PROCEEDINGS OF SCIENCE

Evaluation on Momentum Contrastive Learning with 3D Local Parts

Xuanmeng Sha[a,*] and Tomohiro Mashita[b]

[a] *Graduate School of Information Science and Technology, Osaka University,*
Yamadaoka 1-5, Suita city, Osaka 565-0871, Japan

[b] *Faculty of Information and Communication Engineering, Osaka Electro-Communication University,*
18-8 Hatsucho, Neyagawa-shi, Osaka 572-8530, Japan

E-mail: u418576a@ecs.osaka-u.ac.jp, mashita@osakac.ac.jp

Self-supervised learning speeds up the representation learning process in lots of computer vision tasks. It also saves time and labor of labelling the dataset. Momentum Contrast (MoCo) is one of efficient contrastive learning methods, which has achieved positive results on different downstream vision tasks with self-supervised learning. However, its performance on extracting 3D local parts representations remains unknown. In our study, we make modifications on the MoCo model to learn the local features of ShapeNet, and design data augmentation methods and local clustering method to randomly generate local clusters. To evaluate proposed method, the evaluation experiments on different scales of local clusters and data augmentation methods with our method are performed, then we perform the 3D object classification downstream task on the local parts with pretrained model. From the results, the modified MoCo model shows great performance on extracting local representations and make the classification downstream task faster with pretrained model.

International Symposium on Grids and Clouds (ISGC2024)
24 -29 March, 2024
Academia Sinica Computing Centre (ASGC), Institute of Physics, Academia Sinica Taipei, Taiwan

[*] Speaker

1. Introduction

In light of the development and application with 2D vision tasks, the 3D vision tasks are also developed for processing 3D data. The main tasks include 3D object detection, 3D object classification, 3D semantic segmentaion, 3D reconstruction, etc. However, unlike the ordered and continuous 2D images, the 3D data has a lot of limitations and unique properties, which gives these tasks unpreventable challenges like huge computation costs, time-consuming annotations and information missing.

Due to the exclusive properties of 3D data, different types of data structure are used for solving 3D vision tasks [2, 10, 13, 21, 28–32]. The point cloud is the most basic format for presenting the real world. The samples of point cloud and image are shown in Figure 1. It is simply a set of 3D data points and each point contains the position coordinates information in the Euclidean space, and some of them contains other attributes like color, intensity, reflectivity, normal information, etc. It is regarded as a set of unstructured 3D points that symbolize the geometry of 3D objects and real scene [33]. With these unique properties, a lot of challenges for 3D vision tasks are derived, including the huge computation cost when processing unordered points, the time and labor cost by large amount of 3D annotation and the information lost [3–6]. Therefore, traditional supervised training mechanism is not enough for extracting simple but informative representations of point cloud to solve 3D vision tasks.

Self-supervised Training achieves great success in natural language processing [7, 8, 14], which provides a novel approach for 2D vision tasks [15–17]. With self-supervised training, a lot of obstacles of 3D vision tasks could be prevented, especially with contrastive learning. The contrastive learning is a popular mode of self-supervised training that encourages augmentation of the same input to have more comparable representations. The general approach is to expand the views of input point clouds by various data augmentation techniques [34]. In this way, the point cloud can be learned by itself with pretext task to gather useful information for downstream 3D vision tasks, which can reduce the computation cost [35]. Furthermore, the pseudo labels presented in the pretext task are a multidimensional matrix carrying collection information and are typically generated using clustering methods such as memory bank [17], which ignores 3D annotations. The missing information can also be prevented with different designed objective of pretext tasks [18, 36].

However, contrastive learning for whole shape of 3D objects lacks of local parts information, the contrastive learning ability for 3D local parts also remains unknown. To this end, a modified momentum contrast pretraining architecture for 3D local features is proposed following MoCo [1]. MoCo is one of the pioneer works that proposed momentum contrast mechanism for self-supervised training in multiple 2D detection and segmentation tasks, which is a suitable contrastive learning structure for 3D point cloud processing due to its simplicity. In the proposed structure, a point cloud feature learning backbone is leveraged. For extracting effective local features for downstream task, several 3D data augmentation methods and a random local clustering method are designed. With these methods, the local parts of 3D objects and their local features can be generated and learned randomly, which would be aligned with the situation of occluded 3D objects that captured in 3D real scenes.

To evaluate the proposed 3D local feature pretraining approach, ShapeNet [9] is chosen as the

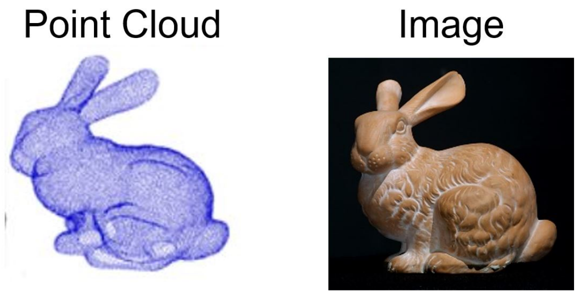

Figure 1: Samples of Point Cloud and Image

training dataset. Two comparison experiments and a downstream experiment are designed. The first comparison experiment evaluates the connection between local cluster size and the 3D local feature learning ability. The second comparison experiment we compare the effect of different data augmentation methods to present the effective data augmentation methods for 3D local feature training. The downstream experiment performs 3D local parts object detection with pretrained model. With these experiments, The 3D local feature learning performance for proposed modified momentum contrastive learning method can be verified and the best 3D local feature pretraining settings can be summarized.

2. Related Work

2.1 Self-supervised Training

Self-supervised training has been increasingly used in vision tasks instead of natural language processing tasks like GPT [7, 14] and BERT [8]. Previous works aims at imposing simple variations on image and extract features by recovering it to original input, like semantic-lable-based methods [37, 38] and cross-modal-based methods [39–41]. However, these methods cost extra computations and memory for generating labels and learning from other modalities. Recently, a lot of works show great interest on contrastive learning, which only relies on dataset itself by generating positive and negative samples with data augmentation method. For example, Wu et al. stores previous feature encodings in a memory bank [17]. Momentum encoders (MoCo) [1] extends the memory bank into an updating queue for negative samples feature with a momentum update encoder, which surpasses ImageNet-supervised counterpart in multiple detection and segmentation tasks. Other advances [15, 16] have also shown that self-supervised training is a successful approach for 2D vision tasks.

For 3D vision tasks, Xie et al. proposed PointContrast [18], which augments 3D scenes with rotations and color transformations then contrast transformed 3D point cloud with contrastive loss function. To address the limitation of requiring multiple views, Zhang et al. presented DepthContrast [19], which learns different formats of representations from depth maps. It uses only single view point cloud data, and constructs two augmented versions using data augmentation. The format-specific encoders generate spatial features with selected input formats of point or voxel. and learns 3D representations that are invariant to point and voxel representations. Then the global features are obtained for instance discrimination.

However, these works focus on pretraining point cloud with whole shape, which ignores the local parts information. In this work, the proposed framework takes local parts of point cloud as input and explores the local feature learning ability with self-supervised training.

2.2 Point Cloud Encoder

In the evolving landscape of 3D data processing based on the 3D structures, point cloud encoders have emerged as pivotal components in numerous 3D vision tasks. Point-based models capture point representations at point-wise level. Qi et al. 's PointNet [11] is a pioneer work to generate features directly from raw point cloud. It learns a spatial encoding of each point and then aggregate all individual point features to a global point cloud signature. However, it suffers the loss of local information. To tackle this problem, PointNet++ [12] introduced a hierarchical structure, which extracts point features while taking local and global features into account with multiple stages of sampling and grouping. Based on PointNet++, Qian et al. introduce a separable MLP and an inverted residual bottleneck design in PointNet++, named as PointNeXt [22]. Hu et al. proposed RandLA-Net, which uses random point sampling method and increase the receptive field through efficient local feature aggregation module [23]. Ma et al. present a residual model named PointMLP, which applies a geometric affine module instead of local geometry extractor [24]. Since the transformer was proposed [25], a lot of encoders apply the transformer structure to present features. Guo et al. proposed a transformer-based point cloud learning framework, Point Cloud Transformer (PCT), which provides permutation invariance and brings transformer to the point cloud feature learning field [26]. Then Pang et al. propose Point-MAE, which presents a masked autoencoder focusing on the local information leakage [27].

The works above propose the point cloud encoder for supervised training. In this work, the point cloud encoder is leveraged in a self-supervised training structure for extracting local features. To simply and effectively represent the 3D local clusters, the PointNet++ [12] is taken as the point cloud feature learning backbone, which can be directly transferred to the downstream tasks like 3D object detection.

3. Method

To train 3D local feature with contrastive learning, a modified momentum contrastive pre-training architecture for 3D local features is proposed. In this section, the proposed momentum contrastive learning structure for 3D features is explained, including the contrastive learning loss and momentum update mechanism. Furthermore, the point cloud feature learning backbone is presented. Finally, the data augmentation methods and random local cluster sampling method are discussed.

3.1 Momentum Contrastive Learning Structure

The proposed framework is shown in Figure 2. The structure is separated as two modules. The former part is the data augmentation module, which processes the input 3D Point clouds to the transformed 3D local clusters with queries and keys. The latter part is the local feature learning module, which learns the 3D local representations from separated query local cluster and key local cluster with contrastive learning.

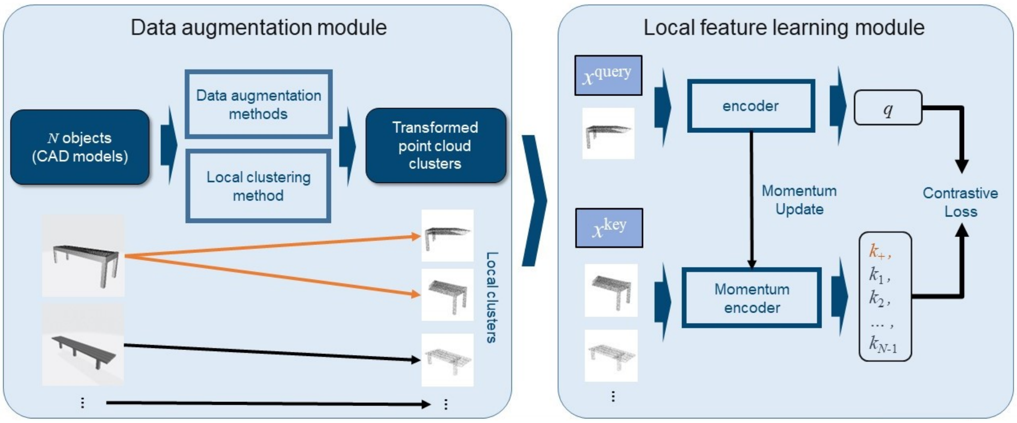

Figure 2: Momentum Contrastive Learning Structure for 3D Local Features

For the data augmentation module, it takes point cloud as input data. With the transform of data augmentation methods and local clustering method, a batch of transformed point cloud local clusters are outputted. The outputted batch of local clusters are considered as query and keys, a query and a key are a positive pair if they are originated from the same CAD model, others are negative pairs accordingly.

The data augmentation methods are used to generate 3D local clusters with multiple shapes from input CAD models with preprocessing, which helps the momentum contrast structure learn the local features more completely and robustly. The local clustering method is used to randomly sample from the transformed CAD models to generate 3D local clusters. Processed by these methods, the 3D local clusters with multiple shapes are presented.

For the local feature learning module, it learns local representations from query clusters and key clusters and update encoders with momentum contrast. To contrast the query feature and key features, the encoder and momentum encoder share the same model to keep their feature consistency. During training, the encoder is updated with back-propagation but the momentum encoder is momentum updated, which makes them the same structure but different parameters. With the generated query feature and key features, the contrastive loss tries to classify the query feature as its positive key feature, during this process, the encoders are updated.

3.1.1 Contrastive Loss

The local clusters are processed into query local features and key local features by encoder and momentum encoder respectively. Consider that there are N samples in one batch size, one query feature q and a set of key features $\{k_+, k_1, k_2, \ldots, k_{N-1}\}$ are generated from encoders. The keys are considered as the dictionary. k_+ and q are positive pairs because they originated from the same point cloud. q and the other key features $\{k_1, k_2, \ldots, k_{N-1}\}$ are considered as negative pairs accordingly. To make the query local feature q similar to its positive key k_+ and dissimilar to other negative keys, a form of contrastive loss function, called InfoNCE [20] is used as loss function:

$$\mathcal{L}_q = -\log \frac{exp(q \cdot k_+/\tau)}{exp(q \cdot k_+/\tau) + \sum_{i=1}^{N-1} exp(q \cdot k_i/\tau)}, \tag{1}$$

where τ is a temperature hyper-parameter that controls the smoothness of the softmax distribution. This loss refers to the form of classifier, which is the log loss of a $(K + 1)$-way softmax-based classifier that tries to classify the query q as its positive key k_+. It will minimize the distance between local features that are from the same point cloud and maximize it that are from different point cloud in the latent space. With this loss, it serves as an unsupervised objective function for training the encoders to represent the query and key features with better local information based on the contrastive learning from local parts of 3D object.

3.1.2 Momentum Update

With the above contrastive loss function, the encoders can be trained to present informative local feature of query cluster and key cluster. The key samples in the dictionary are presented as a queue, the samples in the dictionary are progressively replaced. The current mini-batch is enqueued to the dictionary and the oldest mini-batch is removed, which keeps the dictionary relatively new to train the encoders and supports larger size of the data subset.

However, the gradient cannot propagate to all samples in the dictionary with back-propagation, like the way updating the query encoder. The reason is that the query encoder updates rapidly, if the parameters of key encoder synchronize to the query encoder, it will cause inconsistency between the key representations and query representation. To prevent this situation, a momentum update is used for the momentum encoder:

$$\theta_k \longleftarrow m\theta_k + (1 - m)\theta_q, \tag{2}$$

where θ_q is the parameters of encoder f_q, θ_k is the parameters of momentum encoder f_k, and $m \in [0, 1)$ is a momentum coefficient, which gives f_k a slowly update based on the encoder f_q instead of back-propagation. With this momentum update step, the difference between the key features can be made small, and keeps relatively the same level for represent the features from the slowly updating queue. Following MoCo, $m = 0.999$ works much better.

3.2 Point Cloud Feature Learning Backbone

To learn the informative local features from the point cloud, a proper effective point cloud feature extracting backbone stands an important role. This work takes the PointNet++ [12] as the feature learning backbone. It takes the XYZ coordinates of the 3D data as input, and employs a U-Net [42] structure which has four layers of feature extraction and down-sampling, with two layers of feature aggregation and up-sampling. In this work, it takes 1024 points as the input from local cluster which presents as 1024×3 matrix of XYZ coordinates. Its final layer produces C dimensional per-point features for 128 points after the aggregation.

3.3 Data Processing

3.3.1 Data Random Augmentation Methods

For the data random augmentation methods, some standard 3D augmentation methods are adopted, which are the random input dropout, random rotation and random translation. The random input dropout involves randomly dropping out some of the points in the point cloud, the random rotation gives the different orientations to the point cloud and the random translation gives different positions to the point cloud. These methods are used to improve the robustness of the model to noise and partial occlusions, which is enough for the framework to completely learn 3D local features from randomly transformed point cloud. We also set experiments to evaluate the effects of these three augmentation methods.

3.3.2 Local Clustering Method

For the local clusters sampling method, the K-dimensional Tree (KD-Tree) algorithm is leveraged, which is a space-partitioning data structure that stores a set of k-dimensional points in a tree structure that enables efficient range searches and nearest neighbor searches. It is suitable for sampling local clusters from original point cloud due to its small time complexity, which can greatly speed up the pretraining process.

4. Experiments

Two steps of experiments are presented. The first part is the momentum contrast pretraining, different size of local point cloud and different data augmentation methods are compared to evaluate how these two factors influence the momentum contrastive pretraining. The second part is the downstream task, the best pretrained model is used to perform 3D local parts object classification task. The result is compared with the model without pretraining.

4.1 Dataset

ShapeNet [9] is used as the training dataset. As shown in Figure 3, it is a collection of single-object CAD models that contains 57448 objects from 55 categories developed by researchers from Stanford University, Princeton University, and the Toyota Technological Institute at Chicago, USA. The dataset is widely used for 3D vision tasks and is a rich source of information for computer graphics and vision research.

In both experiments, we choose 30% of ShapeNet dataset as the training set randomly. In the downstream task, only 5 typical classes of the ShapeNet is selected to perform the 3D object classification with pretrained model to evaluate the method, including table, chair, lamp, bench and bookshelf. Each local cluster is downsampled to 1024 points for training from input point cloud. The batch size is equal to 64, and learning rate is 0.007. 200 epochs are trained for each training.

4.2 Momentum Contrast Pretraining

Two comparison experiments are set to evaluate the proposed momentum contrast pretraining method. The first one is pretrined with different partial scales of sampled local clusters. The purpose is to conclude the smallest size of the local clusters that can be trained with momentum contrast. The second one is pretrained with different data augmentation methods. It is to find the efficient data augmentation method that help extracting the local features.

4.2.1 Training with Different Partial Scales

This experiment learns the local features for different size of local clusters with momentum contrast and compares the loss value and accuracy for each local cluster size. From the results,

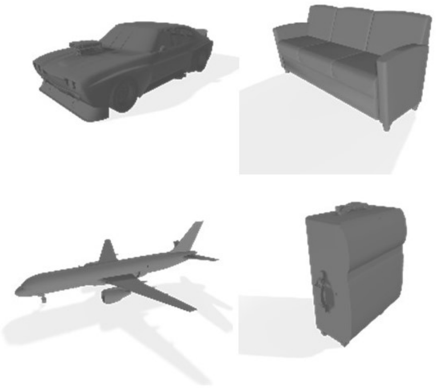

Figure 3: Samples of ShapeNet

it can be concluded if the momentum contrast is effective for learning 3D local features and the smallest size of local clusters for learning 3D local features.

The model is trained with the size of local clusters from 20% to 90%. The random data augmentation methods are the same. The final training result is shown in Table 1. The Acc@1 means the top-1 classification accuracy, which means the highest probability for classification is the positive key. The Acc@5 means the top-5 classification accuracy, which means the highest 5 probabilities for classification include the positive key. As the table 1 shows, The Acc@1 and Acc@5 increase when the local cluster size gets larger, which proves that the local features can be trained with the proposed contrastive learning framework. However, when the size of local cluster is smaller than 30%, the performance decrease drastically. It proves that over 30% of the local parts is effective for the proposed method learning from 3D local features.

4.2.2 Training with Different Data Augmentation Methods

This experiment learns the 3D local features with different data augmentation methods and compares the loss value and accuracy. Each training removes one of the data augmentation methods

Table 1: Comparison Result with Different Size of Local Cluster

Percentage of local cluster size	Loss	Acc@1 [%]	Acc@5 [%]
90%	8.05	44.71	50.57
80%	7.90	50.83	51.06
66%	8.03	51.68	51.68
50%	8.13	50.65	51.03
33%	8.00	50.59	50.62
30%	8.03	25.62	25.98
25%	8.52	0.31	1.24
20%	8.21	0.28	1.08

Table 2: Comparison Result with Different Data Augmentation Methods

Data augmentation methods	Loss	Acc@1 [%]	Acc@5 [%]
All	8.13	50.65	51.03
w/o random input dropout	8.34	0.59	1.45
w/o random rotation	8.19	49.85	50.98
w/o random translation	8.36	11.29	12.37

to evaluate which data augmentation method combination is the best for learning 3D local features.

Each training tasks 50% local parts of the cluster as input. The comparison result is shown in Table 2. From the result, removing random input dropout and random translation makes the classification accuracy decrease, which means these two methods are effective for the proposed method to learn more local details from transformed point cloud. However, without random rotation, the accuracy stays the same level, which proves that the random rotation cannot contribute much for the proposed method learning 3D local features.

4.3 Downstream Tasks

To further prove that the proposed momentum contrastive pretraining method is effective for extracting great 3D local features. The downstream 3D object classification task is performed with and without the pretrained model to make a comparison. Before the downstream experiment, the momentum contrast pretrained model should be outputted first. In the 3D object classification experiment, the pretrained model is verified with a common protocol following [12]. The setting of the dataset and the data processing is the same as momentum contrast pretraining experiment. The training epoch is 200 with the 0.001 learning rate. The batch size is 24 and the optimizer is Adam.

4.3.1 3D Object Classification on Local Parts

We first use the best setting of data augmentation methods and local clustering method to train the classification model with proposed momentum contrast method from 5 categories of dataset. The best top-1 pretraining accuracy is 51.42% in 132 epoch. The whole training takes about 6 and a half hours.

Two comparison experiments of 3D local parts classification are conducted. The first one trains the original initialized PointNet++ classification model to classify the local clusters of point cloud from 5 categories of ShapeNet. The second one trains the pretrained PointNet++ model from the best momentum contrast pretraining result above. The classification result is shown in Table 3.

From the result, after 200 epochs of training, the best instance accuracy and best class accuracy stays the same level in both experiment. However, the pretrained model can be trained much faster than the original classification model. It can be concluded that the 3D object classification on local parts can be trained with less time when using proposed momentum contrast pretraining method, which proves that the proposed momentum contrast pretraining method possesses the ability of learning local features from 3D point cloud.

Table 3: Comparison Result of 3D Object Classification on Local Parts

	Best Instance Accuracy [%]	Best Class Accuracy [%]	Epoch Number for Best Result
w/ Pretrained Model	81.35	75.70	EP168
w/o Pretrained Model	81.57	76.08	EP199

5. Conclusion

This work proposed a momentum contrastive learning framework for 3D local parts of point cloud and designed experiments to evaluate the method. From the result, it can be proved that the 3D local parts of point cloud can be trained with proposed method. The best setting for pretraining the 3D local parts is concluded, and also it shows great performance on speeding up the downstream task.

In future work, we intend to adjust our training set. Currently, ShapeNet has a lot of data that come from the same category, which cause great limitations to our training experiments. We believe that the dataset with more categories and less data that come from the same label will raise our proposed method to a new level, then we also intend to leverage our method to a small supervised dataset with low annotation cost to make a comparison. Finally, we can summarize the pretrained local features as a dictionary for off-line 3D object classification system.

References

[1] 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.

[2] 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.

[3] D. Ghadiyaram, D. Tran, and D. Mahajan, "Large-scale weakly-supervised pre-training for video action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 046–12 055.

[4] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 2020, pp. 491–507.

[5] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Van Der Maaten, "Exploring the limits of weakly supervised pretraining," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 181–196.

[6] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.

[7] 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.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[9] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, “A scalable active framework for region annotation in 3d shape collections,” *ACM Transactions on Graphics (ToG)*, vol. 35, no. 6, pp. 1–12, 2016.

[10] G. Shi, R. Li, and C. Ma, “Pillarnet: High-performance pillar-based 3d object detection,” *arXiv preprint arXiv:2205.07403*, 2022.

[11] 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.

[12] 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.

[13] S. Shi, X. Wang, and H. Li, “Pointrcnn: 3d object proposal generation and detection from point cloud,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 770–779.

[14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[15] 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.

[16] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020.

[17] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.

[18] 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.

[19] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra, “Self-supervised pretraining of 3d features on any point-cloud,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 252–10 263.

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

[21] C. R. Qi, O. Litany, K. He, and L. J. Guibas, “Deep hough voting for 3d object detection in point clouds,” in *proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9277–9286.

[22] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem, “Pointnext: Revisiting pointnet++ with improved training and scaling strategies,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 192–23 204, 2022.

[23] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, “Randla-net: Efficient semantic segmentation of large-scale point clouds,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 108–11 117.

[24] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, “Rethinking network design and local geometry in point cloud: A simple residual mlp framework,” *arXiv preprint arXiv:2202.07123*, 2022.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[26] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, “Pct: Point cloud transformer,” *Computational Visual Media*, vol. 7, pp. 187–199, 2021.

[27] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, “Masked autoencoders for point cloud self-supervised learning,” in *European conference on computer vision*. Springer, 2022, pp. 604–621.

[28] 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.

[29] 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.

[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] 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*, vol. 131, no. 2, pp. 531–551, 2023.

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

[33] A. S. Gezawa, Y. Zhang, Q. Wang, and L. Yunqi, "A review on deep learning approaches for 3d data representations in retrieval and classifications," *IEEE access*, vol. 8, pp. 57 566–57 593, 2020.

[34] C. Zeng, W. Wang, A. Nguyen, and Y. Yue, "Self-supervised learning for point cloud data: A survey," *Expert Systems with Applications*, p. 121354, 2023.

[35] J. Hou, B. Graham, M. Nießner, and S. Xie, "Exploring data-efficient 3d scene understanding with contrastive scene contexts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 587–15 597.

[36] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," *arXiv preprint arXiv:1809.10341*, 2018.

[37] A. Faktor and M. Irani, "Video segmentation by non-local consensus voting." in *BMVC*, vol. 2, no. 7, 2014, p. 8.

[38] I. Croitoru, S.-V. Bogolin, and M. Leordeanu, "Unsupervised learning from video to detect foreground objects in single images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4335–4343.

[39] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 609–617.

[40] P. Agrawal, J. Carreira, and J. Malik, "Learning to see by moving," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 37–45.

[41] D. Jayaraman and K. Grauman, "Learning image representations tied to ego-motion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1413–1421.

[42] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.