2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

VoxelContrast: Voxel Contrast-Based Unsupervised Learning for 3D Point Clouds

Yuxiang Qin[1] and Hao Sun[2]

Abstract— **The annotation process for 3D point cloud data is more complex than for image data, and training with a small amount of annotated data significantly reduces the performance of deep learning models. Unsupervised learning can better utilize large amounts of unlabeled point cloud data for model pretraining, thereby achieving excellent performance on small-scale datasets. However, many existing 3D point cloud unsupervised learning methods are primarily focused on single-object CAD point clouds and may not be suitable for larger-scale autonomous driving LiDAR point clouds. To address this challenging problem, we propose a voxel contrast-based unsupervised learning method (VoxelContrast), which adapts well to different types of point cloud data through voxelization and can be seamlessly integrated with existing model frameworks. Specifically, we utilize voxelization methods to preprocess point cloud data. Then, we incorporate voxel information into contrastive learning, facilitating the creation of more meaningful positive and negative sample pairs. Finally, we conduct unsupervised training of the model using instance discrimination as the proxy task. Our method was validated in two downstream tasks: point cloud shape classification and 3D object detection. Experimental results demonstrated that models pretrained using a substantial amount of unlabeled data can further enhance the effectiveness of existing supervised learning methods.**

I. INTRODUCTION

Deep learning models have achieved remarkable advancements in various domains, including computer vision and natural language processing. Regarding model training, it is common practice to utilize labeled data directly. However, due to the escalating costs of annotation, obtaining large-scale precisely annotated data often becomes a challenging endeavor. Unsupervised learning is a current research hotspot. In scenarios with large-scale unlabeled data, unsupervised pretraining enables models to learn meaningful feature representations, thereby improving the performance of downstream tasks. Currently, unsupervised learning finds wide applications in various domains, such as GPT [1] and BERT [2] in natural language processing, and MOCO [3] and SimCLR [4] in computer vision.

In recent years, with the rapid development of autonomous driving and robotics technology, 3D point clouds have been widely applied [5]–[7]. Unlike image data, point cloud data provides better representation of objects in terms of their three-dimensional coordinates, shapes, contours, and depths. However, point clouds are unstructured data, often sparse,

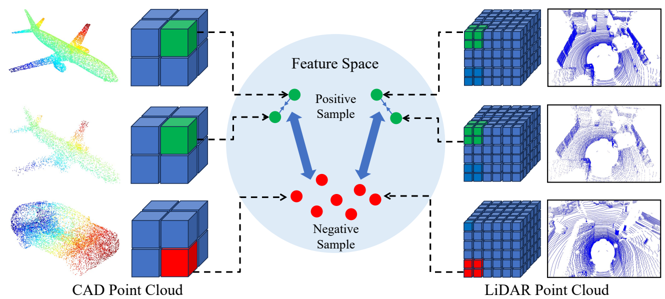

Fig. 1. The main idea of this method. For different types of point cloud samples, we use voxels of varying scales to partition the samples. All samples are processed using random data augmentation. Green voxels come from the same sample. Red voxels come from any sample other than the current one. In the feature space, the feature distances between similar samples are more compact through contrastive learning, while the feature distances between different samples are more separated.

and annotating them at a large scale is challenging and time-consuming. Therefore, leveraging extensive unlabeled data for unsupervised learning pretraining enables achieving favorable results in downstream tasks even when utilizing only a small quantity of labeled data.

Currently, several unsupervised learning methods for 3D point clouds have been proposed [8]–[15]. The main idea of these methods is contrastive learning. By utilizing the model to extract features from positive and negative samples and combining them with a contrastive loss function [16] for parameter updates, the model can better learn sample features. However, existing methods still have some problems. Firstly, in terms of constructing positive and negative samples, due to the randomness of point cloud sampling rules, it can lead to insufficient discriminative features during the contrastive phase, resulting in poor robustness of the trained model. Secondly, in terms of downstream tasks, existing methods primarily focus on shape classification and point cloud segmentation tasks for 3D point clouds, while constructing unsupervised learning models for point cloud scenes in autonomous driving that use LiDAR remains challenging.

In this paper, we propose the VoxelContrast unsupervised learning method for 3D point clouds, as shown in Fig. 1. Our approach takes a voxel-based perspective, using voxels to uniformly partition point cloud data, thereby constructing positive and negative voxel samples and achieving unsupervised learning for point clouds of any size. Compared to other methods that utilize point [10] or chunk [9] feature information, using voxels allows for the adaptation to point

[1]Equal contributions. Yuxiang Qin is with the School of Electronics and Information Engineering, Tongji University, Caoan Road 4800, Shanghai 201804, China. (e-mail: yuxiangqin@163.com)
[2]Equal contributions. Hao Sun is with BOSCH Corporate Research. (e-mail: sun_hao@u.nus.edu)

cloud datasets of various sizes by adjusting the voxel size. Additionally, it can avoid the ambiguity in feature contrast caused by random sampling. We conducted experiments on four datasets, ModelNet40 [17], ShapeNet [18], KITTI [19], and nuScenes [20], training the backbone network using unlabeled data. The experimental results demonstrate that our method outperforms the original supervised learning methods and other unsupervised learning methods when transferred to downstream tasks.

Our contributions are as follows:

- Our method further expands the adaptability of unsupervised methods on downstream tasks through voxelization of point clouds, and has good results in training point cloud data of different scales.
- In constructing positive and negative samples, to facilitate better learning of hard samples by the model, we introduce various 3D point cloud data augmentation methods, enhancing the discriminative challenge of positive samples. Additionally, through voxel feature fusion, we address the issue of poor discriminability of negative samples.
- Extend the MOCO [3] unsupervised learning framework to the point cloud field, further increase the number of negative samples in the contrastive learning, and train a more robust unsupervised model.
- We demonstrate the performance of method VoxelContrast on the downstream tasks of point cloud shape classification and 3D object detection. We use large-scale datasets for unsupervised pre-training, and are able to achieve better results on small-scale datasets.

II. Related Works

A. Representation Learning on Point Clouds

In images, Convolutional Neural Networks (CNNs) serve as the primary method for extracting sample features. Methods like VGGNet [21] and ResNet [22] effectively capture both local and global features of samples by employing multiple layers of convolution and pooling. Point cloud data is typically sparse and irregular, containing not only three-dimensional coordinate information but also potentially including color, normals, reflectance, and other data. Therefore, PointNet [23] and PointNet++ [24] excel in handling the permutation and rotation invariance of point clouds, achieving notable results in shape classification and semantic segmentation tasks. Methods [25]–[27] combine Graph Neural Networks (GNN) to process point cloud data by iteratively aggregating neighboring node information to update node feature representations, effectively extracting both local and global features of the point cloud. For large-scale point cloud data acquired from LiDAR, feature extraction is typically performed using voxel-based methods [28]–[31]. This involves discretizing the point cloud data into a voxel grid, making it easier to process and analyze while preserving the spatial distribution information of points in three-dimensional space, which allows for better capture of local features.

B. Unsupervised Learning on Point Clouds

Abundant and accurate data annotation is a prerequisite for the outstanding performance of deep learning in multiple fields. However, in most real-world scenarios, the quantity and quality of data are often insufficient for model training. In the domain of images, the design of proxy tasks can enable large-scale unsupervised pretraining of unlabeled data. Currently, there are also corresponding unsupervised learning methods proposed for 3D point clouds [8]–[15]. These methods typically focus on model pretraining for single objects or small-scale point cloud scenes, with downstream tasks including shape classification and part segmentation of point clouds. CrossPoint [14] introduces cross-modal data, enabling simultaneous unsupervised learning of both images and point cloud data. Methods [9], [11], [12] achieves this by partitioning point cloud chunk and constructing positive and negative samples for contrastive learning. PointContrast [10] accomplishes point-to-point contrastive learning through a reconstruction network. On the other hand, methods [8], [13] approach the contrast between samples from an instance perspective, utilizing different feature layers and viewpoints.

C. Contrastive Learning

Contrastive learning is widely applied in both supervised and unsupervised learning domains. Its goal is to make samples of the same class have more compact distances in feature space, while increasing the separation distance between samples of different classes [16]. In unsupervised learning for images, MOCO [3] and SimCLR [4] are typical methods of contrastive learning, and they both exhibit significant performance improvements when transferred to other downstream tasks. Regarding measuring the similarity between samples, they employ instance discrimination as a proxy task, which involves predicting positive samples that are similar to the current sample from a majority of negative samples. Here, positive samples are defined as samples obtained through data augmentation of the current sample, while negative samples are defined as any sample other than itself. To increase the number of negative samples, MOCO employs a dynamic queue to store and update them, and it introduces a momentum encoder to maintain the consistency of negative samples. SimCLR enhances the performance of contrastive learning by using larger mini-batches to increase the number of negative samples and adding linear layers to map sample features. Therefore, the quantity of negative samples has a significant impact on the performance of the model in contrastive learning.

III. Method

The core of 3D point cloud understanding is to learning discriminative, generic and robust representations that can capture the underlying shape. To achieve this goal in an unsupervised learning, we propose using the voxel representation to solve the problem of sample constrast in instance discrimination tasks. The overall framework of our method is presented in Fig. 2.

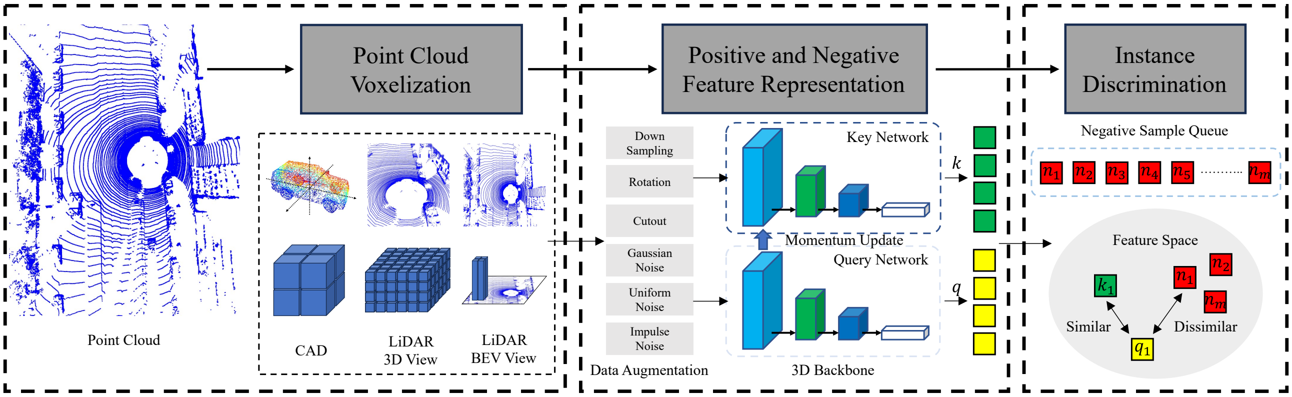

Fig. 2. Overview of our proposed VoxelContrast framework. The framework mainly consists of point clouds voxelization, positive and negative samples feature representation and instance discrimination. Firstly, the data is voxelized based on the type and view of point cloud to meet the requirements of different tasks. Then, various point cloud data augmentation methods are used to construct positive and negative samples, and the corresponding features are extracted through 3D backbone networks. Finally, instance discrimination is used as a proxy task, extending the MOCO [3] unsupervised learning method to the point cloud domain, training a backbone network with good feature representation effects.

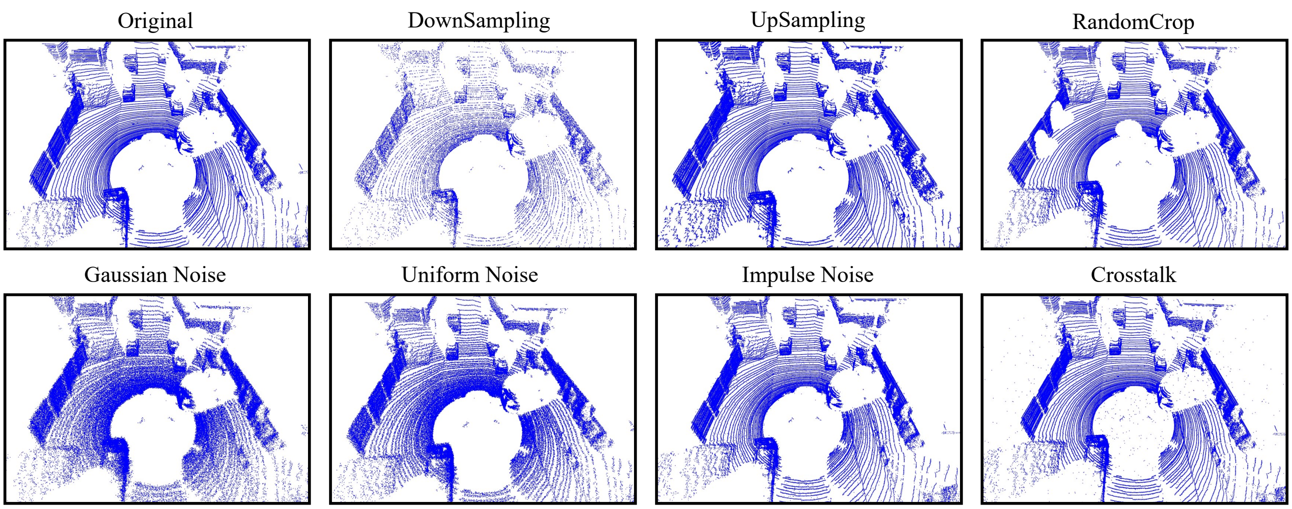

Fig. 3. Visualization of data augmentation for 3D point cloud. Taking the KITTI [19] dataset as an example to show the best view after partial magnification.

A. Point Cloud Voxelization

In point cloud shape classification tasks, since the data only contains a single object, directly extracting global features is an effective approach. However, for point cloud object detection, due to the large scene range and the large number of points, in order to avoid losing target information, the voxelization method can effectively retain local information. Voxelization is a method that can define the local structure of point clouds. Among them, the method of random sampling, as one of voxelization, produces relatively rough results. Therefore, through grid partitioning, a point cloud can be evenly divided into multiple regions.

Our proposed VoxelContrast utilizes voxel features as the main basis for contrast between samples. The voxelization method employed is consistent with current state-of-the-art 3D object detection algorithms, allowing the unsupervised learning of the backbone network to better adapt to downstream tasks. As illustrated in Fig. 2, we adopt different voxelization strategies for point clouds based on their type. For CAD point clouds, since the range of point cloud data is compressed within $[-1, 1]$, we can divide a point cloud sample into 8 voxels using the origin $(0, 0, 0)$. While considering more voxel divisions, the CAD point cloud contains only one target, leading to numerous empty voxels with uniform voxelization. Meanwhile, smaller voxels introduce difficulties in distinguishing between samples. Therefore, in this study, we adopt 8 voxels for partitioning this type of data. For LiDAR point clouds, the data originates from real-world scenes, encompassing a variety of objects and backgrounds, with a large range of point cloud coverage. Existing 3D object detection propose different voxelization approaches. VoxelNet [28] and SECOND [29] uniformly partition the point cloud from a 3D view using a consistent voxel size (v_D, v_W, v_H), dividing the 3D space (D, W, H) into feature space $(D^{'} = D/v_D, W^{'} = W/v_W, H^{'} = H/v_H)$. PointPillars [30] partitions the point cloud data from Bird's Eye View (BEV), distinct from VoxelNet and SECOND, as it does not consider partitioning along the Z-axis direction.

Therefore, it divides the 3D space (D, W, H) into feature space $(D^{'} = D/v_D, W^{'} = W/v_W, 1)$ in the form of pillar (v_D, v_W, H). In this paper, we consider the pillar as a special form of voxel.

B. Positive and Negative Feature Representation

In unsupervised contrastive learning, constructing reasonable positive and negative samples is a key condition for improving the robustness of unsupervised models. In this paper, instance discrimination is employed as a proxy task. Since there are no corresponding labels for the samples, we achieve unsupervised training by constructing positive and negative samples. Here, positive samples are defined as new samples generated through data augmentation from the current sample, while negative samples are defined as other samples excluding the current one. If a model can identify a unique positive sample from a large number of negative samples, it indicates good performance in feature extraction. Therefore, we need to address the following two issues: 1) The identification of positive samples should be challenging. 2) The features of negative samples should not be overly similar to those of positive samples.

For the generation of positive samples, we employed various data augmentation methods for point clouds, as shown in Fig. 3. These methods primarily include *DownSampling*, *UpSampling*, *RandomCrop*, *Gaussian Noise*, *Uniform Noise*, *Impulse Noise*, and *LiDAR Crosstalk*. Downsampling and upsampling primarily manipulate the point cloud density, uniformly reducing or increasing the density using *KNN* algorithms. Random cropping focuses on local point clouds, randomly selecting point cloud positions to crop the points within a certain range. Regarding point cloud noise, we introduced Gaussian noise, uniform noise, and impulse noise, each of which visibly affects point cloud data differently based on visualizations. LiDAR crosstalk [32] primarily arises from multiple sensors being close in proximity, causing numerous outlier points in the point cloud space. We implemented crosstalk effects by applying strong Gaussian noise to a subset of points. Moreover, for CAD point clouds, considering rotational and scale invariance, we introduce additional processing methods such as scaling and rotation apart from the previous approach. We control the degree of the above data augmentation through hyperparameters to ensure the reliability of generating positive samples.

For the sample distinguishability, we primarily consider that the voxel subdivision in 3D object detection methods for point clouds is too fine, resulting in a majority of highly similar samples during instance discrimination, thus reducing model robustness. We propose a voxel feature fusion method, as illustrated in Fig. 4. For voxel features $f_i \, (i = 1, ..., N)$, new voxel features $f_j = \sum_{i \in G_j} f_i$ is obtained through feature fusion, where $G_j \, (j = 1, ..., M)$ represents the set of indices for the j^{th} group of features. Taking PointPillars [30] as an example, the 3D backbone network extracts voxel features and generates corresponding pseudo images. At this stage, each voxel feature contains a relatively small range of point cloud data, leading to high similarity between voxel

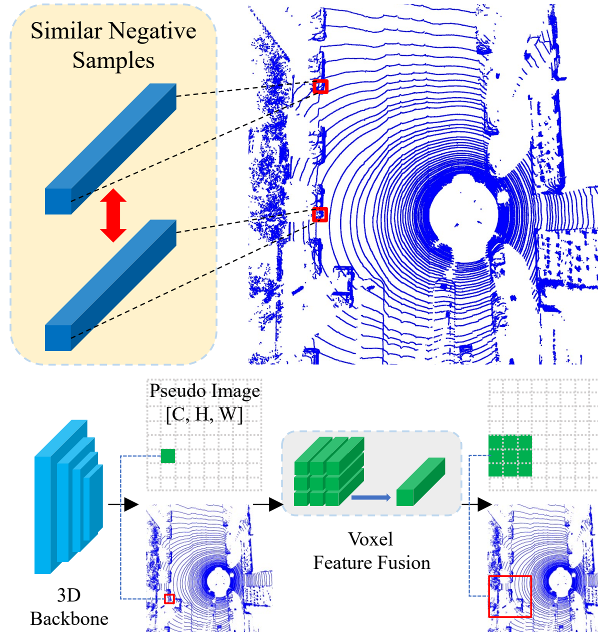

Fig. 4. Voxel feature fusion method. In PointPillars [30], voxels are processed through a 3D backbone network to extract features and generate pseudo images. The voxel feature fusion method integrates local voxel features to create new voxel features, enabling representation of a larger extent of the point cloud and avoiding highly similar features among negative samples.

features. Through voxel feature fusion, we recombine the original voxel features, reducing the number of voxels so that each voxel feature can represent a larger range of point clouds, thereby avoiding high similarity between sample features. Similarly, for the voxelization based on 3D view in both VoxelNet [28] and SECOND [29], we employed the same approach to reassemble the features extracted by the 3D backbone network.

C. Point Cloud Instance Discrimination

Instance discrimination is a common proxy task in unsupervised contrastive learning. MOCO [3] has achieved significant success in the field of unsupervised learning for images. We extend the MOCO framework to the point cloud domain, further improving the performance of VoxelContrast. For point cloud P, utilizing voxelization, data augmentation and 3D backbone feature extraction, we can obtain positive sample pairs f^q and f^k with 128-dim voxel features. Meanwhile, increasing the number of negative samples can enhance the difficulty of recognizing positive samples and strengthen the model's ability to extract features. We use a queue Q^{Neg} to store f^k as a set of negative samples. The objective of instance discrimination is to maximize the similarity among positive samples. Therefore, we introduced InfoNCE [33] for mutual information calculation, which can be expressed as:

$$L = \frac{1}{N} \sum_{i} - \log \left(\frac{e^{\boldsymbol{f}_i^q \cdot \boldsymbol{f}_i^k}}{e^{\boldsymbol{f}_i^q \cdot \boldsymbol{f}_i^k} + \sum_{z=1}^{Z} e^{\boldsymbol{f}_i^q \cdot \boldsymbol{Q}_z^{Neg}}} \right) \tag{1}$$

Z represents the number of negative samples, and it's evident that Eq. 1 is similar to softmax cross-entropy loss. The difference lies in the fact that each sample here represents a separate class, with a total of $Z+1$ classes. Additionally, at the end of each batch data training, we update the data in Q^{Neg} to avoid inconsistency in feature distribution during training.

Moreover, for the training of the 3D backbone network, MOCO adopts a double-channel network structure, namely θ_q and θ_k. The structures of the two networks are the same, with θ_q network being our training target, while the training for θ_k network adopts a momentum update method, which can be expressed as:

$$\theta_k \leftarrow \beta * \theta_k + (1-\beta) * \theta_q \tag{2}$$

where β denotes the extent to which the θ_k are updated each iteration, with larger values of β corresponding to smaller updates. To ensure the consistency of negative sample features, β is set to 0.999.

In order to better understand the calculation process of the VoxelContrast, Algorithm 1 summarizes the specific details.

Algorithm 1 VoxelConstrast's main algorithm.

Require: P = point cloud data
Require: $V(x)$ = rule of voxelization
Require: $S(x, y)$ = compute the similarity
Require: $\theta_q(x)$ = parameters of query network
Require: $\theta_k(x)$ = parameters of key network
initialize queue Q^{Neg}
initialize network parameters $\theta_k \leftarrow \theta_q$
for each data P **do**
  $v_i = V(P),\ i \in \{1, ..., N\}$
  $v_i^q, v_i^k = DataAugmentation(v_i)$
  $f_i^q = \theta_q(v_i^q)$
  $f_i^k = \theta_k(v_i^k)$
  if $Fusion$ **then**
    $f_j^q, f_j^k = \sum_{i \in G_j}(f_i^q, f_i^k),\ j \in \{1, ..., M\}$
  else
    $f_j^q, f_j^k = f_i^q, f_i^k,\ j \in \{1, ..., M\},\ M = N$
  end if
  $pos, neg = S(f_j^q, f_j^k), S(f_j^q, \sum_{z=1}^{Z} Q_z^{Neg})$
  $logits = cat([pos, neg], dim = 1)$
  $labels = zeros(M)$
  $loss \leftarrow L_{SoftmaxCrossEntropy}(logits, labels)$
  update θ_q using optimizer, e.g., SGD
  update $\theta_k \leftarrow \beta * \theta_k + (1-\beta) * \theta_q$
  $Q^{Neg}.put(f_j^k)$
end for
return θ_q

IV. EXPERIMENTS

A. Datasets

1) Shape Classification: For point cloud shape classification, we use the ModelNet40 [17] and ShapeNet [18] datasets to validate our method. ModelNet40 contains 40 shape categories and 12,311 data, 9,843 (80%) of which are train set and 2,468 (20%) are test set. ShapeNet contains 55 shape categories and 51,127 data. We merge the train set and validation set, and the train set and test set are 40,866 (80%) and 10,261 (20%), respectively. Since each sample contains a single object, we normalize the coordinate information of sample points to the unit sphere, which facilitates voxel partitioning.

TABLE I
COMPARE SUPERVISED LEARNING AND UNSUPERVISED LEARNING METHODS CLASSIFICATION ACCURACY (%) ON THE MODELNET40 DATASET. THE UNSUPERVISED METHODS ARE PRE-TRAINED ON THE SHAPENET DATASET.

Method	Points	Sup. or Unsup.	Instance Acc.	Class Acc.
PointNet [23]	1,024	Supervised	89.76	85.86
PointNet++ [24]	1,024	Supervised	90.94	87.43
SO-Net [34]	1,024	Supervised	92.68	88.42
PointCNN [35]	1,024	Supervised	92.26	88.01
DGCNN [25]	1,024	Supervised	93.25	89.47
DensePoint [36]	1,024	Supervised	92.72	88.52
RSCNN [37]	1,024	Supervised	92.84	88.54
Info3D [9]	1,024	Unsupervised	91.46	88.16
PointGLR [8]	1,024	Unsupervised	91.94	88.82
PointContrast [10]	1,024	Unsupervised	92.15	89.58
Ours	1,024	Unsupervised	**93.07**	**90.71**

2) 3D Object Detection: We conduct experiments on two widely used LiDAR 3D object detection datasets: KITTI [19] and nuScenes [20]. The KITTI dataset comprises 7,481 training samples and 7,518 test samples. Since only the training set is annotated, we further divide it into 3,712 samples for training and 3,769 samples for validation. This dataset provides annotations for three categories of objects: car, pedestrian, and cyclist. Additionally, for each sample, the dataset categorizes the level of difficulty into three levels: easy, moderate, and hard. The nuScenes dataset collected 1000 driving sequences, with 700, 150, and 150 of them allocated for training, validation, and testing, respectively. Each sequence is approximately 20 seconds long, with keyframes annotated at 0.5s intervals. In our experiments, we utilized 28,130 training samples and 6,019 validation samples from this dataset. It includes annotations for 10 common object categories.

B. Implementation Details

All the code in this paper is based on the deep learning library PyTorch [38]. All experiments are accomplished on NVIDIA A40 GPUs. In unsupervised learning training, for the shape classification data, we randomly sampled 1024 points from within 8 voxels to train the PointNet++ [24] backbone. For 3D object detection data, we employ the voxel feature fusion to obtain 64 voxel features for training the 3D backbone networks of VoxelNet [28], SECOND [29], and PointPillars [30]. For all models, we use the SGD optimizer with momentum set to 0.9 and weight decay set to 1. We

TABLE II

COMPARE SUPERVISED LEARNING AND UNSUPERVISED LEARNING METHODS IN 3D OBJECT DETECTION: AVERAGE PRECISION (%) ON THE KITTI VALIDATION SET. THE UNSUPERVISED METHODS ARE PRE-TRAINED ON THE NUSCENES DATASET.

Method	Unsupervised Training	Car (AP@0.7)			Pedestrian (AP@0.5)			Cyclist (AP@0.5)		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
VoxelNet [28]	Supervised	78.54	61.28	59.42	53.93	49.03	45.69	57.46	46.20	40.89
	PointGLR [8]	79.25	61.71	**61.13**	54.41	50.36	45.98	59.07	46.17	41.05
	Ours	**80.33**	**62.65**	61.05	**56.12**	**50.49**	**47.36**	**59.78**	**47.89**	**43.21**
SECOND [29]	Supervised	86.06	75.48	73.52	52.23	47.79	42.96	75.95	63.27	57.60
	PointGLR [8]	86.71	76.12	73.44	53.19	48.46	42.59	77.09	63.25	57.98
	Ours	**88.64**	**77.91**	**75.92**	**55.67**	**49.68**	**44.55**	**78.35**	**63.34**	**58.88**
PointPillars [30]	Supervised	82.02	76.23	70.74	49.64	43.17	40.10	74.35	59.49	56.08
	PointGLR [8]	82.42	76.79	71.12	50.13	42.91	**41.02**	74.82	60.73	55.61
	Ours	**83.82**	**76.98**	**72.09**	**50.92**	**45.09**	40.64	**76.94**	**61.25**	**57.54**

set the initial learning rate to 0.01 and employed the cosine dynamic learning rate adjustment method. The model is trained for 300 epochs, and we save the parameters of the model with the lowest loss. The length of the negative sample queue, denoted as Z, is set to 65,536 [3]. In the downstream task of shape classification, we add a layer of C-dim fully connected layer to the backbone, where C represents the number of categories. Then, we fine-tune the network using labeled data. For 3D object detection, we imported the parameters of the unsupervised trained 3D backbone into the original model and fine-tuned the parameters using annotated data. By conducting 100 epochs of training on the labeled data, we can effectively compare the performance of different unsupervised methods in extracting point cloud features.

C. Comparison with State-of-the-art Methods

1) Shape Classification: We conducted experiments on point cloud shape classification using ModelNet40 [17] and ShapeNet [18] datasets. Among them, ShapeNet has a larger amount of data, so we used the ShapeNet dataset for unsupervised learning, with PointNet++ [24] as the backbone network. As shown in Table I, the experimental results indicate that unsupervised pretraining achieves better results compared to supervised learning methods, suggesting that unsupervised methods can learn good point cloud representations from a large amount of unlabeled data, thus enabling better generalization of the model on small-scale datasets. Our proposed VoxelContrast can improve the accuracy by 2.13% over PointNet++, and surpassing other unsupervised learning methods.

2) 3D object detection: In the downstream 3D object detection task, we separately evaluated the results of models VoxelNet [28], SECOND [29], and PointPillars [30] under supervised and unsupervised learning conditions. We utilized a large dataset, nuScenes [20], as unlabeled data for unsupervised pre-training and tested the 3D object detection performance on the KITTI [19] dataset. We compared our unsupervised learning method with PointGLR [8]. Although the PointGLR paper did not include experiments related to 3D object detection, we made certain adjustments to its open-source project, while preserving the core methods as much as possible, enabling it to perform unsupervised training on LiDAR data. As shown in Table II, our proposed VoxelContrast can improve existing methods results. In VoxelNet [28] and SECOND [29], our method outperforms supervised learning by 2% to 3% in average precision (AP) across various categories. Similarly, in PointPillars [30], there is a 1% to 2% improvement. Our method outperforms PointGLR across multiple categories. We also observed a decrease in average precision for PointGLR on certain subclasses compared to supervised learning. This suggests that our proposed voxel-based contrastive method offers better feature representation for LiDAR point clouds. Moreover, We visualized some results of 3D object detection, as shown in Fig. 5. We can see that training the model with a large amount of unlabeled data further improves its robustness on small datasets. Our method achieves more accurate results for some occluded objects, such as vehicles and pedestrians, compared to supervised learning, while also reducing false positive detections.

TABLE III

THE IMPACT OF FREEZING THE BACKBONE NETWORK ON UNSUPERVISED LEARNING.

Method	Points	Sup. or Unsup.	Instance Acc.	Class Acc.
PointNet++ [24]	1,024	Supervised	90.94	87.43
Info3D [9]	1,024	Unsupervised	84.83	78.21
PointGLR [8]	1,024	Unsupervised	86.42	81.02
PointContrast [10]	1,024	Unsupervised	86.89	82.98
Ours	1,024	Unsupervised	**89.12**	**84.02**

D. Ablation Studies

1) Freezing the Backbone Network: In unsupervised learning, the effective extraction of features from samples is a crucial metric for evaluating methods. To further validate whether our proposed method extracts key features from samples, we froze the parameters of the backbone network and only trained the classification layer parameters during the downstream task. As shown in Table III, compared

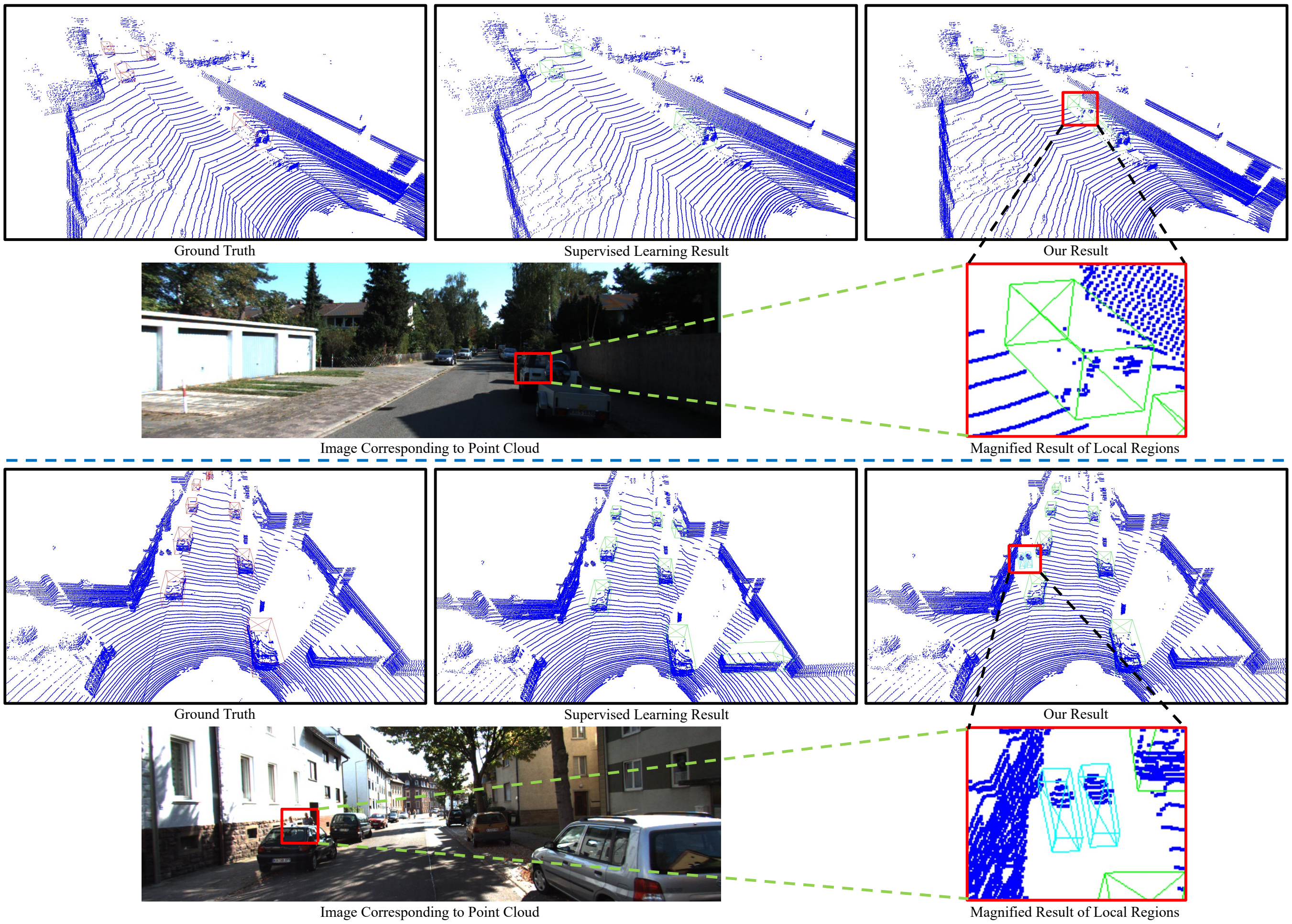

Fig. 5. Visualization of 3D object detection results using the SECOND method on the KITTI dataset. The top row presents the point cloud from left to right: ground truth, supervised learning result, and our result. The bottom row shows image corresponding to point cloud and magnified result of local regions.

TABLE IV
THE IMPACT OF DATA AUGMENTATION ON UNSUPERVISED LEARNING.

Randomly Select Data Augmentation	Car (mAP@0.7)	Pedestrian (mAP@0.5)	Cyclist (mAP@0.5)
1 method	78.72	47.86	65.97
4 methods	80.04	49.51	66.49
7 methods	80.82	49.96	66.85

to existing unsupervised methods, our method achieves a classification accuracy of 89.12% even when the backbone network is frozen, outperforming other unsupervised learning methods.

2) Data Augmentation: This experiment illustrates the impact of data augmentation methods on unsupervised learning. If the degree of data augmentation is small, the similarity between current samples and positive samples is high, making it easier for the model to identify positive samples from negative samples, resulting in ineffective learning of valid point cloud feature representations. In Section IV-C.2, we employed 7 data augmentation methods specifically designed for LiDAR point clouds. In this section, we tested the effects of unsupervised learning with the random selection of 1 and 4 data augmentation methods, as shown in Table IV. Under the SECOND [29] method, we can observe that the combination of multiple data augmentation methods further enhances the effectiveness of unsupervised learning.

V. CONCLUSIONS

Our proposed method, VoxelContrast, enables pretraining with large-scale unlabeled data to obtain more robust models. Voxelized point cloud data allows better focus on local features while adapting to different types of point cloud data. For feature representation of positive and negative samples, we employ various data augmentation techniques and voxel feature fusion to mitigate the decrease in model generalization caused by high similarity between samples. Additionally, to introduce more negative samples in instance discrimination tasks, we incorporate MOCO as the primary contrastive learning framework. Our experimental results demonstrate that unsupervised pretraining with VoxelContrast further improves the effectiveness of existing methods in both point cloud shape classification and 3D object detection downstream tasks.

REFERENCES

[1] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[2] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[3] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[5] H. Sun, Z. Meng, X. Du, and M. H. Ang, "A 3d convolutional neural network towards real-time amodal 3d object detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 8331–8338.

[6] Q. Jiang and H. Sun, "Semanticbevfusion: Rethinking lidar-camera fusion in unified bird's-eye view representation for 3d object detection," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 5707–5714.

[7] J. Wang, H. Zhu, H. Guo, A. A. Mamun, C. Xiang, and T. H. Lee, "Few-shot point cloud semantic segmentation via contrastive self-supervision and multi-resolution attention," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2811–2817.

[8] Y. Rao, J. Lu, and J. Zhou, "Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5376–5385.

[9] A. Sanghi, "Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*. Springer, 2020, pp. 626–642.

[10] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 574–591.

[11] M. Gadelha, A. RoyChowdhury, G. Sharma, E. Kalogerakis, L. Cao, E. Learned-Miller, R. Wang, and S. Maji, "Label-efficient learning on point clouds using approximate convex decompositions," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer, 2020, pp. 473–491.

[12] B. Du, X. Gao, W. Hu, and X. Li, "Self-contrastive learning with hard negative sampling for self-supervised point cloud learning," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3133–3142.

[13] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu, "Spatio-temporal self-supervised representation learning for 3d point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6535–6545.

[14] M. Afham, I. Dissanayake, D. Dissanayake, A. Dharmasiri, K. Thilakarathna, and R. Rodrigo, "Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9902–9912.

[15] Y. Wu, J. Liu, M. Gong, P. Gong, X. Fan, A. K. Qin, Q. Miao, and W. Ma, "Self-supervised intra-modal and cross-modal contrastive learning for point cloud understanding," *IEEE Transactions on Multimedia*, vol. 26, pp. 1626–1638, 2024.

[16] Y. Qin, C. Yan, G. Liu, Z. Li, and C. Jiang, "Pairwise gaussian loss for convolutional neural networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6324–6333, 2020.

[17] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[18] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[19] 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.

[20] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, 2015, pp. 1–14.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[24] 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.

[25] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[26] G. Li, M. Muller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9267–9276.

[27] W. Shi and R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1711–1719.

[28] 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.

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

[30] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[31] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-end multi-view fusion for 3d object detection in lidar point clouds," in *Conference on Robot Learning*. PMLR, 2020, pp. 923–932.

[32] L. Briñón-Arranz, T. Rakotovao, T. Creuzet, C. Karaoguz, and O. El-Hamzaoui, "A methodology for analyzing the impact of crosstalk on lidar measurements," in *2021 IEEE Sensors*. IEEE, 2021, pp. 1–4.

[33] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[34] J. Li, B. M. Chen, and G. H. Lee, "So-net: Self-organizing network for point cloud analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9397–9406.

[35] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, 2018.

[36] Y. Liu, B. Fan, G. Meng, J. Lu, S. Xiang, and C. Pan, "Densepoint: Learning densely contextual representation for efficient point cloud processing," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5239–5248.

[37] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8895–8904.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.