This CVPR 2021 workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Self-supervised Feature Learning by Cross-modality and Cross-view Correspondences

Longlong Jing Ling Zhang Yingli Tian
The City University of New York

Abstract

The success of supervised learning requires large-scale ground truth labels which are very expensive, time-consuming, or may need special skills to annotate. To address this issue, many self- or un-supervised methods are developed. Unlike most existing self-supervised methods to learn only 2D image features or only 3D point cloud features, this paper presents a novel and effective self-supervised learning approach to jointly learn both 2D image features and 3D point cloud features by exploiting cross-modality and cross-view correspondences without using any human annotated labels. Specifically, 2D image features of rendered images from different views are extracted by a 2D convolutional neural network, and 3D point cloud features are extracted by a graph convolution neural network. Two types of features are fed into a two-layer fully connected neural network to estimate the cross-modality correspondence. The three networks are jointly trained (i.e. cross-modality) by verifying whether two sampled data of different modalities belong to the same object, meanwhile, the 2D convolutional neural network is additionally optimized through minimizing intra-object distance while maximizing inter-object distance of rendered images in different views (i.e. cross-view). The effectiveness of the learned 2D and 3D features is evaluated by transferring them on five different tasks including multi-view 2D shape recognition, 3D shape recognition, multi-view 2D shape retrieval, 3D shape retrieval, and 3D part-segmentation. Extensive evaluations on all the five different tasks across different datasets demonstrate strong generalization and effectiveness of the learned 2D and 3D features by the proposed self-supervised method.

1. Introduction

The deep convolutional neural networks for computer vision tasks (e.g. classification [37, 39], detection [26], segmentation [2], etc.) are highly relied on large-scale labeled datasets [19, 40]. Collecting and annotating the large-scale datasets are usually expensive and time-consuming. To facilitate 3D computer vision research, more and more 3D datasets such as mesh and point cloud data have been recently proposed. Compared to the annotation process of 2D image data, 3D point cloud data are especially harder to annotate and the cost is more expensive.

To learn features from unlabeled data, many self-/un-supervised learning methods are proposed for images, videos [11, 18, 24], and 3D point cloud data [13] by training deep neural networks to solve pretext tasks with automatically generated labels based on attributes of the data such as clustering images [3, 33], playing image jigsaw [32], predicting geometric transformation of images or videos [9, 17], image inpainting [35], reconstructing point cloud [57], etc. The learned features through these processes are then used as pre-trained models for other tasks to overcome over-fitting and speed up convergence especially when training data is limited.

Recently self-supervised feature learning on 3D point cloud data attract more attention including auto-encoders-based methods [1, 8, 57, 59], generative model-based methods [28, 48, 54], and context-based pretext task method [13, 58]. The auto-encoders-based and generative-based methods learn features by generating or reconstructing the point cloud data and have obtained very competitive performance on the 3D recognition benchmark [57]. However, by optimizing the loss for generation or reconstruction tasks, these networks suffer from modeling low-level features and compromising their ability to capture high-level features from the point cloud data.

In this paper, as shown in Fig. 1, we propose a novel idea to explore how to use the abundant relations of different views and modalities of 3D data (e.g. mesh, point cloud, rendered shading images, rendered depth images, etc.) as supervision signal to learn both 2D and 3D features without using any human annotated labels. The main contributions of this paper are summarized as follows:

- We design a new schema to jointly learn both 2D and 3D features through solving two parallel pre-defined pretext tasks: 1) Cross-modality task - to recognize whether two data in different modalities (3D point cloud and 2D image) belong to the same object; 2)

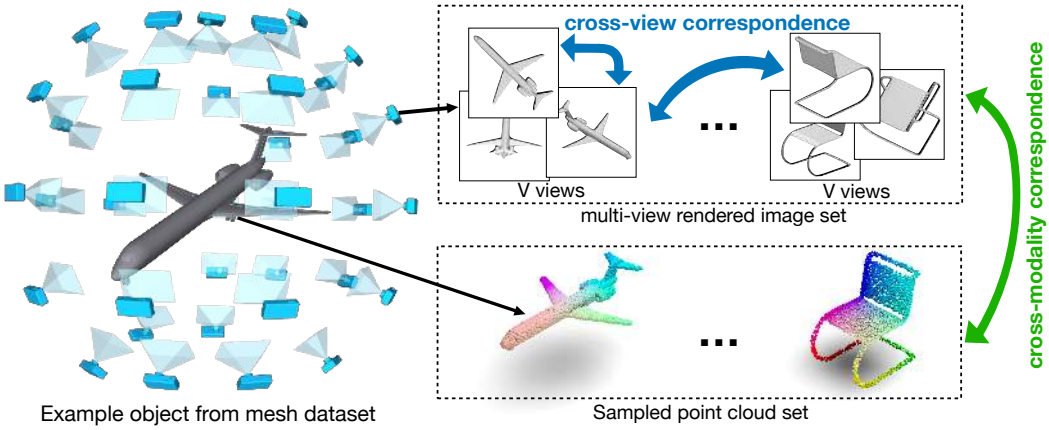

Figure 1. Training set generation. From 3D mesh datasets, multi-view rendered image set and sampled point cloud set are generated. The relations of the different data representations are employed as supervision signal (cross-view and cross-modality correspondences) to learn both 2D and 3D features without using any human annotated labels.

Cross-view task - to minimize the distance of 2D image features in different views of the same object while maximizing the distance of 2D image features from different objects.

- The discriminative 2D and 3D features learned by the self-supervised schema are used as pre-trained models for other down-stream tasks such as classification, retrieval, and 3D part segmentation, etc.

- Extensive experiments on five different tasks (i.e. multi-view 2D shape recognition, 3D shape recognition, multi-view 2D shape retrieval, 3D shape retrieval, and 3D part-segmentation) demonstrate the effectiveness and generalization of the proposed framework. For the recognition tasks, our 2D and 3D models outperform the existing state-of-the-art unsupervised methods and achieve comparable performance as the supervised methods on the ModelNet40.

2. Related Work

3D Point Cloud Understanding: Various methods have been proposed for point cloud data understanding and they can be categorized into three types: hand-crafted methods [5, 20] which use hand-designed feature extractors to model the geometric features; deep neural networks on regular 3D data [4, 7, 12, 22, 31, 38, 46, 47, 49] in which the network usually operates on multi-view rendered images [46, 47] or volumetric voxelized data [4, 22, 31, 38, 49]; and deep neural networks on unordered 3D data in which the network operates directly on the unordered point cloud data [27, 29, 37, 39, 51, 53, 55]. 3D point cloud data can be rendered into 2D images from different views to create multi-modality data. To utilize the multi-view images, Su *et al.* proposed to tackle the 3D shape recognition by multi-view CNN operating on multiple 2D images that rendered from different views of the 3D data [46]. To directly learn 3D features on unordered point cloud data, Qi *et al.* proposed the milestone work PointNet by using a deep neural network to classify 3D shape data, and later this work was extended to many other networks [29, 39, 53]. Wang *et al.* proposed the EdgeConv with Multilayer Perceptron (MLP) to modal local features for each point from its k nearest neighbor (KNN) points.

2D Unsupervised Feature Learning: Recently, many self-supervised learning methods (also known as unsupervised learning) have been proposed to learn features from unlabeled data [3, 9, 11, 17, 18, 24, 32, 33, 35]. Usually, a pretext task is defined to train a network with automatically generated labels based on the attributes of the data. These methods fall into four groups: correspondence-based method (i.e. using the correspondence of two different modalities like visual and audio streams in videos as supervision signals) [25]; context-based methods (i.e. using context structure or similarity of the data as supervision signals) [3, 9, 32, 33]; generation-based methods (i.e. using the learned features in the process of generating images or videos such as Generative Adversarial Networks and Auto-encoder) [35]; and free semantic label-based methods (i.e. using the automatically generated labels by game engines or some traditional methods) [34]. The 2D self-supervised learning has been well studied recently, and some methods have been successfully adapted to the 3D self-supervised feature learning [13, 42, 58].

3D Self-supervised Feature Learning: Several self-supervised learning methods have been proposed to model features from unlabeled 3D point cloud data [5, 8, 20, 28, 48, 54, 57, 59]. Most of these methods are auto-encoder based [1, 8, 57, 59] to learn the features in the process of reconstructing the point cloud data or generative-based

methods [28, 48, 50, 54] to learn the features in the process of generating plausible point cloud data. Recently, a few work attempted to learn features by designing novel pretext tasks [13, 42, 58]. Sauder *et al.* proposed to learn features by recognizing the relative position of two segments from point cloud data [42]. Zhang *et al.* proposed EdgeConv to learn features by verifying whether two segments are from the same object and then boosting the performance of a cluster task [58]. Hassani *et al.* proposed a multi-task learning framework to learn features by optimizing three different tasks including clustering, prediction, and reconstruction [13]. However, all these methods only focus on learning one type of feature for 3D shape data while ignoring the inherent multi-modalities of different data representations. In this paper, we propose to learn two different types of features, 2D image features, and 3D point cloud features, by exploiting the correspondences of cross-modality and cross-view attributes of 3D data.

3. Method

Preparing 2D images in multiple views and 3D point cloud data from mesh objects is essential for our proposed self-supervised 2D and 3D feature learning. The details of the data generation, the architecture of the framework, and model parameterization are introduced in the following sections.

3.1. Data Generation

As shown in Fig. 1, two types of training sets are generated from 3D object datasets, i.e., multi-view rendered image set and sampled point cloud set, for learning 2D and 3D features. 3D objects are typically represented in polygon meshes as collections of vertices, edges, and faces, etc. See Section 3.3 for specific input samples for the framework.

Multi-view image generation: Following [46], the Phong reflection model [36] is employed as the rendering engine to generate rendered images in different views from 3D polygon meshes. By given a 3D polygon mesh m from a 3D object set M, a spherical coordinate system is defined with the centroid of m as the center for the system. The centroid for each m is calculated as the average of all mesh face geometric centers of m, while the mesh face centers are weighted by the corresponding mesh face areas. To project m to multi-view 2D planes, V virtual cameras (viewpoints) around m are randomly placed for each object along a sphere surface with radius R (see Fig. 1). Each virtual camera is arranged by an azimuthal angle (randomly selected from 10 to 340 degrees) and a polar angle (randomly selected from 10 to 165 degrees) of the spherical coordinate system. All virtual cameras point toward the centroid of m, and one image is rendered form each camera. The intensities of pixels in the rendered images are determined by interpolating the reflected intensities of the polygon vertices. Due to the randomness of the sampled views, some parts of objects would be dark following the traditional settings if only one light source is placed during rendering. To avoid the problem, in our rendering process, two light sources are placed facing each other, while the mesh object is in between. The model shapes are uniformly scaled to fit into the perspective view. Note that V images at different views are rendered for each 3D object, and up to two of the rendered images are used in each input training sample, and $v \leq V$ images are used in the testing phase.

Point cloud sampling: Following [37], we adopt the Farthest Point Sampling (FPS) algorithm to sample point clouds from each mesh object surface in the mesh datasets. Starting from a randomly chosen point, the next point is sampled in turn according to the average distance to all sampled points, that is, the farthest point. Each mesh object is uniformly sampled 2,048 points to keep the shape information of the object as much as possible. All sampled points are then normalized into a unit sphere.

3.2. Framework Architecture

As illustrated in Figure 2, there are three networks in our framework: a 2DCNN (F_{img}) to extract 2D features from images cross different views, a graph neural network (F_p) to extract 3D features from unordered point cloud data, and a two-layer fully connected neural network F_f to predict the cross-modality correspondence based on the two types of features extracted by F_{img} and F_p. The three networks are jointly optimized by cross-modality correspondence, meanwhile, the network F_{img} is optimized by cross-view correspondence (see details in Section 3.3).

The 2D image feature learning network (F_{img}) employs ResNet18 [14] as the backbone network with four convolution blocks with a number of $\{64, 128, 256, \text{and } 512\}$ 3×3 kernels. Each convolution block includes two convolution layers followed by a batch-normalization layer and a ReLU layer, except the first convolution block which consists of one convolution layer, one batch-normalization layer, and one max-pooling layer. A global average pooling layer, after the fourth convolution blocks, is used to obtain the global features for each image. Unless specifically pointed out, a 512-dimensional vector after the global average pooling layer is used for all our experiments.

The 3D point cloud feature learning network (F_p) employs dynamic graph convolutional neural network (DGCNN) [53] as the backbone model due to its capability to model local structures of each point by dynamically constructed graphs and its good performance on classification and segmentation tasks. There are four EdgeConv layers and the number of kernels in each layer is 64, 64, 64, and 128, respectively. Each convolution graph consists of one KNN graph layer which builds the KNN graph for each point and two convolution layers. Each convolution layer is

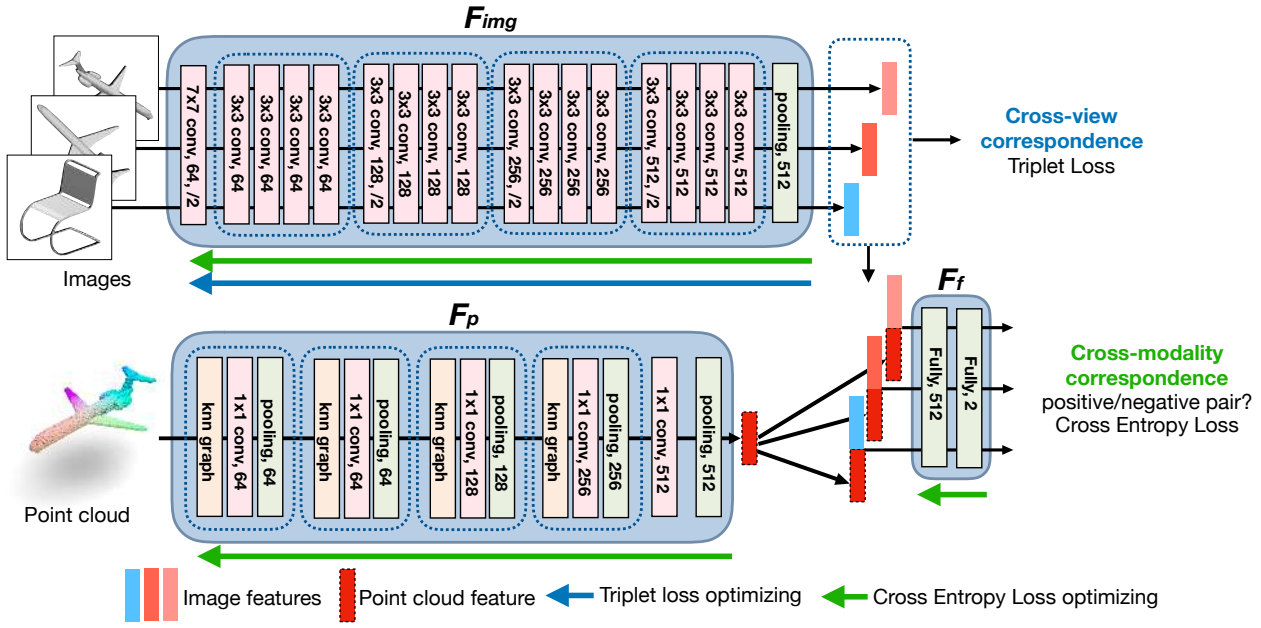

Figure 2. The proposed framework for self-supervised 2D and 3D feature learning by cross-modality and cross-view correspondences. It consists of an image feature extracting 2DCNN (F_{img}) taking different rendered views, a graph neural network (F_p) taking unordered point cloud data, and a two-layer fully connected neural network (F_f) taking the concatenation of two types of features extracted by F_{img} and F_p to predict the cross-modality correspondence. F_{img}, F_p, and F_f are jointly trained (i.e. cross-modality, the blue solid arrow) by verifying whether two sampled data of different modalities belong to same object, meanwhile, F_{img} is additionally optimized through minimizing intra-object distance while maximizing inter-object distance of rendered images in different views (i.e. cross-view, the green solid arrow).

followed by a batch-normalization layer and a leaky ReLU layer. The EdgeConv layers aim to construct graphs over k nearest neighbors calculated by KNN and the features for each point are calculated by an MLP over all the k closest points. After the four EdgeConv blocks, a 512-dimension fully connected layer is used to extract per-point features for each point and then a max-pooling layer is employed to extract global features for each object.

The two-layer fully connected neural network F_f is employed for cross-modality classification, which consists of a 256-dimensional fully connected layer and a 2-dimensional fully connected layer. Each feature vector feeding into F_f is extracted by F_{img} and F_p and concatenated together as a 1024-dimension vector. The output of F_f is a binary classification value.

3.3. Model Parameterization

In our proposed self-supervised learning schema, two types of constraints are used as supervision signals to optimize the networks: cross-modality correspondence and cross-view correspondence. The cross-modality correspondence requires networks to learn modality-invariant features extracted from two different modalities F_{img} and F_p, while the cross-view correspondence requires the subnetwork F_{img} to capture semantic 2D image features to match objects from random views. We formulae the cross-modality task as a classification task and the cross-view task as a metric learning task.

Let $\mathcal{D} = \{sample^{(1)}, ..., sample^{(N)}\}$ denotes training data of size N. The i-th input sample $sample^{(i)} = \{p^{(i)}, img_1^{(i)}, img_2^{(i)}, img_3^{(i)}, y_1^{(i)}, y_2^{(i)}, y_3^{(i)}\}$, where $p^{(i)}$ and $img_1^{(i)}$, $img_2^{(i)}$ represent the point cloud and two different rendered views generated from the same 3D mesh object respectively, and $img_3^{(i)}$ is an image rendered from a different object. The labels $y_j^{(i)} \in \{0, 1\}$ indicates whether the point cloud $p^{(i)}$ and the rendered image $img_j^{(i)}$ are from same object where 1 for same object and 0 for different objects. Note that $img_1^{(i)}$ and $img_2^{(i)}$ are randomly selected in V rendered views from a 3D mesh object same as the sampled point cloud $p^{(i)}$, while $img_3^{(i)}$ is from a different one.

Cross-view correspondence: The objective of the cross-view task is to train the network F_{img} to learn view invariant features from rendered images. When an object observed from different views, the visible parts may look differently, however, the semantic features for images in different views should be similar. Therefore, triplet loss [43] is employed here to train the network to minimize distance of features of positive pairs (i.e. from same object) and maxi-

mize distance of features of negative pairs (i.e. from different objects):

$$L_{triplet} = max(\|F_{img}(img_1^{(i)}) - F_{img}(img_2^{(i)})\|^2 - \|F_{img}(img_1^{(i)}) - F_{img}(img_3^{(i)})\|^2 + \alpha, 0), \quad (1)$$

where the triple samples $img_1^{(i)}$, $img_2^{(i)}$ and $img_3^{(i)}$ correspond to anchor, positive and negative rendered images, α is the margin hyper-parameter to control the differences of intra- and inter- objects.

Cross-modality correspondence: The cross-modality learning is modeled as a binary classification task by employing the cross-entropy loss to optimize all the three networks. After obtaining image features by F_{img} from rendered images and point cloud features by F_p from point clouds, the network F_f predicts whether the two input data of different modalities are from same object by discovering the high-level modality invariant features. The positive samples are the point cloud and image pairs from same 3D mesh object, while the negative samples are from different objects. The loss function for jointly optimizing networks F_{img}, F_p, and F_f is:

$$L_{cross} = -\sum_{j=1}^{3}(y_j^{(i)} \log(F_f(F_{img}(img_j^{(i)}), F_p(p^{(i)}))) + (1 - y_j^{(i)}) \log(1 - F_f(F_{img}(img_j^{(i)}), F_p(p^{(i)})))), \quad (2)$$

The input features of F_f are extracted by F_{img} and F_p, and F_f learns the correlation of the features extracted from two different data modalities.

When jointly train the three networks, a linear weighted combination of the loss functions $L_{triplet}$ and L_{cross} are employed to optimize the whole framework. The final self-learning loss is combined as:

$$L_{self} = L_{triplet} + \beta L_{cross}, \quad (3)$$

where β is the weight for the cross-modality loss.

The details of the joint training process are illustrated in *Algorithm 1*. After the jointly training finished, two networks F_{img} and F_p are obtained as pre-trained models for two different modalities. The joint training enables the two feature extractors to learn more discriminative and robust features cross different data domains.

Algorithm 1 The proposed self-supervised feature learning algorithm.

mini-batch size: B; 2D image features: fi; 3D point cloud features: fp; binary prediction: $\hat{y}$;
for all sampled mini-batch $\{sample^{(b)}\}_{b=1}^{B}$ **do**
  for all $b \in \{1, \ldots, B\}$ **do**
    # feature extraction
    $fi_1^{(b)} = F_{img}(img_1^{(b)})$; $fi_2^{(b)} = F_{img}(img_2^{(b)})$; $fi_3^{(b)} = F_{img}(img_3^{(b)})$;
    $fp^{(b)} = F_p(p^{(b)})$;
    # classification prediction
    $\hat{y}_1^{(b)} = F_f(fi_1^{(b)}, fp^{(b)})$; $\hat{y}_2^{(b)} = F_f(fi_2^{(b)}, fp^{(b)})$; $\hat{y}_3^{(b)} = F_f(fi_3^{(b)}, fp^{(b)})$;
    # loss calculation
    $\mathcal{L}_{triplet}^{(b)} = max(\|fi_1^{(b)} - fi_2^{(b)})\|^2 - \|fi_1^{(b)} - fi_3^{(b)}\|^2 + \alpha, 0)$
    $L_{cross}^{(b)} = -\sum_{j=1}^{3}(y_j^{(b)} \log(\hat{y}_j^{(b)}) + (1 - y_j^{(b)}) \log(1 - \hat{y}_j^{(b)}))$
    $L_{self}^{(b)} = L_{triplet}^{(b)} + \beta L_{cross}^{(b)}$
  end for
  $\mathcal{L} = \frac{1}{B}\sum_{b=1}^{B} L_{self}^{(b)}$
  update networks F_{img}, F_p and F_f to minimize $\mathcal{L}$
end for
return pre-trained networks F_{img} and F_p

4. Experimental Results

4.1. Experimental Setup

Self-supervised learning: The proposed framework is optimized end-to-end using the SGD optimizer with an initial learning rate of 0.001, the moment of 0.9, and weight decay of 0.0005. The learning rate decreases by 90% every $40,000$ iteration. The networks for self-supervised learning are trained on the ModelNet40 dataset for $120,000$ iterations using a mini-batch size of 32. To learn more robust features, data augmentation is applied to both images and point clouds. The images are randomly cropped and randomly flipped with 50% probability in the horizontal direction, while the point clouds are randomly rotated between $[0, 2\pi]$ degrees along the up-axis, randomly jittered the position of each point by Gaussian noise with zero mean and 0.02 standard deviation. The rendering views V is 180 for each 3D mesh object in the dataset. During the testing, we randomly select 2D-2D and 2D-3D testing pairs from the test split of ModelNet40 and ModelNet10. The amount of two types of pairs is ten times the test split including half positive pairs and half negative pairs.

Evaluation of learned 2D and 3D features: To evaluate the effectiveness and generalization of the learned 2D and 3D features by the proposed self-supervised learning schema, five different tasks are designed as follows. For the multi-view 2D shape recognition and 3D shape recognition tasks, the image and point cloud features are extracted by two pre-trained networks F_{img} and F_p, then trained on corresponding SVMs with one class linear kernel, respectively. For the 3D part segmentation task, additional fully connected layers are added on top of the pre-trained F_p and then fine-tuned on the ShapeNet [4] dataset. The network is optimized with Adam optimizer [21] using an initial learning rate of 0.003 and decreased by 90% every 20 epochs. For the 2D and 3D shape retrieval tasks, Euclidean distance over the global features of two objects is used as a metric to measure the similarity of two objects.

Datasets: All the experiments are conducted on two 3D object benchmarks: ModelNet40 [56] and ShapeNet [4].

The ModelNet40 dataset contains $12,311$ meshed models covering 40 classes, of which $9,843$ are used for training and $2,468$ for testing. The ModelNet40 is used to train our proposed self-supervised learning framework as well as for the evaluation tasks of multi-view 2D shape recognition and 3D shape recognition. The ModelNet10, a subset of ModelNet40, is also used as a testing set, which contains 10 classes. The ShapeNet contains 16 object categories including $12,137$ models for training and $2,874$ for testing and it is employed to evaluate the task of 3D part segmentation. In all experiments, $2,048$ points are sampled for each 3D mesh object as the input point cloud data.

4.2. Cross-modality and Cross-view Correspondence Evaluation

A straightforward evaluation of the effectiveness of our proposed self-supervised learning framework is to recognize the cross-modality and cross-view correspondence with ModelNet40 and ModelNet10 datasets. Table 1 and 2 report the cross-modality recognition accuracy and cross-view feature Euclidean distance of testing image pairs.

Table 1. Performance on pretext task: cross-modality recognition. CM indicates network training with cross-modality correspondence. CV indicates network training with cross-view correspondence.

Testing Set	Network	Cross-modality Acc (%)
ModelNet40	F_p-CM	93.5
	F_p-CM-CV	91.8
ModelNet10	F_p-CM	92.0
	F_p-CM-CV	91.5

Table 2. Performance on pretext task: cross-view feature distance analysis. mPD indicates mean Pair Distance with corresponding standard deviation in brackets.

Testing set	Network	Positive mPD	Negative mPD
ModelNet40	F_{img}-CM	6.43 (2.38)	12.07 (3.46)
	F_{img}-CM-CV	2.56 (0.56)	4.33 (1.37)
ModelNet10	F_{img}-CM	6.83 (2.36)	11.29 (3.15)
	F_{img}-CM-CV	2.571 (0.52)	4.304 (1.06)

For the cross-modality recognition task in Table 1, our networks accomplish over 90% accuracy which shows that the self-supervised learning successfully learns modality invariant features. For the cross-view correspondence recognition in Table 2, the margins between the mean distance of positive pairs and that of negative pairs are very large which demonstrates that the networks indeed learn the view-invariant features. When the networks trained jointly with cross-modality and cross-view correspondence, although the performance of cross-modality recognition decreases a little bit, the standard deviations for the distances of both positives and negatives are significantly improved (see rows 2 and 4) which validate that the cross-view correspondence enforces the learning of view-invariant features.

One common problem of self-supervised learning is that the network can easily learn trivial features (e.g. corners, edges, or other low-level features) instead of high-level semantic features. To further analyze the features extracted by F_{img} and F_p, we use T-distributed Stochastic Neighbor Embedding (TSNE) [30] to visualize the learned 2D and 3D features of the top 10 object categories in ten different colors on ModelNet40 as shown in Fig. 3. Each point indicates one feature that is max-pooled from v extracted features of v views. In the feature space, the features belong to the same class are closer than the features from different object classes, which show that the network indeed can learn high-level semantic features.

4.3. Transfer to 2D and 3D shape recognition tasks

Our proposed framework effectively learns both 2D and 3D features and achieves high performance on the pretext task. Here, we further evaluate the learned 2D and 3D features (i.e. F_{img} and F_p) as pre-trained models on other down-stream supervised tasks: 2D and 3D shape recognition on ModelNet40 dataset. Two linear SVM classifiers are trained based on the extracted 2D and 3D features by F_{img} and F_p, respectively. Same as in subsection 4.2, each extracted feature for 2D recognition task is max pooled from v extracted features of v random views, except when $v = 1$.

Table 3. The performance of using the self-supervised learned models as feature extractors on the 2D and 3D shape recognition tasks on the ModelNet40 dataset. Both 2D and 3D shape recognition tasks are benefited from jointly training with cross-view and cross-modality correspondences. When multiple views (#Views = 12, 36, or 80) are available for testing, the performance of 2D shape recognition is significantly improved.

Modality	Network	Testing #Views	Recognition Acc (%)
2D Image	F_{img}-CM	1	66.1
	F_{img}-CM-CV	1	72.5 (+6.4)
	F_{img}-CM-CV	12	87.3 (+21.1)
	F_{img}-CM-CV	36	88.7 (+22.6)
	F_{img}-CM-CV	80	**89.3** (+23.2)
3D Point Cloud	F_p-CM	-	87.5
	F_p-CM-CV	-	**89.8** (+2.3)

As shown in Table 3, both the pre-trained F_{img} and F_p can achieve high accuracy on the 2D and 3D shape recognition tasks (89.3% and 89.8%) to recognize 40 object categories on ModelNet40 dataset which show that the two networks learn discriminative semantic features through the

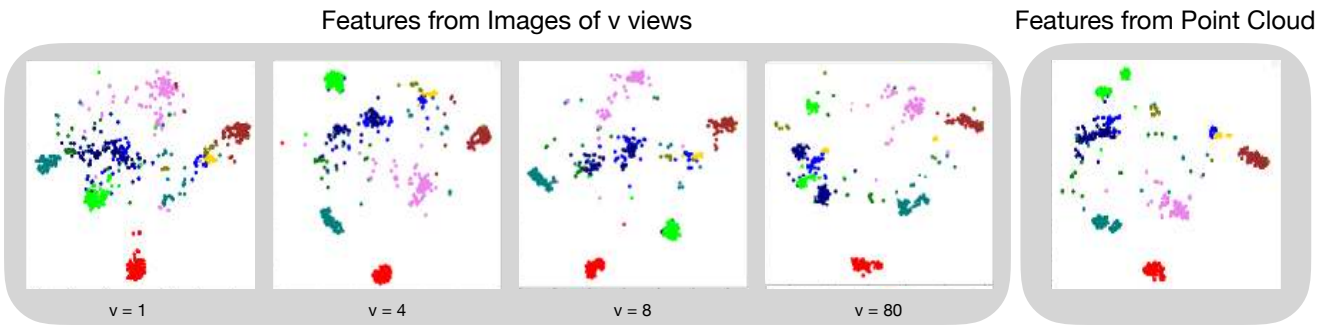

Figure 3. Visualization of 2D and 3D features of the top 10 object categories (ten different colors) on the ModelNet40 test set. When more views in the testing phase are used to represent 3D objects, the distribution of 2D image features for different category objects is more discriminative, and is more similar to 3D point cloud feature distribution.

self-supervised learning process. When only trained with cross-modality correspondence, the 3D features learned by F_p-CM achieve 87.5% accuracy while the performance of 2D features by F_{img}-CM is only 66.1%. The joint training of cross-view and cross-modality correspondence significantly improves the performance of F_{img} on 2D recognition ($+6.4\%$) and F_p on 3D recognition ($+2.3\%$). The accuracy of 2D recognition is further boosted by more discriminative image features max pooled from multi-testing-view features, achieving 89.3% with 80 views from each data.

4.4. Transfer to 2D and 3D shape retrieval tasks

To evaluate the generalization ability of the learned features, we further evaluate both 2D and 3D features extracted by F_{img} and F_p on shape retrieval tasks on the ModelNet40 dataset and Top-K accuracy are reported in Table 4.

Table 4. Performance of the learned 2D and 3D features on the 2D and 3D shape retrieval tasks on ModelNet40 dataset. When only using 1 view for each image, our self-supervised model F_{img}-CM-CV outperforms the ImageNet pre-trained model.

Network	#Views	Top1 (%)	Top5 (%)	Top10 (%)
F_p-CM	—	82.9	94.2	96.0
F_p-CM-CV	—	84.0	94.3	96.5
ImageNet [14]	1	61.4	82.1	88.5
F_{img}-CM	1	54.6	79.2	87.1
F_{img}-CM-CV	1	**66.9**	**85.8**	**91.1**
ImageNet [14]	12	83.6	94.7	96.7
F_{img}-CM	12	75.5	91.2	95.0
F_{img}-CM-CV	12	83.5	94.2	96.2
ImageNet [14]	80	87.6	95.7	97.4
F_{img}-CM	80	82.7	93.9	96.4
F_{img}-CM-CV	80	84.7	94.7	96.6

Since no other self-supervised learning methods for point cloud or multi-view images have reported performance on this task, we directly compare with ImageNet pre-trained models on the retrieval task. The 3D network F_p-CM and F_p-CM-CV can accomplish the retrieval task with high accuracy. As for the 2D network F_{img}, the performance is significantly improved when more views are used to represent each object. When only using 1 view for each image, our self-supervised model F_{img}-CM-CV outperforms the ImageNet pre-trained model. When more views (12 or 80) are available, our model achieves comparable performance with the supervised model which is pre-trained on the ImageNet dataset.

4.5. Transfer to 3D part segmentation task

To further verify the quality of 3D features learning by the pre-trained F_p for point cloud data, we conduct the transfer learning on the 3D part segmentation task with the ShapeNet dataset. To adapt F_p on the 3D part segmentation task, four fully connected layers are added on the top of F_p, and the output from all the four blocks and the global features are used to predict the pixel-wise labels. Three sets of experiments are studied: (1) Only update the four newly added layers with frozen F_p, (2) F_p and newly added layers are randomly initialized and supervised trained from scratch [37], (3) The learned features by F_p are used as pre-trained models and all the layers are fine-tuned (unfrozen). The extensive studies of train/fine-tune strategies with different amounts of training data on the ShapeNet dataset for the 3D part segmentation are shown in Table 5.

As shown in Table 5, training with cross-view correspondence can improve the ability of F_p to recognize object parts. When 100% of the training data are available, even without updating the parameters of F_p on the new task, it still achieves 80.8% instance mIOU which is only 2.2% lower than the supervised model. It validates that F_p can learn semantic features from the proposed pretext task and transfer them across datasets and tasks. When the full network is initialized with the pre-training weights and further fine-tuned, the instance mIOU improves by 0.4% and the class mIOU improves by 0.5% showing that the learned

Table 5. The performance of the three types of settings on different amount of data from the ShapeNet dataset. F_p with parameter-unfrozen setup outperforms the supervised method. When only a very small amount of data (2%) is available for training, all our models outperform the supervised model.

Network	Training data	Class mIOU (%)	Instance mIOU (%)
F_p-CM-Frozen	100%	71.2	78.6
F_p-CM-CV-Frozen	100%	74.7	80.8
F_p-Supervised [37]	100%	77.6	83.0
F_p-CM-Unfrozen	100%	78.1 (+0.5)	83.4 (+0.4)
F_p-CM-CV-Unfrozen	100%	**79.1 (+1.5)**	**83.7 (+0.7)**
F_p-CM-Frozen	20%	65.6	75.4
F_p-CM-CV-Frozen	20%	68.5	77.8
F_p-Supervised [37]	20%	69.9	79.1
F_p-CM-Unfrozen	20%	70.9 (+1.0)	80.0 (+0.9)
F_p-CM-CV-Unfrozen	20%	**72.2 (+2.3)**	**80.3 (+1.2)**
F_p-CM-Frozen	2%	57.1 (+0.9)	69.2 (+0.2)
F_p-CM-CV-Frozen	2%	58.4 (+2.2)	72.1 (+3.1)
F_p-Supervised [37]	2%	56.2	69.0
F_p-CM-Unfrozen	2%	60.6(+4.4)	72.6 (+3.6)
F_p-CM-CV-Unfrozen	2%	**60.7 (+4.5)**	**74.0 (+5.0)**

weights for F_p from self-supervised pretext task can be served as a good starting point for the optimization. When using only 20% data, the parameter-unfrozen setup can significantly (+2.3% on class mIOU, and +1.2% on instance mIOU) boost up the performance than the supervised setup. When using only 2% of the data, the performance of both parameter-frozen setup (+2.2% on class mIOU, and +3.1% on instance mIOU) and parameter-unfrozen setup (+4.5% on class mIOU, and +5.0% on instance mIOU) are better than the supervised setup. Our pre-trained F_p performs well when fine-tuned on small-scale 3D shape datasets.

4.6. Comparison with the State-of-the-Art methods

In this section, we further compare our pre-trained F_{img} and F_p with the state-of-the-art methods for 3D shape recognition on ModelNet40 dataset including 2D image-based methods [41, 41, 46] and 3D methods of both unsupervised learning models [1, 5, 8, 10, 13, 20, 44, 54, 57, 59] and supervised learning models [8, 15, 23, 28, 39, 45, 52, 53]. The setups of our models are same as in subsection 4.3. The comparisons are shown in Table 6.

Compared to other unsupervised feature learning methods in Table 6, our approach achieves the state-of-the-art accuracy on the ModelNet40 shape recognition task with pre-trained F_p and a linear SVM. The performance of F_p trained with both cross-modality and cross-view correspondences is 89.8% which is 0.7% higher than the previous state-of-the-art method. Even trained without using any human-annotated labels, the features learned by our network achieve comparable performance as the supervised methods on the ModelNet40 dataset. Moreover, almost all

Table 6. The comparison with the state-of-the-art methods for 3D shape recognition on ModelNet40 dataset. * indicates the image-based methods.

Unsupervised feature learning		Supervised feature learning	
Network	Acc (%)	Network	Acc (%)
SPH [20]	68.2	PointNet [28]	89.2
LFD [5]	75.5	PointNet++ [39]	90.7
T-L Network [10]	74.4	PointCNN [15]	86.1
VConv-DAE [44]	75.5	DGCNN [53]	**92.2**
Fisher Vector* [41]	78.8	KCNet [45]	91.0
3D-GAN [54]	83.3	KDNet [23]	91.8
Latent-GAN [1]	85.7	MRTNet [8]	91.7
MRTNet-VAE [8]	86.4	SpecGCN [52]	91.5
Contrast-Cluster [58]	86.8	DeCAF* [6]	88.6
FoldingNet [57]	88.4	MVCNN* [46]	90.1
PointCapsNet [59]	88.9		
MultiTask [13]	89.1		
MVI [16]	89.3		
F_p-CM	87.5		
F_{img}-CM-CV*	**89.3**		
F_p-CM-CV	**89.8**		

the existing self-supervised learning methods can learn only 2D image features or only 3D point cloud features, while our method can jointly learn both the discriminative 2D and 3D features that outperform previous state-of-the-art self-supervised learning methods.

5. Conclusion

In this paper, we have proposed a self-supervised learning schema that can jointly learn discriminative 2D and 3D features by using the cross-view and cross-modality correspondences on the 3D point cloud datasets. The learned features from both the 2D image-based network and the 3D point cloud-based graph neural network have been extensively tested across different tasks including multi-view 2D shape recognition, 3D shape recognition, multi-view 2D shape retrieval, 3D shape retrieval, and 3D part-segmentation, showing strong generalization abilities of the learned features. Our results demonstrate a promising direction to learn features by exploiting cross-modality correspondence among different modalities derived from 3D data including mesh, rendered multi-view data, voxel, point cloud, Phong, depth, Silhouette, etc.

Acknowledgement.

This material is partially based upon the work supported by National Science Foundation (NSF) under award number IIS-2041307. We thank Yucheng Chen for the discussions during his visit of the City College of New York.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017.

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9297–9307, 2019.

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[5] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3d model retrieval. In *Computer graphics forum*, volume 22, pages 223–232. Wiley Online Library, 2003.

[6] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

[7] Yi Fang, Jin Xie, Guoxian Dai, Meng Wang, Fan Zhu, Tiantian Xu, and Edward Wong. 3d deep shape descriptor. In *CVPR*, pages 2319–2328, 2015.

[8] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018.

[9] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[10] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016.

[11] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6391–6400, 2019.

[12] Kan Guo, Dongqing Zou, and Xiaowu Chen. 3d mesh labeling via deep convolutional neural networks. *TOG*, 35(1):3, 2015.

[13] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8160–8171, 2019.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993, 2018.

[16] Longlong Jing, Yucheng Chen, Ling Zhang, Mingyi He, and Yingli Tian. Self-supervised modal and view invariant feature learning. *arXiv preprint arXiv:2005.14169*, 2020.

[17] Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2(7):8, 2018.

[18] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *arXiv preprint arXiv:1902.06162*, 2019.

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[20] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *Symposium on geometry processing*, volume 6, pages 156–164, 2003.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *ICCV*, pages 863–872, 2017.

[23] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 863–872, 2017.

[24] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019.

[25] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Advances in Neural Information Processing Systems*, pages 7763–7774, 2018.

[26] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[27] Huan Lei, Naveed Akhtar, and Ajmal Mian. Spherical kernel for efficient graph convolution on 3d point clouds. *arXiv preprint arXiv:1909.09287*, 2019.

[28] Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018.

[29] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9267–9276, 2019.

[30] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[31] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*, pages 922–928. IEEE, 2015.

[32] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[33] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9359–9367, 2018.

[34] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2701–2710, 2017.

[35] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[36] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[38] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *CVPR*, pages 5648–5656, 2016.

[39] 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*, pages 5099–5108, 2017.

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[41] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[42] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *Advances in Neural Information Processing Systems*, pages 12942–12952, 2019.

[43] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[44] Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *European Conference on Computer Vision*, pages 236–250. Springer, 2016.

[45] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4548–4557, 2018.

[46] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[47] Jong-Chyi Su, Matheus Gadelha, Rui Wang, and Subhransu Maji. A deeper look at 3d shape classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[48] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua E Siegel, and Sanjay E Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. *arXiv preprint arXiv:1810.05591*, 2018.

[49] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*, pages 2088–2096, 2017.

[50] Ali Thabet, Humam Alwassel, and Bernard Ghanem. Mortonnet: Self-supervised learning of local features in 3d point clouds. *arXiv preprint arXiv:1904.00230*, 2019.

[51] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019.

[52] Chu Wang, Babak Samari, and Kaleem Siddiqi. Local spectral graph convolution for point set feature learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–66, 2018.

[53] 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(5):1–12, 2019.

[54] 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*, pages 82–90, 2016.

[55] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019.

[56] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[57] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018.

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

[59] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1009–1018, 2019.