mao, J., Niu, M., Jiang, C., Liang, H., Chen, J., Liang, X., Li, Y., Ye, C., Zhang, W., Li, Z., Yu, J., Xu, H., Xu, C., 2021. One Million Scenes for Autonomous Driving: ONCE Dataset. arXiv 2106.11037, 1-21.
Meyer, M., Kuschk, G., 2019. Automotive Radar Dataset for Deep Learning Based 3D Object Detection, Proceedings of the 16th European Radar Conference, pp. 1-4.
Mostajabi, M., Wang, C.M., Ranjan, D., Hsyu, G., 2020. High Resolution Radar Dataset for Semi-Supervised Learning of Dynamic Objects, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 450-457.
OpenPCDet-Development-Team, 2020. OpenPCDet: An opensource toolbox for 3D object detection from point clouds.
Palffy, A., Pool, E., Baratam, S., Kooij, J.F.P., Gavrila, D.M., 2022. Multi-class road user detection with 3+1D radar in the view-of-Delft dataset. IEEE Rob. Autom. Lett. 7, 4961–4968. https://doi.org/10.1109/lra.2022.3147324.
Park, J.-I., Park, J., Kim, K.-S., 2020. Fast and accurate desnowing algorithm for LiDAR point clouds. IEEE Access 8, 160202–160212. https://doi.org/10.1109/access.2020.3020266.
Pitropov, M., Garcia, D., Rebello, J., Smart, M., Wang, C., Czarnecki, K., Waslander, S., 2021. Canadian adverse driving conditions dataset. Int. J. Robot. Res. 40, 681–690.
Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. PointNet++: Deep hierarchical feature learning on point sets in a metric space. arXiv 1706.02413.
Qi, C.R., Chen, X., Litany, O., Guibas, L.J., 2020. ImVoteNet: Boosting 3D Object Detection in Point Clouds With Image Votes, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4403-4412.
Rusu, R.B., Cousins, S., 2011. 3D is here: Point cloud library (PCL), 2011 IEEE International Conference on Robotics and Automation, Shanghai,China.
Sheeny, M., Pellegrin, E.D., Mukherjee, S., Ahrabian, A., Wang, S., Wallace, A., 2021. RADIATE: A radar dataset for automotive perception in bad weather, IEEE International Conference on Robotics and Automation, pp. 1-7.
Shi, S., Wang, X., Li, H., 2019. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-779.
Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H., 2020. PV-RCNN: Point-voxel feature set abstraction for 3d object detection, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) IEEE, pp. 10526-10535.
Shi, S., Jiang, L., Deng, J., Wang, Z., Guo, C., Shi, J., Wang, X., Li, H., 2022. PV-RCNN++: point-voxel feature set abstraction with local vector representation for 3D object detection. Int. J. Comput. Vis. 131, 531–551. https://doi.org/10.1007/s11263-022-01710-9.
Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D., 2020. Scalability in Perception for Autonomous Driving: Waymo Open Dataset, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2443-2451.
Transportation, U.S.D.o., 2019. U.S. Department of Transportation: Federal Highway Administration. How do weather events impact roads?, https : / / ops . fhwa . dot . gov / weather / q1_roadimpact.html.
Uijlenhoet, R., Stricker, J., 1999. A consistent rainfall parameterization based on the exponential raindrop size distribution. 218, 101-127.
Vora, S., Lang, A.H., Helou, B., Beijbom, O., 2020. PointPainting: Sequential fusion for 3D object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4603–4611.
Wang, Z., Zhan, J., Duan, C., Guan, X., Lu, P., Yang, K., 2022. A Review of Vehicle Detection Techniques for Intelligent Vehicles. IEEE Trans Neural Netw Learn Syst PP.10.1109/TNNLS.2021.3128968.
Wu, H., Deng, J., Wen, C., Li, X., Wang, C., Li, J., 2022. CasA: A Cascade attention network for 3-D object detection from LiDAR point clouds. IEEE Trans. Geosci. Remote Sens. 60, 1–11. https://doi.org/10.1109/tgrs.2022.3203163.
Wu, H., Wen, C., Shi, S., Li, X., Wang, C., 2023. Virtual sparse convolution for multimodal 3D object detection. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 21653–21662.
Xia, Q., Chen, Y., Cai, G., Chen, G., Xie, D., Su, J., Wang, Z., 2023. 3-D HANet: A flexible 3-D heatmap auxiliary network for object detection. IEEE Trans. Geosci. Remote Sens. 61, 1–13. https://doi.org/10.1109/TGRS.2023.3250229.
Xu, Q., Zhou, Y., Wang, W., Qi, C.R., Anguelov, D., 2021. SPG: unsupervised domain adaptation for 3D object detection via semantic point generation. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 15426–15436.
Yan, Y., Mao, Y., Li, B., 2018. SECOND: Sparsely embedded convolutional detection. Sensors (Basel) 18. https://doi.org/10.3390/s18103337.
Yang, H., Cai, J., Liu, C., Ke, R., Wang, Y., 2023a. Cooperative multi-camera vehicle tracking and traffic surveillance with edge artificial intelligence and representation learning. Transp. Res. Part C: Emerg. Technol. 148 https://doi.org/10.1016/j.trc.2022.103982.
Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J., 2019. STD: Sparse-to-dense 3D object detector for point cloud, 2019 IEEE/CVF International Confercence on Computer Vision, pp. 1951–1960.
Yang, J., Levin, M.W., Hu, L., Li, H., Jiang, Y., 2023b. Fleet sizing and charging infrastructure design for electric autonomous mobility-on-demand systems with endogenous congestion and limited link space. Transp. Res. Part C: Emerg. Technol. 152 https://doi.org/10.1016/j.trc.2023.104172.
Yin, T., Zhou, X., Krahenbuhl, P., 2021. Center-based 3D object detection and tracking. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11779–11788.
Yu, C., Lei, J., Peng, B., Shen, H., Huang, Q., 2022. SIEV-net: a structure-information enhanced voxel network for 3D object detection from LiDAR point clouds. IEEE Trans. Geosci. Remote Sens. 60, 1–11. https://doi.org/10.1109/tgrs.2022.3174483.
Zhang, Y., Carballo, A., Yang, H., Takeda, K., 2023b. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. ISPRS J. Photogramm. Remote Sens. 196, 146–177. https://doi.org/10.1016/j.isprsjprs.2022.12.021.
Zhang, Y., Trenberth, K.E., 2018. How often does it really rain? Bull. Am. Meteorol. Soc. 99, 289–298. https://doi.org/10.1175/bams-d-17-0107.1.
Zhang, E., Zhang, R., Masoud, N., 2023a. Predictive trajectory planning for autonomous vehicles at intersections using reinforcement learning. Transp. Res. Part C: Emerg. Technol. 149 https://doi.org/10.1016/j.trc.2023.104063.
Zhao, J., Xu, H., Liu, H., Wu, J., Zheng, Y., Wu, D., 2019. Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors. Transp. Res. Part C: Emerg. Technol. 100, 68–87. https://doi.org/10.1016/j.trc.2019.01.007.
Zhou, Y., Tuzel, O., 2018. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4490-4499.
Zhou, D., Fang, J., Song, X., Liu, L., Yin, J., Dai, Y., Li, H., Yang, R., 2020. Joint 3D Instance Segmentation and Object Detection for Autonomous Driving, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1836-1846.
Zhu, Y., Xu, R., An, H., Tao, C., Lu, K., 2022. Anti-noise 3D object detection of multimodal feature attention fusion based on PV-RCNN. Sensors (Basel) 23. https://doi.org/10.3390/s23010233.