PointCMC: Cross-Modal Multi-Scale Correspondences Learning for Point Cloud Understanding

Honggu Zhou  
School of Media and Design  
Hangzhou Dianzi University, China  
hongguzhou@hdu.edu.cn

Xiaogang Peng  
School of Media and Design  
Hangzhou Dianzi University, China  
pengxiaogang@hdu.edu.cn

Jiawei Mao  
School of Media and Design  
Hangzhou Dianzi University, China  
jiaweima0@hdu.edu.cn

Zizhao Wu*  
School of Media and Design  
Hangzhou Dianzi University, China  
wuzizhao@hdu.edu.cn

Ming Zeng  
School of Informatics  
Xiamen University, Xiamen, China  
zengming@xmu.edu.cn

Abstract

Some self-supervised cross-modal learning approaches have recently demonstrated the potential of image signals for enhancing point cloud representation. However, it remains a question on how to directly model cross-modal local and global correspondences in a self-supervised fashion. To solve it, we proposed PointCMC, a novel cross-modal method to model multi-scale correspondences across modalities for self-supervised point cloud representation learning. In particular, PointCMC is composed of: (1) a local-to-local (L2L) module that learns local correspondences through optimized cross-modal local geometric features, (2) a local-to-global (L2G) module that aims to learn the correspondences between local and global features across modalities via local-global discrimination, and (3) a global-to-global (G2G) module, which leverages auxiliary global contrastive loss between the point cloud and image to learn high-level semantic correspondences. Extensive experiment results show that our approach outperforms existing state-of-the-art methods in various downstream tasks such as 3D object classification and segmentation. Code will be made publicly available upon acceptance.

Keywords: Self-Supervised Representation Learning, Contrastive Learning, Cross-Modal Learning, Point Cloud Understanding

*Corresponding author

1. Introduction

Helping machines understand the 3D world is crucial in many real-world applications, such as autonomous driving, AR, VR, and other fields. This has given rise to intermediate forms, such as point clouds, meshes, vox-
We specify this question into three aspects: (1) local multi-scale correspondence between 2D and 3D modalities called PointCMC, which aims to effectively establish the representations? 

Additionally, local correspondences can relate cross-modal local geometries and high-level semantics in the reconstruction process and a lack of local correspondence approach may result in a loss of 2D semantics, i.e., by maximizing agreement between corresponding global representations across modalities. Other levels, such as Point-Pixel level correspondences, are investigated in an indirect way [31], i.e., by reconstructing the 2D images into point clouds to perform point cloud correspondence. However, the global correspondence methods sacrifice a large number of local representations because their backbones obtain global representations based on the consumption of points and pixels. In addition, the point-pixel correspondence approach may result in a loss of 2D semantics in the reconstruction process and a lack of local geometric perception. We note that local geometric features and high-level semantics are crucial in learning point cloud and image representations. Modelling cross-modal local correspondences can relate cross-modal local geometric features while enforcing the correspondences of cross-modal global features facilitates high-level shared semantics learning. However, multi-scale correspondences have not been modelled by the above methods. In this paper, we propose a question: How to directly model cross-modal local and global correspondence to learn point cloud representations?

To answer this question, we propose a novel method called PointCMC, which aims to effectively establish the multi-scale correspondence between 2D and 3D modalities. We specify this question into three aspects: (1) local correspondence, (2) local-to-global correspondence, and (3) global correspondence. Firstly, we need to establish local correspondence between point clouds and images. To this end, we exploit the L2L module with a two-branch local attention block to optimize cross-modal local features from corresponding abstraction hierarchies. We were then maximizing local similarity on mismatched local features and minimizing local similarity on mismatched local features. Secondly, to model local-to-global correspondence, we utilize the L2G module to capture the shared properties between local parts and global shapes in a self-supervised way. The L2G enables the local features of one object to be closer to its global features. Hence, the unique semantic information of each object can be learned through local representations. Thirdly, to model global correspondence, we utilize the G2G module to strengthen the association of global representations between different modalities by embedding global representations of the same object to be closer.

We have validated the effectiveness of our method with a series of downstream tasks on several generic point cloud backbone networks. Furthermore, a sequence of ablation experiments has validated the contribution of the proposed three modules to our method.

The main contributions of our approach are as follows:

- We propose a novel self-supervised method for point cloud representation learning by contrasting the multi-scale representations, i.e., local representations between different modalities, local and global representations between different modalities, and global representations between different modalities.
- We propose a novel L2L module that does not require annotating the local correspondence between point clouds and images but instead performs local similarity learning after optimizing local representations with two-branch local attention blocks. In addition, we also propose the L2G module to learn the cross-modal global-local correspondence in a self-supervised way.
- We evaluate our method in three downstream tasks: (1) 3D object classification, (2) few-shot object classification, and (3) 3D Object part segmentation. Quantitative and qualitative results on real-world and synthetic benchmarks have shown our methods can learn discriminative representations and outperform previous self-supervised learning methods.

2. Related Work

2.1. Supervised methods for point cloud

Point cloud understanding is still challenging compared to NLP and 2D. On the one hand, point clouds lack a highly regular structure: there are images for 2D and word embeddings for NLP. On the other hand, point clouds also need to be permutation invariance when processing. Unlike 2D
images, which have a unified architecture like convolutional neural networks, there are many network structures to handle point clouds from different perspectives. Point-based networks [35][37] directly consume raw point clouds, and the pioneering work is Pointnet [35], which proposes an architecture that stacks MLP layers to extract point-wise features independently, then aggregates them by max pooling. However, PointNet fails to capture local information. To address this issue, Qi et al. [37] then proposes PointNet++ to learn global and local information through the hierarchical aggregation of neighborhood points. Graph-based networks model the relationship between points as a graph. Wang et al. [51] proposes pioneering work dubbed DGCNN, which uses a graph convolution named EdgeConv to capture local features for each point from its k nearest neighbor points. Voxel-based networks [36][23] voxelize irregular point clouds into regular 3D grids and use 3D convolution for feature learning. Spatial CNN-based networks [19][30] directly apply Spatial-convolutions on irregular point clouds. RSCNN [30] is proposed to explicitly encode the geometric relation of points and achieve contextual shape-aware learning of point clouds.

2.2. Self-supervised methods for point cloud

Motivated by the success of self-supervised representation learning [6][67][3][5][15][40][11] in the field of 2D, some recent works have investigated self-supervised methods for 3D representation learning, which can be roughly divided into two categories.

**Classical learning methods.** Classical methods learn 3D representation by performing generative tasks mainly based on autoencoder [22] and generative adversarial networks (GANs) [10]. Autoencoder networks aim to train the encoder to learn the representation of the input point cloud, while the decoder reconstructs the point cloud from the learned representation. The shape information is captured during the training process, and based on this idea, many self-supervised learning methods are based on autoencoder networks. Self-reconstruction [63][8][25][69][13] is a classical self-supervised learning task based on autoencoder networks, where the encoder is trained by optimizing the reconstruction results of the learned representations. Furthermore, autoencoder-based point cloud upsampling [26][65][27] and completion [49][18][42][58] put more emphasis on learning more complete representations, as finer local structures of the point cloud need to be predicted. In addition to autoencoder networks, GANs learn representations in an unsupervised manner by adversarially training generators and discriminators. Recent GANs-based methods [1][45][24] have also shown great potential for representation extraction in voxels and point clouds.

**Contrastive learning methods.** Contrastive learning uses predefined positive and negative samples to maximise agreements between positive pairs and minimise agreements between positive and negative samples. Xie et al. [59] proposes to learn representations by performing point-level discriminations via scene point clouds data from two viewpoints. Zhang et al. [66] proposes a method based on deep graph convolutional neural networks performed part contrast, which could verify whether two randomly sampled parts belong to the same object. Du et al. [7] proposes a method based on the nonlocal self-similarity of point clouds to learn representation. Rao et al. [39] combines contrastive learning and self-reconstruction to formulate a task that focuses on global and local representations reasoning. Huang et al. [17] introduces BYOL [11] into the point cloud and extracts spatial and temporal representation from point clouds. Different from the prior works, we attempt to exploit contrastive learning in modelling 2D-3D correspondence and further explore the multi-scale correspondences across modalities.

2.3. Cross-modal learning for point cloud understanding

Cross-modality (e.g., image and natural language) [70][38][2][55] has been shown to provide additional training signals, such as correspondences, to help to learn representation. Some recent works have shown the success of cross-modal learning between 2D and natural language. However, as pointed out in [57], there are relatively few methods for learning 3D representations using 2D images as auxiliary inputs. Xu et al. [60] directly transfers 2D convolutional models to the point cloud model by inflating the 2D convolution or by direct parameter replication. Liu et al. [31] proposes a cross-modal knowledge distillation, which maps 2D images into point clouds in a reconstruction manner to enforce point correspondences. However, this mapping approach may result in the loss of 2D semantics. Jing et al. [20] learns 3D point cloud representations by predicting the correspondence between cross-modal global representations. Afham et al. [2] introduces auxiliary cross-modal instance discrimination to enforce 3D-2D correspondences while preserving the model being invariant to affine and spatial transformations by inner-modal instance discrimination. Our method is inspired by the work [2], but we note that the hierarchical encoder structures of images and point clouds capture global features by consuming points and pixels, sacrificing a significant amount of local information. To reduce the loss of local signals, we directly model the multi-scale correspondences across modalities.

3. Method

The proposed PointCMC method aims to directly model cross-modal local and global correspondences in a self-supervised way. The key components of PointCMC are comprised of the L2L module, the L2G module, and the G2G module. In this section, we detail the method overview
and the three key components. The overall framework is illustrated in Figure 2.

3.1. Method Overview

The input can be represented as \{P_i, I_i\}, with \(P_i \in \mathbb{R}^{N \times 3}\) and \(I_i \in \mathbb{R}^{H \times W \times 3}\), where \(I_i\) is a random-view rendered image from point cloud \(P_i\). Our architecture consists of two Encoders \{\(E^P\), \(E^{img}\)\} from different modalities, two L2L modules, one L2G module, and one G2G module. We use \(E^P\) to extract multi-scale point cloud features from augmented \(P_i\), and use \(E^{img}\) to extract multi-scale features from augmented \(I_i\). Based on these extracted features, firstly, we model the correspondences of cross-modal local features by the L2L modules, which adopt the attention mechanism to consider the correspondences across different modalities. Secondly, we model the cross-modal local-global relationships with the L2G module by considering shared semantic properties between cross-modal local and global features. Lastly, we use the general G2G module to enforce global features of images as a centroid to global features of point clouds to learn global correspondences.

3.2. Local-to-Local Module

In this subsection, we introduce how to utilize the local attention blocks to model the correspondence for image patches and point cloud patches through the L2L module. To be specific, the process of the L2L module is to feed the local representations captured by the two backbones into the local attention block to obtain their optimized versions, and then local discriminations will be performed on these optimized versions.

**Local attention block.** The local attention block is based on the attention mechanism, which requires three specified inputs Q(query), K(key), and V(value). The affinity matrix is obtained by calculating the correlation between Q and K. Then, the affinity matrix is weighted and summed with the corresponding V to obtain the final output. When the queries, keys, and values are all from the same object, it is called the self-attention mechanism. Following by [48][56], we utilize the Transformer blocks to implement the attention modules, as shown in Figure 3. The multi-head attention layer consists of \(h\) self-attention blocks. First, pass the inputs to each of the \(h\) different Self-Attention and compute the \(h\) output matrices, then contact outputs together and pass into a Linear layer to obtain the final output of Multi-Head Attention. To be specific, let \(X = \{x_1, x_2, ..., x_k\}\) represents inputs of local attention block, where \(x_i \in \mathbb{R}^{L \times d_x}\) and \(X \in \mathbb{R}^{K \times d_x}\), we firstly calculate the query, key and value for the input: \(Q^X_i = X \cdot W^Q_i\), \(K^X_i = X \cdot W^K_i\), \(V^X_i = X \cdot W^V_i\), where \(W^Q_i \in \mathbb{R}^{d_x \times d_k}\), \(W^K_i \in \mathbb{R}^{d_x \times d_k}\), \(W^V_i \in \mathbb{R}^{d_x \times d_v}\), \(i \in h\). Then we get the affinity matrix through \(A(Q^X_i, K^X_i) = Q^X_i \cdot (K^X_i)^T\). Furthermore, the output of Self-Attention blocks can be formulated as follows:

\[
SA(Q^X_i, K^X_i, V^X_i) = \text{softmax}\left(\frac{A(Q^X_i, K^X_i)}{\sqrt{d_k}}\right) \cdot V^X_i, \tag{1}
\]

after that, we compute all values in all the Self-Attention blocks and concatenate them together to get the output of the multi-head attention layer:

\[
MA(X) = \text{concat}(SA_1, ..., SA_h) \cdot W^O, \tag{2}
\]

where \(W^O \in \mathbb{R}^{h \times d_x \times d_k}\). In order to prevent the gradient from disappearing and preserve its stability, the output of the multi-head attention layer is also connected by residual connections with layer normalizations, and it can be described as:

\[
FFN(MA) = \text{ReLU}(MA \cdot W_1 + b_1) \cdot W_2 + b_2, \tag{3}
\]

where \(W_1 \in \mathbb{R}^{1 \times d_{MA}}\), \(W_2 \in \mathbb{R}^{d_{MA} \times d_{MA}}\), \(b_1 \in \mathbb{R}^{1 \times d_{MA}}\) and \(b_2 \in \mathbb{R}^{d_{MA} \times d_{MA}}\). In our works, local attention blocks are embedded in the corresponding abstract layers of the point cloud and image encoder to optimize local representations.

**Cross-modal local correspondences.** With the above local attention blocks, every image patch or point cloud patch can pay more attention to representations of interesting areas in the same modality so that we can obtain more fine-grained optimized local representations. In our work, optimized local representations are \(\{f^{img}_l, f^p_l\}\), where \(l\) is the number of hierarchies. We cannot directly measure the similarity of cross-modal local features, etc., due to the varying number of channels of local features across different modalities. Informed by the work of [2][39], we embed them separately into the same shared feature space using prediction networks that can be implemented using multilayer perceptions (MLP) projection heads \(\phi^{img}_l(\cdot)\) and \(\phi^p_l(\cdot)\). New local representations are formulated as:

\[
F^{img}_l = \phi^{img}_l(f^{img}_l), \quad F^p_l = \phi^p_l(f^p_l), \tag{4}
\]

where \(l\) is the number of hierarchies.

Our goal is to enforce \(F^{img}_l\) to be closer to \(F^p_l\) of the same object than other objects. Inspired by instance discrimination, we design local discrimination, which maximizes the similarity between \(F^{img}_l\) and \(F^p_l\), minimizing the similarity between others. Specifically, we propose Local-to-Local correspondence loss \(L_{l2l}\), which can be viewed as a variant of InfoNCE [46]. Our Local-to-Local correspondence loss function \(L_{l2l}\) can be described as:

\[
f(a, b) = \exp(\text{sim}(a, b)/\tau), \tag{4}
\]
Figure 2: PointCMC: Cross-Modal Multi-Scale Correspondences Learning for Point Cloud Understanding. PointCMC works by embedding three types of extra modules between 3D and 2D backbones, including: (1) Local-to-Local (L2L) module, (2) Local-to-Global (L2G) module, and (3) Global-to-Global (G2G) module. L2L module models the correspondences between multi-scale cross-modal local features by local discrimination. L2G module models the correspondence between multi-scale cross-modal local and global features by local-global discrimination. G2G module models the correspondence between cross-modal global features by general instance discrimination. Once trained, these modules will be discarded, and the point cloud encoders will be retained for downstream tasks.

\[
c(x_i, g_i) = \frac{\mathcal{f}(x_i, g_i)}{\sum_{k=1, k \neq i}^{N} \mathcal{f}(x_i, x_k) + \sum_{k=1}^{N} \mathcal{f}(x_i, g_k)},
\]

(5)

\[
L_{l2l} = -\frac{1}{2N} \sum_{l=1}^{M} \sum_{i=1}^{N} \log(c(F_{l}^{img, i}, F_{l}^{p, i}) + c(F_{l}^{p}, F_{l}^{img, i})),
\]

(6)

where \(N, \tau\) and \(M\) are the numbers of local representations, temperature coefficient and the numbers of hierarchies. We choose cosine similarity as \(\text{sim}(\cdot)\).

3.3. Local-to-Global Module

In this subsection, we introduce how to model local-to-global correspondence across modalities, which is not explored in recent works. As [39] mentioned, a strong semantic connection is between local and global features of the 3D point cloud. We exploit this insight into cross-modal learning through the L2G module. Furthermore, the effectiveness of the L2G module has been demonstrated in ablation experiments.

To be specific, we explore shared cross-modal semantic knowledge by local-global discrimination. Hierarchical
representations of point-cloud modality are shaped as \( f^p \) as the above. The global representations captured from two encoders are \( \{g^p, g^{img}\} \). Since local and global representations represent the same object at different hierarchies in different modalities, we fuse the cross-modal global representations, and then we map the local and fused representations to a shared feature space by prediction functions to present, the same object in the shared space as a positive sample. Our approach encourages learning from these harder positive and negative samples as [2], which can enhance the capability of representation learning. In this way, our cross-modal local-to-global correspondence loss function \( L_{l2g} \) can be expressed as follows:

$$L_{l2g} = -\frac{1}{2N} \sum_{i=1}^{N} \sum_{l=1}^{M} \log \left( c \left( g, f^p_l, i \right) \right),$$  \hspace{1cm} (7)

where \( c \left( \cdot \right) \) is same as Eq.5 and \( N, M \) refers to the same parameters in Eq.6.

### 3.4. Global-to-Global Module

In addition to the multi-scale correspondence learning in the above sections, we introduce a general auxiliary contrastive objective across modalities to enforce high-level semantic correspondence. Specifically, the Global representations \( \{g^p, g^{img}\} \) of point-cloud and image modalities are the same as above, then we set the cross-modal global representation of the same object in the shared space as a positive sample. Our approach encourages learning from these harder positive and negative samples as [2], which can enhance the capability of representation learning. In this way, our cross-modal global-to-global correspondence loss function \( L_{g2g} \) can be expressed as follows:

$$L_{g2g} = -\frac{1}{2N} \sum_{i=1}^{N} \log \left( c \left( g^{img}, g^p, i \right) \right),$$  \hspace{1cm} (8)

where \( c \left( \cdot \right) \) and \( N \) are same as Eq.5.

### 3.5. Overall Objective

Finally, our final joint learning objective combines the losses of the above three sections to achieve multi-scale 2D-3D correspondence. Hyperparameter \( \alpha \) is to regulate the balance between the losses.

$$L_{total} = \alpha \cdot L_{l2l} + L_{l2g} + L_{g2g}.$$  \hspace{1cm} (9)

### 4. Experiments

To evaluate the effectivity of the point cloud representations learned with our model, we set three widely used downstream tasks in point cloud representation learning and validate the usefulness of the module. We also design a series of ablation experiments. In the following, we first introduce our pre-training in section 4.1. We then demonstrate the effectiveness of our model via the results of these downstream tasks in section 4.2. Finally, in section 4.3, we show the results of our ablation experiments.

#### 4.1. Pre-Training

**Dataset.** In the pre-training stage, we use ShapeNet [4] as the point cloud dataset, a 3D dataset containing 55 classes with more than 50,000 CAD models. The image dataset [61] contains 43,783 images of 13 classes rendered from ShapeNet at random angles. To keep the input in the form of point cloud-image pairs, we use 13 classes of ShapeNet. For the input point cloud, we sample 2048 points; for the corresponding rendered images, we resize them to 224*224. In addition, we perform data augmentation on these input data. We use transformations such as scaling, translation, and rotation for point cloud data. We use transformations such as random crop, color jittering, and horizontal flips for image data.

**Implementation Details.** Our proposed PointCMC network is implemented through the PyTorch [33] framework and uses one GTX3090 GPU. For a fair comparison with existing methods, we select DGCNN[51] and RSCNN [30] methods, PointCMC has made progress on both DGCNN and RSCNN backbones.

| Method           | ModelNet40 |
|------------------|------------|
| 3D-GAN [52]      | 83.3       |
| Latent-GAN [1]   | 85.7       |
| SO-Net [25]      | 87.3       |
| FoldingNet [63]  | 88.4       |
| MRTNet [9]       | 86.4       |
| 3D-PointCapsNet [69] | 88.9       |
| DepthContrast [68] | 85.4       |
| ClusterNet [66]  | 86.8       |
| VIP-GAN [12]     | 90.2       |
| DGCNN + Multi-Task [14] | 89.1       |
| DGCNN + Self-Contrast [7] | 89.6       |
| DGCNN + Jigsaw [41] | 90.6       |
| DGCNN + STRL [17] | 90.9       |
| DGCNN + Rotation [34] | 90.8       |
| DGCNN + OcCo [49] | 89.2       |
| DGCNN + CrossPoint [2] | 91.2       |
| DGCNN + PointCMC | 91.7       |
| RSCNN + GLR [39] | 89.5       |
| RSCNN + CrossPoint [2] | 91.1       |
| RSCNN + PointCMC | 91.5       |

### 4.2. Downstream Tasks

#### 4.2.1. Table 1: Linear classification results on the ModelNet40 dataset. After training the model, we fit a linear SVM classifier onto the ModelNet40 test dataset to report the overall accuracy(%). Compared to previous self-supervised methods, PointCMC has made progress on both DGCNN and RSCNN backbones.
as the encoders of the point clouds and ResNet50 [16] as the encoder of the images. The local representation of the point cloud extracted from an intermediate layer is shaped as \( \{B, C, N\} \), where \( B \) represents the batch size, \( C \) represents the feature dimension, and \( N \) represents the part of the point cloud; the local representation of the image is shaped as \( \{B, C, H, W\} \), where \( B, C \) have the same meaning as above, and \( H, W \) represent the number of patches per row and column respectively, so we fuse the \( H, W \) dimensions to get \( N_2 \), which represents the number of patches. We use different 2-layer MLP as prediction networks to map local representations and global representations to the same invariant space. We use the Adam optimizer [21] with a decay rate of \( 1 \times 10^{-4} \), a learning rate of \( 1 \times 10^{-3} \), and the hyperparameter \( \alpha \) is set to 0.1. At the same time, we also apply the Cosine annealings [32] as the learning rate scheduler, and the model is trained for 100 epochs. After the pretraining, all our downstream tasks are performed based on the point cloud encoder \( E^p \).

### 4.2. Downstream Tasks

We extensively evaluate our method in three downstream tasks using the pre-trained point cloud encoders. The downstream tasks are: (1) 3D object classification, (2) Few-shot object classification, and (3) 3D object part segmentation. Performance is evaluated by two metrics: overall accuracy (OA) and mean intersection over union (Mean IoU).

**Transfer to 3D object classification.** We compare classification overall accuracy (OA) on the ModelNet40 [54] (synthetic) and ScanObjectnn [44] (real-world) benchmarks with state-of-the-art methods. The ModelNet40 dataset is a dataset containing 40 classes with 12331 3D CAD models, of which 9843 are training sets and 2468 are test sets; the ScanObjectNN dataset is a real 3D object dataset, which is extracted from real-world indoor scene scans, and it contains a total of 2880 objects in 15 classes, of which 2304 are the training set, and 576 are the test set. We sample 1024 points per object for training and test classification. Classification is performed on DGCNN and RSCNN backbones, where DGCNN is built on Edge-Convolution networks, and RSCNN consists of Relation-Shape Convolution. We follow the same setup as in [2][49], i.e., freeze the point cloud encoder and fit a simple SVM classifier on the downstream task training sets.

Table 1 shows the results of our method for classification on ModelNet40 dataset, and Table 2 shows the results of our method for classification on ScanObjectNN dataset. Our method achieves state-of-the-art classification accuracy on both point cloud frameworks. Although our method achieves great results in self-supervised contrastive learning, some methods cannot be fairly compared with us due to factors such as pre-training methods, selection of backbone networks, etc.

| Method | ScanObjectNN |
|--------|--------------|
| DGCNN + Jigsaw [41] | 70.5 |
| DGCNN + OcCo [49] | 78.3 |
| DGCNN + STRL [17] | 77.9 |
| DGCNN + CrossPoint [2] | 81.7 |
| DGCNN + PointCMC | **84.4** |
| RSCNN + GLR [39] | 80.3 |
| RSCNN + CrossPoint [2] | 84.6 |
| RSCNN + PointCMC | **85.1** |

**Transfer to Few-shot object classification.** We conduct the few-shot object classification experiments on the ModelNet40 dataset and ScanObjectNN dataset to validate the performance of our method with limited fine-tuning data. In the few-shot classification (N-way K-shot learning), \( N \) represents the number of classes to be evaluated, and \( K \) represents the number of samples in each class. We follow the settings in [2][49] and compare four settings. Table 3 shows the few-shot classification results on the ModelNet40 test dataset, where our method outperforms the previous methods in three of the four settings for both DGCNN and RSCNN backbones. Table 4 shows the few-shot classification results on the ScanObjectNN test dataset, where our method outperforms the previous methods in three of the two settings for both DGCNN and RSCNN backbones. Our method does not reach the first in some settings. However, it only exposes the small margin between our method and the state-of-the-art method, indicating that our method can still learn to produce general point cloud representations.

| Method | 5-way | 10-way |
|--------|-------|--------|
| DGCNN + Rand | 70.6 \pm 2.4 | 40.8 \pm 4.6 |
| DGCNN + Jigsaw [41] | 34.3 ± 1.3 | 42.3 ± 2.5 |
| DGCNN + STRL [17] | 60.0 ± 6.8 | 67.7 ± 2.6 |
| DGCNN + OcCo [49] | 90.6 ± 2.8 | 92.5 ± 1.9 |
| DGCNN + CrossPoint [2] | **92.5 ± 3.0** | 94.9 ± 2.1 |
| DGCNN + PointCMC | 92.2 ± 2.0 | 95.5 ± 3.3 |
| RSCNN + Rand | 40.2 ± 2.9 | 49.8 ± 3.2 |
| RSCNN + GLR [39] | 91.5 ± 4.9 | 94.7 ± 3.6 |
| RSCNN + CrossPoint [2] | **93.9 ± 4.2** | 95.6 ± 4.0 |
| RSCNN + PointCMC | 93.5 ± 5.1 | **95.8 ± 3.4** |

**Transfer to 3D Object part segmentation.** We evaluate...
Table 4: Few-shot classification results in ScanObjectNN dataset. We report the mean accuracy(%) and standard deviation(%) of 10 independent experiments.

| Method                  | 5 way          |                  | 10 way          |                  |
|-------------------------|----------------|------------------|------------------|------------------|
|                         | 10 shot        | 20 shot          | 10 shot          | 20 shot          |
| DGCNN + Rand            | 62.0±5.6       | 72.2±8.3        | 67.8±5.1        | 77.2±8.3        |
| DGCNN + Jigsaw [41]     | 65.2±3.8       | 77.2±8.1        | 72.2±3.7        | 82.2±8.2        |
| DGCNN + cTree [43]      | 68.4±3.4       | 77.2±8.1        | 72.4±2.7        | 83.2±3.0        |
| DGCNN + OcCo [49]       | 72.4±1.4       | 77.2±8.1        | 72.4±1.3        | 81.6±1.2        |
| DGCNN + CrossPoint [5]  | 74.8±1.5       | 79.0±1.2        | 62.9±1.7        | 73.9±2.2        |
| DGCNN + PointCMC        | 78.4±3.4       | 84.4±5.9        | 68.4±2.4        | 76.3±3.8        |
| RSCNN + Rand            | 69.2±5.6       | 73.3±6.3        | 43.7±5.1        | 48.8±4.6        |
| RSCNN + GLR [39]        | 77.2±7.2       | 83.4±5.7        | 65.2±4.9        | 72.0±4.4        |
| RSCNN + CrossPoint [2]  | 83.5±6.7       | 88.3±4.3        | 78.8±4.3        | 79.6±3.8        |
| RSCNN + PointCMC        | 84.0±5.7       | 88.4±4.8        | 78.4±4.1        | 79.2±3.6        |

Table 5: Part Segmentation result on ShapeNetPart dataset. ‘mIoU’ denotes the mean IoU across all object classes in the dataset. Compared with the current supervised and self-supervised methods, PointCMC can achieve the best performance.

| Category     | Method      | mIoU(%) |
|--------------|-------------|---------|
| Supervised   | PointNet [35]| 83.7    |
|              | PointNet++ [37]| 85.1    |
|              | DGCNN [51]  | 85.1    |
| Unsupervised | Self-Contrast [7]| 82.3    |
|              | Jigsaw [41] | 85.3    |
|              | OcCo [49]   | 85.0    |
|              | PointContrast [59]| 85.1    |
|              | Liu et al. [29]| 85.3    |
|              | CrossPoint [2]| 85.5    |
|              | PointCMC    | 85.7    |

Table 6: Linear classification results on ModelNet40 with different modules, where A represents the DGCNN feature extractor and B represents the RSCNN feature extractor. In both backbones, the joint-module network outperforms the single-module network.

| Backbones | L2L | L2G | G2G | accuracy(%) |
|-----------|-----|-----|-----|-------------|
| A         | ✓   | ✓   | ✓   | 90.9        |
| A         | ✓   |     |     | 90.1        |
| A         | ✓   |     | ✓   | 90.2        |
| A         | ✓   | ✓   | ✓   | 91.7        |
| B         | ✓   |     |     | 90.6        |
| B         | ✓   |     | ✓   | 89.9        |
| B         | ✓   | ✓   | ✓   | 90.1        |
| B         | ✓   | ✓   | ✓   | 91.5        |

4.3. Ablation study

We conduct ablation experiments based on DGCNN [51] and RSCNN [30] to demonstrate the efficacy of our model. We evaluate our model thoroughly from two perspectives: 1) the impact of modules and 2) the impact of the number of corresponding 2D images.

Impact of modules. Our method maps local and global features of images and point clouds to the same Euclidean space to extract different fine-grained semantic features, respectively, and the cross-modal correspondence establishes a hard positive sample in contrast learning. We hypothesize that multiscale cross-modal correspondence can learn representations better than a single scale. We validate our hypothesis by training a network with only a single module and evaluating the accuracy of the linear SVM classifier on the ModelNet40 and ScanObjectNN datasets. As shown in Table 6, our joint-module network outperforms the single-module network in all settings. Our joint-module network obtains accuracy gains of 0.9% and 0.8% over the second-best approach in ModelNet40 with the DGCNN feature extractor and the RSCNN feature extractor.

In particular, we observe that the best classification performance among the single-module networks is the L2L-only network, and we believe that embedding local features of the pictures near the point cloud features promotes fine-grained semantic understanding. Figure 4 visualizes
Figure 4: **T-SNE [47] visualization on the ModelNet10 test dataset.** We show the feature distribution extracted by different module networks: (a) L2L-only network; (b) L2G-only network; (c) G2G-only network; and (d) Joint-module network. Our proposed joint-module network can better distinguish between different classes than single-module networks.

Figure 5: **Visualization of part segmentation results.** We visualize the part segmentation results from (a) Self-Contrast; (b) OcCo; (c) CrossPoint; (d) PointCMC; and (e) Ground-Truth(GT).

Impact of images. We also verify the effect of different numbers of rendered images on our network by selecting $n$ rendered images from different random views and by calculating the average of the $n$ image features as the input for the T-SNE [47] plots of the methods in the ablation experiments described above for the test split of ModelNet10. Although the single-module network can classify different categories of features well, there are still categories with obscure boundaries (e.g., table, chair), while the joint-module network is better at distinguishing between categories.

| Numbers of rendered images | 1  | 2  | 3  | 4  | 5  |
|----------------------------|----|----|----|----|----|
| Linear accuracy(%)         | 91.7 | 91.6 | 91.3 | 91.3 | 90.9 |
the loss calculation. Table 7 records the results with different numbers of images on the linear SVM classifier. The results show that the best results are obtained when only one image input. We believe the information collected from images is redundant when more than two image inputs.

5. Conclusion

In this paper, we introduce PointCMC, a novel self-supervised learning method for point cloud representation learning. Comprehensive downstream experiments show that enforcing the correspondences of potential semantics across modalities leads to better point cloud representations. Meanwhile, the ablation experiments validate our hypothesis that joint correspondence of cross-modal low-level and high-level semantics enables the model to achieve the best performance. In future works, transferring the learned image knowledge to more efficient models and applying our method to point cloud scenario tasks such as segmentation and detection is an interesting direction. Furthermore, leveraging point cloud geometric knowledge as auxiliary inputs for image learning is another direction that is not explored.

References

[1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. J. Guibas. Learning representations and generative models for 3d point clouds. Learning, 2017. 2, 3, 6, 7

[2] 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, pages 9902–9912, 2022. 2, 3, 4, 6, 7, 8

[3] P. Bachman, R. D. Hjelm, and W. Buchwalter. Learning representations by maximizing mutual information across views. neural information processing systems, 2019. 3

[4] A. X. Chang, T. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. arXiv: Graphics, 2015. 6

[5] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. international conference on machine learning, 2020. 3

[6] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. International conference on computer vision, 2015. 3

[7] B. Du, X. Gao, W. Hu, and X. Li. Self-contrastive learning with hard negative sampling for self-supervised point cloud learning. arXiv: Computer Vision and Pattern Recognition, 2021. 3, 6, 8

[8] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. Computer vision and pattern recognition, 2016. 3

[9] M. Gadelha, R. Wang, and S. Maji. Multiresolution tree networks for 3d point cloud processing. European conference on computer vision, 2018. 6

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. Neural information processing systems, 2014. 3

[11] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent: A new approach to self-supervised learning. Neural information processing systems, 2020. 3

[12] Z. Han, M. Shang, Y.-S. Liu, and M. Zwicker. View inter-prediction gan: Unsupervised representation learning for 3d shapes by learning global shape memories to support local view predictions. National conference on artificial intelligence, 2019. 6

[13] Z. Han, X. Wang, Y.-S. Liu, and M. Zwicker. Multi-angle point cloud-vae: Unsupervised feature learning for 3d point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. International conference on computer vision, 2019. 3

[14] K. Hassani and M. Haley. Unsupervised multi-task feature learning on point clouds. arXiv: Computer Vision and Pattern Recognition, 2019. 6

[15] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. Computer vision and pattern recognition, 2022. 3

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv: Computer Vision and Pattern Recognition, 2015. 7

[17] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. International conference on computer vision, 2021. 3, 6, 7

[18] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le. Pf-net: Point fractal network for 3d point cloud completion. Computer vision and pattern recognition, 2020. 3

[19] T. Hugues, R. Q. Charles, D. Jean-Emmanuel, M. Beatriz, G. Francois, and G. Leonidas. Kpconv: Flexible and deformable convolution for point clouds. IEEE Conference Proceedings, 2019. 3

[20] L. Jing, L. Zhang, and Y. Tian. Self-supervised feature learning by cross-modality and cross-view correspondences. Computer vision and pattern recognition, 2021. 2, 3

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv: Learning, 2014. 7

[22] M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. Aiche Journal, 1991. 3

[23] T. Le and Y. Duan. Pointgrid: A deep network for 3d shape understanding. Computer vision and pattern recognition, 2018. 3

[24] C.-L. Li, M. Zaheer, Y. Zhang, B. Póczos, and R. Salakhutdinov. Point cloud gan. Learning, 2018. 3

[25] J. Li, B. M. Chen, and G. H. Lee. So-net: Self-organizing network for point cloud analysis. Computer vision and pattern recognition, 2018. 2, 3, 6
[26] R. Li, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng. Pugan: A point cloud upsampling adversarial network. *International conference on computer vision*, 2019. 3

[27] R. Li, X. Li, P.-A. Heng, and C.-W. Fu. Point cloud upsampling via disentangled refinement. *Computer vision and pattern recognition*, 2021. 3

[28] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. Pointcnn: convolution on x-transformed points. *Neural information processing systems*, 2018. 2, 7

[29] F. Liu, G. Lin, and C.-S. Foo. Point discriminative learning for unsupervised representation learning on 3d point clouds. *arXiv: Computer Vision and Pattern Recognition*, 2021. 8

[30] Y. Liu, B. Fan, S. Xiang, and C. Pan. Relation-shape convolutional neural network for point cloud analysis. *arXiv: Computer Vision and Pattern Recognition*, 2019. 2, 3, 6, 7, 8

[31] Y.-C. Liu, Y.-K. Huang, H.-Y. Chiang, H.-T. Su, Z.-Y. Liu, C.-T. Chen, C.-Y. Tseng, and W. H. Hsu. Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pre-training. *arXiv: Computer Vision and Pattern Recognition*, 2021. 2, 3

[32] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *Learning*, 2016. 7

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. *Neural information processing systems*, 2019. 6

[34] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim. Self-supervised learning of point clouds via orientation estimation. *International conference on 3d vision*, 2020. 6

[35] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Computer vision and pattern recognition*, 2016. 2, 3, 8

[36] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. *Computer vision and pattern recognition*, 2016. 3

[37] 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*, 30, 2017. 2, 3, 7, 8

[38] A. Radford, J. W. Kim, C. Hallac, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *International conference on machine learning*, 2021. 3

[39] Y. Rao, J. Lu, and J. Zhou. Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds. *Computer vision and pattern recognition*, 2020. 3, 4, 5, 6, 7, 8

[40] A. Sanghi. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. *European conference on computer vision*, 2020. 3

[41] J. Sauder and B. Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Neural information processing systems*, 2019. 6, 7, 8

[42] A. Sharma, O. Grau, and M. Fritz. Vconv-dae: Deep volumetric shape learning without object labels. *arXiv: Computer Vision and Pattern Recognition*, 2016. 3

[43] C. Sharma and M. Kaul. Self-supervised few-shot learning on point clouds. *Neural information processing systems*, 2020. 7, 8

[44] M. A. Uy, Q.-H. Pham, B.-S. Hua, D. T. Nguyen, and S.-K. Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. *International conference on computer vision*, 2019. 7

[45] D. Valsesia, G. Facastoro, and E. Magli. Learning localized generative models for 3d point clouds via graph convolution. *International conference on learning representations*, 2018. 3

[46] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *Learning*, 2018. 4

[47] L. van der Maaten and G. E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008. 9

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[49] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner. Unsupervised point cloud pre-training via occlusion completion. *International conference on computer vision*, 2021. 3, 6, 7, 8

[50] P.-S. Wang, Y.-Q. Yang, Q.-F. Zou, Z. Wu, Y. Liu, and X. Tong. Unsupervised 3d learning for shape analysis via multiresolution instance discrimination. *International conference on artificial intelligence*, 2020. 2

[51] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 2018. 2, 3, 6, 8

[52] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Neural information processing systems*, 2016. 6, 7

[53] W. Wu, Z. Qi, and L. Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. *Computer vision and pattern recognition*, 2018. 2

[54] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. *Computer vision and pattern recognition*, 2014. 7

[55] W. Wu, Z. Zhang, M. Zeng, F. Qin, and Y. Wang. Joint analysis of shapes and images via deep domain adaptation. *Computer Graph.*, 70:140–147, 2018. 3

[56] W. Xi, T. Zhang, Y. Li, Y. Zhang, and F. Wu. Multi-modality cross attention network for image and sentence matching. *Computer vision and pattern recognition*, 2020. 4

[57] A. Xia, J. Huang, D. Guan, and S. Lu. Unsupervised representation learning for point clouds: A survey. *arXiv preprint arXiv:2002.13589*, 2022. 3

[58] J. Xie, Z. Zheng, R. Gao, W. Wang, S.-C. Zhu, and Y. N. Wu. Learning descriptor networks for 3d shape synthesis and analysis. *Computer vision and pattern recognition*, 2018. 3
[59] S. Xie, J. Gu, D. Guo, C. R. Qi, L. J. Guibas, and O. Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. *European Conference on Computer Vision*, 2020. 2, 3, 8

[60] C. Xu, S. Yang, B. Zhai, B. Wu, X. Yue, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka. Image2point: 3d point-cloud understanding with pretrained 2d convnets. *arXiv: Computer Vision and Pattern Recognition*, 2021. 3

[61] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Neural Information Processing Systems*, 2019. 6

[62] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. *arXiv: Computer Vision and Pattern Recognition*, 2018. 2

[63] Y. Yang, C. Feng, Y. Shen, and D. Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. *Computer Vision and Pattern Recognition*, 2017. 3, 6, 7

[64] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. J. Guibas. A scalable active framework for region annotation in 3d shape collections. *International Conference on Computer Graphics and Interactive Techniques*, 2016. 8

[65] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng. Pu-net: Point cloud upsampling network. *Computer Vision and Pattern Recognition*, 2018. 3

[66] L. Zhang and Z. Zhu. Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks. *International Conference on 3d Vision*, 2019. 2, 3, 6

[67] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. *Computer Vision and Pattern Recognition*, 2016. 3

[68] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra. Self-supervised pretraining of 3d features on any point-cloud. *International Conference on Computer Vision*, 2021. 6

[69] Y. Zhao, T. Birdal, H. Deng, and F. Tombari. 3d point capsule networks. *Computer Vision and Pattern Recognition*, 2018. 3, 6, 7

[70] K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Learning to prompt for vision-language models. *arXiv: Computer Vision and Pattern Recognition*, 2021. 3