A Weakly Supervised Vehicle Detection Method from LiDAR Point Clouds

Yiyuan Li[1], Yuhang Lu[1], Xun Huang[1], Siqi Shen[1], Cheng Wang[1], Chenglu Wen [1]

[1]Xiamen University, Xiamen, China - {liyiyuan, 31520201153885, huangxun}@stu.xmu.edu.cn,
{siqishen, cwang, clwen}@xmu.edu.cn

Keywords: LiDAR, Object Detection, Weakly Supervised Learning.

Abstract

Training LiDAR point clouds object detectors requires a significant amount of annotated data, which is time-consuming and effort-demanding. Although weakly supervised 3D LiDAR-based methods have been proposed to reduce the annotation cost, their performance could be further improved. In this work, we propose a weakly supervised LiDAR-based point clouds vehicle detector that does not require any labels for the proposal generation stage and needs only a few labels for the refinement stage. It comprises two primary modules. The first is an unsupervised proposal generation module based on the geometry of point clouds. The second is the pseudo-label refinement module. We validate our method on two point clouds based object detection datasets, namely KITTI and ONCE, and compare it with various existing weakly supervised point clouds object detection methods. The experimental results demonstrate the method’s effectiveness with a small amount of labeled LiDAR point clouds.

1. Introduction

3D object detection is a fundamental task in the field of computer vision, encompassing a broad spectrum of applications in areas such as autonomous driving robot navigation, etc. It involves the precise identification and localization of 3D objects within a scene using point cloud data acquired from advanced 3D sensors such as LiDAR or depth cameras. In recent years, with the powerful capabilities of deep neural networks, 3D object detection frameworks (Li et al., 2021), (Chen et al., 2022) based on lidar point clouds have emerged and demonstrated high performance on various public benchmark datasets (Geiger et al., 2012), (Mao et al., 2021). Despite receiving significant attention, 3D object detection using LiDAR point clouds continues to encounter substantial challenges, such as the requirement for extensive labeled data. The process of labeling point cloud data is laborious and time-consuming. Consequently, there exists considerable research significance in exploring weakly supervised 3D object detection methods.

Existing weakly supervised methods for 3D object detection often rely on weak forms of supervision rather than fully annotated 3D bounding boxes, such as image-level labels (Peng et al., 2022), a pre-trained 2D detector (Qin et al., 2020), or partial annotations like object center in bird’s eye view (Meng et al., 2020), to reduce the annotation burden. Weakly supervised 3D object detection reduces the labor cost of data annotation. However, due to the lack of sufficient accurate annotations, these methods have a performance gap with fully supervised methods. Some works focus on semi-supervised learning (Caine et al., 2021), which usually assists model training by generating pseudo-labels. Nevertheless, how to fully exploit the geometric information of point clouds to generate high-quality pseudo-labels with limited annotations remains an open question.

In order to reduce the burden of data annotation and promote the development of weakly supervised object detection, we propose a simple yet effective weakly supervised point clouds-based vehicle detection method, which can generate high-quality pseudo-labels to aid in training. As a two-stage method, it consists of an unsupervised proposal generation (UPG) module for the proposal generation stage and a pseudo-label refinement (PLR) module for the refinement stage. At the core of the UPG module is the estimation of potential locations and orientations of vehicles based on LiDAR point cloud geometry. Through the UPG module, we can obtain 3D object proposals in an unsupervised way without requiring annotation information. Due to potential noise in the generated predictions, they cannot be the final output. In order to improve the accuracy of pseudo-labels, in the refinement stage, the PLR module first refines pseudo-labels with confidence. Next, a clustering-based approach is employed to classify and filter redundant bounding boxes. Finally, the obtained high-quality pseudo-labels can be used to train the 3D detector. We have conducted extensive experiments on the KITTI (Geiger et al., 2012) and the ONCE (Mao et al., 2021) dataset to demonstrate the effectiveness of our approach. In summary, the primary contributions of our work are as follows:

1. We propose a proposal generation module that can obtain 3D object proposals in an unsupervised way.

2. We introduce a pseudo-label refinement module consisting of two stages to obtain accurate pseudo-labels for training.

3. We present an effective weakly-supervised 3D vehicle detection framework and conduct extensive experiments on two mainstream 3D object detection datasets. The results demonstrate the good performance of our method.

The rest of this work is organized as follows: Section 2 outlines the related works of this study. Section 3 details our framework including two modules. Section 4 evaluates our method on several benchmarks. Finally, Section 5 concludes this work.

2. Related Works

2.1 3D object detection

In recent years, a multitude of methods for 3D object detection in point cloud data have been proposed. The majority of these methods can be categorized into three distinct groups: point-based methods, grid-based methods, and multi-modal fusion-based methods.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-1-2024-123-2024 | © Author(s) 2024. CC BY 4.0 License.

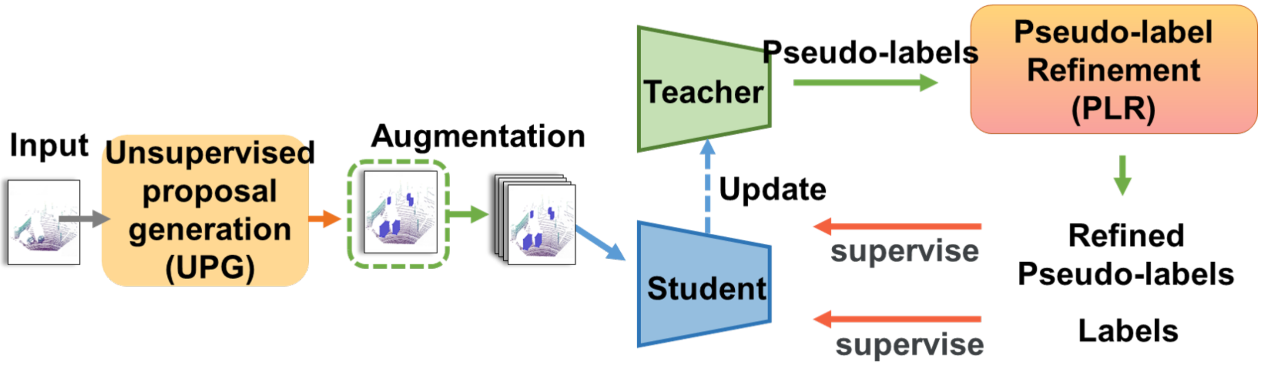


Figure 1. The overview of our approach.

Point-based methods (Qi et al., 2017) employ multi-layer perceptrons to directly extract features from unordered point clouds, thus better preserving the original data information. Shi et al. (Shi et al., 2019) partitioned the point cloud into foreground and background points, and integrated the semantic information and local features of the point cloud for precise localization. Grid-based methods utilize rasterization techniques to convert point clouds into discrete grid representations, such as voxels (Deng et al., n.d.) or bird's-eye view (Lang et al., 2019), and then extract features from these representations. Multi-modal fusion methods integrate data from different sensors containing both texture and spatial position information (Shi et al., 2020). Some approaches integrated 3D point clouds with images to acquire fused data containing both texture and spatial location information (Xu et al., 2018). Other approaches integrated various forms of point cloud inputs, including points and voxels, to enhance detection efficiency and achieve a broader receptive field (Shi et al., 2020), (Qian et al., 2022).

2.2 Weakly-supervised 3D object detection

Exploring weakly supervised 3D object detection methods is of significant research importance. Weakly supervised methods utilize weak supervisory signals to train 3D detectors. The weak supervisions include BEV object centers (Meng et al., 2020), pre-trained 2D detectors (Qin et al., 2020), 2D image annotations (Wei et al., 2021), and sparse annotations (Liu et al., 2022). These weakly supervised methods effectively reduce the annotation requirements, but there still exists a performance gap compared to fully supervised methods. We believe that within a 3D point cloud scene, objects themselves carry unique geometric properties, such as shape, size, and so on. However, how to fully exploit these geometric properties for accurate object recognition remains an open issue. This paper aims to use the geometric information of point clouds for weakly supervised object detection. Therefore, we initially generate region proposals in an unsupervised manner based on the density information of instances in the scene, which are then classified and refined in subsequent steps.

2.3 Semi-supervised 3D object detection

Semi-supervised object detection aims to train detectors using a small amount of labeled data and a large amount of unlabeled data to approach or even reach the performance of fully supervised methods. In semi-supervised 3D object detection, there are two main approaches: the teacher-student model and the pseudo-labeling technique. The teacher-student model (Zhao et al., 2020) typically follows a mean teacher paradigm, which includes a teacher network and a student network. Specifically, the teacher network is first trained using labeled data and then guides the training of the student network on unlabeled data. The pseudo-labeling method (Caine et al., 2021) first utilizes annotated data to train a 3D detector. Subsequently, this detector is used to predict unlabeled data and generate pseudo-labels. Finally, the detector is retrained on the unlabeled scenes with the pseudo-labels to enhance detection performance. Chen et al. (Chen et al., 2023) transformed the issue of noise in pseudo-labels into a noise learning problem and proposed a noise-resistant supervision module and a feature consistency constraint module, which helps improve the model's generalization ability and eliminate the impact of noisy annotations. Semi-supervised methods can mine useful information from unlabeled data. At the same time, we find that relying solely on the geometric information of point clouds is insufficient for object recognition and localization. Therefore, our approach uses semi-supervised learning based on the mean-teacher network and employs a pseudo-label refinement module to remove low-quality pseudo-labels for providing the model with additional accurate supervision signals.

3. Methodology

In this section, we introduce the proposed 3D object detection method in detail. Our method is a two-stage, weakly supervised LiDAR point clouds vehicle detection method. Its overview is shown in Fig. 1. It consists of two major modules: the unsupervised proposal generation (UPG) module and the pseudo-labels Refinement (PLR) module. In the proposal generation stage, the UPG module generates proposals based on the geometry of LiDAR point clouds. After data augmentation, the data are used to train the student model (Tarvainen and Valpola, 2017). The teacher model is initialized using the student model. In the proposal refinement stage, the pseudo-label refinement (PLR) module adopts a two-stage approach to filter low-quality pseudo-labels generated by the teacher model. The precise pseudo-labels and labels are used to supervise the student model. Then, the teacher model is updated according to (Tarvainen and Valpola, 2017). The final student model is used as the object detector.

3.1 Unsupervised proposal generation (UPG) module

3.1.1 Anchors initialization In the absence of annotations, the object's location within the scene can be arbitrary. To address this, we leverage prior knowledge to identify potential object regions. Most vehicles are typically situated on

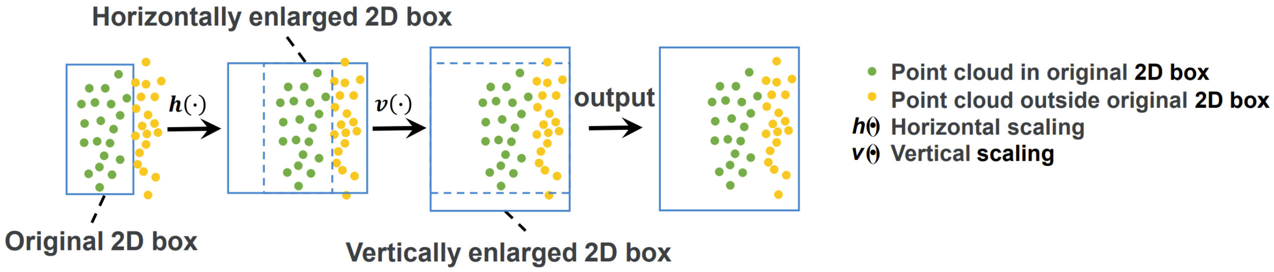

Horizontally enlarged 2D box
$h(\cdot)$
$v(\cdot)$
output
Original 2D box
Vertically enlarged 2D box
● Point cloud in original 2D box
● Point cloud outside original 2D box
$h(\cdot)$ Horizontal scaling
$v(\cdot)$ Vertical scaling

Figure 2. 2D box scaling. The green and yellow points are the points of the same object. The transformation consists of two components: $h(\cdot)$ horizontal scaling and $v(\cdot)$ vertical scaling. The dashed rectangle and solid rectangle represent the box before and after transformation, respectively.

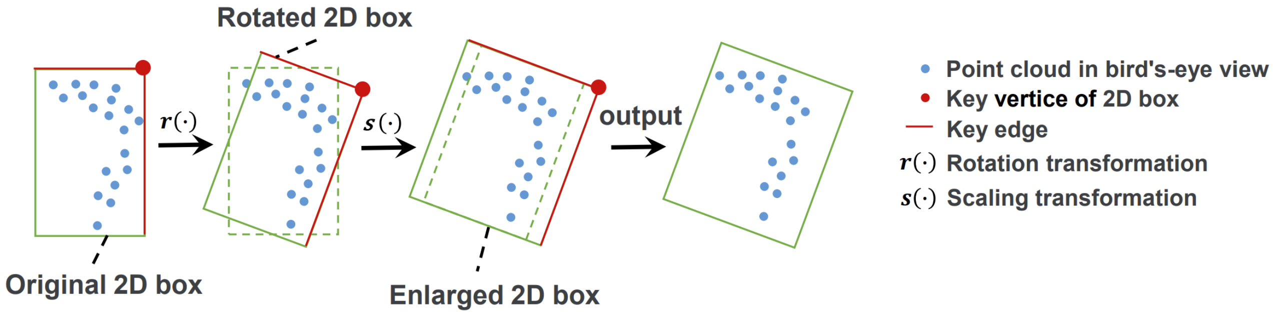


Figure 3. 2D box rotation and scaling for object orientation estimation. When the 2D box fails to accurately estimate the object's orientation, we initially rotate the 2D box and then enlarge the 2D box to encompass the entire object. The dashed rectangle and solid rectangle, respectively, represent the box before and after transformation.

the ground, given a point clouds scene, we employ the random sampling consistency method(Fischler and Bolles, 1981) to eliminate point clouds of ground. Subsequently, we generate anchor boxes at intervals of 0.2m within the range of 0 to 80m in front of the LiDAR sensor and 0 to 40m in the left and right directions. Based on the geometry of point clouds, the anchor size is 3.9m×1.6m×1.56m, and its spatial orientation angles are 0 degrees and 90 degrees, which are perpendicular to each other. We then project both the 3D anchor and point cloud to the front view according to (Qin et al., 2020), which enables direct processing of the 2D box corresponding to the 3D anchor box.

3.1.2 2D box scaling In 3D scenes, it is known that the regions of interest typically contain a greater number of points. We first select the anchor to make sure that the anchor contains most of the points of the point clouds of the vehicle based on geometry (i.e., point cloud density). The density of point clouds is influenced by the number of points inside a 3D anchor and the sampling distance of LiDAR. As the distance between LiDAR and a 3D anchor increases, the resulting point clouds become sparse. Meanwhile, given the fixed size, position, and orientation of the preset anchors, we utilize the strategy introduced in (Qin et al., 2020) to resize the 2D box into a square through interpolation. Hence, we quantify the LiDAR point cloud density within a 2D box as follows.

$$D = \frac{\alpha N}{R_c^2} \tag{1}$$

where D is the point cloud density and α is a weight coefficient, which increases linearly with the increase of the distance between the vehicle and LiDAR. N is the number of points inside the 3D anchor, and R_c represents the width and height of the resized 2D box. 2D boxes with a density below a threshold τ are filtered out.

Since the placement of the 3D anchor box is determined by predetermined parameters, it may not entirely enclose the vehicle. Consequently, the positions of points tend to be relatively close to the boundary of the 3D anchor. To make sure that a 3D anchor entirely encloses the point clouds of a vehicle, we perform horizontal scaling $h(\cdot)$ and vertical scaling $v(\cdot)$ on the 3D anchor's 2D box to accurately estimate the bounding boxes of vehicles. Specifically, the process consists of two steps. First, we horizontally enlarge the 2D box. When the enlarged 2D box contains additional points, we choose the enlarged 2D box to fully encompass the object. Otherwise, we select the original 2D box. Next, we perform vertical enlargement. Similarly, we only choose the enlarged 2D box when it contains additional points after enlargement. On the other hand, for 2D boxes containing sparse edge regions, we apply the same method to scale down the 2D box in both horizontal and vertical directions to make it more compact. We only choose the scaled-down 2D box when the number of points contained has not decreased. The schematic plot of this enlargement process is depicted in Fig. 2. After the 2D box operations, the corresponding 3D transformations are applied to the 3D anchors. The 3D visualization result of this transformation process is shown in Fig. 4(a).

3.1.3 Object orientation estimation In addition to determining the object's position, it is essential to estimate the object's orientation. The UPG module rotates the 2D box and the corresponding 3D anchor if the orientation of the 2D box in a bird's-eye view does not match the vehicle's orientation well. The UPG module performs 2D box rotation according to the key vertex and key edge of the 2D box. The key vertex is a vertex among the four vertices of the 2D box. It exhibits the minimal sum of the distance from each point within the box. The key edges are 2D box edges that are connected to the key vertex. A higher concentration of points close to key edges corresponds

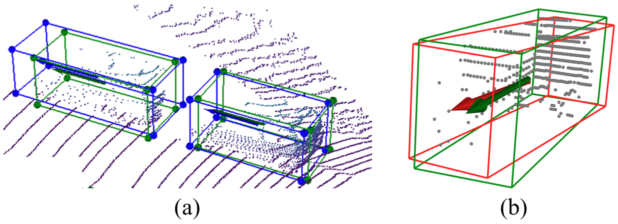


Figure 4. Visualization results of 3D anchor transformation. (a) Before translation and scaling (blue box) and after (green box). (b) Before rotation (green box) and after object rotation estimation (red box).

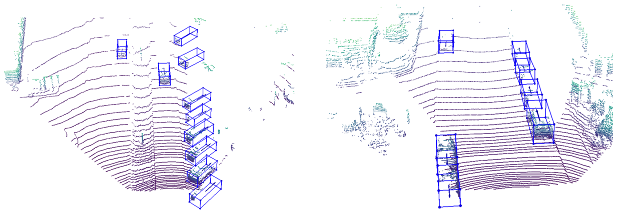


Figure 5. Visualization results of unsupervised proposals generation on the KITTI dataset.

to a potentially more accurate estimation of the object's orientation. We calculate the point density f around key edges as follows.

$$f = \frac{|\mathcal{Q}_0|}{|\mathcal{Q}|} \tag{2}$$

where Q_0 denotes:

$$\mathcal{Q}_0 = \{q | q \in \mathcal{Q}, \text{dist}(q, l_1) > \lambda \wedge \text{dist}(q, l_2) > \lambda\} \tag{3}$$

and $\mathcal{Q}$ is the set of all the cloud points within the 2D box. The $dist(q, l)$ is the distance between point q and key edge l, and λ is the distance threshold. We rotate the 2D box incrementally, at 0.5-degree intervals, from 0 to 90 degrees. The rotation $r(\cdot)$ with the smallest f is chosen as the direction of the proposal. In the end, the UPG module adjusts the size and location of the 2D box through a scaling transformation $s(\cdot)$ (includes $h(\cdot)$ and $v(\cdot)$) in the same way as Sec. 3.1.2. This process is illustrated in Fig. 3, and the 3D visualization result of the process is shown in Fig. 4(b). By performing the aforementioned steps of anchor selection and transformation, we obtain unsupervised object proposals for vehicles. A part of the results as shown in Fig. 5.

3.2 Data augmentation

Data augmentation is important for the generalization of the 3D detectors. Besides typical augmentation methods such as random rotation, random scaling, and random flipping, we augment data by randomly sampling annotated vehicles and concatenating them to the point clouds of a scene. The augmented data are then fed into the teacher-student network to output predictions, which are further refined via the following PLR module.

3.3 Pseudo-label refinement (PLR) module

As a weakly-supervised method, our method adopts the mean-teacher paradigm (Tarvainen and Valpola, 2017). The teacher model produces pseudo-labels for training 3D detectors. However, the pseudo-labels generated by the teacher model are noisy, which could hurt model performance. The PLR module refines the generated pseudo-labels using the following procedure.

3.3.1 Confidence-threshold refinement As the augmented data may be noisy, we use the teacher model to detect these generated objects to obtain their classes and bounding boxes. These results may suffer from wrong classification or inaccurate regression. The PLR module uses a classification confidence threshold and an intersection over union (IoU) threshold to refine these augmented data. Taking the KITTI dataset (Geiger et al., 2012) as an example, we define predictions with classification confidence scores below 0.1 or IOU values below 0.3 as negative samples, and the rest as positive samples.

3.3.2 Clustering-based pseudo-label refinement Albeit we have refined pseudo-labels according to the confidence score and IoU threshold, there still exist redundant bounding boxes that may cause false positives. The PLR module adapts a clustering-based method proposed in (Yin et al., 2022) to filter these redundant bounding boxes. The boxes are firstly sorted according to confidence score. Then, the PLR modules aggregate boxes that have large IoU with the box with the highest score. The process is performed iteratively for all the boxes. After that, all the boxes are grouped into different clusters. For a cluster with multiple boxes, the features of boxes are extracted, and they are fed into a 2-layer MLP, which regresses the box position. Then, the PLR module uses the non-maximum suppression (NMS) method to obtain an accurate box position for each cluster. In the end, the PLR module obtains high-quality pseudo-labels, which can be used to train the 3D detector.

3.4 Loss function

Given a series of labeled point clouds and pseudo-labeled point clouds, our objective is to optimize the model parameters to accurately predict object categories and locations. We use loss functions similar to SECOND(Yan et al., 2018) for both the annotated and pseudo-labeled data. For the object classification task, we employ the focal loss(Lin et al., 2017), which is defined as follows.

$$\mathcal{L}_{cls} = -\alpha(1 - q_t)^{\gamma} log(q_t) \tag{4}$$

where α represents the weight coefficient assigned to positive and negative samples, γ denotes the weight coefficient assigned to difficult and easy samples, and q_t signifies the class prediction probability. We use α = 0.25 and γ = 2. For parameter estimation of object regression, such as target center (x, y, z), size (w, h, l), and angle (θ), we use the loss:

$$\mathcal{L}_{reg} = \sum_{q \in \{x,y,z,l,w,h,\theta\}} SmoothL1(\Delta q) \tag{5}$$

where q denotes the predicted result, and Δq represents the prediction errors between ground truth and the predicted result.

For annotated data, the loss function is denoted as:

$$\mathcal{L}^L = \beta_1 \mathcal{L}^L_{cls} + \beta_2 \mathcal{L}^L_{reg} \tag{6}$$

where β_1 and β_2 are hyper-parameters. For pseudo-labeled data, its loss function is defined similarly:

$$\mathcal{L}^U = \beta_3 \mathcal{L}^U_{cls} + \beta_4 \mathcal{L}^U_{reg} \tag{7}$$

Table 1. Average precision (AP) results of two proposal generation methods on the validation set of KITTI (Car).

	AP (IoU=0.3)			AP (IoU=0.5)		
	Easy	Moderate	Hard	Easy	Moderate	Hard
VS3D (Qin et al., 2020) + UPM	65.82	60.43	50.35	40.32	38.64	32.06
VS3D (Qin et al., 2020) + UPG	**67.26**	**63.12**	**53.04**	**43.31**	**40.52**	**35.76**

Table 2. Average Orientation Similarity (AoS) results of two proposal generation methods on validation set of KITTI (Car).

	AP (IoU=0.3)			AP (IoU=0.5)		
	Easy	Moderate	Hard	Easy	Moderate	Hard
UPM	73.64	66.56	58.46	61.04	58.76	42.93
UPG	**81.89**	**73.71**	**68.24**	**76.92**	**70.54**	**65.74**

where β_3 and β_4 are hyper-parameters. We also utilize the consistency loss introduced by SESS(Zhao et al., 2020), denoted as $\mathcal{L}_{consistency}$. The overall loss function can be represented as follows.

$$\mathcal{L} = \mathcal{L}^L + \mathcal{L}^U + \mathcal{L}_{consistency} \tag{8}$$

4. Experiments

We evaluate the proposed LiDAR-based vehicle detector based on the popular LiDAR-based object detection dataset: KITTI (Geiger et al., 2012) and the ONCE (Mao et al., 2021), with multiple weakly supervised point clouds object detection methods.

4.1 Dataset

The KITTI dataset(Geiger et al., 2012) serves as a benchmark for evaluating computer vision techniques and comprises a comprehensive collection of resources, including 389 pairs of stereo images, visual odometry sequences spanning 39.2 kilometers, and over 200,000 annotated 3D objects for object detection. The original dataset is classified into distinct categories, namely 'Road', 'City', 'Residential', 'Campus', and 'Person'. Regarding 3D object detection, the labels are further divided into car, van, truck, pedestrian, seated pedestrian, cyclist, tram, and miscellaneous objects. The ONCE dataset(Mao et al., 2021) consists of one million LiDAR scenes and seven million corresponding camera images, covering various regions, periods, and weather conditions. Notably, there are annotated 16,000 scenes with 3D ground truth boxes encompassing 5 categories: car, bus, truck, pedestrian, and cyclist. We train our method with different amounts of labeled and unlabeled data on the training sets of two datasets and report results on the KITTI validation set and ONCE test set, respectively.

4.2 Evaluation metrics

To assess the performance of the proposed methods, we report the popular average precision(AP) with Intersection over Union (IoU) 0.5 threshold. We can plot the Precision-Recall curve and determine the average precision by integrating the area under the curve. Additionally, we use the Average Orientation Similarity indicator (AoS) (Geiger et al., 2012) for object orientation estimation. AOS is defined as:

$$AoS = \frac{1}{11} \sum_{r \in \{0,0.1,..,1\}} \max_{\widetilde{r}:\widetilde{r} \geq r} s(\widetilde{r}) \tag{9}$$

where r denotes the recall rate, and the orientation similarity $s(\widetilde{r})$ is the normalized cosine distance between all predicted samples and the ground truth.

4.3 Implementation details

The presented model is implemented utilizing the OpenPCDet framework(Team, 2020), while the experimental datasets employed are KITTI(Geiger et al., 2012) and ONCE(Mao et al., 2021) datasets. We pre-train the student network with all available labeled data and initialize the teacher network with the pretrained weights. Then, we train the student network on both labeled and unlabeled data and update the parameters of the teacher network using the exponential moving average (EMA) of the student network's parameters. For the EMA decay value, we follow (Tarvainen and Valpola, 2017). The training is conducted on NVIDIA GeForce RTX 3090 GPU.

4.4 Experimental results

4.4.1 Experimental results on kITTI dataset We initially validate the effectiveness of the unsupervised proposal generation (UPG) module and compare it with the object proposal module (UPM) of the baseline VS3D (Qin et al., 2020). We utilize the VS3D (Qin et al., 2020) framework and substitute the UPM module with our UPG module. Tables 1 and 2 present the experimental results of the Average Precision (AP) metric and the Average Orientation Similarity (AOS) metric with IOU thresholds of 0.3 and 0.5 on the validation set of KITTI. By introducing a distance threshold to reduce the interference of sparse point clouds and employing key edge-based rotation to estimate the direction of objects, our UPG module demonstrates superior performance in proposal generation.

To further experiment, we compare our approach with the fully supervised method PVRCNN(Shi et al., 2020) on the car category of the KITTI(Geiger et al., 2012) validation set. The Intersection over Union (IoU) threshold is set to 0.5. The evaluation metric is the average precision for the car category. To investigate the effectiveness of our proposed weakly supervised method, we gradually reduce the number of labeled scenes from 100% to 1%, with the number of labeled cars from 15,654 to 178. As it is shown in Table 3, our method performs slightly worse than PVRCNN(Shi et al., 2020) in the fully supervised case. However, for the 50% labeled case, the proposed method performs slightly better than PVRCNN. When there are only 20%, 10% and 1% labeled point clouds scenes, our method consistently outperforms PVRCNN. The primary reason for this

Table 3. Average precision (AP) results on the validation set of KITTI (Car).

Learning paradigm	Detector	Training setting	Car Easy	Moderate	Hard
Fully supervised Weakly supervised	PVRCNN(Shi et al., 2020) Proposed	Trained on 100% labeled scenes with 15654 vehicle instances	**88.45** 86.44	**77.67** 74.21	**76.30** 69.85
Fully supervised Weakly supervised	PVRCNN(Shi et al., 2020) Proposed	Trained on 50% labeled scenes with 7312 vehicle instances	83.62 **84.36**	69.93 **71.05**	**66.43** 65.69
Fully supervised Weakly supervised	PVRCNN(Shi et al., 2020) Proposed	Trained on 20% labeled scenes with 3634 vehicle instances	72.43 **74.23**	62.69 **65.14**	54.81 **56.67**
Fully supervised Weakly supervised	PVRCNN(Shi et al., 2020) Proposed	Trained on 10% labeled scenes with 1362 vehicle instances	68.85 **70.82**	60.41 **62.06**	53.26 **54.09**
Fully supervised Weakly supervised	PVRCNN(Shi et al., 2020) Proposed	Trained on 1% labeled scenes with 178 vehicle instances	66.14 **68.47**	60.05 **61.72**	52.87 **53.30**

Table 4. Average precision (AP) results of weakly supervised methods on validation set of KITTI (Car).

Learning Paradigm	Detector	Training setting	Car Easy	Moderate	Hard
Fully supervised	SECOND(Yan et al., 2018)	Trained on 500/3712 labeled frames with 2716 vehicle instances	**76.12**	**67.43**	**58.30**
Weakly supervised	VS3D(Qin et al., 2020)		42.43	40.67	32.15
Weakly supervised	Proposed		75.39	66.74	56.17

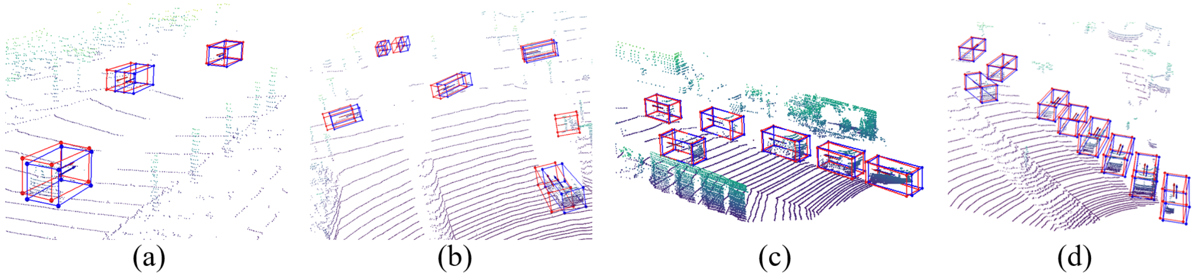


Figure 6. Visual detection results on the KITTI dataset. (a) and (b) represent sparse scenes, while (c) and (d) represent dense scenes. The red boxes depict the ground truth, while the blue boxes represent the detection results.

improvement lies in our method's ability to accurately identify candidate areas based on the geometry of point clouds, ensuring the quality of the generated pseudo-labels. The refinement of these pseudo-labels further provides the model with more precise supervision signals, ultimately improving its performance.

Table 4 depicts the comparison results of our method with the weakly supervised method VS3D (Qin et al., 2020) and the fully supervised method SECOND on the car category of the KITTI(Geiger et al., 2012) dataset. The experiment consists of a total of 3712 point cloud scenes. We experiment the condition of 500 labeled frames with 2176 labeled objects. Our method outperforms the weakly supervised method VS3D(Qin et al., 2020). Furthermore, our proposed method performs comparably to the fully supervised method SECOND(Yan et al., 2018). This demonstrates the effective utilization of our method on unlabeled point clouds. The detection results are visualized in Fig. 6., including both sparse and dense scenes.

4.4.2 Experimental Results on ONCE Dataset To validate the effective utilization of unlabeled data, we compare our approach with several semi-supervised point cloud object detection methods on the ONCE (Mao et al., 2021) dataset (vehicle category). We use the experimental steps introduced in ONCE (Mao et al., 2021). Firstly, we establish a baseline by training a detector SECOND(Yan et al., 2018) using labeled data. Secondly, we use different weakly supervised learning methods to train the baseline method. We report the detection performance on the test set for models trained with the different unlabeled subsets: U_{small}, U_{Medium}, and U_{large}, respectively.

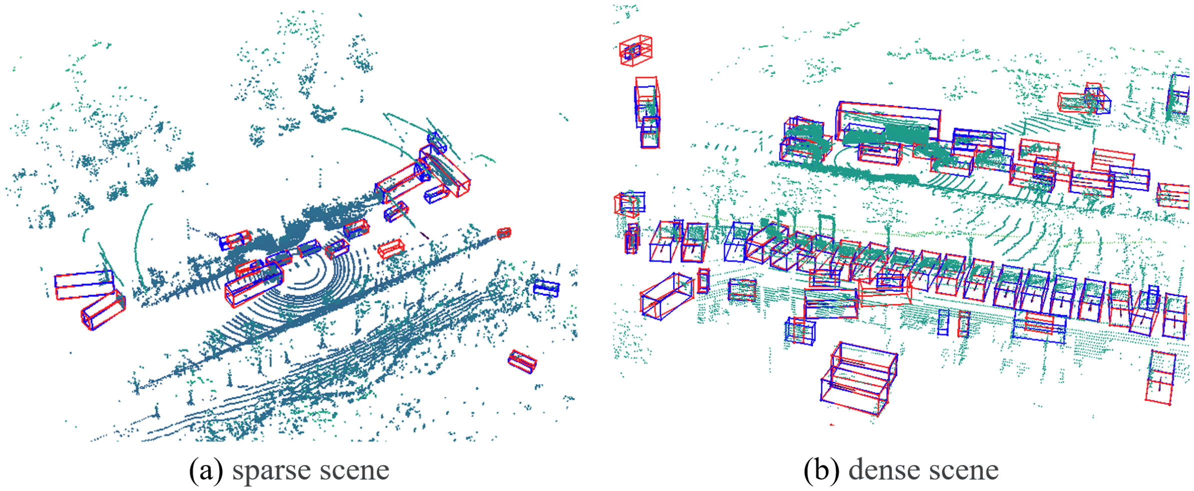
As it is shown in Table 5, most weakly supervised methods demonstrate the capability to enhance detection performance by leveraging unlabeled data, in contrast to SECOND[45], which relies solely on labeled data for training. Moreover, as the quantity of unlabeled data increases, all methods except for Pseudo-label(Lee, 2013) can improve detection results. Pseudo-label(Lee, 2013) initially obtains performance gain from 72.37% to 73.06%, followed by a subsequent decline. our proposed approach obtains better results than other semi-supervised methods, which demonstrates the high quality of the generated pseudo-labels. Fig. 7 illustrates two exemplar results on ONCE (Mao et al., 2021) dataset.

4.5 Ablation Study

To study the impact of different modules, we conduct experiments on the KITTI dataset. We studied two different ways to implement the unsupervised proposal generation (UPG) module. The first one, point clouds-density (PCD), estimates the

Table 5. Average precision (AP) results of weakly supervised learning methods on test set of ONCE (Vehicle).

Detector	Training setting	Vehicle			
		overall	0-30m	30-50m	50m-inf
baseline(Yan et al., 2018)		70.21	85.32	61.47	43.83
Pseudo-label(Lee, 2013)	Trained on 100,000 unlabeled set U_{small}	72.37	87.51	61.69	46.32
SESS(Zhao et al., 2020)		72.35	87.23	63.35	49.65
3DIoUMatch(Wang et al., 2021)		72.13	**88.34**	65.55	50.39
Proposed		**72.41**	86.06	**68.79**	**53.59**
Pseudo-label(Lee, 2013)	Trained on 500,000 unlabeled set U_{Medium}	73.06	83.52	66.05	51.82
SESS(Zhao et al., 2020)		74.11	86.78	70.21	56.14
3DIoUMatch(Wang et al., 2021)		75.07	**86.94**	**70.61**	56.06
Proposed		**76.07**	86.83	70.38	**56.56**
Pseudo-label(Lee, 2013)	Trained on 1 million unlabeled set U_{large}	72.80	84.46	64.97	51.46
SESS(Zhao et al., 2020)		75.38	**86.67**	70.48	**56.60**
3DIoUMatch(Wang et al., 2021)		75.11	86.46	70.44	56.06
Proposed		**75.69**	86.54	**71.74**	56.43


(a) sparse scene

(b) dense scene

Figure 7. Example of detection results on the ONCE dataset, with sparse scene on the left and dense scene on the right. The red boxes are the ground truth, while the blue boxes represent the detection results.

Table 6. Average precision (AP) results of different proposal generation strategies on validation set of KITTI (Car).

Ablations	Car		
	Easy	Moderate	Hard
UPG (PCD)	49.36	41.12	30.97
UPG (DPCD)	**50.12**	**52.87**	**34.77**
UPG (DPCD)+PLR	**74.23**	**65.14**	**56.67**

Table 7. Average Orientation Similarity (AoS) results of orientation estimation (OE) on validation set of KITTI (Car).

Proposal Generation Strategy	Car		
	Easy	Moderate	Hard
UPG w/o OE	65.13	48.82	43.96
UPG	**76.92**	**70.54**	**65.74**

density D proposed in Formula (1) as $D = N/(R_c^2)$. The second one, distance-based point cloud density (DPCD), estimates the density D as Formula (1). As it is shown in the first two rows of Table 6, UPG(DPCD) performs better than UPG(PCD). This justifies the design choice of the UPG module, which can address the interference caused by distance in point cloud density computation. Further, to enhance the accuracy of pseudo-labels, the PLR module employs confidence-threshold and clustering-based methods to refine predictions generated by the teacher model. We conduct experiments on the KITTI(Geiger et al., 2012) validation set. As shown in the third row of the table, the UPG(DPCD)+PLR method performs better than using the UPG(DPCD) module alone. This indicates the PLR module is important for good performance.

For object orientation estimation, we employ the AOS evaluation metric to validate the accuracy in predicting object direction. As shown in Table 7, without using the object orientation estimation module, the performance of the UPG module drops.

5. Conclusion

In this work, we present an efficient approach for weakly supervised LiDAR point cloud vehicle detection, which consists of two main modules. First, the unsupervised proposal generation (UPG) module generates proposals based on the point cloud geometry of vehicles without any annotations. Second, the pseudo-label refinement (PLR) module initially refines pseudo-labels based on the confidence threshold and then uses clustering methods to obtain accurate pseudo-labels for model training, effectively reducing the dependence on labeled data. We adopt the mean-teacher paradigm to train a point cloud detector. Experimental results based on LiDAR-based KITTI and ONCE datasets show that the proposed method demonstrates superior performance compared to existing approaches. Thus, our research can contribute to reducing the annotation workload and advancing the development of 3D object detection.

References

Caine, B., Roelofs, R., Vasudevan, V., Ngiam, J., Chai, Y., Chen, Z., Shlens, J., 2021. Pseudo-labeling for Scalable 3D Object Detection. *Cornell University - arXiv,Cornell University - arXiv*.

Chen, Z., Li, Z., Wang, S., Fu, D., Zhao, F., 2023. Learning from noisy data for semi-supervised 3d object detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6929–6939.

Chen, Z., Li, Z., Zhang, S., Fang, L., Jiang, Q., Zhao, F., 2022. Bevdistill: Cross-modal bev distillation for multi-view 3d object detection. *arXiv preprint arXiv:2211.09386*.

Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H., n.d. Voxel r-cnn: Towards high performance voxel-based 3d object detection. *AAAI*.

Fischler, M. A., Bolles, R. C., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6), 381–395.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. *CVPR*, 3354–3361.

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O., 2019. Pointpillars: Fast encoders for object detection from point clouds. *CVPR*, 12697–12705.

Lee, D.-H., 2013. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on challenges in representation learning*, 3number 2, 896.

Li, Z., Wang, F., Wang, N., 2021. Lidar r-cnn: An efficient and universal 3d object detector. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7546–7555.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. *CVPR*, 2980–2988.

Liu, C., Gao, C., Liu, F., Liu, J., Meng, D., Gao, X., 2022. SS3D: sparsely-supervised 3d object detection from point cloud. *CVPR*, 8418–8427.

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 e-prints*, arXiv:2106.11037.

Meng, Q., Wang, W., Zhou, T., Shen, J., Van Gool, L., Dai, D., 2020. Weakly supervised 3d object detection from lidar point cloud. *ECCV*, Springer, 515–531.

Peng, L., Yan, S., Wu, B., Yang, Z., He, X., Cai, D., 2022. Weakm3d: Towards weakly supervised monocular 3d object detection. *arXiv preprint arXiv:2203.08332*.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*, 652–660.

Qian, R., Lai, X., Li, X., 2022. BADet: Boundary-aware 3D object detection from point clouds. *Pattern Recognition*, 125, 108524.

Qin, Z., Wang, J., Lu, Y., 2020. Weakly supervised 3d object detection from point clouds. *MM*, 4144–4152.

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. *CVPR*, 10529–10538.

Shi, S., Wang, X., Li, H., 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. *CVPR*, 770–779.

Tarvainen, A., Valpola, H., 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NeurIPS*, 30, Curran Associates, Inc., 1195–1204.

Team, O. D., 2020. Openpcdet: An open-source toolbox for 3d object detection from point clouds. `https://github.com/open-mmlab/OpenPCDet`.

Wang, H., Cong, Y., Litany, O., Gao, Y., Guibas, L. J., 2021. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. *CVPR*, 14615–14624.

Wei, Y., Su, S., Lu, J., Zhou, J., 2021. Fgr: Frustum-aware geometric reasoning for weakly supervised 3d vehicle detection. *ICRA*, 4348–4354.

Xu, D., Anguelov, D., Jain, A., 2018. Pointfusion: Deep sensor fusion for 3d bounding box estimation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 244–253.

Yan, Y., Mao, Y., Li, B., 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 3337.

Yin, J., Fang, J., Zhou, D., Zhang, L., Xu, C.-Z., Shen, J., Wang, W., 2022. Semi-supervised 3d object detection with proficient teachers. *ECCV*, Springer-Verlag, Berlin, Heidelberg, 727–743.

Zhao, N., Chua, T., Lee, G., 2020. Sess: Self-ensembling semi-supervised 3d object detection. *CVPR*, 11076–11084.