Object-Level Contrast Learning for 3D Sparse Object Detection in Ocean Scene

Yuheng He, Wenbin Yang, Xiangfeng Luo*, Liyan Ma, Shaorong Xie
College of Computer Engineering and Science,Shanghai University
Shanghai, China
rianh@shu.edu.cn, youngwb@shu.edu.cn, *luoxf@shu.edu.cn, liyanma@shu.edu.cn srxie@shu.edu.cn

***Abstract*—LiDAR-based 3D object detection provides the necessary high-precision environmental sensing information for the safe navigation of smart ships. However, relying on viewpoint projections, voxelized point clouds, or using inefficient point sampling methods, current LiDAR 3D object detection methods treat all objects uniformly and quantitatively while ignoring the specificity of sparse objects in the scene, which leaves less useful information about sparse objects. In this paper, we propose an end-to-end two-stage architecture, Object-Level Contrast Learning 3D Object Detection network (OCL), for better construction of sparse object features and improving the ability of model to detect sparse objects. In the first stage, the Contrast Learning based Sparse Object Feature Enhancement training strategy is proposed to decrease the feature discrepancy between sparse and regular objects in object-level. In the second stage, we use the Point-level Feature Multiple Aggregation Strategy to aggregate finer point-level features of sparse objects. Extensive experiments show that OCL achieves excellent performance on both Ship dataset and KITTI dataset. Furthermore, our work proposes a promising new idea for applying contrast learning to 3D object detection.**

I. Introduction

Accurate 3D object detection is critical to ensure safe navigation of smart ships, since the precise localization information it provides directly affects the effectiveness of downstream tasks. LiDAR, as a high-precision detection instrument, offers more precise position information for objects than conventional images, and the LiDAR-based object detection keeps a promising research area with significant potential.

However, there is a high variability in the sparsity of LiDAR data due to the distance, the factors in LiDAR itself and the environment, as shown in Figure 1. Objects closer to the sensor are denser, while more distant objects are sparser. Moreover, high-powered LiDAR used for ocean navigation may also overheat, resulting in sparse objects. As a consequence, objects with more points and distinct contours are more easily detected, while those with fewer points and weaker contours are more challenging to detect.The topic of how to efficiently identify sparse objects remains a challenging research subject.

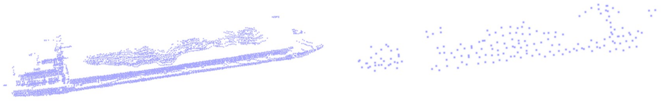

Fig. 1. Dense Ship Object vs. Sparse Ship Object.

The methods for 3D object detection have been extensively studied from various perspectives, including 2D projection-based methods [1], [2], 3D voxel-based methods [3]–[6], and point-based methods [7]–[10]. These methods quantifies all objects uniformly and ignores the specificity of sparse objects in the scene, which makes sparse objects have even less useful information. In an effort to generate finer feature representations of objects, some researchers obtains voxel-level features and then aggregates more precise point-level features [11]–[15]. This kind of framework has been very popular in recent studies, showing strong performance across a variety of scenarios, and presenting opportunities for further advancements, particularly in the detection of sparse objects.

In this paper, We propose Object-Level Contrast Learning 3D Object Detection network (OCL), an end-to-end two-stage detector that addresses the unique feature specificity of sparse objects in LiDAR data. OCL consists of two crucial modules: Contrast Learning based Sparse Object Feature Enhancement(CLFE) and Point-level Feature Multiple Aggregation(PFMA). To achieve better voxel-level feature representation of sparse objects, we propose CLFE in the first stage, which decreases the feature discrepancy between sparse and regular objects by contrast learning. The feature of sparse objects keeps more abstract than regular objects. Therefore, in the second stage, we propose a finner feature aggregation method PFMA to better perform point-level feature aggregation on sparse objects.

The effectiveness of our proposed theories has been demonstrated through sufficient experimentation. Notably, the OCL has proven to be highly effective for detecting sparse objects in both the Ship dataset and the KITTI dataset, surpassing the performance of existing methods. There is a 2.04% and 1.77% improvement in the mAP metric on the Ship dataset and KITTI dataset, respectively, compared to the current methods. Furthermore, our work proposes a promising new idea for applying contrast learning to 3D object detection. We summarize the following contributions of our method:

- By using contrast learning, we propose a sparse object feature enhancement strategy to decrease the voxel-level feature discrepancy between sparse and regular objects.
- Building upon prior research, we propose a multiple aggregation strategy for finner point-level feature aggre-

* is the corresponding author.

DOI reference number: 10.18293/SEKE2023-137

gation of sparse object.
- Our proposed method shows excellent performance on both Ship dataset and KITTI dataset compared to the current methods.

II. Related work

3D Object Detection. Voxel-based methods [3]–[6] represent point cloud quantitatively with voxels for rapid analysis of disordered point clouds using 3D Convolution. Point-based methods [7]–[10] usually sample key points first to reduce computation, and then use a symmetric function to extract point-level features. Point-voxel-based methods [11]–[15] use both voxel-level and point-level features of the point cloud to combine their advantages for better detection. SA-SSD [11] proposes an auxiliary network, which interpolates point-level features to intermediate voxel layers, to auxiliary supervise the training of backbone network. PV-RCNN [12] and PV-RCNN++ [13] use a two-stage structure to get more precise bounding box. In the first stage, getting the rough estimated ROI region through the voxel-level backbone network and RPN. In the second stage, aggregating the point-level feature for precise bounding box generation. PDV [15] uses the correlation between density and distance in point cloud to aggregates more information in the second stage. Point-Voxel-based methods are a popular research trend and have demonstrated strong performance across various scenes, particularly in larger-scale scenes. However, these methods still have great potential for detecting sparse objects.

Contrastive learning. Contrast learning [16] is a popular methodology that has been used in various fields recently [17]–[21]. The basic concept of contrast learning is to create a metric space where similar pairs of samples are pushed closer together and dissimilar pairs are pushed further apart. The most prevalent approach for implementing contrastive learning is to apply some global transformations to the image and then use a siamese network [20], [21]. The siamese network can be trained to learn differences and similarities between pairs of inputs. This apporach has proven to be simple and effective for downstream tasks such as classification and regression that rely on global features. However, object-level features are more important for 3D object detection task than global features of scene. How to effectively use contrast learning method in 3D object detection task is still an area that needs to be explored.

III. Method

A. Network Architecture

Our network designed as a two-stage detector, as shown in Figure 2. In the first stage, we utilize the backbone network, refer to Second [4], with both 3D sparse convolution layers and 2D convolution layers to generate ROI regions from the point cloud. The Contrast Learning based Sparse Object Feature Enhancement module is also build on the feature extraction part, which will be detailed in section III-B. In the second stage, we use the Point-level Feature Multiple Aggregation (PFMA) module to create more precise local features for sparse objects, which will be detailed in section III-C. Besides, we also fuse BEV-level ROI features to add global features for objects. Finally, the parameters associated with the bounding box of object will be optimized by classification and regression.

B. Contrast Learning based Sparse Object Feature Enhancement

There is a significant difference in the distribution of points between sparse and regular objects, which results in a large semantic gap in their feature representation on the feature map and limits the detection performance of sparse objects. To address this issue, we propose the CLFE module, which is specifically designed in object-level to decrease the feature discrepancy between sparse and regular objects. This module is only activated during training and does not add any time cost burden to inference. Here we call the points in ground-truth(GT) bounding box as info-points, and the features used to judge the class of objects as intrinsic-feature. We perform object intrinsic-feature invariant transformations on the info-points of all objects in the input point cloud, such as local down-sampling, center rotation and scaling, to simulate the points distribution of sparse objects in the scene. The generated point cloud scene is input to the backbone network together with the raw point cloud scene as a pair. Due to the characteristics of the organization of the voxel space, we can derive the position of each object on the feature map from the step information in the convolution process. As mentioned above, the corresponding position $\mathcal{P}$ and size $\mathcal{S}$ of each GT on the feature map can be formulated as:

$$\mathcal{P}_i = (GT_i^x/stride_x, GT_i^y/stride_y) \tag{1}$$

$$\mathcal{S}_i = (GT_i^l/stride_x, GT_i^w/stride_y) \tag{2}$$

where $GT_i^x, GT_i^y, GT_i^l, GT_i^w$ denote the coordinates along X-axis, Y-axis, and length, width of the ith GT, respectively; $stride_x, stride_y$ denote the spatial scaling of the current feature map relative to the original point cloud space.

For computational convenience, we take the smallest square region containing the ith object on the feature map as the contrast loss denoted as $\Gamma_i \in R^{max(\mathcal{S}_i)\times max(\mathcal{S}_i)\times C}$. The ith object feature region in the original scene and the transformed scene are denoted as Γ_i^O and Γ_i^T. Conventional contrast methods, which directly calculating the feature gap between Γ_i^O and Γ_i^T , may cause the network to fall into extreme local optimization, resulting in the network not learning any useful information. Inspired by SimSiam [21], We apply a linear projection ς to a portion of the feature pair to enable the training of the network to proceed as we expect, and the contrast loss $L_{contrast}$ can be formulated as:

$$L_{contrast} = \mathcal{C}(\Gamma^O, \varsigma(\Gamma^T)) \tag{3}$$

where $\mathcal{C}$ denotes the calculation method of feature distance.

The feature region needs to be transformed into feature vectors before it is used for contrast loss, and we propose several feature transformation methods: direct contrast mode, average value mode, maximum activation mode, and mean-maximum activation mode, which we will discuss specifically in the

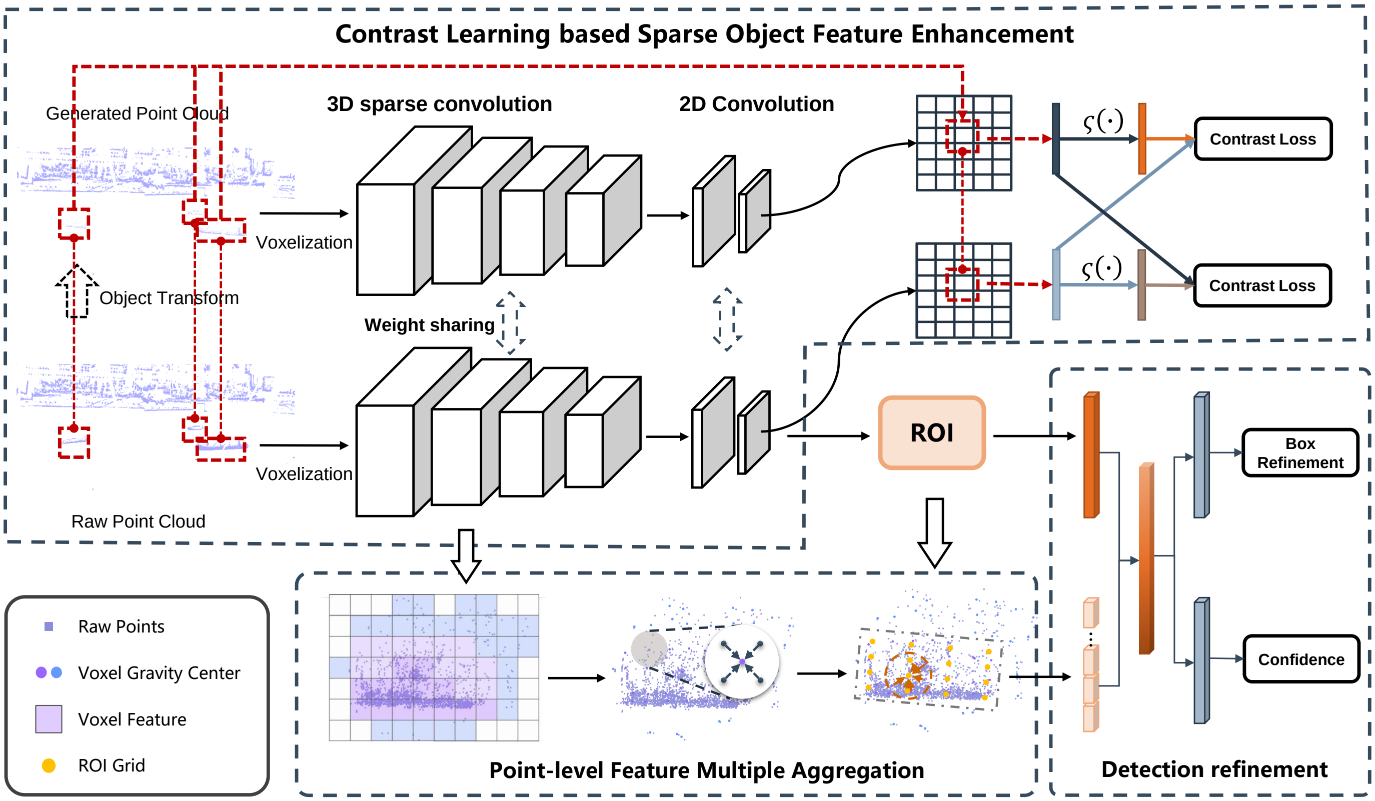

Fig. 2. The Architecture of Object-Level Contrast Learning 3D Object Detection Network

ablation experiments. The total loss of this part $L_{auxiliary}$ can be formulated as:

$$L_{auxiliary} = \frac{\sum_i^{N^{GT}}(W^O \times \mathcal{D}_i^O + W^T \times \mathcal{D}_i^T)}{N^{GT}} \tag{4}$$

$$\mathcal{D}_i^O = 1 - \mathcal{C}(\Upsilon(\Gamma_i^T), \Upsilon(\varsigma(\Gamma_i^O))) \tag{5}$$

$$\mathcal{D}_i^T = 1 - \mathcal{C}(\Upsilon(\Gamma_i^O), \Upsilon(\varsigma(\Gamma_i^T))) \tag{6}$$

where W^O, W^T is the predefined loss weight corresponding to original and transformed objects, $\mathcal{D}^O, \mathcal{D}^T$ is the calculated feature distance of original and transformed objects, Υ is the feature transformation method, and N^{GT} denotes the number of GT in the current scene. We choose cosine similarity function as the distance calculation method $\mathcal{C}$,

C. Point-level Feature Multiple Aggregation

Numerous studies have investigated ROI feature aggregation method. PV-RCNN [12] uses the Farthest Point Sampling (FPS) [22] to globally sample key points for feature aggregation. PDV [15] directly attaches voxel features to the gravity centor of voxels (GCV) and uses them as key points for subsequent feature aggregation to reduce the complexity of FPS sampling and original point feature extraction. However, these methods reduce the effective information of sparse objects in the process of sampling or quantizing. Based on the existing works, we propose some finner ROI feature aggregation strategies aimed at improving the point-level feature representation of sparse objects.

Specifically, we propagate the voxel-level features obtained in the previous stage to the raw points within each voxel. The point-level features are then aggregated at a fine-grained level for the original points to form the local features of the object using the following strategy:

Vanilla Aggregation. For raw points that have been assigned voxel-level features, we operate on them directly through grid-based feature aggregation used in PDV [15].

Voxel Gravity Points Aggregation. First, The raw points with voxel-level features attached are interpolated to the GCV using a method similar to that used in PV-RCNN [12]. Then, the features on the GCV are aggregated to the ROI by Vanilla Aggregation. In this way, the feature offset problem caused by attaching voxel features directly to the GCV can be well corrected, and the features on GCV will contain more detailed information.

Voxel Gravity Grid Points Aggregation. First, the raw points with voxel-level features attached are aggregated to the GCV by Vanilla Aggregation. Then, the features of the GCV are aggregated to the ROI by Vanilla Aggregation again. The ROI features obtained by this two-step aggregation strategy will contain more detailed and deeper information about the semantic features of the objects.

The effectiveness of the above strategies will be verified in detail in the ablation study. Meanwhile, we impose a simple graph convolution module for enhancing the feature correlation between GCV to mine more sparse object features.

D. Training Losses

We use an end-to-end training strategy. The total loss L_{total} consists of three parts, which are the contrast learning auxiliary loss $L_{auxiliary}$ for the feature extraction part, the region proposed loss L_{RPN} for the first stage, and the suggested optimization loss L_{refine} for the second stage:

$$L_{total} = L_{auxiliary} + L_{RPN} + L_{refine} \tag{7}$$

Where $L_{auxiliary}$ has been explained in detail in section III-B, the region proposal loss L_{RPN} is composed of classification loss and box regression loss. The classification loss here adopts focal loss. The box regression loss adopts smooth-L1 loss. Therefore, the region proposal loss can be formulated as:

$$L_{RPN} = L_{focal} + L_{smooth-L1} \tag{8}$$

The proposal refinement loss L_{refine} is also composed of classification loss and residual box regression loss. The classification loss here adopts IoU loss as same as PV-RCNN [12]. The residual box regression loss also adopts smooth-L1 loss. Thus, the proposal refinement loss can be formulated as:

$$L_{refine} = L_{IoU} + L_{smooth-L1} \tag{9}$$

IV. Experiments

A. Datasets and Implementation Details

Datasets. We evaluate our model on both Ship dataset and KITTI [23] dataset. The Ship dataset is composed of actual ship data obtained from various locations such as ports and shoreside, utilizing 32-line and 128-line LiDAR. The LiDAR data is filtered, labeled, and divided into a training set (3427 samples) and a test set (856 samples) with a ratio of 8:2. The training set is used to train the detection model, while the test set is used to verify the effectiveness of our model. The 3D autonomous driving dataset KITTI also be divided into a training set (3712 samples) and a validation set (3769 samples) to further verify our theory. For the Ship dataset, the detection range is [-400m, 400m] for the X axis, [-50m, 450m] for the Y axis, and [-10m, 20m] for the Z axis. We divide the raw point cloud into voxels of size (0.5 m, 0.36m, 0.75m). For the KITTI dataset, the detection range is set to be [0, 70.4m] for the X axis, [-40m, 40m] for the Y axis, and [-3m, 1m] for the Z axis. We set the voxel size to be (0.05m, 0.05m, 0.1m).

Training and Inference Details. For the second stage point-level feature aggregation, we use the last two intermediate voxel-level features, and the spherical query radius is set to [3,6] in Ship dataset, [0.4,0.6] in KITTI dataset. For generating transformed point cloud data, we set the object scaling to [0.95,1.05], the object points down-sampling range to [0.1,0.4], and the minimum number of the object points threshold to 10. Our model is trained using Adam [24] optimizer with initial learning rate set to 0.01 and one-cycle strategy [25] for learning rate update. The training environment is RTX 3090 for a total of 80 epochs. For the preprocessing of the training data, we apply some 3D object detection data augmentation strategies, including global rotation, random flip, global scaling, and ground truth data augmentation [4].

B. 3D Detection on the Ship Dataset

Table I presents the performance of our proposed method on the Ship test set, where our method achieves optimal performance on CargoShip, EngineeringShip, and all categories. Specifically, the results on $AP|_{R40}$ have shown a significant improvement of 2.92%, 2.41%, and 2.04% for CargoShip, EngineeringShip, and all categories, respectively, when compared to the current optimal results.In comparison to our benchmark method PDV, our proposed method outperforms PDV by 2.92%, 2.82%, 2.41%, and 2.04% on CargoShip, TourBoat, EngineeringShip, and mAP, respectively. Figure 3 intuitively shows the detection results of our proposed method and the benchmark model PDV on the Ship test set. The lidar in the scene is located in the lower center of the image. Our method accurately detects sparse objects, which effectively demonstrates the effectiveness of our method.

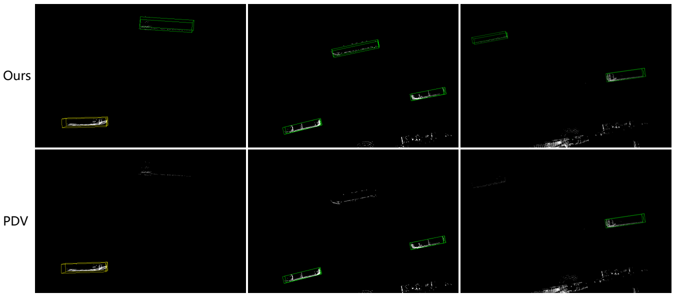

Fig. 3. Snapshots of our 3D detection results on the Ship test set. Green boxes for CargoShip and yellow for ContainerShip.

C. 3D Detection on the KITTI Dataset

We also evaluated the effectiveness of our proposed method on the KITTI validation set, and the results are shown in Table II. Our method achieves state-of-the-art multiclass results, with 3D $AP|_{R40}$ improving 0.07%, 1.98%, and 2.47% on the moderately difficult car, pedestrian, cyclist categories, respectively, and a 1.77% improvement on the average multiclass accuracy mAP. Figure 4 visualizes the detection results of our proposed method and the benchmark model PDV on the KITTI validation set. Compared with the benchmark model, our method is also able to accurately detect some sparse objects at longer distances in the scene, which have fewer points than the sparse ship objects. This result further demonstrate the effectiveness of our method for detecting sparse objects.

D. Ablation Studies

We performed ablation experiments on the Ship dataset for each of our modules and validated the effectiveness of our modules using mAP.

TABLE I

THE SHIP TEST SET FOR MULTI-CLASS DETECTION, WITH 3D AVERAGE PRECISION OF 40 SAMPLING RECALL POINTS AND 0.7 INTERSECTION OVER UNION.

Method	CargoShip	ContainerShip	TourBoat	EngineeringShip	mAP
Second(2019)	51.52	98.89	89.47	55.13	73.75
PV-RCNN(2020)	82.88	100.00	96.71	84.18	90.94
IA-SSD(2022)	64.67	100.00	**96.78**	81.96	85.85
PDV(2022)	83.18	100.00	91.50	90.00	91.17
Ours	**86.10**	100.00	94.32	**92.41**	**93.21**
Improvement	2.92			2.41	2.04

TABLE II

THE KITTI VAL SET FOR MULTI-CLASS DETECTION, WITH 3D AVERAGE PRECISION OF 40 SAMPLING RECALL POINTS.

Method	AP3D@Car-R40 (IoU=0.7)			AP3D@Pedestrian-R40 (IoU = 0.5)			AP3D@Cyclist-R40 (IoU = 0.5)			mAP
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	
VoxelNet(2018)	81.97	65.46	62.85	57.86	53.42	48.87	67.17	47.65	45.11	58.93
Second(2019)	90.71	81.73	78.79	57.46	52.86	47.95	81.29	65.80	62.94	68.84
PV-RCNN(2020)	92.04	84.06	81.99	65.39	58.06	53.31	90.47	72.42	68.26	74.00
IA-SSD(2022)	90.94	83.00	80.04	56.98	52.50	47.74	92.80	71.81	67.47	71.48
PDV(2022)	92.39	85.11	82.78	64.77	58.10	53.41	89.94	72.03	67.78	74.03
Ours	**92.76**	**85.18**	**83.00**	**66.70**	**60.08**	**55.34**	**93.91**	**74.89**	**70.36**	**75.80**
Improvement	0.37	0.07	0.22	1.31	1.98	1.93	1.11	2.47	2.10	1.77

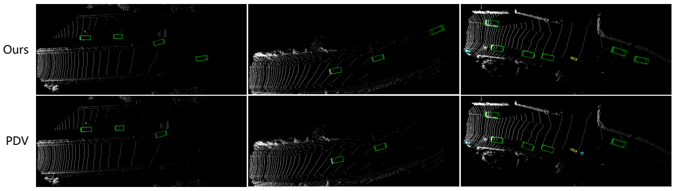

Fig. 4. Snapshots of our 3D detection results on the KITTI val set. Green boxes for car, cyan for pedestrian and yellow for cyclist.

Components Ablation. We first conducted ablation experiments for each module in our network. As show in Table III, CLFE represent the Contrast Learning based Sparse Object Feature Enhancement strategy, PFMA represent the Point-level Feature Multiple Aggregation strategy. Exp.1 is the benchmark model PDV, while Exp.2 adds the CLFE strategy, resulting in a 1.88% improvement on mAP compared to the benchmark model, which validates the effectiveness of the CLFE training strategy. Exp.3 uses the PFMA as the ROI feature aggregation strategy, resulting in a 0.64% improvement on mAP compared to the benchmark model, which validates the effectiveness of the PFMA strategy. Experiment 4 is our proposed method, which improves mAP by 2.04% compared to the benchmark model.

TABLE III

ABLATION EXPERIMENTS OF TWO STRATEGIES ON NETWORK PERFORMANCE.

Exp.	CLFE	PFMA	mAP
1			91.17
2	√		93.05
3		√	91.81
4	√	√	93.21

Methods For Measuring Feature Discrepancy Between Objects. In this part, we conducted ablation experiments on the methods that can be used to measure feature discrepancy between objects. In Table IV, Flatten indicates that the values of the two feature regions are flattened directly, and the flattened vector is used to calculate the contrast loss; Avg and Max indicates that we use an extra average-pooling layer and max-pooling layer, respectively, to further abstract the object features and compress them into a vector, which is then used to calculate the contrast loss; Avg-Max indicates that the vector used to calculate the contrast loss is obtained by concatenating the abstract object vector used in the Avg and Max strategies. use the concatenate feature of avg and max to calculate the contrast loss. As shown in Table IV, Exp.2 and Exp.3 are improved by 1.31% and 1.83%, respectively, compared with Exp.1, which indicates that the object feature vector after further abstraction by the pooling layer can better measure the feature discrepancy between sparse and regular objects, among which the max-pooling can relatively better express the object features. On the contrary, the performance of Exp.4 on mAP are decreased by 0.13% compared with Exp.3, which indicates that the object feature representations obtained from the two pooling layers mentioned above have some conflicts that cannot be used directly.

TABLE IV

ABLATION ANALYSIS OF DIFFERENT METHODS FOR MEASURING FEATURE DISCREPANCY BETWEEN OBJECTS.

Exp.	Flatten	Avg	Max	Avg-Max	mAP
1	√				91.22
2		√			92.53
3			√		93.05
4				√	92.92

Point-level Feature Multiple Aggregation Strategy. In this part, we validate the effectiveness of three different ROI feature aggregation strategy, which have detailed in section III-C. As shown in Table V, VA represent the Vanilla Aggregation Strategy, VGPA represent the Voxel Gravity Points Aggregation Strategy, VGGPA represent the Voxel Gravity Grid Points Aggregation Strategy. The experimental results have 0.69% and 0.58% improvement on mAP for VGPA and VGGPA respectively, which well validate our previously proposed theory.

TABLE V
ABLATION ANALYSIS OF DIFFERENT STRATEGIES FOR ROI FEATURE AGGREGATION.

Exp.	VA	VGPA	VGGPA	mAP
1	√			90.54
2		√		91.23
3			√	91.81

V. Conclusion

Current methods treat all objects uniformly, ignoring the specificity of sparse objects in the scene, which leaves less useful information about sparse objects and is not conducive to the detection of 3D sparse objects in ocean scene. To address this limitation, Object-Level Contrast Learning 3D Object Detection network (OCL) is proposed, which is an end-to-end two-stage architecture that takes into account the feature specificity of sparse objects. In the first stage, the Contrast Learning based Sparse Object Feature Enhancement training strategy is designed in object-level to decrease the feature discrepancy between sparse and regular objects. In this procedure, The voxel-level features of sparse object can be enhanced. In the second stage, the Point-level Feature Multiple Aggregation strategy is utilized to better aggregate the point-level features of sparse object. The effectiveness of our proposed strategies are verified in extensive experiments on Ship dataset and KITTI dataset. Our work uses the characteristics of LiDAR data to generate sample pairs in a relatively simple way for contrast learning. However, the relationships between more objects in the scene have not been fully explored. Furthermore, we believe that there keeps a great deal of opportunity to explore the application of contrast learning in 3D object detection.

Acknowledgement

This work is supported by the Development Proiect of Ship Situational Intelligent Awareness System under the Grant MC-201920-X01, the National Natural Science Foundation of China under the Grant No.61991415.

References

[1] M. Simony, S. Milzy, K. Amendey, and H.-M. Gross, "Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[2] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660.

[3] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[4] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[5] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel r-cnn: Towards high performance voxel-based 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1201–1209.

[6] J. Mao, Y. Xue, M. Niu, H. Bai, J. Feng, X. Liang, H. Xu, and C. Xu, "Voxel transformer for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3164–3173.

[7] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[8] Y. Zhang, Q. Hu, G. Xu, Y. Ma, J. Wan, and Y. Guo, "Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 953–18 962.

[9] W. Yang, S. Sheng, X. Luo, and S. Xie, "Geometric relation based point clouds classification and segmentation," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 11, p. e6845, 2022.

[10] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 040–11 048.

[11] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3d object detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 873–11 882.

[12] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 529–10 538.

[13] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li, "Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection," *International Journal of Computer Vision*, pp. 1–21, 2022.

[14] J. Mao, M. Niu, H. Bai, X. Liang, H. Xu, and C. Xu, "Pyramid r-cnn: Towards better performance and adaptability for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2723–2732.

[15] J. S. Hu, T. Kuai, and S. L. Waslander, "Point density-aware voxels for lidar 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8469–8478.

[16] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[17] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *International conference on machine learning*. PMLR, 2020, pp. 4182–4192.

[18] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *Advances in neural information processing systems*, vol. 32, 2019.

[19] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *European conference on computer vision*. Springer, 2020, pp. 776–794.

[20] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6707–6717.

[21] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 750–15 758.

[22] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1305–1315, 1997.

[23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay," *arXiv preprint arXiv:1803.09820*, 2018.