Exploiting Label Uncertainty for Enhanced 3D Object Detection from Point Clouds

Yang Sun, Bin Lu, Yonghuai Liu, *IEEE Senior Member*, Zhenyu Yang, Ardhendu Behera, *IEEE Member*, Ran Song, *IEEE Senior Member*, Hejin Yuan, Haiyan Jiang

***Abstract*—Accurate detection of objects from LiDAR point clouds is crucial for autonomous driving and environment modeling. However, uncertainties in ground truth labels due to occlusions, sparsity, and truncation can hinder model training and performance. This paper introduces two strategies to address these issues: 1) Soft Regression Loss (SoRL) and 2) Discrete Quantization Sampling (DQS). SoRL utilizes Gaussian distributions for object predictions, measuring uncertainty based on the probability of ground truth labels within these distributions. This method effectively accounts for deviations in object location and orientation. Meanwhile, DQS introduces uncertainty scores for dynamic sample selection, aiming to refine the quality of positive samples for regression. Based on the proposed modules, we design a lightweight multi-stage object detection framework. Notably, these modules can enhance existing 3D object detection methods without affecting significantly inference speeds. Experiments over benchmark datasets show the effectiveness of our method, especially for cars in sparse point clouds.**

***Index Terms*—3D object detection, deep learning, point clouds, soft regression loss, dynamic sample selection**

I. Introduction

Recent advancements in 3D object detection technology, spurred by the growth of autonomous driving and environment monitoring, have significantly enhanced detection accuracy. Nevertheless, the task remains challenging and demands further improvement to fulfill the practical requirements for autonomous driving and environment modeling applications. While datasets with annotations are commonly used as benchmarks for object detection method evaluations, the uncertainty of these annotations is often overlooked, resulting in a negative impact on their performance.

Traditional object detection methods consider bounding box label distributions as Dirac Delta distribution [1], [2] and treat ground truth labels as deterministic regression targets. In 2D image-based detection, uniform pixel-wise labeled objects and dense pixels surrounding the object are employed for localization. In contrast, LiDAR-generated point clouds exhibit inherent challenges such as sparsity, occlusions, and truncation. This results in label uncertainties and ambiguities, and potential gaps in data annotation. Consequently, label uncertainty often adversely affects the training and performance of the detection methods.

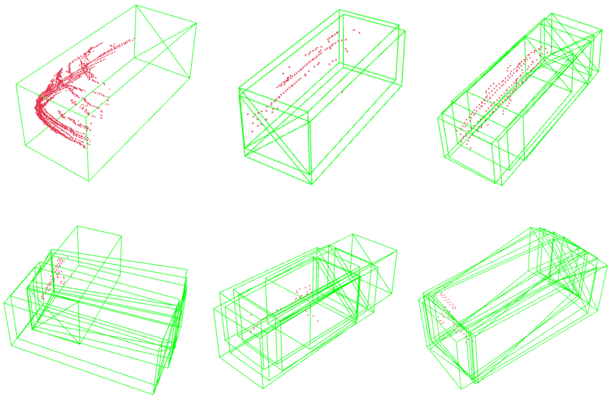


Fig. 1: Uncertainty in the annotation of potential ground truths for the cars in green with various sparse point clouds in red from the KITTI dataset.

To further illustrate the issue of label uncertainty in 3D object detection, Fig. 1 presents several examples from the KITTI dataset [3], where the green box denotes the potential ground truth labels. As point clouds become sparser, the number of plausible ground truth labels increases. Due to the incompleteness of the point clouds, the label thus does not correspond to a definite object/part, complicating the accurate identification of location and parts on the side of the object without any points. To investigate this observation, we statistically analyzed the length and width of the cars in the KITTI dataset, as depicted in Fig. 2. The x axis represents the car length, and the y axis represents the car width. Each point in Fig. 2 demonstrates a ground truth label. Overall, the distribution can be approximated as a two-dimensional Gaussian distribution, which inspired us to represent the predicted bounding boxes as Gaussian distributions. Specifically, Fig. 2 underscores that even the cars with identical lengths can exhibit varied widths following a Gaussian distribution.

Recent advancements categorize 3D object detection methods primarily into single-stage and multi-stage detectors. Single-stage methods extract features directly from the raw point clouds for box classification and regression, achieving high inference speed [4]–[6]. Multi-stage methods [7]–[9] adopt a coarse-to-fine pipeline to enhance performance. They

(Corresponding author: Bin Lu.)

Yang Sun, Bin Lu, Zhenyu Yang, and Hejin Yuan are with the Engineering Research Center of Intelligent Computing for Complex Energy Systems, Ministry of Education, North China Electric Power University, Baoding 071003, China (e-mail: sun.yang@ncepu.edu.cn; lubin@ncepu.edu.cn; yangzhenyu@ncepu.edu.cn; yuanhejin@ncepu.edu.cn).

Yonghuai Liu and Ardhendu Behera are with the Intelligent Visual Computing Research Centre, Edge Hill University, Ormskirk Lancashire L39 4QP, UK (e-mail: yonghuai.liu@edgehill.ac.uk, beheraa@edgehill.ac.uk).

Song Ran is with the School of Control Science and Engineering, Shandong University, Jinan 250100, China (e-mail: ransong@sdu.edu.cn).

Haiyan Jiang is with the College of Artificial Intelligence, Nanjing Agricultural University, Nanjing 210095, China (e-mail: jianghy@njau.edu.cn).

execute a fine-grained feature extraction algorithm on the proposals generated in the first stage and jointly optimize objective functions across all the stages later. Despite the increased computational consumption, such methods usually achieve superior performance compared to the aforementioned single-stage ones. Although these methods utilize different feature extraction techniques and employ discretized location deviations for regression, they often do not account for the ambiguity caused by label uncertainty.

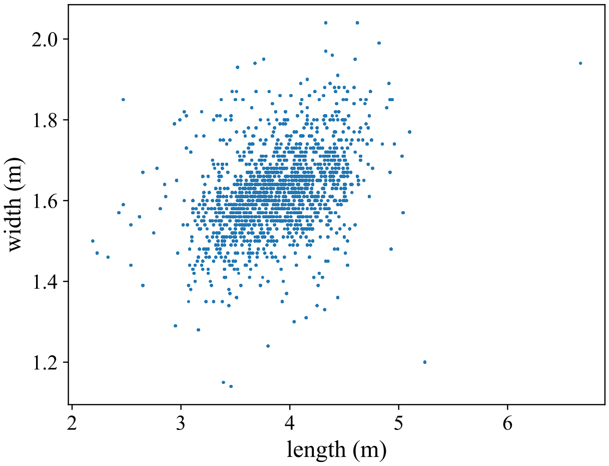


Fig. 2: The statistics in length and width of the cars in the KITTI dataset.

Despite their excellent performance, there still exists room for improvement in terms of incorporating uncertainty information into the object detection process. While recent researchers tap into uncertainty information for 2D image-based object detection enhancement [10], such methods are unsuitable for 3D point cloud data, primarily due to inherent sparsity and unevenness and lack of order in points. To address these issues, new methods are needed to model the uncertainty information in 3D point clouds for more robust and accurate object detection. However, it is challenging to incorporate such uncertainty information into the multi-stage methods.

To handle the problem of label uncertainty in the multi-stage method, we propose a novel Soft Regression Loss (SoRL) designed to model the uncertainty of predicted localization. This innovative metric computes the similarity between the predicted bounding box and its corresponding ground truth while taking the uncertainty information into account. To consider the impact of uncertainty on sample selection, we then propose the Discrete Quantization Sampling (DQS) method. DQS can adjust the Intersection over the Union (IoU) threshold dynamically, which makes the selection of positive and negative samples more balanced and diverse. Specifically, DQS incorporates the uncertainty information into the confidence scores and sorts the proposals accordingly. Then, the mean and the variance of the IoU distribution are employed to dynamically adjust the threshold, thereby ensuring that the features of various objects are adequately learned. By combining SoRL and DQS, our method achieves significant improvements in handling annotation uncertainty and enhancing localization accuracy while effectively reducing the impact of low-quality samples.

Our proposed method offers a new perspective on enhancing 3D object detection performance in LiDAR point clouds by incorporating annotation uncertainty information into the object detection framework. The experimental results show that the consideration of annotation uncertainty can significantly improve the object detection performance. We believe that our work will shed light on the future development of 3D object detection methods and inspire further studies in this field.

In summary, the main contributions of our work are as follows:

1) We propose an end-to-end framework that takes label uncertainty into account for accurate object detection in point clouds.
2) We propose the SoRL approach, which treats the object predictions as distributions for the measurement of their uncertainty to obtain a more robust regression loss.
3) We propose a DQS module that can dynamically select samples within proposals by leveraging uncertainty scores to acquire higher-quality positive samples for regression.
4) Both the comparison with the state-of-the-art and ablation studies show the efficacy and potential of the proposed methods for object detection under various scenarios.

II. Related Works

To address the challenges posed by occlusions, sparsity, and truncation of LiDAR point clouds for object detection, there are two primary strategies for point cloud analysis: point-based, and voxel and projection-based methods.

Point-based methods. Hierarchical sampling and feature aggregation of raw points with accurate coordinates are typically involved in point-based methods. The advantage is that the point clouds can be aggregated into a specific number of points, making it possible to design various feature extraction algorithms. Early works in this area include PointNet [11] and PointNet++ [12], which map points into high-dimensional feature space and gradually aggregate them through hierarchical downsampling. Point-RCNN [13] uses a Farthest Point Sampling (FPS) to aggregate points, and then refines proposals by learning the semantic features of each point using foreground points. 3DSSD [14] improves point sampling by using farthest feature distance sampling to better preserve foreground points, while IASSD [15] leverages the semantics of each point for downsampling from different categories.

Some recent works have integrated attention mechanisms to extract more robust features. For example, PointFormer [16] employs the Transformer architecture [17] as its backbone network to capture the dependencies between both local and global point features. CT3D [18] applies the Transformer in its second-stage refinement process and introduces the channel-wise attention mechanism to extract more sophisticated features after random point sampling in the Regions of Interest (RoIs).

The key challenge faced by point-based detectors is how to balance efficacy and efficiency. Increasing the ball query radius

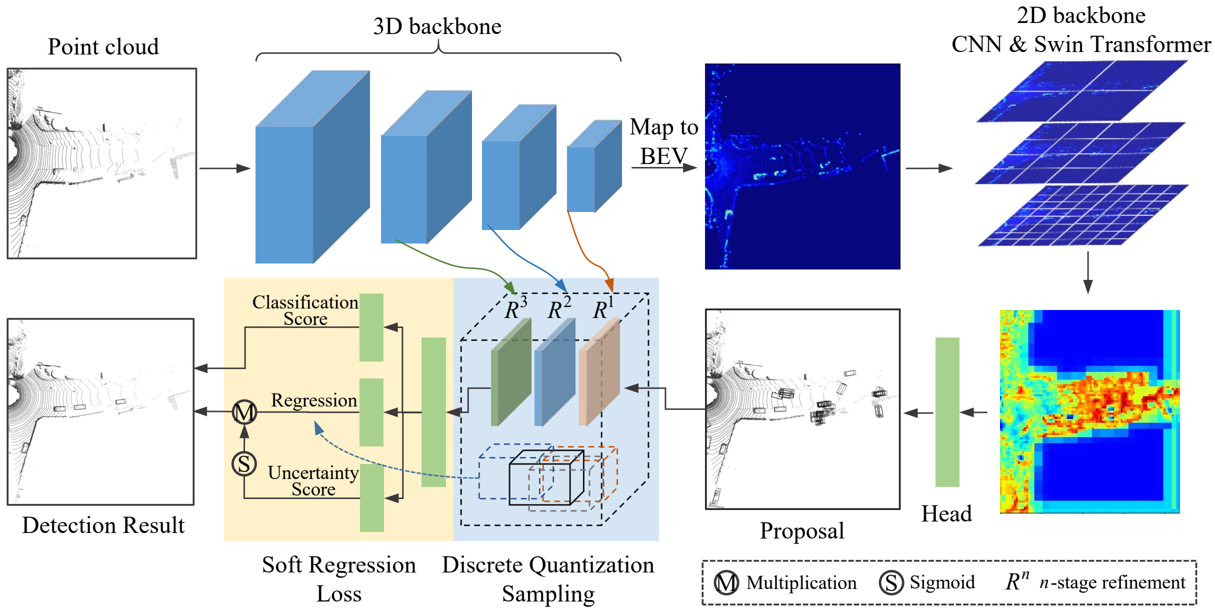


Fig. 3: Network structure of our proposed method with the usage of label uncertainty information.

for point aggregation can improve contextual information capture but lead to low inference speed and increased memory consumption. For example, FPS is a common sampling method, but the distance-based principle makes it computationally inefficient. To address this challenge, 3DSSD and IASSD have improved their point sampling methods to preserve more foreground points without significantly increasing computational consumption. As multi-line LiDAR technology continues to evolve, the number of points in scenes is expected to increase, further exacerbating computational consumption. The selection of an appropriate point sampling method is critical to ensure the effectiveness of the sampled points and extracted features for object representation.

Voxel and projection-based methods. To reduce sparsity and improve computational efficiency, voxel-based methods convert raw points into voxels with regular and discretized spatial distributions. Voxels are then processed using 3D Convolutional Neural Networks (CNNs) for the extraction of semantic information and the generation of object proposals. VoxelNet [19] is a pioneering work, that employs 3D convolution for feature extraction, and then compresses features to Bird's Eye View (BEV) for the generation of object proposals. Similarly, SECOND [4] leverages 3D sparse convolution to improve efficiency through processing only non-empty voxels. Another method, PointPillars [5], converts point clouds into 2D pseudo-images and performs feature extraction using 2D convolution. These advances have inspired the development of several voxel-based two-stage detectors. For example, Voxel R-CNN [7] builds upon SECOND as the first-stage backbone network and aggregates multi-scale voxel features for refinement in the second stage. Meanwhile, CIA-SSD [20] fuses multi-layer spatial and semantic features and supervises predictions using IoU. Furthermore, Voxel-FPN [21] aggregates voxel features of different sizes, demonstrating improved performance over the methods using a single voxel size.

The challenge of voxel and projection-based methods is the loss of precise spatial information and the increase in computational complexity and storage memory. After converting points into voxels or pixels, the precise coordinate information of each point is inevitably lost, which affects the accuracy of the detected objects. Additionally, the computational cost of 3D CNNs is significantly higher than that of 2D counterparts. Therefore, effectively reducing computational complexity and preserving precise spatial information in point clouds is a crucial issue for these methods.

Label uncertainty-based methods. Several methods have been proposed to address label uncertainty. For instance, General Focal Loss (GFL) [22] proposes Distribution Focal Loss (DFL) to study the discretized probability distribution of bounding boxes, resulting in more robust and accurate bounding box estimates. He et al. [23] model the predictions and ground truth boxes as Gaussian distributions and Dirac Delta functions, and then calculate regression loss with the KL divergence. Xu et al. [24] treat the bounding boxes as two-dimensional Gaussian distributions and calculate the similarity between predictions and corresponding ground truth using the proposed Normalized Wasserstein Distance (NWD). Choi et al. [25] model the bounding box coordinates as Gaussian parameters to estimate localization uncertainty for box regression. Yang et al. [26] regard the arbitrarily oriented bounding boxes as Gaussian distributions, then use the proposed Gaussian Wasserstein Distance (GWD) as IoU loss for object detection. GLENet [27] first estimates a distribution from the dataset and then calculates the KL divergence between the predicted distribution and the target distribution as the loss. However, due to the sparsity of point clouds and significant differences in data distribution, directly calculating the differences between distributions relies on prior knowledge of the dataset distribution and can lead to difficulties in model training.

Considering the complexity of the 3D object detection task and the characteristics of the point cloud data, there are limitations in using uncertainty information to measure the similarity of ground truth and predictions. The distribution of bounding boxes can vary significantly due to differences in object size, aspect ratio, orientation, and the level of occlusion and truncation. This wide disparity in bounding box distributions can pose challenges for object detection from point clouds.

III. Method

In this section, we detail the proposed SoRL and DQS modules, which are part of the framework, as shown in Fig. 3. Our framework is a voxel-based lightweight multi-stage detector consisting of a 3D backbone network, a 2D backbone network, and a detection head. Firstly, the point clouds are voxelized through quantization and fed into the 3D sparse CNN for feature extraction. Then, the 3D sparse feature is mapped to BEV for the generation of object proposals using the 2D backbone network. Subsequently, the proposed SoRL module is then used to handle the label uncertainty of each bounding box. The DQS module is employed to select samples from the proposals. Finally, an additional uncertainty loss is added to improve the performance of the box regression branch. Each of these modules will be explained individually in the following sections.

A. Symbol Definition

We define the point cloud as $P = \{p_i\}_1^N$, where $p_i = \{x_i, y_i, z_i, r_i\}$ represents each point with 3-dimensional coordinates x, y, and z, and an additional feature r such as reflectivity. i denotes the index of each point within a point cloud. The predicted bounding box B_R and ground truth bounding box B_G can be formulated as $\{x, y, z, l, w, h, \theta\}$, where x, y, and z denote the box center coordinates; l, w, and h represent length, width, and height of the box relative to its center, respectively; θ denotes the rotation of the 3D box relative to the x axis. We denote the predicted box as $\{x_R, y_R, z_R, l_R, w_R, h_R, \theta_R\}$ and the ground truth box as $\{x_G, y_G, z_G, l_G, w_G, h_G, \theta_G\}$. The subscript notation R represents the output of the regression branch in the detection head, and G denotes the ground truth.

B. Backbone network

Voxelization and encoding. Point clouds are characterized by disorder, irregularity, and permutation invariance, making it challenging to extract features by using traditional convolutional operations. The voxel-based method first divides the irregular point cloud space into uniformly stacked voxels and processes the point cloud at the voxel level. Then, 3D convolution is utilized to extract multi-scale spatial features. Initially, the point cloud is voxelized as $R^{D\times W\times H}$ via quantization, where $D = \lfloor\frac{x_{max}-x_{min}}{v_x}\rfloor$, $W = \lfloor\frac{y_{max}-y_{min}}{v_y}\rfloor$, and $H = \lfloor\frac{z_{max}-z_{min}}{v_z}\rfloor$ represent the number of voxels along x, y and z directions, respectively. v_x, v_y, and v_z denote the sizes of each voxel along the three directions. During the training process, the number V of total voxels is limited to a maximum of 16,000 following [7], [8].

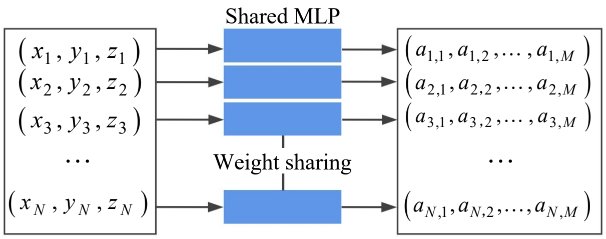

Shared MLP
(x_1, y_1, z_1)
(x_2, y_2, z_2)
(x_3, y_3, z_3)
$\cdots$
(x_N, y_N, z_N)
Weight sharing
$(a_{1,1}, a_{1,2}, \ldots, a_{1,M})$
$(a_{2,1}, a_{2,2}, \ldots, a_{2,M})$
$(a_{3,1}, a_{3,2}, \ldots, a_{3,M})$
$\cdots$
$(a_{N,1}, a_{N,2}, \ldots, a_{N,M})$

Fig. 4: Shared MLP for point encoding

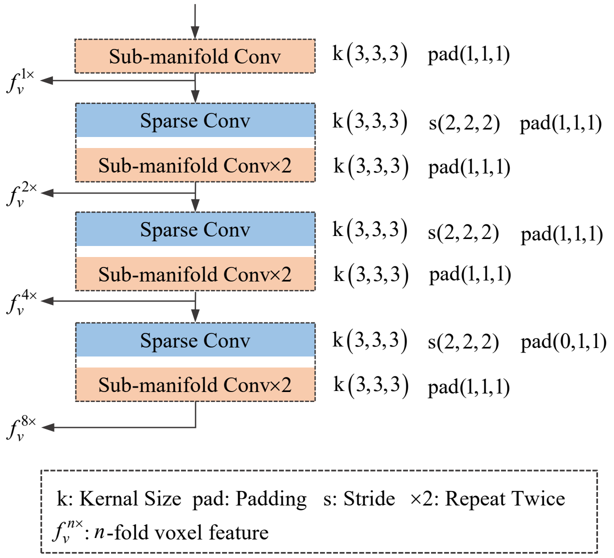


Fig. 5: Efficient sparse and sub-manifold convolution blocks with multi-scale downsamplings at different resolutions.

Although the point clouds were filtered according to the range $\{x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max}\}$ during the preprocessing stage, the space still contains a large number of non-uniformly distributed points. Moreover, after voxelizing the point clouds, the number of points contained in each voxel grid is uneven. To reduce the computational complexity of the voxel encoding, we limit the number of points contained in each voxel. If the number exceeds the threshold T, then T points will be randomly sampled, and the unselected points will be discarded. Conversely, if the number is less than T, it will be padded with zeros to complete. Since the lists of points contained in each voxel are initialized with 0, we record the number of points in each voxel, which can serve as a mask to ensure that the max-pooling is performed only on the original points. This helps maintain the order invariance of the point cloud.

The coordinates of individual points are encoded through a shared Multilayer Perceptron (MLP), and then mapped to a high-dimensional feature space, as shown in Fig. 4, where $\{a_{i,1}, a_{i,2} \ldots a_{i,M}\}$ represents the feature vector after feature transformation.

3D backbone. We apply four convolution blocks consisting of sub-manifold convolution and 3D sparse convolution to learn multi-scale voxel features, as shown in Fig. 5. We utilize four downsampling blocks to obtain multi-scale dense features. Each block consists of several sparse convolution layers and sub-manifold layers. Sub-manifold convolution is used for channel transformation of voxel features to maintain the sparsity of the point clouds, and 3D sparse convolution only performs convolution calculations over non-empty voxels, which is known to improve convolutional efficiency. Therefore, each block first performs sparse convolution to obtain features at different scales and then applies channel transformations with several sub-manifold convolution layers to enhance the non-linear representation of the features. The output features from each downsampling block can be denoted as $f_v = \{f_v^{1\times}, f_v^{2\times}, f_v^{4\times}, f_v^{8\times}\}$, where $f_v^{n\times}$ represents multi-scale voxel features by n-fold downsampling. For example, $f_v^{1\times}$ typically contains more geometric information and $f_v^{8\times}$ are more semantically informative. Each scale feature is obtained by downsampling the previous feature by a factor of 2. Therefore, our 3D backbone network is capable of obtaining features from different levels to form a more comprehensive representation of the point clouds.

Subsequently, as suggested in [4], we compress $f_v^{8\times}$ onto BEV as the map-view features, and then feed them into the 2D backbone network to generate object proposals.

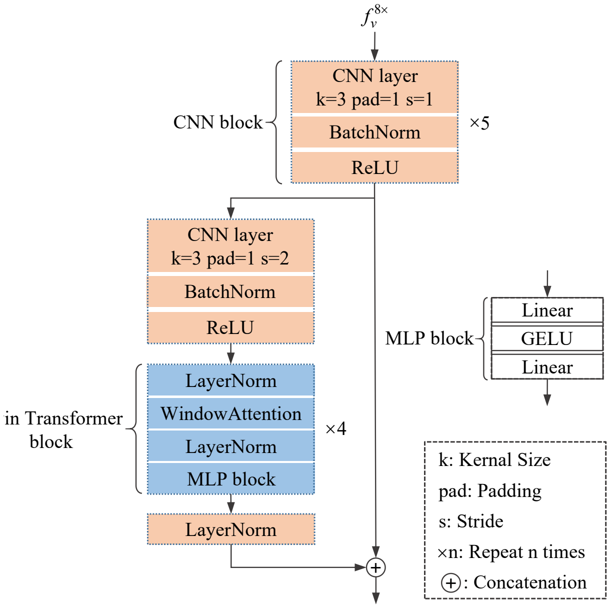


Fig. 6: 2D backbone network structure.

2D backbone. Our 2D backbone network architecture consists of two main components: the CNN feature extraction module and the Swin Transformer [28] as a global feature aggregation module, as shown in Fig. 6. The CNN feature extraction module comprises a series of convolutional layers that extract local features from the BEV feature map. The output of the CNN feature extraction module is then fed into the Swin Transformer to capture their global dependencies with the self-attention mechanism. Specifically, the output feature maps of the CNN module and the Swin Transformer module can be denoted as $F_{CNN}(x)$ and $F_{ST}(x)$, respectively. We use a skip-connection between the two feature extraction modules for feature aggregation, which can be represented as $F_{cat} = [F_{CNN}(x), F_{ST}(x)]$, where $[\cdot]$ denotes the concatenation operation.

C. Soft regression loss (SoRL)

Traditional methods often treat the regression target as a Dirac Delta distribution [7], [8], as illustrated in Fig. 7. This means that the coordinates of both the predicted and ground truth boxes are treated as fixed values, even though objects of the same size may result in different regression losses due to sparsity or truncation in point clouds. In this section, we propose a Gaussian distribution-based soft regression loss to assist in training the regression branch that consists of a probabilistic object detection head and a probability weighted regression loss function.

1) Probabilistic Object Detection Head: This method can be employed to quantify the similarity between the predicted bounding box and its corresponding ground truth, thus mitigating the inherent uncertainty in the data annotation process from point clouds. The current 3D detection paradigm is inspired by Faster-RCNN [29], which consists of a classification branch and a regression branch.

The regression loss $\{\Delta x, \Delta y, \Delta z, \Delta l, \Delta w, \Delta h, \Delta\theta\}$ of the target B_R relative to the ground truth B_G can be calculated as follows:

$$\begin{aligned}
&\Delta x = \frac{x_G - x_R}{d}, \Delta y = \frac{y_G - y_R}{d}, \Delta z = \frac{z_G - z_R}{h},\\
&\Delta l = log\frac{l_G}{l_R}, \Delta w = log\frac{w_G}{w_R}, \Delta h = log\frac{h_G}{h_R},\\
&\Delta\theta = \theta_G - \theta_R,
\end{aligned} \tag{1}$$

where $d = \sqrt{l^2 + w^2}$ is the diagonal length of the ground truth bounding box. l_R, w_R, and h_R denote the output length, width, and height from the regression branch respectively.

The regression loss is calculated by comparing the predictions and regression targets using the commonly employed smooth-L1 function for backward propagation.

The common regression branch outputs the estimated coordinates of the bounding boxes, which does not consider the uncertainty information from the ground truth. Directly learning the uncertainty from the ground truth labels presents a challenge, as it requires additional data statistics from the dataset and hinders the end-to-end training of the model.

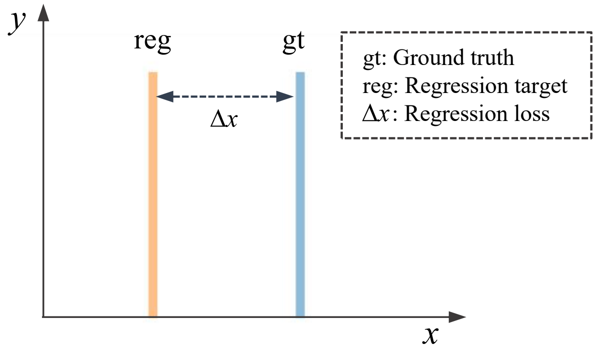


Fig. 7: Dirac delta distribution based target regression.

To elegantly incorporate uncertainty information into our framework and allow it to be easily integrated into other detectors, we formulate the uncertainty of the predicted bounding box as the Gaussian distribution D_R. The label uncertainty is expressed as the probability of appearance within the predictive distribution. This method provides a quantitative metric of similarity between the predicted box and the ground truth, as shown in Fig. 8. The uncertainty can be modeled as:

$$\rho(B_G|D_R) = \rho(B_G|\mathcal{N}(\mu, \sigma)), \tag{2}$$

where ρ denotes the probability density, and $\mathcal{N}$ denotes the 2D Gaussian distribution. When focusing specifically on the x-dimension in the ground truth, the equation can be formulated as:

$$\mathcal{N}(\mu_x, \sigma_x) = \frac{1}{\boldsymbol{\sigma_x}\sqrt{2\pi}} e^{-\frac{(x-\boldsymbol{\mu_x})^2}{2\boldsymbol{\sigma_x}^2}}, \tag{3}$$

where $\boldsymbol{\mu_x}$ and $\boldsymbol{\sigma_x}$ denote the predicted mean and standard deviation of the target in the x-dimension, respectively. The formulations of the other y, z, l, w, h and θ dimensions are similar.

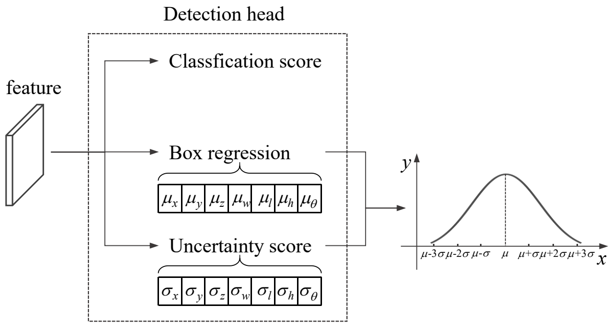

Detection head
feature
Classfication score
Box regression
μ_x μ_y μ_z μ_w μ_l μ_h μ_θ
Uncertainty score
σ_x σ_y σ_z σ_w σ_l σ_h σ_θ
y
x
μ-3σ μ-2σ μ-σ μ μ+σ μ+2σ μ+3σ

Fig. 8: Detection Head.

Specifically, the output vector of the regression branch is considered as the expectation μ, which consists of a convolutional layer with a kernel size of 1×1 and a fully connected layer. We incorporate an additional branch in the detection head to predict the parameter σ as the uncertainty score identical to the regression branch, denoted as $\{\sigma_x, \sigma_y, \sigma_z, \sigma_l, \sigma_w, \sigma_h, \sigma_\theta\}$, which corresponds to each value in R, and then estimate the uncertainty of the bounding box by the distribution of its center and size instead of corner points, which can better handle variations in object size, aspect ratio, and orientation. As a result, it becomes easier to achieve more accurate and reliable object detection across different scenarios and integrate it into existing object detection frameworks.

2) Probability weighted regression loss: After obtaining μ and σ, we can derive the distribution of the predicted box, and then calculate the probability of the ground truth B_G in this distribution as the weight of the regression loss. Finally, we normalize the probability with the Softmax function:

$$\rho_s = Softmax(\rho(B_G|D_R)), \tag{4}$$

where ρ_s is used to weight the regression loss in order to mitigate the problem of being unable to dynamically adjust based on label uncertainty. Taking the x dimension as an example, as shown in Fig. 9, Δx represents the Euclidean distance between the ground truth and prediction. We replace Δx with $\rho_s \Delta x$ as the final regression loss to train the regression branch more effectively.

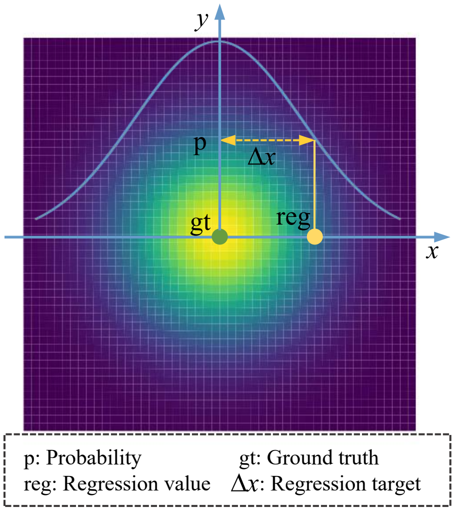


Fig. 9: Diagram of soft regression loss based on Gaussian distribution.

D. Discrete Quantization Sampling (DQS)

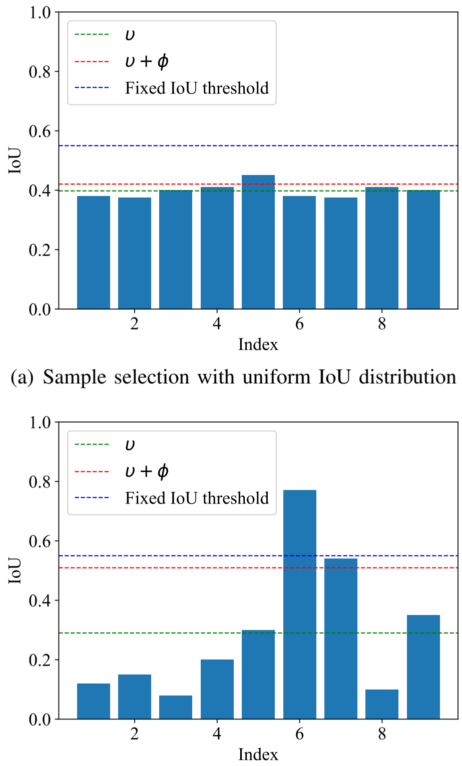
During the training process, the proposals are divided into positive and negative samples for classification and regression, while only positive samples are used for regression. Conventional methods usually use a fixed IoU threshold to perform sample selection. Proposals with IoU greater than a certain positive threshold (t_1) are considered positive, while those with IoU less than a certain threshold (t_2) are considered negative. Usually, t_2 is smaller than t_1, and remaining unselected proposals are ignored. Subsequently, the positive samples are used in the calculation of the regression loss for the improvement of the localization accuracy. Thus, the quality of positive samples is critical for determining the performance of the model.

However, the fixed IoU threshold cannot perform effective sample selection for proposals generated from objects with different sparsities. Sparsity refers to the density of data points or samples in a given space for the representation of an object as a whole or its parts. Different sparsities imply how unevenly the data points are distributed in the RoI. To address this issue, we propose the Discrete Quantization Sampling (DQS) module, which incorporates a discretized sampling strategy into the training process.

The uncertainty estimation vector σ is encoded by fully connected layers (FC) and a non-linear activation function to generate proposal-wise quality scores. Subsequently, the classification scores p_{cls^*} are recalculated by adding the proposal-wise quality scores to the original ones p_{cls}:

$$p_{cls^*} = \mathfrak{N}(\mathcal{A}(\mathcal{F}(\sigma)) + p_{cls}), \tag{5}$$

where $\mathfrak{N}$ denotes a normalization function, $\mathcal{A}$ denotes the Sigmoid activation function and $\mathcal{F}$ denotes the FC.



(a) Sample selection with uniform IoU distribution

(b) Sample selection with uneven IoU distribution

Fig. 10: Sample selection with DQS module.

The proposals are firstly sorted by the p_{cls^*} to obtain the valid proposals associated with each ground truth. Then, we dynamically calculate the mean and variance of 3D IoUs for the proposals belonging to the same ground truth as υ and ϕ. Finally, we take $\tau = \upsilon + \phi$ as the final IoU threshold of each ground truth to select positive samples for each point cloud. So, proposals with IoU greater than τ can be marked with positive samples for the calculation of the regression loss.

Our motivation for introducing DQS in sample selection is to filter out proposals with low-quality scores and low IoU. These low-quality proposals significantly decrease the average IoU value, resulting in the selection of an excessive number of positive samples with low IoU. To capture features of objects with varying sparsity, we assign IoU thresholds for each ground truth to maintain diversity among positive samples. Half of the IoUs are greater than υ in Gaussian distribution. When the points within an object are sparse, the variance of the IoU distribution increases. To account for this, we introduce ϕ as a measure of sparsity, which adjusts the IoU threshold accordingly, preventing the learning of invalid features. This further enhances the quality of positive samples.

Fig. 10 presents the sample selection strategies for objects with both dense and sparse point clouds. The green line denotes υ, representing the mean IoU associated with the same ground truth. The red line denotes τ, calculated by $\upsilon+\phi$. The blue line represents the fixed IoU threshold, widely used in the previous methods. We set the threshold to 0.55 for comparison, which is a commonly used threshold in empirical studies. As shown in Fig. 10(a), the proposals for objects with dense point clouds usually have a more uniform IoU distribution. If the fixed IoU threshold is set unreasonably high, all the proposals for some ground truths will be treated as negative samples, making it impossible for the models to improve their localization ability. Nevertheless, when using $\upsilon+\phi$, the boxes with the highest IoU among all the proposals belonging to the same ground truth can be selected as the positive samples. This increases the diversity of the positive samples, which is beneficial for the model to learn features of different objects. Fig. 10(b) presents another object with a sparser point cloud, where the variance of IoUs is larger than that in Fig. 10(a). Although $\upsilon+\phi$ is closer to the fixed IoU threshold, the former selects two proposals with the highest IoU as positive samples, while the latter selects only one. This demonstrates that our DQS can select more balanced positive samples adaptively for each ground truth.

E. Detection head and loss function

1) Detection head: In this section, we introduce the detection head utilized in our proposed method, which consists of classification, regression, and uncertainty estimation branches. The classification branch employs the Binary Cross Entropy (BCE) as the loss function, while the regression branch adopts the smooth-L1 loss function. The overall loss is a weighted combination of these components. The classification branch generates a probability distribution over a set of discrete classes, while the regression branch outputs the offset of the coordinates between the proposals and the ground truth according to Equation 1.

2) Loss function: For the classification task, we employ the BCE loss, a widely used loss function for binary classification problems. Given the predicted classification score B_R^{cls} and the ground truth class label B_G^{cls}, the BCE loss is defined as:

$$L_{cls} = \sum -(B_G^{cls} * log(B_R^{cls}) + (1 - B_G^{cls}) * log(1 - B_R^{cls})). \tag{6}$$

This loss function effectively penalizes misclassifications and encourages the model to output probabilities closer to the ground truth labels, thereby improving the classification performance.

We adopt the smooth-L1 loss function for the regression task, which can be formulated as:

$$\begin{aligned} L_{reg} &= smooth - L1(B_R^{reg}, B_G^{reg}) \\ &= \begin{cases} 0.5 * (B_R^{reg} - B_G^{reg})^2, & if |B_R^{reg} - B_G^{reg}| < 1, \\ |B_R^{reg} - B_G^{reg}| - 0.5, & otherwise, \end{cases} \end{aligned} \tag{7}$$

where B_R^{reg} denotes the predicted proposal position and B_G^{reg} denotes the ground truth localization label. Then, we weight L_{reg} with ρ_s to obtain our soft regression loss L_{sorl} as:

$$L_{sorl} = \rho_s * L_{reg}. \tag{8}$$

The overall loss function for our proposed detection head is a weighted combination of L_{cls} and L_{sorl}. The weights w_{cls} and w_{sorl} are hyperparameters that can be adjusted according to the specific task and dataset requirements. We use 1 as the weight here following [7], [8]. The overall loss function is thus defined as:

TABLE I: 3D AP and mAP in percentage (%) of the original methods and their SoRL-enhanced counterparts.

Method	3D AP recall 40			
	Easy	Mod.	Hard	mAP
PV-RCNN [8]	92.11	84.39	82.50	86.33
SoRL-PV(Ours)	**92.27**	**85.22**	**83.15**	**86.88**
Voxel R-CNN [7]	92.38	85.29	82.86	86.84
SoRL-V(Ours)	**92.52**	**85.39**	**83.04**	**86.98**
CT3D [18]	92.85	**85.82**	83.46	87.38
SoRL-C(Ours)	**94.21**	85.41	**84.79**	**88.13**

$$L_{total} = w_{cls} * L_{cls} + w_{sorl} * L_{sorl}. \tag{9}$$

TABLE II: AP and mAP in percentage (%) of different methods for the car detection on the KITTI test set over 40 recall positions. All results are reported from the KITTI official benchmark. The top-1 result is in bold and the second is marked with an underscore.

Modality	Method	Car - 3D Detection			
		Easy	Mod.	Hard	mAP
Single-stage	CenterNet3D [6]	88.23	79.23	75.34	80.93
	3DSSD [14]	88.36	79.57	74.55	80.83
	SA-SSD [30]	88.75	79.79	74.16	80.90
	SE-SSD [31]	91.49	82.54	77.15	83.73
	CIA-SSD [20]	89.59	80.28	72.87	80.91
	3D-CenterNet [32]	86.83	80.17	75.96	80.99
	IA-SSD [15]	88.87	80.32	75.10	81.43
	VIC-Net [33]	88.25	80.61	75.83	81.56
	HVPR [34]	86.38	77.92	73.04	79.11
	SECOND [4]	84.65	75.96	68.71	76.44
	PointPillars [5]	82.58	74.31	68.99	75.29
	PVB-SSD [1]	89.99	80.68	76.23	82.30
	3D IoU Loss [35]	86.16	76.50	71.39	78.02
	Associate-3Ddet [36]	85.99	77.40	70.53	77.97
Multi-stage	Point RCNN [13]	86.96	75.64	70.70	77.77
	CT3D [18]	87.83	81.77	77.16	82.25
	Graph-Po [37]	91.79	83.18	77.98	83.45
	Fast Point R-CNN [38]	85.29	77.40	70.24	77.64
	STD [39]	87.95	79.71	75.09	80.92
	PV-RCNN [8]	90.25	81.43	76.82	82.83
	FV2P [40]	88.53	81.58	77.37	82.49
	BADet [41]	89.28	81.61	76.58	82.49
	VPGA [42]	90.97	81.62	76.90	83.16
	Focals Conv [43]	90.20	82.12	77.50	83.27
	GraR-Vo [37]	91.29	82.77	77.20	83.75
	3D Cascade RCNN [44]	90.46	82.16	77.31	83.31
	CasA [45]	91.58	83.06	**80.08**	**84.91**
	OcTr [46]	90.43	81.86	77.36	83.22
	3D HANet [47]	90.79	**84.18**	77.57	84.18
	GLENet [27]	91.67	83.23	78.43	84.44
	Ours	**91.83**	82.95	78.12	84.30

IV. Experiments

In this section, we first introduce the experimental setup and training details of the proposed method. Then, we report comparisons with the SOTA methods on both the KITTI validation and test sets and the nuScenes dataset [50]. After that, we integrate SoRL into several popular baseline models for the revelation of its effectiveness and universality. Finally, we conduct ablation studies on the KITTI benchmark for the demonstration of the validity of the proposed modules.

A. Experiment Settings

1) Dataset: The KITTI dataset contains 7,481 training images/point clouds and 7,518 test images/point clouds with several categories such as car, pedestrian, and cyclist. The level of difficulty is classified as easy, moderate, and hard according to the number of points contained in the object, occlusion, and truncation level of each category. A common way to split the training images/point clouds results in a training set containing 3,712 point clouds and a validation set containing 3,769 point clouds. We train our model on the training set, and then conduct experiments on the validation and test sets, respectively. For a fair comparison, we use the average precision (AP) recommended by the official KITTI with both 11 and 40 recall positions to evaluate performance.

The nuScenes dataset [50] presents a greater level of difficulty for autonomous driving compared to other datasets. It comprises 380,000 LiDAR sweeps gathered from 1,000 scenes, with annotations for up to 10 object categories, including 3D bounding boxes, object velocity, and attributes, across the full 360° detection range, which is a considerable improvement over the 90° range offered by KITTI. This dataset contains 1,000 scenes for multiple object categories such as cars, pedestrians, cyclists, and so on. We use the official metrics for the evaluation of our method, including mean average precision (mAP), which is similar to the KITTI dataset, and nuScenes Detection Score (NDS). The NDS is formulated as a weighted sum of a range of metrics, including the mean average precision (mAP), thus providing a more comprehensive evaluation of object detection methods.

2) Setup Details: For the KITTI dataset, the detection range of point clouds is set as $[0, 70.4]$m, $[-40.0, 40.0]$m, and $[-3.0, 1.0]$m on x, y, and z axes, respectively, and the voxel size is set as $(0.05, 0.05, 0.05)$m. For the nuScenes dataset, the detection range of point clouds is set as $[-51.2, 51.2]$m, $[-51.2, 51.2]$m, and $[-5.0, 3.0]$m on x, y, and z axes, respectively, and the voxel size is set as $(0.1, 0.1, 0.2)$m instead. We conduct all the experiments based on the OpenPCDet[1] toolbox.

3) Backbone Network: We build a multi-stage LiDAR-based 3D object detection framework. Following previous works, we voxelize the point clouds via quantization for resolution reduction, and points within a voxel are represented by the average coordinates. Subsequently, the voxels are fed into the 3D sparse CNN for feature extraction following [4] which can obtain multi-dimensional features, and then the 3D sparse features are mapped to BEV for the generation of the object proposals through a 2D backbone network. Motivated by the success of Swin Transformer in 2D image downstream tasks, we incorporate it into our 2D backbone for feature extraction.

4) Training and Inference: We train the model for 80 epochs on 8 RTX 3090 GPUs with a batch size of 2. During the training process, we use $Adam\_onecycle$ as the optimizer, set the learning rate as 0.001, the division factor as 10, the momentum ranging from 0.95 to 0.85, and the weight decay as 0.01.

[1] https://github.com/open-mmlab/OpenPCDet

TABLE III: AP and mAP in percentage (%) of CasA and SoRL for the pedestrian and cyclist detection on the KITTI test set over 40 recall positions.

Method	Pedestrian - 3D Detection				Cyclist - 3D Detection			
	Easy	Mod.	Hard	mAP	Easy	Mod.	Hard	mAP
CasA [45]	48.92	40.29	36.74	41.98	80.99	63.76	57.35	67.37
SoRL(Ours)	48.94	41.17	38.81	42.97	80.26	65.13	58.76	68.05

TABLE IV: Inference speeds in milliseconds (ms) of our and other SOTA methods.

Method	Speed (ms)
PDV [9]	180
PV-RCNN [8]	200
CasA [45]	300
Ours	**175**

To avoid overfitting, four commonly used data augmentation strategies are employed in our model: 1) ground truth sampling; 2) global scaling with a random scaling factor in $[0.95, 1.05]$, global rotation around the z-axis with a random angle in $[-\frac{\pi}{4}, +\frac{\pi}{4}]$; 3) rotating the ground truth with a random angle in $[-\frac{\pi}{2}, +\frac{\pi}{2}]$ around the z-axis to simulate steering; and 4) random flipping along the x-axis.

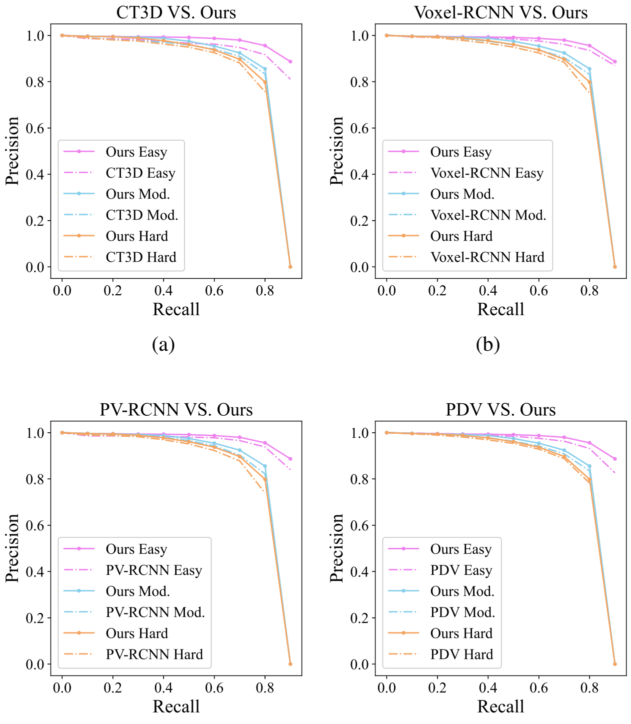
During the inference process, the Non-maximum Suppression (NMS) threshold is set as 0.7 for the car category and 0.5 for cyclists, respectively. We use predicted boxes with the top 160 classification scores as input for the second stage of refinement. When performing post-processing, we remove the predicted boxes with classification scores below 0.55, and then filter redundant predicted boxes with an NMS threshold of 0.1. The inference process is performed on a single RTX 3090 GPU with a batch size of 1.

5) Base Detectors: To demonstrate the universality of SoRL, we integrate SoRL into several advanced 3D object detection frameworks, as shown in Table I. Specifically, an additional detection branch is added to predict the uncertainty information of the box. Then, the original box regression branch is replaced by SoRL to improve the box positioning performance.

B. Experiments on the KITTI Dataset

We submitted our model to the KITTI official server on the test set for evaluation, which allows us to obtain the 3D AP with 40 recall positions for the car detection. Table II reports its performance compared with other advanced LiDAR-based methods. Our method achieves 91.83%, 82.95%, and 78.12% in 3D AP at the easy, moderate, and hard difficulty levels, respectively. It especially outperforms other methods at the easy difficulty level and exhibits competitive results at the moderate and hard difficulty levels.

The relatively superior performance of CasA [45] can be attributed to its multi-stage structure. Table III presents further the detection results of SoRL and CasA in the Pedestrian and Cyclist categories on the KITTI test set, with the method re-implemented as much as possible under the same software and hardware environment for fair comparison. SoRL outperforms CasA with an average precision improvement of 0.99% and 0.68% for the pedestrian and cyclist detection, respectively. However, the advantage of CasA comes at the expense of significantly slower training and inference time as shown in Table IV, limiting its practical applicability. In contrast, our proposed SoRL and DQS modules are employed only during the training phase to assist model training and do not participate in the inference stage. This results in higher performance and ease of integration with other methods without incurring significant computational overhead.



(a)

(b)

(c)

(d)

Fig. 11: The recall-precision curves of different methods for the car detection over the KITTI test set.

Fig. 11 displays the recall-precision curves of our proposed method in comparison to other leading approaches for the car detection over the KITTI test set. Four well-known methods are chosen for this comparison. Voxel R-CNN [7] is a highly-performing object detector based on voxel feature aggregation with a straightforward network architecture, while CT3D [18] is the first technique to employ a Transformer for extracting point-wise features for object detection. PV-RCNN [8] is a pioneering method to merge the benefits of points and voxels for the generation of high-quality proposals. PDV [9] refines the proposals using point cloud density information, representing a prominent work that introduces additional data for object detection. The solid line signifies our method, while

TABLE V: 3D AP and mAP in percentage (%) of different methods for the car detection over the KITTI val set with 11 and 40 recall positions.

Modality	Method	3D AP recall 11				3D AP recall 40			
		Easy	Mod.	Hard	mAP	Easy	Mod.	Hard	mAP
Single-stage	CenterNet3D [6]	86.27	76.45	71.11	77.94	-	-	-	-
	PVB-SSD [1]	-	-	-	-	90.98	82.06	79.34	84.13
	SA-SSD [30]	90.15	79.91	78.78	82.95	93.23	84.30	81.36	86.30
	SE-SSD [31]	90.21	85.71	79.22	85.05	93.19	86.12	83.31	87.54
	CIA-SSD [20]	90.04	79.81	78.80	82.88	-	-	-	-
	3D-CenterNet [32]	86.83	80.17	75.96	80.99	-	-	-	-
	IA-SSD [15]	-	-	-	79.57	-	-	-	-
	VIC-Net [33]	89.58	84.40	78.86	84.28	-	-	-	-
	HVPR [34]	-	-	-	-	91.14	82.05	79.49	84.23
	Associate-3Ddet [36]	-	-	-	-	89.29	79.17	77.76	82.07
Multi-stage	CT3D [18]	89.54	86.06	78.99	84.86	92.85	85.82	83.46	87.38
	Graph-Po [37]	-	-	-	-	93.27	86.50	83.87	87.88
	PV-RCNN [8]	89.35	83.69	78.70	83.91	-	-	-	-
	FV2P [40]	-	-	-	-	93.00	85.61	83.43	87.35
	BADet [41]	**90.06**	85.77	79.00	84.94	-	-	-	-
	VPGA [42]	-	-	-	-	92.95	85.31	82.64	86.97
	Part-A2 [49]	89.47	79.47	78.54	82.49	-	-	-	-
	Voxel R-CNN [7]	89.41	84.52	78.93	84.29	92.38	85.29	82.86	86.84
	PDV [9]	-	-	-	-	92.56	85.29	83.05	86.97
	Focals Conv [43]	89.52	84.93	79.18	84.54	-	-	-	-
	GraR-Vo [37]	-	-	-	-	93.33	86.12	83.29	87.58
	3D Cascade RCNN [44]	90.05	86.02	79.27	85.11	93.20	86.19	83.48	87.62
	CasA [45]	89.88	86.58	79.38	85.28	93.21	86.37	83.93	87.84
	GLENet [27]	89.93	86.46	79.19	85.19	93.51	86.10	83.60	87.74
	Ours	90.02	**86.92**	**79.54**	**85.49**	**93.54**	**86.57**	**84.21**	**88.11**

TABLE VI: 3D AP and mAP in percentage (%) of different methods for the cyclist detection over the KITTI validation set.

Method	3D AP recall 40			
	Easy	Mod.	Hard	mAP
CT3D [18]	91.99	71.60	67.34	76.98
Voxel R-CNN [7]	91.28	72.54	68.46	77.43
PV-RCNN [8]	88.88	71.95	66.78	75.87
PDV [9]	92.72	74.23	69.60	78.85
Ours	**92.77**	**75.25**	**69.95**	**79.32**

the dashed lines represent the others. A curve closer to the top-right corner signifies a detector with higher accuracy and recall rates, indicating superior performance. Among the compared detectors, our proposed method demonstrates the best results.

We also conduct experiments on the KITTI validation set. Since October 8, 2019, KITTI changed its calculation of AP from using 11 recall positions to 40 instead. For a fair comparison with previous methods, we adopt both the 11 and 40 recall positions on the validation set. As presented in Table V, the proposed method achieves the highest performances under almost all the metrics and yields 3D AP of 90.02%, 86.92%, and 79.54% at the easy, moderate, and hard difficulty levels with 11 recall positions, respectively, which achieves significant improvement compared with the other SOTA methods.

To further assess the effectiveness, we present the detection performance for the cyclist category on the KITTI validation set in Table VI. The accuracy is computed with 40 recall positions, providing a comprehensive evaluation of the method's performance. The experimental outcomes reveal that our method not only achieves competitive detection results but also excels across all the three difficulty levels. These results provide additional evidence of the strong generalization ability of the proposed method, highlighting its potential for application in a wide range of object detection tasks.

In addition to the above experiments, we also conduct an experiment to evaluate the inference speed of our proposed method. Since our proposed modules are both auxiliary training components that do not increase the computational complexity during the inference phase, our method has sufficient potential to exhibit high inference speed. We select three representative SOTA methods for comparison, including PDV (an advanced transformer-based detector), PV-RCNN (a classic point-voxel fusion method), and CasA (a method with high accuracy).

Table IV presents the inference speeds of different methods. Each model was evaluated with a single GTX3090 GPU, AMD EPYC 7543 32-Core Processor and the batch size was set as 1. The experimental results demonstrate that our proposed method achieves both high accuracy and high inference efficiency, making it more valuable for practical applications.

C. *Experiments on the NuScenes Dataset*

We conduct experiments on the NuScenes dataset to further validate the generalisability of the proposed method, which is a large-scale and diverse dataset for autonomous driving. Table VII reports the detection results on the NuScenes validation set compared with PointPillars, 3D-CVF, 3DSSD and SASA, which are advanced state-of-the-art (SOTA) methods with point clouds only as input. In this more challenging multi-category object detection task, our proposed method achieves

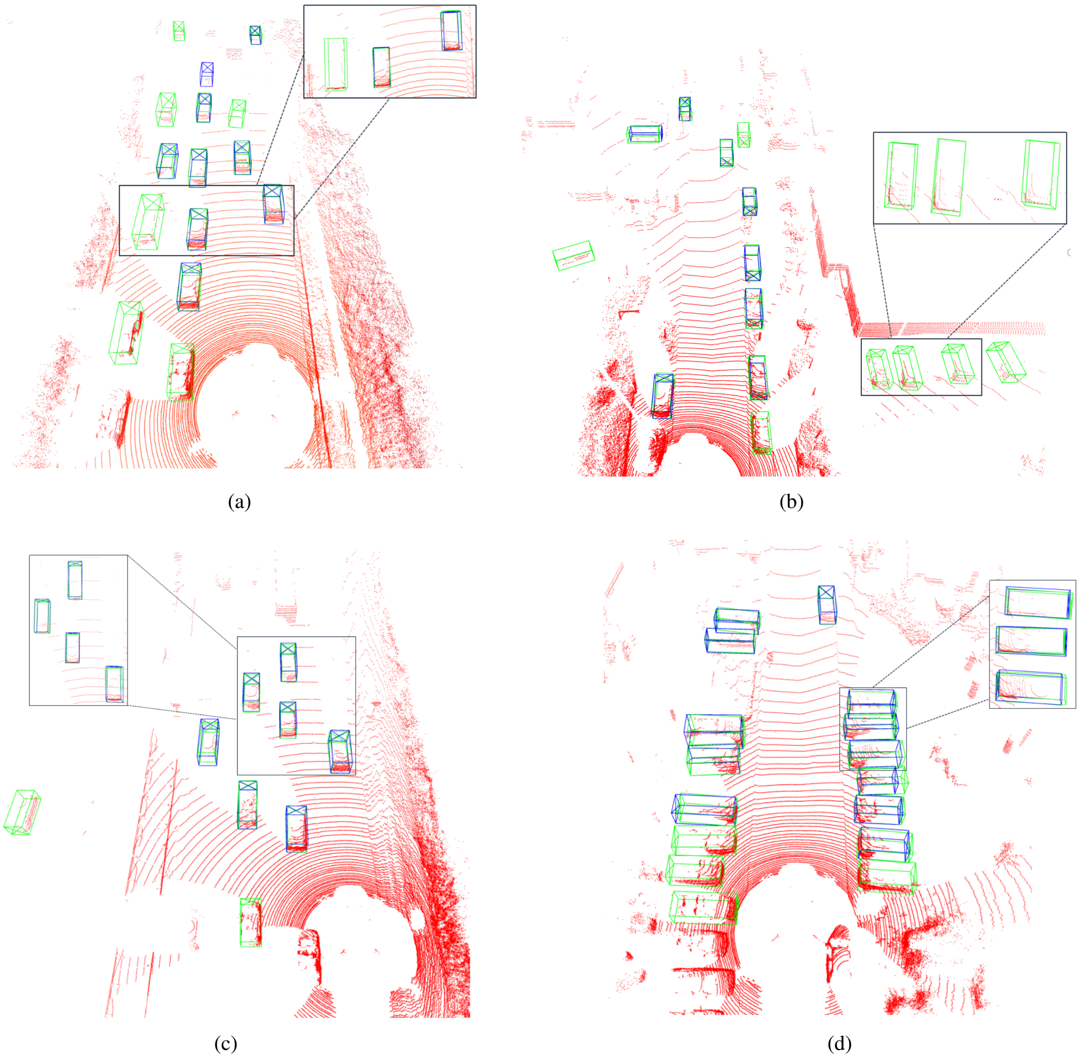

(a)
(b)
(c)
(d)

Fig. 12: Visualization of our proposed method in four different scenarios. The zoomed-in views in each subfigure highlight the accurate detection of objects in various scenarios, demonstrating the robustness and effectiveness of our proposed method.

superior detection performance in either NDS or mAP, which demonstrates that it is capable of handling various objects within large-scale scenarios.

Specifically, our model achieves the best results in NDS and mAP evaluation metrics at 65.2% and 55.3% without any ensembling or testing augmentation, respectively, which indicates the good performance of the model. However, the model did not perform optimally in the detection of the Car and Trailer categories. This may be related to the working principle of SoRL. The nuScenes dataset includes a wide variety of object categories, and some show small differences while the others have more significant disparities. The results shown in Table VII indicate that the model has a strong ability to distinguish between different distinct categories, but the ability to differentiate between similar categories is somewhat limited. This limitation may be tied to our SoRL, which uses a distributional form to represent the regression loss. The distribution can lead to minor dissimilarities being overlooked, to some extent thus limiting its ability to learn expressive features. In some instances, the model may struggle to identify subtle differences, which hinder its feature learning effectiveness.

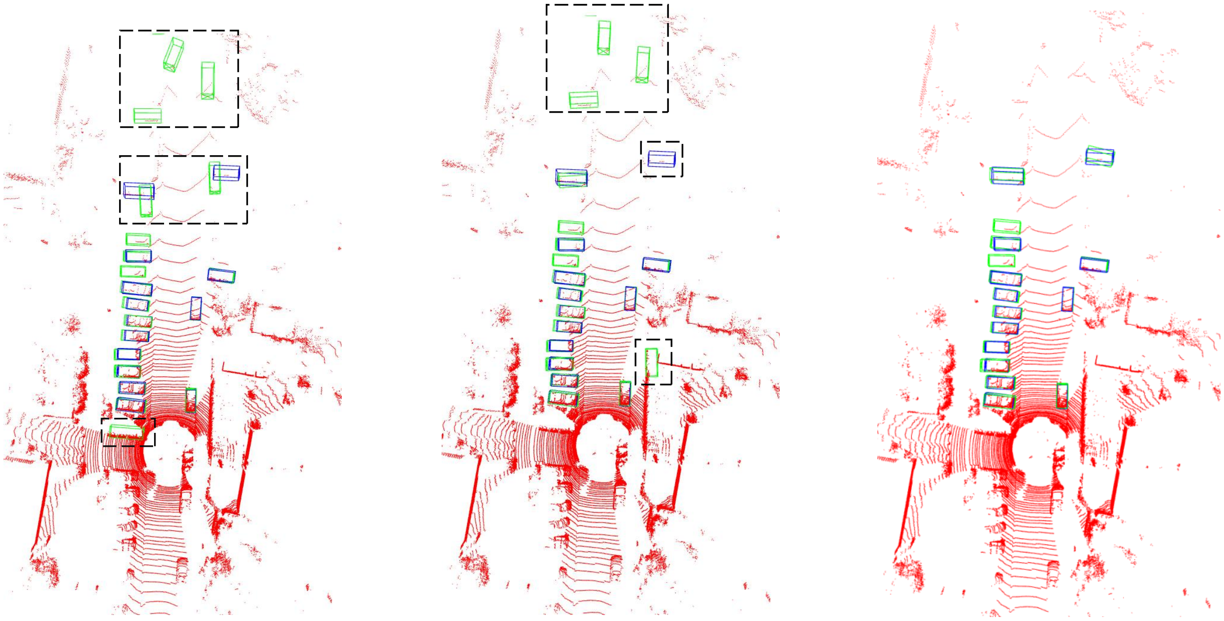
D. *Integration of SoRL into Existing Models*

To further demonstrate the effectiveness and versatility of the proposed SoRL method, we conduct additional experiments by incorporating SoRL into three widely adopted 3D object detection models: PV-RCNN, CT3D, and Voxel R-CNN. The training and evaluation processes are conducted for the car detection in the KITTI validation dataset, following the same experimental protocols and settings in the previous experiments. We compare the performance of the original models with the enhanced ones, which have incorporated the SoRL detection head.

Table I presents the results. It can be observed that the integration of SoRL consistently improves both the AP and mAP of all the three models. Specifically, the PV-RCNN+SoRL, CT3D+SoRL, and Voxel R-CNN+SoRL models exhibit an improvement of 0.55%, 0.14%, and 0.75% in mAP, respectively. These results indicate that our proposed SoRL method can be

TABLE VII: Comparison of performance with state-of-the-art methods on the NuScenes validation set. Evaluation metrics consist of NDS, mAP and AP in percentage (%) across 10 categories. Abbreviations: Pedestrian (Ped.), Traffic cone (T.C.), Construction vehicle (C.V.).

Method	NDS	mAP	Car	Truck	Bus	Trailer	C.V.	Ped.	Motor	Bicycle	T.C.	Barrier
PointPillars [5]	46.8	28.2	75.5	31.6	44.9	23.7	4.0	49.6	14.6	0.4	8.0	30.0
3D-CVF [51]	49.8	42.2	79.7	37.9	55.0	36.3	-	36.3	37.2	-	40.8	47.1
3DSSD [14]	56.4	42.6	**81.2**	47.2	61.4	30.5	12.6	70.2	36.0	8.6	31.1	47.9
SASA [52]	61.0	45.0	76.8	45.0	66.2	**36.5**	16.1	69.1	39.6	16.9	29.9	53.6
Ours	**65.2**	**55.3**	81.1	**53.1**	**68.0**	28.5	**18.8**	**82.9**	**58.1**	**42.2**	**62.6**	**58.4**



(a) Voxel R-CNN (b) CT3D (c) Ours

Fig. 13: Visualization of detection results for Voxel R-CNN, CT3D, and our proposed method in sparse point cloud scenarios.

effectively integrated into existing 3D object detection models, leading to notable performance gains.

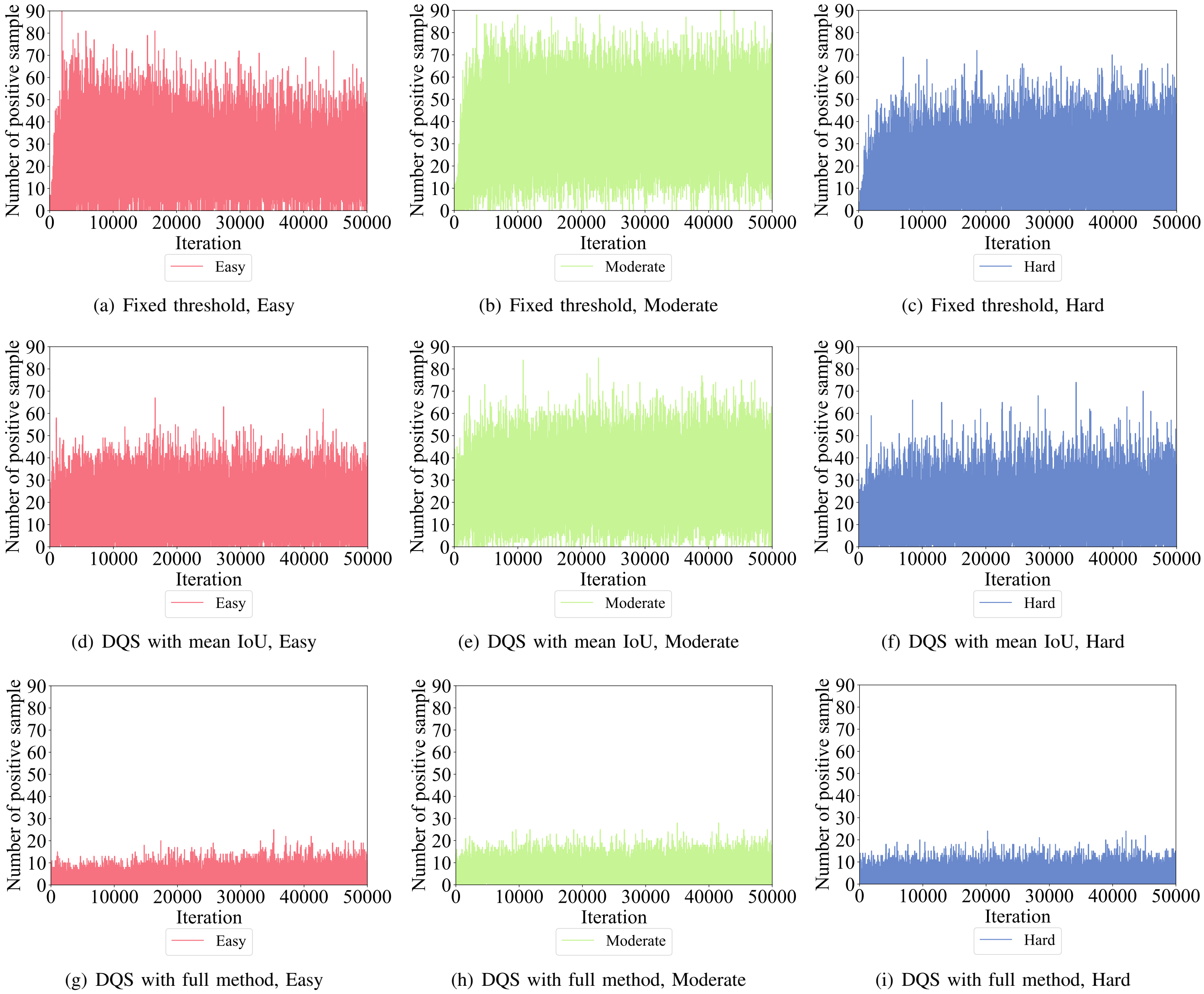
In conclusion, our experiments demonstrate that the proposed method is not only competitive as a standalone method but also capable of enhancing the performance of various existing 3D object detection models. The versatility of our method makes it a valuable addition to the field of 3D object detection, paving the way for its integration into more advanced models and applications.

E. Model Visualization Analysis

In this section, we present the visualization analysis of our model, which consists of two subsections. The first subsection compares the visualization results of our model under different scenarios. The second subsection focuses on its comparison with the state-of-the-art methods: Voxel R-CNN and CT3D.

1) Visualization Experiment: We conduct visualization experiments in which we showcase the performance of our algorithm in four different scenarios, as shown in Fig. 12. The point clouds are rendered in red. The blue boxes represent the ground truths, and the green boxes represent the predicted boxes. The zoomed-in views in each subfigure highlight the accurate detection of objects in various scenarios. In practical scenarios, there are significant differences in point cloud distribution, which makes the detector susceptible to missed and false detections. Meanwhile, it challenges the accuracy of object localization. As can be seen from the figure, the proposed method has a high detection accuracy for distant objects and the predicted boxes have a high degree of overlap with the ground truth label. Additionally, the model demonstrates strong generalization capabilities, which can even effectively detect unannotated objects in the scenarios.

2) Comparisons with other SOTA methods: We conduct visual comparison experiments of our model with Voxel R-CNN and CT3D to evaluate the performance of each method in challenging scenarios with sparse point clouds. As shown in Fig. 13, the blue box and the green box represent the ground truth and the predicted box, respectively. The dashed lines in the figure represent false detection cases. Both Voxel R-CNN and CT3D exhibit a certain degree of false detections when the point clouds are sparse, particularly at far distances, highlighted in the larger dashed line boxes, indicating lower robustness, while the Voxel R-CNN also inaccurately estimates the orientations of the objects and CT3D misses the detection of one object. In contrast, our proposed model can accurately recognize and detect objects even in sparse point cloud conditions at a distance.



(a) Fixed threshold, Easy

(b) Fixed threshold, Moderate

(c) Fixed threshold, Hard

(d) DQS with mean IoU, Easy

(e) DQS with mean IoU, Moderate

(f) DQS with mean IoU, Hard

(g) DQS with full method, Easy

(h) DQS with full method, Moderate

(i) DQS with full method, Hard

Fig. 14: Experimental results of the number of positive samples at different difficulty levels during training iterations with different sample selection strategies.

F. Ablation Experiments

In this section, we conduct three ablation experiments to investigate the contributions of SoRL and DQS to the overall performance of our model and how DQS would impact the sample selection process. We first present the ablation study for SoRL, followed by an analysis of the effect of DQS on the model's performance and sample selection in various scenarios.

TABLE VIII: The impact of different modules over the KITTI val set with AP in percentage (%) calculated over 40 recall positions for the car category.

SoRL	DQS	3D AP		
		Easy	Mod.	Hard
		92.54	85.32	81.72
✓		93.17	86.35	83.51
✓	✓	93.54	86.57	84.21

As presented in Table VIII, the SoRL significantly improves the detection performance compared to the baseline model, demonstrating its effectiveness in addressing the issue of label uncertainty in object detection. The DQS also contributes to the improved performance of the model, indicating its importance in considering the impact of uncertainty on sample selection. The combination of SoRL and DQS leads to the highest detection performance, highlighting the complementary nature of these two components and their joint contribution to the overall performance of our proposed method for object detection in point clouds.

We conduct another experiment to investigate the effectiveness of our DQS module. During the training phase, we recorded the number of positive samples generated at each iteration, categorized by easy, moderate, and hard difficulty levels. The experiment was conducted in three scenarios: using a fixed IoU threshold, our DQS with υ only, and DQS with $\upsilon + \phi$.

Fig. 14 illustrates the results of these experiments, where the red, green, and blue colors represent the results for the

TABLE IX: The average variance of positive samples at different difficulty levels with different sample selection strategies.

Fix IoU Threshold	DQS with mean IoU only	Full DQS
207.6	115.8	7.3

easy, moderate and hard difficulty levels, respectively. Fig. 14(a) ∼ Fig. 14(c) illustrate the results using a fixed threshold of 0.55 [7], [9], [45]. Fig. 14(d) ∼ Fig. 14(f) illustrate the results using DQS with the mean IoU only. Fig. 14(g) ∼ Fig. 14(i) illustrate the results using full DQS. Overall, using a fixed threshold, the model obtains the highest numbers of positive samples for all the three difficulty levels. However, the numbers of positive samples for easy and moderate difficulty are noticeably higher than that for the hard difficulty, with the moderate difficulty level having the most. This suggests that the model pays more attention to the objects of moderate difficulty. When DQS has been applied only with the mean IoU, the number of positive samples begins to decrease, and the quality of the positive samples starts to increase. When the full DQS has been utilized, the number of positive samples is significantly reduced, but as seen in Fig. 14(g) ∼ Fig. 14(i), the numbers of positive samples for the three difficulty levels are relatively stable and balanced. Considering the high accuracy of the model, it can be deduced that the model has learned more effective features from the selected higher-quality positive samples.

We also computed the average variance of positive sample numbers across different difficulty levels with different sample selection strategies, as shown in Table IX. The fixed threshold produces an imbalance in sample selection, while DQS, especially with the full method, ensures a more balanced and stable sample selection. This demonstrates that the model can effectively focus on various scenarios when calculating the regression loss, utilizing fewer high-quality positive samples to train the model, thereby achieving better results. It also indicates that traditional methods using a fixed threshold to divide proposals might introduce a larger number of low-quality positive samples, leading to instability in the model's feature learning and object detection.

V. Conclusion

This paper has proposed a high-performance 3D object detector based on raw point clouds, which improves model performance by taking label uncertainty into account. We developed two novel strategies to address the issue of label uncertainty. The Soft Regression Loss was used to account for uncertainty information in the calculation of regression loss, and the Discrete Quantization Sampling was used to account for the uncertainty information in the sample selection process. We have conducted a series of experiments to demonstrate the effectiveness of the proposed modules. Although our method makes progress in mitigating label uncertainty, there are still many potential research directions to explore in terms of measuring and utilizing uncertainty information. For example, uncertainty information can be used as a more effective metric than IoU, which can be realized through Bayesian neural networks or reinforcement learning methods. In addition, combining label uncertainty with prior knowledge is also a promising research direction. Research in these areas will lead to more effective models and algorithms in the future for object detection tasks.

References

[1] K. Ning, Y. Liu, Y. Su, et al. Point-Voxel and Bird-Eye-View representation aggregation network for single stage 3D object detection, in: IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 3, pp. 3223-3235.

[2] Z. Ouyang, X. Dong, J. Cui, et al. PV-EncoNet: Fast object detection based on colored point cloud, in: IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pp. 12439-12450.

[3] A. Geiger, et al. Are we ready for autonomous driving? the KITTI vision benchmark suite, in: *Proc. CVPR*, 2012, pp. 3354-3361.

[4] Y. Yan, Y. Mao, B. Li, SECOND: Sparsely embedded convolutional detection, in: Sensors, vol. 18, 2018, p. 3337.

[5] A. H. Lang, et al. PointPillars: Fast encoders for object detection from point clouds, in: *Proc. CVPR*, 2019, pp. 12697-12705.

[6] G. Wang, J. Wu, B. Tian, et al., CenterNet3D: An anchor free object detector for point cloud, in: IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pp. 12953-12965.

[7] J. Deng, et al. Voxel R-CNN: towards high performance voxel-based 3D object detection, in: *Proc. AAAI*, 2021, pp. 1201-1209.

[8] S. Shi, et al. PV-RCNN: point-Voxel feature set abstraction for 3D object detection, in: *Proc. CVPR*, 2020, pp. 10526-10535.

[9] J. S. K. Hu, T. Kuai, S. L. Waslander, Point Density-Aware Voxels for LiDAR 3D Object Detection, in: *Proc. CVPR*, 2022, pp. 8469-8478.

[10] D. Yu, S. Ji, A New Spatial-Oriented Object Detection Framework for Remote Sensing Images, in IEEE Transactions on Geoscience and Remote Sensing, 2022, vol. 60, pp. 1-16.

[11] C. R. Qi, H. Su, K. Mo, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation, in *Proc. CVPR*, 2017, pp. 652-660.

[12] C. R. Qi, L. Yi, H. Su, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, *Proc. NIPS*, 2017, 30.

[13] S. Shi, X. Wang, H. Li, PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud, in: *Proc. CVPR*, 2019, pp. 770-779.

[14] Z. Yang, Y. Sun, S. Liu, J. Jia, 3DSSD: Point-based 3D single stage object detector, in: *Proc. CVPR*, 2020, pp. 11037-11045.

[15] Y. Zhang, Q. Hu, G. Xu, Y. Ma, J. Wan, Y. Guo, Not All Points Are Equal: Learning Highly Efficient Point-Based Detectors for 3D LiDAR Point Clouds, in: *Proc. CVPR*, 2022, pp. 18953-18962.

[16] X. Pan, Z. Xia, S. Song, L. E. Li and G. Huang, 3d object detection with pointformer, in *Proc. CVPR*, 2021, pp. 7463-7472.

[17] A. Vaswani, et al. Attention is All you Need, in: *Proc. NIPS*, 2017, pp. 5998-6008.

[18] H. Shenga, et al. Improving 3D Object Detection with Channel-wise Transformer, in: *Proc. ICCV*, 2021, pp. 2723-2732.

[19] Y. Zhou, O. Tuzel, Voxelnet: End-to-end learning for point cloud based 3d object detection, in: *Proc. CVPR*, pp. 4490-4499, Jun. 2018.

[20] W. Zheng, et al. CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud, in: *Proc. AAAI*, 2021, pp. 3555-3562.

[21] H. Kuang, B. Wang, J. An, M. Zhang and Z. Zhang, Voxel-FPN: Multi-scale voxel feature aggregation for 3D object detection from LIDAR point clouds, Sensors, vol. 20, no. 3, p. 704, Jan. 2020.

[22] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang, Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. in: *Proc. NIPS*, 2020, pp. 21002-21012.

[23] He Y, Zhu C, Wang J, et al. Bounding box regression with uncertainty for accurate object detection, in *Proc. CVPR*, 2019: 2888-2897.

[24] C. Xu, J. Wang, W. Yang, H. Yu, L. Yu, G. Xia, Detecting tiny objects in aerial images: A normalized Wasserstein distance and a new benchmark, ISPRS Journal of Photogrammetry and Remote Sensing, V. 190, 2022, pp. 79-93.

[25] J. Choi, D. Chun, H. Kim, et al., Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving, in *Proc. ICCV*, 2019: pp. 502-511.

[26] X. Yang, J. Yan, Q. Ming, W. Wang, X. Zhang, Q. Tian, Rethinking rotated object detection with gaussian wasserstein distance loss, in *Proc. PMLR*, 2021, pp. 11830-11841.

[27] Zhang Y, Zhang Q, Zhu Z, et al. GLENet: Boosting 3D object detectors with generative label uncertainty estimation[J]. International Journal of Computer Vision, 2023, pp. 1-21.

[28] Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows, in: *Proc. ICCV*, 2021, pp. 10012-10022.

[29] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems, 2015, 28.

[30] C. He, et al. Structure Aware Single-Stage 3D Object Detection From Point Cloud, in: *Proc. CVPR*, 2020, pp. 11873-11882.

[31] W. Zheng, et al. SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud, in: *Proc. CVPR*, 2021, pp. 14494-14503.

[32] Q. Wang, J. Chen, J. Deng, X. Zhang, 3D-CenterNet: 3D object detection network for point clouds with center estimation priority, in: Pattern Recognition, vol. 115, 2021, p. 107884.

[33] T. Jiang, N. Song, H. Liu, R. Yin, Y. Gong, J. Yao, VIC-Net: Voxelization Information Compensation Network for Point Cloud 3D Object Detection, in: *Proc. ICRA*, 2021, pp. 13408-13414.

[34] J. Noh, et al. HVPR: Hybrid Voxel-Point Representation for Single-Stage 3D Object Detection, in: *Proc. CVPR*, 2021, pp. 14605-14614.

[35] D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, R. Yang, IoU Loss for 2D/3D Object Detection, in: *Proc. 3DV*, 2019, pp. 85-94.

[36] L. Du, et al. Associate-3Ddet: Perceptual-to-Conceptual Association for 3D Point Cloud Object Detection, in: *Proc. CVPR*, 2020, pp. 13329-13338.

[37] H. Yang, et al. Graph R-CNN: Towards Accurate 3D Object Detection with Semantic-Decorated Local Graph, in: *Proc. ECCV*, 2022, pp. 662-679.

[38] Y. Chen, et al. Fast Point R-CNN, in: *Proc. ICCV*, 2019, pp. 9775-9784.

[39] Z. Yang, Y. Sun, S. Liu, X. Shen, J. Jia, STD: Sparse-to-dense 3D object detector for point cloud, in: *Proc. CVPR*, 2019, pp. 1951-1960.

[40] J. Li, H. Dai, L. Shao, Y. Ding, From voxel to point: IoU-guided 3D object detection for point cloud with voxel-to-point decoder, in: *Proc. ACM*, 2021, pp. 4622-4631.

[41] R. Qian, X. Lai, X. Li, BADet: Boundary-Aware 3D Object Detection from Point Clouds, in: Pattern Recognition, vol. 125, 2022, p. 108524.

[42] G. Shi, K. Wang, R. Li, et al. Real-Time point cloud object detection via voxel-point geometry abstraction, in: IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2023.3259582.

[43] Y. Chen, et al. Focal Sparse Convolutional Networks for 3D Object Detection, in: *Proc. CVPR*, 2022, pp. 5428-5437.

[44] Q. Cai, et al. 3D Cascade RCNN: High Quality Object Detection in Point Clouds, in: IEEE TIP, vol. 31 , 2022, pp. 5706-5719.

[45] H. Wu, et al. CasA: A Cascade Attention Network for 3-D Object Detection From LiDAR Point Clouds, IEEE Trans. Geoscience and Remote Sensing, vol. 60, 2022, pp. 1-11.

[46] C. Zhou, Y. Zhang, J. Chen, et al. OcTr: Octree-Based Transformer for 3D Object Detection, in: *Proc. CVPR*, 2023, pp. 5166-5175.

[47] X. Qiming, C. Yidong, C. Guorong, et al. 3-D HANet: A flexible 3-D heatmap auxiliary network for object detection, in: IEEE Trans. GRS, vol. 61, no. 5701113, pp. 1-13.

[48] Z. Li, Y. Yao, Z. Quan, J. Xie, W. Yang, Spatial information enhancement network for 3D object detection from point cloud, in: Pattern Recognition, vol. 128, 2022, p. 108684.

[49] S. Shi, et al. From Points to Parts: 3D Object Detection From Point Cloud With Part-Aware and Part-Aggregation Network, in: IEEE Trans. PAMI, vol. 43, no. 8, 2020, pp. 2647-2664.

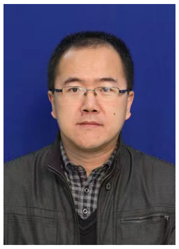
[50] H. Caesar, V. Bankiti, A. H. Lang, et al., nuscenes: A multimodal dataset for autonomous driving, in: *Proc. CVPR*, 2020, pp. 11618–11628.

[51] J. H. Yoo, Y. Kim, J. Kim, et al., 3D-CVF: Generating Joint Camera and LiDAR Features Using Cross-view Spatial Feature Fusion for 3D Object Detection, in *Proc. ECCV*, 2020, vol 12372.

[52] C. Chen, Z. Chen, J. Zhang, et al., SASA: Semantics-Augmented Set Abstraction for Point-Based 3D Object Detection, in *Proc. AAAI*, 2022, vol 36, no 1, pp. 221-229.

Yang Sun received his master degree from Hebei University, Baoding, China, in 2017.

He is currently a Ph.D. student at the North China Electric Power University, Baoding, China. His main research interests include machine learning and computer vision.



Bin Lu received his Ph.D. from Northwestern Polytechnical University, Xian, China, in 2003.

He is currently a Professor with the Engineering Research Center of Intelligent Computing for Complex Energy Systems, Ministry of Education, North China Electric Power University, Baoding, China. His main research interests include intelligence computation and computer vision, integrated energy systems, and big data analytics.

Yonghuai Liu obtained his first PhD degree in 1997 from Northwestern Polytechnical University, Xian, China, and second PhD degree in 2001 from The University of Hull, United Kingdom.

He is currently a professor and director of the Intelligent Visual Computing Research Centre at Edge Hill University since 2018. He is currently an area/associate editor or editorial board member for a number of international journals and conference proceedings. His primary research interests lie in 3D computer vision, image processing, pattern recognition, machine learning, artificial intelligence, and intelligent systems. He is a senior member of IEEE, a Fellow of the British Computer Society, and a Fellow of the Higher Education Academy of the United Kingdom.

Zhenyu Yang received his bachelor degree from North China Electric Power University, Baoding, China, in 2021.

He is currently a Master degree student at the North China Electric Power University, Baoding, China. His main research interests include machine learning and computer vision.

Ardhendu Behera received his PhD in Computer Science from the University of Fribourg and MEng in System Science and Automation from the Indian Institute Science Bangalore, India. He is a Professor of Computer Vision & AI, and co-director of the Intelligent Visual Computing Research Centre at Edge Hill University, UK. He has worked as a Research Fellow and Senior Research Fellow in the Computer Vision Group at the University of Leeds. He is a Fellow of HEA and a member of IEEE, BMVA, AVA, BCS, affiliated member of IAPR and ECAI. His main interests include computer vision, deep learning, human-robot social interaction, activity analysis, and recognition.

Ran Song received the BEng degree in telecommunications engineering from Shandong University, Jinan, China, in 2005, and the Ph.D. degree in computer vision from the University of York, UK, in 2009.

He is currently a professor with the School of Control Science and Engineering, Shandong University. His research interests include 3D shape analysis and 3D visual perception.

Hejin Yuan received his Ph.D. from Northwestern Polytechnical University, Xian, China, in 2009.

He is currently an Associate Professor with the Engineering Research Center of Intelligent Computing for Complex Energy Systems, Ministry of Education, North China Electric Power University, Baoding, China. His main research interests include pattern recognition and computer vision.

Haiyan Jiang received her Ph.D. degree in Agricultural Information from Nanjing Agricultural University, Nanjing, China, in 2007.

She is currently a professor at the National Engineering Research Center for Information Technology in Agriculture and the Key Laboratory of Crop System Analysis and Decision Making, Ministry of Agriculture, Nanjing Agricultural University, Nanjing, China. Her research interests include computer vision, the measurement of phenotypes in animals and plants, and the analysis of big data and intelligent computing in crop systems.