A Cross Branch Fusion-Based Contrastive Learning Framework for Point Cloud Self-supervised Learning

Chengzhi Wu[1] Qianliang Huang[1] Kun Jin[2] Julius Pfrommer[3] Jürgen Beyerer[1,3]

[1]Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Germany
[2]Department of Automation, Tsinghua University, China
[3]Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Germany

chengzhi.wu@kit.edu, qianliang.huang@student.kit.edu, kun6jin@gmail.com,
{julius.pfrommer, juergen.beyerer}@iosb.fraunhofer.de

Abstract

*Contrastive learning is an essential method in self-supervised learning. It primarily employs a multi-branch strategy to compare latent representations obtained from different branches and train the encoder. In the case of multi-modal input, diverse modalities of the same object are fed into distinct branches. When using single-modal data, the same input undergoes various augmentations before being fed into different branches. However, all existing contrastive learning frameworks have so far only performed contrastive operations on the learned features at the final loss end, with no information exchange between different branches prior to this stage. In this paper, for point cloud unsupervised learning without the use of extra training data, we propose a **C**ontrastive **C**ross-branch **A**ttention-based framework for **Po**int cloud data (termed PoCCA), to learn rich 3D point cloud representations. By introducing sub-branches, PoCCA allows information exchange between different branches before the loss end. Experimental results demonstrate that in the case of using no extra training data, the representations learned with our self-supervised model achieve state-of-the-art performances when used for downstream tasks on point clouds.*

1. Introduction

Contrastive learning stands as a pivotal method for learning latent representations, especially in the domain of computer vision and natural language processing. However, while its success has been widely demonstrated in 2D image-based tasks [9, 10, 27, 49], its application to self-supervised contrastive learning on 3D point cloud data has remained relatively underexplored. Point clouds possess unique characteristics and structural complexities that necessitate tailored approaches for representation learning. Despite the challenges, and while the majority still use reconstruction-based methods [3, 5, 11, 28, 32, 48, 65], recent research has started to address this gap. Promising advancements have been made in the domain of contrastive learning-based point cloud self-supervised learning. For example, STRL [29] extends BYOL [49] to the 3D domain and performs contrastive learning on global representations directly. PointContrast [64] performs contrastive learning on the point-wise level yet is computationally expensive. While Info3D [50] uses the shape part and the full shape as positive pairs directly, HSN [17] only considers the part pair information and ignores the information of the whole shape.

On the other hand, in recent years, the surge of large models has sparked significant interest in multi-modality-based approaches that leverage extra training data for contrastive-based point cloud representation learning. For example, CrossPoint [2] and I2P-MAE [72] use multi-view images of the objects as extra training data, while ReCon [47] further introduces additional text information for richer latent representation learning. However, there are still many cases that only single modality data is available for the training. Moreover, in all the above methods, latent representations are learned solely on each branch, with no information exchange before the loss end. In this paper, we rethink the way of applying contrastive learning to point clouds without extra training data, and explore the new possibility of incorporating information from different branches when using a multi-branch framework.

The key idea of contrastive learning is to use multiple branches to learn multiple latent representations for the same input of different variants, and the network is trained by minimizing their latent representation differences. When multi-modality is used, it is natural to use one branch for each modality. For example, CrossPoint [2] uses one branch for images and the other branch for point clouds. The encoders for each modality on the perspective branches are totally different and the whole framework can be trained stably. When only one modality is used, the most widely used method is to perform different augmentations on the same input and use augmented variants for different branches [29]. However, the model may collapse easily if the vanilla model of both branches sharing a same encoder is used, i.e., the encoder encodes everything into a same latent representation. Various methods have been proposed to deal with this problem, including introducing negative pairs [9, 27], using memory bank [59, 63], adding a predictor on single branch [10, 49], and using momentum update for certain branch encoder [27, 29, 49], etc. In our case, we only use a single modality of point cloud data and only use positive pairs for training. Following BYOL [49], the single-branch predictor strategy and the momentum update strategy are adopted to prevent model collapse.

By using each branch to learn one latent representation solely, all existing contrastive learning frameworks only performed contrastive operations on the learned features at the final loss end, with no information exchange between different branches prior to this stage. At this point, a question arises: is it possible to incorporate information from different branches before the loss end? In this paper, we explore this possibility by introducing sub-branches on the common frameworks. To this end, we propose a self-supervised contrastive learning framework for point clouds by fusing (i) the online branch information and the target branch information; (ii) the global sub-branch information and the local sub-branch information, as illustrated in Figure 1. PoCCA is a symmetric 3D point cloud representation learning framework. It augments the raw point clouds and then samples them globally or locally on different sub-branches. The features from different branches are fused subsequently. Using different augmentations of the same raw point cloud as a positive pair, the contrastive loss is defined by their latent representation difference. Further details of the framework are given in Section 3. Experimental results demonstrate that the representations learned with our self-supervised model achieve excellent performances when used for downstream tasks on point clouds.

We summarize our contributions as follows:

- A contrastive learning framework that enables information exchange between the online branch and the target branch by introducing sub-branches.
- Both global and local features of the input point cloud are extracted. Cross-attention modules are used for global-local feature fusion.
- Multiple variants of the proposed contrastive learning framework are evaluated in the ablation study, as well as various local patch sampling methods.
- Excellent experimental results on multiple downstream tasks. Among the point cloud unsupervised learning methods that do not use extra training data, PoCCA achieves state-of-the-art results.

2. Related Work

2.1. Contrastive Learning

Self-supervised learning methods have made great strides in recent years. They are usually either generative-based [14, 33] or contrastive-based [6, 10, 27, 49]. Contrastive learning, in contrast to generative models, is a discriminative strategy that tries to separate varied samples while grouping similar samples. As a pioneering work, InstDisc [63] separates the extracted features in high dimensional space and categorizes the features as positive and negative samples. Contrastive Multiview Coding [54] extends the definition of positive samples by considering different views from the same object as positive pairs. MoCo [27] proposes a strategy of stopping-gradient on the target branch and instead uses a moving-averaged encoder. SwAV [6] further discards the negative samples and compares new sample features with the center of positive features clustering. SimCLR [9] demonstrates that unsupervised contrastive learning benefits from stronger data augmentations, and a non-linear transformation between the representation and the contrastive loss can significantly improve the quality of the learned representations. SimSam [10] and BYOL [49] further add a predictor at the end and both use only positive pairs for self-supervised learning.

2.2. Self-supervised Learning on Point Cloud

In order to accomplish self-supervised representation learning on 3D point clouds, various methods have been proposed. Pretext tasks-based self-supervised learning methods are first explored. Jigsaw [52] is trained by reconstructing point clouds from randomly rearranged parts. PointRotation [44] learns point cloud representations by predicting their rotation. STRL [29] takes two temporally-correlated frames from a 3D point cloud sequence as the input, transforms it with the spatial data augmentation, and learns the invariant representation in a self-supervised manner. Contrastive-based learning has also been used in several works. Wang et al. [57] pre-trains an encoder with occluded points for downstream tasks. Point-level invariant mapping is carried out by PointContrast [64] on two transformed views of the input point cloud. CrossPoint [2] performs cross-modality contrastive learning between point clouds and their corresponding rendered images. More recently, MAE-based reconstruction methods have shown promising results on point cloud self-supervised learning. Point-BERT [69] and Point-MAE [43] are the first two methods that transfer the idea of MAE to point clouds by masking point patches. A similar encoder is used in MaskPoint [39],

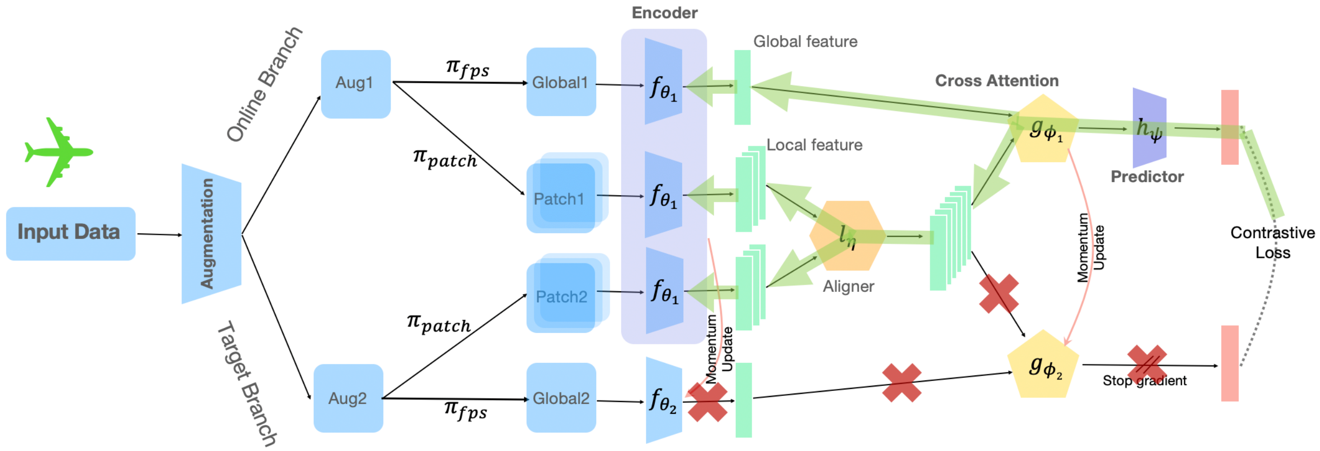

Figure 1. The framework of our proposed PoCCA. Given an input point cloud, it is first augmented with two different augmentation operations. After global sampling and local patching, global and local features are obtained respectively with a pre-designed encoder (e.g. PointNet, DGCNN). The local features from the two branches are then aligned and sent to the cross-attention module to enhance the online- and target global features, respectively. Finally, by comparing the difference between the output representations, we obtain a contrastive loss and train the whole model. See more module details in subsection 3.3.

but instead of using the transformer as a decoder for reconstruction, it uses the transformer as a discriminator. Point-M2AE [71] further updates the encoder with a hierarchical Transformer by introducing multi-scale masking. Using extra training data, I2P-MAE [72] and ReCon [47] obtain superior 3D representations via cross-modal training.

2.3. Attention Mechanism on Point Cloud

In recent years, attention-based methods start dominating the image learning domain since ViT [15]. More recently, the attention mechanism [56] has also been proven to be effective for point cloud learning. PCT [24] employs self-attention for point cloud understanding with proposed offset-attention. By constructing a residual point transformer block with self-attention-based layers and linear projections, PT [73] builds a U-Net-like network structure. Pointformer [42] proposes a local-global Transformer to integrate features from different levels. 3DPCT [41] designs a dual transformer approach and builds a hierarchical encoder-decoder network. SA-Det3D [4] proposes a generic globally-adaptive context aggregation module and a scalable self-attention variant is designed. Pyramid Point Cloud Transformer [30] develops a pyramid module to aggregate multi-scale features. SeedFormer [75] and PoinTr [68] employ attention-based methods for point cloud completion. APES [60] uses the attention map for sampling edge points of the point clouds. PatchFormer [13] proposes patch-attention and a lightweight multi-scale attention block. Stratified Transformer [35] is proposed to additionally sample distant points as keys to capture long-range contexts and demonstrates strong generalization ability.

3. Methodology

3.1. Preliminaries

Denote two main branches as branch A and B (in most papers, they are referred to as online branch and target branch), they have sub-branches A_1, A_2 and B_1, B_2. By introducing sub-branches, we can fuse the information on sub-branches, e.g. A_2 and B_2, before the loss end, and share this fused feature for A_1 and B_1 for further operations. In this case, the fused feature should be useful for A_1 and B_1 in generating richer latent representations. In the point cloud deep learning domain, local-to-global cross-attention for feature fusion has been proven to be an effective operation [13, 20, 42] and is ideal in our case. Therefore, we use sub-branches A_1 and B_1 for global feature learning, and sub-branches A_2 and B_2 for local feature learning. In practice, since the patch features are from different augmentations, for one certain branch, using the local features from the other branch as the key and value input directly is not ideal for the subsequent step of local-to-global cross attention. We hence further propose an additional aligner module to align the local features from different augmentations.

As the basics of the framework have been established, the only question that remains to be answered now is: For the vanilla mode that only has two main branches, it is quite clear that the encoder on one branch updates with the gradient backpropagation normally, while the encoder on the other branch (gradient stopped) updates with momentum update. But in our framework, there are four encoders on four sub-branches, which ones should be updated normally, and which ones should be momentum updated? Our proposal and also the best practice is: follow the gradient. The proposed framework is illustrated in Figure 1, in which green arrows indicate the gradient flow, with red crosses indicating that gradients are stopped on these paths. In this case, we update the encoders on sub-branches A_1, A_2, and B_1 normally with backpropagation (which means they share a same encoder), while updating the encoder on sub-branch B_2 with momentum update.

3.2. Overall Framework

Given a point cloud $\mathcal{P}$, the goal of PoCCA is to pre-train a powerful encoder f_θ that can encode it into a good latent representation that can be used for downstream tasks. There are four branches in our network architecture: online global branch, online patch branch, target global branch, and target patch branch, corresponding to the four branches in Figure 1 from top to bottom. The online branch consists of five stages: (i) two sampling sub-branches to obtain a global sample with sampling function π_{fps} and multiple local samples with sampling function π_{patch}; (ii) an encoder f_{θ_1} for shape global/patch encoding; (iii) an aligner l_η for local feature aligning; (iv) a cross-attention module g_{ϕ_1} for merging local and global features; and (v) a predictor h_ψ. The target branch uses the same structure as the online branch but without the predictor. Moreover, while the target patch branch uses a parameter-shared encoder f_{θ_1} with the online branch, the target global branch uses another encoder f_{θ_2} whose parameters are momentum updated with the online parameters in f_{θ_1}. The same goes for the target attention module g_{ϕ_2}, whose parameters are momentum updated with the online parameters in g_{ϕ_1}.

3.3. PoCCA Step by Step

Augmentation. Many successful self-supervised learning approaches cast the prediction problem directly into representation space: the representation of one augmented view of an image should be similar to the representation of another augmented view of the same image [8, 10, 49]. However, compared to 2D image benchmarks that have millions of training samples, 3D datasets are typically much smaller in size, and often have fewer labels and less diversity. Therefore, for 3D vision, data augmentation is a very important step to avoid overfitting and improve the generalization ability of the network. Same as many previous methods [45, 46], we augment input point clouds with random rotation, scaling, translation, and jittering.

Sampling methods. We sample the raw point cloud to capture its global and local features, respectively. Farthest point sampling (FPS) [19] is used for global sampling. For local sampling, given a desired number of patches N_p, such many kernel points are first selected with FPS on the original point cloud. Then for each kernel point, K nearest neighbors are gathered to form a patch of K points.

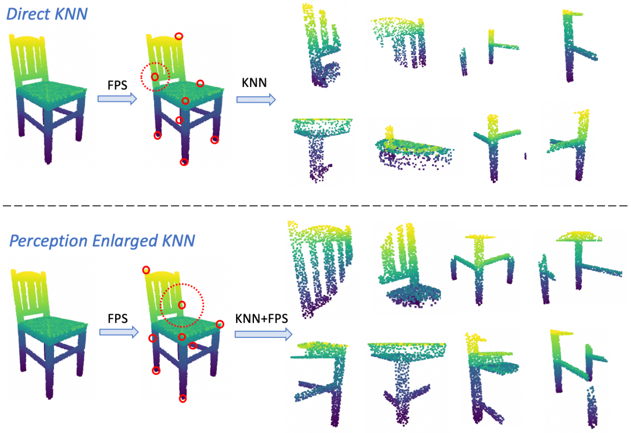

Figure 2. Perception enlarged KNN-based patch sampling. After kernel points are selected, direct-KNN gets patches with Kneighbors for each kernel point directly, while perception-enlarged-KNN gets patches with $2^\alpha K$ neighbors first and then samples them to K points with FPS. α is the scale factor.

In our experiments, we adopt a perception-enlarged multi-scale KNN sampling strategy. A demo is given in Figure 2 for output patch comparison of direct KNN and perception-enlarged KNN. The multi-scale idea has also been used in many other advanced frameworks in other fields, e.g. GCN [36] and Stratified Transformer [34]. Denoting α as the scale factor, after N_p kernel points are first selected, $2^\alpha K$ points are gathered with KNN, and subsequently downsampled to K points with FPS. For scale 0 ($\alpha = 0$), it falls into the normal direct KNN method. We use multi-scale $\alpha = 0, 1, 2$ as the default setting.

Encoder. For a fair comparison with existing self-supervised methods, we use PointNet [45], DGCNN [58] and Transformer as the backbone for extracting point cloud features. To enable contrastive learning, the encoders for the online global branch, the online patch branch, and the target patch branch share the same weights, while the encoder for the target global branch is updated via the momentum update with the weights from other branches. Specifically, we parameterize the online branches with θ_1 and the target branch with θ_2. The target branch is used to train the online branches, and its parameters θ_2 are an exponential moving average of the online parameters θ_1.

$$\theta_2 \leftarrow \tau\theta_2 + (1-\tau)\theta_1 \tag{1}$$

where $\tau \in (0, 1)$ is the decay rate of moving average.

Aligner. The aligner module is an important component of PoCCA. It enables information exchange between the online branch and the target branch. Since the patch features are from different augmentations, for one certain branch, using the patch features from the other branch as the key and value input directly is not ideal for the subsequent step of local-to-global cross attention. We propose to use an aligner module to align the patch features from different augmentations. The aligner module is basically a mixture of self-attention and cross-attention by using the local patch features from two branches as the queries separately, while using them both as the key and value dictionary. Its detailed architecture is given in Figure 3.

Cross-Attention. Cross-attention takes two separate embedding sequences of the same dimension and fuses them asymmetrically. For local-global feature fusion, the global feature serves as the query input, while the local patch features serve as the key and value input. As with updating the encoder, we use the momentum update method to update our cross-attention. Specifically, we denote the parameters for the online cross-attention module as ϕ_1, for the target cross-attention module as ϕ_2, and $\tau \in (0,1)$ keeps up with the former τ.

$$\phi_2 \leftarrow \tau\phi_2 + (1-\tau)\phi_1 \tag{2}$$

Predictor. In particular, inspired by BYOL [49], PoCCA uses symmetric network branches that interact and learn from each other. Since the training pair comes from the same original point cloud, it is possible for the encoder to produce the same representation for all augmented samples, which means that the network falls into a collapsed solution. In order to prevent collapse, and based on the experience of other methods [10, 49], we append a predictor to all online branches, which has been shown to be effective. Our predictor is a simple MLP-based network consisting of two linear layers and a batch norm layer.

3.4. Loss

As described in Section 3.2, for a given point cloud $\mathcal{P}$, PoCCA first augments $\mathcal{P}$ to get two augmentations m_1 and m_2 by using two different augmentations. Then we process m_1 and m_2 with sampling function π_{fps} to get two global samples σ_1, σ_2, and with sampling function π_{patch} to get a set of local samples $\mathcal{A} = \{a_1^1, a_1^2, \ldots, a_1^{N_p}, a_2^1, a_2^2, \ldots, a_2^{N_p}\}$, where each a denotes one certain patch. After that, σ_1 and $\mathcal{A}$ are sent to the online branch encoder f_{θ_1}, while σ_2 is sent to the target branch encoder f_{θ_2}. After being processed with the aligner module l_η, the outputs are subsequently sent to the respective cross-attention module g_{ϕ_1} and g_{ϕ_2}. Denote the output of the online branch as z_{θ_1} and the output of the target branch as z_{θ_2}. h_ψ denotes the predictor for online branches. We minimize the similarity loss between $h_\psi(z_{\theta_1})$ and z_{θ_2}, which is defined by their mean square error:

$$\begin{aligned}\mathcal{L}_{\sigma_1,\sigma_2} &\triangleq \left\| \frac{h_\psi(z_{\theta_1})}{\|h_\psi(z_{\theta_1})\|_2} - \frac{z_{\theta_2}}{\|z_{\theta_2}\|_2} \right\|_2^2 \\ &= 2 - 2 \cdot \frac{\langle h_\psi(z_{\theta_1}), z_{\theta_2} \rangle}{\left\|h_\psi(z_{\theta_1})\right\|_2 \cdot \left\|z_{\theta_2}\right\|_2}\end{aligned} \tag{3}$$

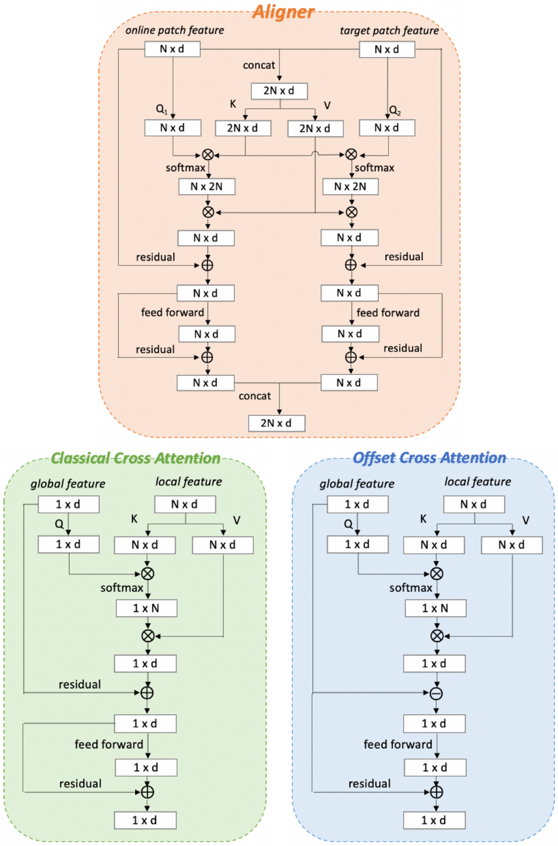

Figure 3. Detailed architecture of the aligner module and two local-to-global cross-attention variants.

During each training step, we swap the 2 shape augmentations for online and target branches to compute the symmetry $\mathcal{L}_{\sigma_2,\sigma_1}$. The total loss is given as:

$$\mathcal{L}_{total} = \mathcal{L}_{\sigma_1,\sigma_2} + \mathcal{L}_{\sigma_2,\sigma_1} \tag{4}$$

4. Experiments

4.1. Pre-training

Dataset. We pre-train our model with the ShapeNet [7] dataset, which contains 57,448 synthetic models from 55 categories. The point clouds are preprocessed following the work of Yang et al. [66] and each contains 2048 points. After inputting into the network, each 3D shape is first augmented with different augmentation operations.

Implementation Details. For each shape augmentation, we sample a global point cloud of 1024 points with FPS and local patches of 256 points (*i.e.*, $K = 256$) of multi-scales (8 patches per scale) with perception-enlarged multi-scale KNN. We use the Adam optimizer with parameter weight decay 1×10^{-4}. The initial learning rate is set as 1×10^{-4}. A cosine annealing schedule is used for the learning rate and the model is trained end-to-end for 100 epochs with a

Category	Method	Backbone	Extra Training Data	Overall Accuracy(%)	
				ModelNet40	ScanObjectNN
Self-supervised (Reconstruction -based)	SO-Net [37]	SO-Net-Encoder	×	87.3	-
	FoldingNet [66]	FoldingNet-Encoder	×	88.4	-
	MRTNet [23]	MRT-Encoder	×	86.4	-
	3D-PointCapsNet [74]	3D Capsule-Encoder	×	88.9	-
	VIP-GAN [25]	EncoderRNN	×	90.2	-
	OcCo [57]	PointNet	×	88.7	69.5
	OcCo [57]	DGCNN	×	89.2	78.3
	Point-BERT [69]	Transformer	×	87.4	-
	Point-MAE [43]	Transformer	×	91.0	77.7
	Point-M2AE [71]	H. Transformer	✓	92.9	84.1
	I2P-MAE [72]	H. Transformer	✓	**93.4**	**87.1**
	ReCon [47]	Transformer	✓	**93.4**	-
Self-supervised (Pretext tasks / Contrastive -based)	Jigsaw [52]	PointNet	×	87.3	55.2
	STRL [29]	PointNet	×	88.3	74.2
	Rotation [44]	PointNet	×	88.6	-
	CrossPoint [2]	PointNet	✓	89.1	**75.6**
	SelfCorrection [12]	PointNet	✓	**89.9**	-
	PoCCA (Ours)	PointNet	×	89.4	**75.6**
	ClusterNet [70]	DGCNN	×	86.8	-
	Multi-Task [26]	DGCNN	×	89.1	-
	Self-Contrast [16]	DGCNN	×	89.6	-
	HSN [17]	DGCNN	×	89.6	-
	Jigsaw [52]	DGCNN	×	90.6	59.5
	STRL [29]	DGCNN	×	90.9	77.9
	Rotation [44]	DGCNN	×	90.8	-
	CrossPoint [2]	DGCNN	✓	91.2	81.7
	PoCCA (Ours)	DGCNN	×	**91.4**	**82.2**
	PoCCA (Ours)	Transformer	×	**92.1**	**83.6**

Table 1. Comparison of linear classification results with previous self-supervised methods on ModelNet40 and ScanObjectNN. A linear classifier is fit onto the shape global representation learned with the pre-trained model (the encoder is frozen). The overall accuracy for the classification task is reported. "H. Transformer" stands for Hierarchical Transformer. PoCCA achieves state-of-the-art results among the methods that do not use extra training data in both backbones.

batch size of 16. We set the exponential moving average parameter $\tau = 0.99$. After pre-training, the target branch encoder, alinger l_η, cross-attention g_ϕ, and predictor h_ψ are all discarded. All downstream tasks are performed on the pre-trained encoder f_θ.

4.2. Justification of Performed Comparison

Although in various fields nowadays, multi-modal approaches often outperform their single-modal counterparts in terms of performance, and this study primarily focuses on research involving single-modal data, we have still included the results of some recent multi-modal methods for comparison. This is because they remain important relevant literature in the field of unsupervised learning. On the other hand, please note that for self-supervised learning frameworks, the key is the framework itself, other than the backbone. That is why MoCo, SimCLR, BYLO, etc. all used ResNet-50 as the backbone for image self-supervised contrastive learning. Using a more advanced backbone would surely improve performance, but they did not for a fair comparison. But for those reconstruction-based self-supervised learning methods, various customized backbones are used since their frameworks include additional generators, especially the recent MAE-based ones. Before these MAE-based methods, in the point cloud contrastive learning subdomain, most papers use PointNet and DGCNN as the backbone for a fair framework comparison. In our experiments, we mainly compare our results with those that use PointNet and DGCNN as the backbone, but meanwhile, the results on a simple Transformer backbone is also reported.

Category	Method	Backbone	Extra Training Data	Overall Accuracy(%)	
				ModelNet40	ScanObjectNN
Supervised	PointNet [45]	-	×	89.2	68.2
	PointNet++ [46]	-	×	90.7	77.9
	PointCNN [38]	-	×	92.2	78.5
	DGCNN [58]	-	×	92.9	78.1
	PCT [24]	-	×	93.2	-
Self-supervised (Reconstruction -based)	OcCo [57]	PointNet	×	90.1	80.0
	OcCo [57]	DGCNN	×	93.0	83.9
	Point-BERT [69]	Transformer	×	92.7	83.1
	Point-MAE [43]	Transformer	×	93.2	85.2
	Point-M2AE [71]	H. Transformer	✓	93.4	-
	I2P-MAE [72]	H. Transformer	✓	**93.7**	90.1
	ReCon [47]	Transformer	✓	**94.1**	**90.6**
Self-supervised (Pretext tasks / Contrastive -based)	Jigsaw [52]	PointNet	×	89.6	76.5
	Info3D [50]	PointNet	×	90.2	-
	SelfCorrection [12]	PointNet	✓	90.0	-
	ParAE [18]	PointNet	✓	**90.5**	-
	PoCCA (Ours)	PointNet	×	90.2	**80.3**
	Jigsaw [52]	DGCNN	×	92.4	82.7
	ParAE [18]	DGCNN	✓	92.9	-
	Info3D [50]	DGCNN	×	93.0	-
	STRL [29]	DGCNN	×	93.1	-
	PoCCA (ours)	DGCNN	×	**93.2**	**84.1**
	PoCCA (ours)	Transformer	×	**93.3**	**84.8**

Table 2. Results of shape classification task network fine-tuned on ModelNet40 and ScanObjectNN. The self-supervised pre-trained backbone encoders serve as the initial weights for supervised downstream tasks.

4.3. Downstream Tasks

Linear SVM Classification. For the classification task, we adopt the protocols of previous work [29] to evaluate the transferability of PoCCA on the ModelNet40 [62] and ScanObjectNN [55] benchmarks. A linear Support Vector Machine (SVM) [21] is used to classify 3D shapes by applying it to the encoded global feature representations. We freeze the pre-trained encoder and fit a simple linear SVM classifier on the train split of ModelNet40 and ScanObjectNN, respectively. Experimental results are presented in Table 1. Note that reconstruction-based methods typically design specific encoder-decoder architectures and do not use common encoders (e.g. PointNet, DGCNN) as the backbone. They also typically incorporate an additional reconstruction loss. As shown in Table 1, in the category of pretext tasks and contrastive-based self-supervised learning method, PoCCA outperforms other state-of-the-art unsupervised methods with both backbones. It even outperforms the reconstruction-based methods that do not use extra training data.

Fine-tuned Classification. The results of transfer learning on the ModelNet40 classification task are reported in Table 2, *i.e.*, the decoder is first initialized with pre-trained weights, then the whole task network is fine-tuned in a supervised manner on the training set. From it, we observe that our method outperforms most other SOTA contrastive-based methods. The t-SNE plots are given in Figure 4 for better visualization of the learned latent representations.

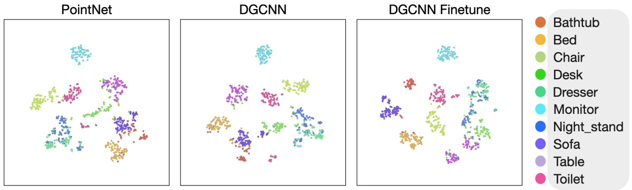

Figure 4. t-SNE visualization of features on the test split of ModelNet10 dataset, with PointNet and DGCNN as the backbone in a self-supervised manner (left and middle), the right one illustrates the latent features learned with DGCNN after fine-tuning.

Method	1%	5%	10%	20%	100%
DGCNN	58.4%	80.7%	85.2%	88.1%	92.9%
STRL + DGCNN	60.5%	82.7%	86.5%	89.7%	93.1%
PoCCA + DGCNN	**65.7%**	**85.8%**	**89.2%**	**91.3%**	**93.2%**

Table 3. Shape classification fine-tuned on ModelNet40 with only partial training samples.

Additional results of fine-tuning on partial training samples are also provided in Table 3 in comparison with STRL [29].

Few-shot Classification. We additionally give the results on the few-shot object classification metric. Few-shot learning (FSL) [22] is a machine learning methodology where models are trained on small datasets, where each category only provides a few instances. We test our models on a standard few-shot task, namely X-way Y-shot learning, where the model is evaluated on X classes, and each class contains Y samples. Like standard 3D object classification. We again use ModelNet40 [62] and ScanObjectNN [55] datasets to carry out FSL experiments. We take 10 random few-shot tasks and report the mean and standard deviation of their results. As presented in Table 4, our FSL results on ModelNet40 show that PoCCA outperforms most prior works for both PointNet and DGCNN backbones. The FSL results for the ScanObjectNN dataset are presented in Table 5. By using PoCCA, the accuracy is also increased significantly in most settings for both feature extractors.

Part Segmentation. The pre-trained model is also used for the 3D part segmentation task on the ShapeNetPart [67] dataset. This dataset contains a total of 16881 3D objects from 16 different categories with 50 annotated semantic parts. We first pre-train our PoCCA framework with the DGCNN or Transformer backbone on the ShapeNet [7] dataset. The model is then fine-tuned on the training set of ShapeNetPart. Both category mIoU and instance mIoU are computed and presented in Table 6. It shows that the backbone pre-trained via PoCCA leads to better part segmentation performance than other self-supervised methods, as well as the randomly initialized DGCNN baseline. Regarding the transformer backbone, PoCCA does not achieve comparable results to methods that use extra training data, yet still outperforms the ones that do not. Overall, PoCCA is a good choice for weight initialization for feature extractors, as proved by the results.

Method	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
3D-GAN [61]	55.8±3.4	65.8±3.1	40.3±2.1	48.4±1.8
FoldingNet [66]	33.4±4.1	35.8±5.8	18.6±1.8	15.4±2.2
Latent-GAN [1]	41.6±5.3	46.2±6.2	32.9±2.9	25.5±3.2
3D-PointCapsNet [74]	42.3±5.5	53.0±5.9	38.0±4.5	27.2±4.7
PointNet++ [46]	38.5±4.4	42.4±4.5	23.1±2.2	18.8±1.7
PointCNN [38]	65.4±2.8	68.6±2.2	46.6±1.5	50.0±2.3
RSCNN [40]	65.4±8.9	68.6±7.0	46.6±4.8	50.0±7.2
PointNet + Rand	52.0±3.8	57.8±4.9	46.6±4.3	35.2±4.8
PointNet + Jigsaw [52]	66.5±2.5	69.2±2.4	56.9±2.5	66.5±1.4
PointNet + cTree [53]	63.2±3.4	68.9±3.0	49.2±1.9	50.1±1.6
PointNet + OcCo [57]	89.7±1.9	92.4±1.6	83.9±1.8	89.7±1.5
PointNet + CrossPoint [2]	90.9±4.8	93.5±4.4	84.6±4.7	90.2±2.2
PointNet + PoCCA (Ours)	**91.7±3.1**	**94.2±3.5**	**87.3±2.9**	**90.9±4.1**
DGCNN + Rand	31.6±2.8	40.8±4.6	19.9±2.1	16.9±1.5
DGCNN + Jigsaw [52]	34.3±1.3	42.2±3.5	26.0±2.4	29.9±2.6
DGCNN + cTree [53]	60.0±2.8	65.7±2.6	48.5±1.8	53.0±1.3
DGCNN + OcCo [57]	90.6±2.8	92.5±1.9	82.9±1.3	86.5±2.2
DGCNN + CrossPoint [2]	92.5±3.0	**94.9±2.1**	83.6±5.3	87.9±4.2
DGCNN + PoCCA (Ours)	**93.5±3.7**	92.1±3.6	**88.1±5.3**	**90.9±4.0**

Table 4. Few-shot object classification results on ModelNet40. We report mean and standard error over 10 runs. Our proposed PoCCA improves the few-shot accuracy in most of the reported settings. Table is extended from [2].

Method	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
PointNet + Rand	57.6±2.5	61.4±2.4	41.3±1.3	43.8±1.9
PointNet + Jigsaw [52]	58.6±1.9	67.6±2.1	53.6±1.7	48.1±1.9
PointNet + cTree [53]	59.6±2.3	61.4±1.4	53.0±1.9	50.9±2.1
PointNet + OcCo [57]	70.4±3.3	72.2±3.0	54.8±1.3	61.8±1.2
PointNet + CrossPoint [2]	68.2±3.3	73.2±2.9	58.7±1.8	64.6±1.2
PointNet + PoCCA (Ours)	**70.5±1.8**	**74.8±3.2**	**60.3±2.3**	**65.2±1.7**
DGCNN + Rand	62.0±5.6	67.8±5.1	37.8±4.3	41.8±2.4
DGCNN + Jigsaw [52]	65.2±3.8	72.2±2.7	45.6±3.1	48.2±2.8
DGCNN + cTree [53]	68.4±3.4	71.6±2.9	42.4±2.7	43.0±3.0
DGCNN + OcCo [57]	72.4±1.4	77.2±1.4	57.0±1.3	61.6±1.2
DGCNN + CrossPoint [2]	74.8±1.5	79.0±1.2	62.9±1.7	73.9±2.2
DGCNN + PoCCA (Ours)	**79.9±4.7**	**83.5±4.2**	**66.0±3.2**	**75.1±2.7**

Table 5. Few-shot object classification results on ScanObjectNN. We report mean and standard error over 10 runs. Our proposed PoCCA improves the few-shot accuracy in all the reported settings. Table is extended from [2].

4.4. Ablation Study

PoCCA variants. Apart from the PoCCA framework used above, its variants of modifying different parts are additionally investigated, including discarding local branches, not aligning local features, using different local-global feature fusion methods, and not using the predictor. Numerical results are given in Table7. From it, we can observe that not using local patches decreases performance significantly. Meanwhile, not merging the patch features

Category	Method	Backbone	Extra Training Data	Cat. mIoU	Ins. mIoU
Supervised	PointNet [45]	-	×	80.4	83.7
	PointNet++ [46]	-	×	81.9	85.1
	DGCNN [58]	-	×	82.3	85.1
	RSCNN [40]	-	×	84.0	86.2
	PCT [24]	-	×	83.1	86.4
Self-supervised (Encoder frozen)	CloudContext [51]	DGCNN	×	-	81.5
	HNS [17]	DGCNN	×	79.9	82.3
	CMCV [31]	DGCNN	×	74.7	80.8
	PoCCA (Ours)	DGCNN	×	**80.8**	**83.7**
Self-supervised (Encoder fine-tuned)	Jigsaw [52]	DGCNN	×	83.1	85.3
	CMCV [31]	DGCNN	×	79.1	83.7
	OcCo [57]	DGCNN	×	84.4	-
	CrossPoint [2]	DGCNN	✓	-	85.5
	PoCCA (Ours)	DGCNN	×	**84.5**	**85.8**
	Point-BERT [69]	Transformer	×	84.1	85.6
	Point-MAE [43]	Transformer	×	-	86.1
	Point-M2AE [71]	H. Transformer	✓	84.9	86.5
	I2P-MAE [72]	H. Transformer	✓	**85.1**	**86.7**
	ReCon [47]	Transformer	✓	84.8	86.4
	PoCCA (Ours)	Transformer	×	84.7	86.1

Table 6. Shape part segmentation results on the ShapeNetPart dataset using DGCNN or Transformer as the backbone. Both category mIoU and instance mIoU are reported. Cases of encoder frozen and encoder fine-tuned are both presented. ”H. Transformer” stands for Hierarchical Transformer.

Sub-branch	Momentum Updated Encoder Branch	Sub-branch Merge	Local-Global Merge	Predictor	Accuracy
✓	Target global	Aligner	Classical CA	✓	**91.4**
✓	Target global	Concat.	Classical CA	✓	91.0
✓	Target global	-	Classical CA	✓	89.7
✓	Target global	Aligner	Offset CA	✓	91.2
✓	Target global	Concat.	Offset CA	✓	90.9
✓	Target global	-	Offset CA	✓	89.5
✓	Target global	Aligner	Concat.	✓	84.2
✓	Target global	Concat.	Concat.	✓	84.3
✓	Target global	-	Concat.	✓	86.1
✓	None	Aligner	Classical CA	✓	90.2
✓	Target both	Aligner	Classical CA	✓	85.8
✓	Target global	Aligner	Classical CA	-	8.3
-	Target	-	-	-	7.7
-	Target	-	-	✓	89.6

Table 7. Ablation study of different modules or settings in PoCCA. ”CA” stands for Cross-Attention.

Patch Sampling Method	Kernel Points for KNN	Test Accuracy
Slice-cut	-	88.7
Cuboid-cut	-	90.3
Sphere-cut	-	90.5
KNN scale 0	random	89.2
KNN scale 1	random	90.1
KNN scale 2	random	90.5
KNN scale 0, 1, 2	random	90.9
KNN scale 0	FPS	89.6
KNN scale 1	FPS	90.5
KNN scale 2	FPS	91.1
KNN scale 0, 1, 2	FPS	**91.4**

Table 8. Numerical results with different patch sampling methods.

from both branches also decreases the performance in most cases. Regarding the local-global feature fusion operation, the cross-attention module outperforms direct concatenation significantly. Direct concatenation even performs worse than not using the patches, indicating the importance of feature fusion. Apart from the classical cross-attention from Transformers, we also have tried the offset attention which was introduced in PCT [24]. Their structures are given in Figure 3. However, while they claimed better performance in supervised learning tasks with offset attention, we observe a performance decrease in our self-supervised learning task. Moreover, using momentum update is helpful to the model, yet should be used wisely. We recommend following the actual backpropagated gradients. Last but not least, same as in BYOL, when the predictor is not used, the pipeline collapses to a minimal solution and thus does not work anymore.

Patch Sampling Methods. Ablation experiments are carried out for the comparison of different patch sampling methods. All experiments are conducted with the DGCNN backbone. From Table 8, we can observe that cuboid-cut and sphere-cut sampling methods achieve similar performances, which is quite reasonable since they produce similar patches. Meanwhile, the KNN-based sampling methods outperform shape-cut-based sampling methods. The additional results of using different settings of the KNN-based sampling methods show that random kernel point selection achieves lower performance, especially when the perceptual field is small. This is because patches that are not around the point cloud contours usually do not contain too much information, let alone it is hard to guarantee a good coverage of shapes with these patches. On the other hand, FPS-based kernel point selection assures better patch acquisition. When multi-scale perception KNN is used to create multi-scale patches, our method achieves the best performance.

5. Conclusion

In this paper, we propose an effective unsupervised framework PoCCA for point cloud representation learning. Compared to common contrastive learning frameworks, PoCCA enables information exchange between the online branch and the target branch by leveraging the local and global features of different sub-branches. We have evaluated our approach on point cloud classification and segmentation benchmarks, and the experimental results show that it achieves state-of-the-art performance between the point cloud contrastive learning methods that do not use extra training data. We have also evaluated the influence of different components of PoCCA through ablation studies. For future work, it would be interesting to investigate point-wise contrastive frameworks. New losses could be designed for better model pre-training. Moreover, better ways to exploit the attention mechanism could be explored.

Acknowledgements

This research is funded by the Carl-Zeiss Foundation.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *Proceedings of the 35th International Conference on Machine Learning*, pages 40–49, 2018. 7

[2] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9892–9902, 2022. 1, 2, 6, 7, 8

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *ArXiv*, abs/2106.08254, 2022. 1

[4] Prarthana Bhattacharyya, Chengjie Huang, and K. Czarnecki. Sa-det3d: Self-attention based context-aware 3d object detection. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3022–3031, 2021. 3

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2019. 1

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *ArXiv*, abs/2006.09882, 2020. 2

[7] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv:1512.03012*, 2015. 5, 7

[8] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 347–356, 2021. 4

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020. 1, 2

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 1, 2, 4, 5

[11] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016. 1

[12] Ye Chen, Jinxian Liu, Bingbing Ni, Hang Wang, Jiancheng Yang, Ning Liu, Teng Li, and Qi Tian. Shape self-correction for unsupervised point cloud understanding. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8362–8371, 2021. 6

[13] Zhang Cheng, Haocheng Wan, Xinyi Shen, and Zizhao Wu. Patchformer: A versatile 3d transformer based on patch attention. *ArXiv*, abs/2111.00207, 2021. 3

[14] Casey Chu, Andrey Zhmoginov, and Mark Sandler. Cyclegan, a master of steganography. *ArXiv*, abs/1712.02950, 2017. 2

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021. 3

[16] Bi'an Du, Xiang Gao, Wei Hu, and Xin Li. Self-contrastive learning with hard negative sampling for self-supervised point cloud learning. In *ACM Multimedia Conference*, pages 3133–3142, 2021. 6

[17] Bianli Du, Xiang Gao, Wei Hu, and Xin Li. Self-contrastive learning with hard negative sampling for self-supervised point cloud learning. *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. 1, 6, 8

[18] Benjamin Eckart, Wentao Yuan, Chao Liu, and Jan Kautz. Self-supervised learning on 3d point clouds by learning discrete generative models. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8244–8253, 2021. 6

[19] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y. Zeevi. The farthest point strategy for progressive image sampling. *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 2 - Conference B: Computer Vision & Image Processing. (Cat. No.94CH3440-5)*, pages 93–97 vol.3, 1994. 4

[20] Nico Engel, Vasileios Belagiannis, and Klaus C. J. Dietmayer. Point transformer. *IEEE Access*, 9:134826–134840, 2021. 3

[21] Theodoros Evgeniou and Massimiliano Pontil. Support vector machines: Theory and applications. In *Machine Learning and Its Applications*, 2001. 6

[22] Michael Fink. Object classification from a single example utilizing class relevance metrics. In *NIPS*, 2004. 7

[23] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3D point cloud processing. In *ECCV*, 2018. 6

[24] Meng-Hao Guo, Junxiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph Robert Martin, and Shimin Hu. Pct: Point cloud transformer. *Comput. Vis. Media*, 7:187–199, 2021. 3, 6, 8

[25] Zhizhong Han, Mingyang Shang, Yu-Shen Liu, and Matthias Zwicker. View inter-prediction gan: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:8376–8384, 2019. 6

[26] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 6

[27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. 1, 2

[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B. Girshick. Masked autoencoders

are scalable vision learners. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2022. 1

[29] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6515–6525, 2021. 1, 2, 6, 7

[30] Le Hui, Hang Yang, Mingmei Cheng, Jin Xie, and Jian Yang. Pyramid point cloud transformer for large-scale place recognition. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6078–6087, 2021. 3

[31] Longlong Jing, Yucheng Chen, Ling Zhang, Mingyi He, and Yingli Tian. Self-supervised feature learning by cross-modality and cross-view correspondences. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1581–1891, 2020. 8

[32] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ArXiv*, abs/1710.10196, 2018. 1

[33] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019. 2

[34] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8490–8499, 2022. 4

[35] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 8490–8499, 2022. 3

[36] G. Li, Matthias Müller, Ali K. Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9266–9275, 2019. 4

[37] Jiaxin Li, Ben M. Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9397–9406, 2018. 6

[38] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, 2018. 6, 7

[39] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *European Conference on Computer Vision*, 2022. 2

[40] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8895–8904, 2019. 7, 8

[41] Dening Lu, Kyle Gao, Qian Xie, Linlin Xu, and Jonathan Li. 3dpct: 3d point cloud transformer with dual self-attention. *ArXiv*, abs/2209.11255, 2022. 3

[42] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7459–7468, 2021. 3

[43] Yatian Pang, Wenxiao Wang, Francis E. H. Tay, W. Liu, Yonghong Tian, and Liuliang Yuan. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision*, 2022. 2, 6, 8

[44] Omid Poursaeed, Tianxing Jiang, Quintessa Qiao, Nayun Xu, and Vladimir G. Kim. Self-supervised learning of point clouds via orientation estimation. *2020 International Conference on 3D Vision (3DV)*, pages 1018–1028, 2020. 2, 6

[45] C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 4, 6, 8

[46] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, 2017. 4, 6, 7, 8

[47] Zekun Qi, Runpei Dong, Guo Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. *ArXiv*, abs/2302.02318, 2023. 1, 3, 6, 8

[48] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016. 1

[49] Pierre H Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, et al. Byol works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020. 1, 2, 4, 5

[50] Aditya Sanghi. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. *ArXiv*, abs/2006.02598, 2020. 1, 6

[51] Jonathan Sauder and Bjarne Sievers. Context prediction for unsupervised deep learning on point clouds. *ArXiv*, abs/1901.08396, 2019. 8

[52] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 32:12962–12972, 2019. 2, 6, 7, 8

[53] Charu Sharma and Manohar Kaul. Self-supervised few-shot learning on point clouds. In *Advances in Neural Information Processing Systems*, pages 7212–7221, 2020. 7

[54] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 2

[55] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1588–1597, 2019. 6, 7

[56] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017. 3

[57] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matthew J. Kusner. Unsupervised point cloud pre-training via occlusion completion. In *International Conference on Computer Vision, ICCV*, 2021. 2, 6, 7, 8

[58] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38:1 – 12, 2019. 4, 6, 8

[59] Xin Wei, Xiang Gu, and Jian Sun. Learning generalizable part-based feature representation for 3d point clouds. In *Neural Information Processing Systems*, 2022. 2

[60] Chengzhi Wu, Junwei Zheng, Julius Pfrommer, and Jürgen Beyerer. Attention-based point cloud edge sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5333–5343, 2023. 3

[61] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, 2016. 7

[62] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015. 6, 7

[63] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 2

[64] Saining Xie, Jiatao Gu, Demi Guo, C. Qi, Leonidas J. Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. *ArXiv*, abs/2007.10985, 2020. 1, 2

[65] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: a simple framework for masked image modeling. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9643–9653, 2022. 1

[66] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 206–215, 2018. 5, 6, 7

[67] L. Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas J. Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (TOG)*, 35:1 – 12, 2016. 7

[68] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12478–12487, 2021. 3

[69] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19291–19300, 2021. 2, 6, 8

[70] Ling Zhang and Zhigang Zhu. Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks. In *International Conference on 3D Vision (3DV)*, pages 395–404, 2019. 6

[71] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bingyan Zhao, Dong Lei Wang, Yu Jiao Qiao, and Hongsheng Li. Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training. *ArXiv*, abs/2205.14401, 2022. 3, 6, 8

[72] Renrui Zhang, Liuhui Wang, Yu Jiao Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. *ArXiv*, abs/2212.06785, 2022. 1, 3, 6, 8

[73] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16239–16248, 2021. 3

[74] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3D point capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6, 7

[75] Hao Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. *ArXiv*, abs/2207.10315, 2022. 3