Unsupervised Point Cloud Representation Learning With Deep Neural Networks: A Survey

Aoran Xiao, Jiaxing Huang, Dayan Guan, Xiaoqin Zhang, *Senior Member, IEEE*, Shijian Lu, and Ling Shao, *Fellow, IEEE*

***Abstract*—Point cloud data have been widely explored due to its superior accuracy and robustness under various adverse situations. Meanwhile, deep neural networks (DNNs) have achieved very impressive success in various applications such as surveillance and autonomous driving. The convergence of point cloud and DNNs has led to many deep point cloud models, largely trained under the supervision of large-scale and densely-labelled point cloud data. Unsupervised point cloud representation learning, which aims to learn general and useful point cloud representations from unlabelled point cloud data, has recently attracted increasing attention due to the constraint in large-scale point cloud labelling. This paper provides a comprehensive review of unsupervised point cloud representation learning using DNNs. It first describes the motivation, general pipelines as well as terminologies of the recent studies. Relevant background including widely adopted point cloud datasets and DNN architectures is then briefly presented. This is followed by an extensive discussion of existing unsupervised point cloud representation learning methods according to their technical approaches. We also quantitatively benchmark and discuss the reviewed methods over multiple widely adopted point cloud datasets. Finally, we share our humble opinion about several challenges and problems that could be pursued in the future research in unsupervised point cloud representation learning.**

***Index Terms*—3D vision, deep learning, deep neural network, point cloud, pre-training, self-supervised learning, transfer learning, unsupervised representation learning.**

Manuscript received 11 June 2022; revised 20 March 2023; accepted 26 March 2023. Date of publication 29 March 2023; date of current version 4 August 2023. This work was funded in part by the Ministry of Education Singapore, under the Tier-1 scheme with project under Grant RG18/22. It is also supported in part under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contributions from Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU). Recommended for acceptance by M. Salzmann. *(Aoran Xiao and Jiaxing Huang are co-first authors.) (Corresponding authors: Shijian Lu; Xiaoqin Zhang.)*

Aoran Xiao, Jiaxing Huang, and Shijian Lu are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: aoran.xiao@ntu.edu.sg; jiaxing.huang@ntu.edu.sg; shijian.lu@ntu.edu.sg).

Dayan Guan is with the Mohamed bin Zayed University of Artificial Intelligence, Masdar City, Abu Dhabi 7909, UAE (e-mail: dayan.guan@mbzuai.ac.ae).

Xiaoqin Zhang is with the Key Laboratory of Intelligent Informatics for Safety and Emergency of Zhejiang Province, Wenzhou University, Wenzhou, Zhejiang 325035, China (e-mail: zhangxiaoqinnan@gmail.com).

Ling Shao is with the UCAS-Terminus AI Lab, University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: ling.shao@ieee.org).

A project associated with this survey has been built at https://github.com/xiaoaoran/3d_url_survey.

Digital Object Identifier 10.1109/TPAMI.2023.3262786

I. Introduction

3D ACQUISITION technologies have experienced fast development in recent years. This can be witnessed by different 3D sensors that have become increasingly popular in both industrial and our daily lives such as LiDAR sensors in autonomous vehicles, RGB-D cameras in Kinect and Apple devices, 3D scanners in various reconstruction tasks, etc. Meanwhile, 3D data of different modalities such as meshes, point clouds, depth images and volumetric grids, which capture accurate geometric information for both objects and scenes, have been collected and widely applied in different areas such as autonomous driving, robotics, medical treatment, remote sensing, etc.

Point cloud as one source of ubiquitous and widely used 3D data can be directly captured with entry-level depth sensors before triangulating into meshes or converting to voxels. This makes it easily applicable to various 3D scene understanding tasks [1] such as 3D object detection and shape analysis, semantic segmentation, etc. With the advance of deep neural networks (DNNs), point cloud understanding has attracted increasing attention as observed by a large number of deep architectures and deep models developed in recent years [2]. On the other hand, effective training of deep networks requires large-scale human-annotated training data such as 3D bounding boxes for object detection and point-wise annotations for semantic segmentation, which are usually laborious and time-consuming to collect due to 3D view changes and visual inconsistency between human perception and point cloud display. Efficient collection of large-scale annotated point clouds has become one bottleneck for effective design, evaluations, and deployment of deep networks while handling various real-world tasks [3].

Unsupervised representation learning (URL), which aims to learn robust and general feature representations from unlabelled data, has recently been studied intensively for mitigating the laborious and time-consuming data annotation challenge. As Fig. 1 shows, URL works in a similar way to pre-training which learns useful knowledge from unlabelled data and transfers the learned knowledge to various downstream tasks [4]. More specifically, URL can provide useful network initialization with which well-performing network models can be trained with a small amount of labelled and task-specific training data without suffering from much over-fitting as compared with training from random initialization. URL can thus help reduce training data and annotations which has demonstrated great effectiveness in

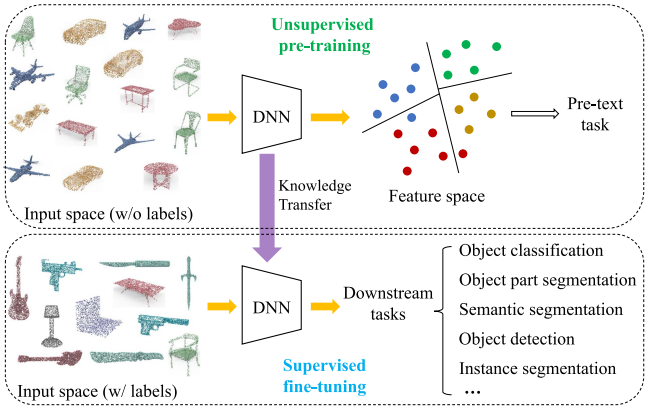


Fig. 1. The general pipeline of unsupervised representation learning on point clouds: Deep neural networks are first pre-trained with unannotated point clouds via unsupervised learning over certain pre-text tasks. The learned unsupervised point cloud representations are then transferred to various downstream tasks to provide network initialization, with which the pre-trained networks are *fine-tuned* with a small amount of annotated task-specific point cloud data.

the areas of natural language processing (NLP) [5], [6], 2D computer vision [7], [8], [9], [10], etc.

Similar to URL from other types of data such as texts and 2D images, URL of point clouds has recently attracted increasing attention in the computer vision research community. A number of URL techniques have been reported which are typically achieved by designing different pre-text tasks such as 3D object reconstruction [11], partial object completion [12], 3D jigsaws solving [13], etc. However, URL of point clouds still lags far behind as compared with its counterparts in NLP and 2D computer vision tasks. For the time being, training from scratch on various target new data is still the prevalent approach in most existing 3D scene understanding development. At the other end, URL from point cloud data is facing increasing problems and challenges, largely due to the lack of large-scale and high-quality point cloud data, unified deep backbone architectures, generalizable technical approaches, as well as comprehensive public benchmarks.

In addition, URL for point clouds is still short of systematic survey that can offer a clear big picture about this new yet challenging task. To fill up this gap, this paper presents a comprehensive survey on the recent progress in unsupervised point cloud representation learning from the perspective of datasets, network architectures, technical approaches, performance benchmarking, and future research directions. As shown in Fig. 2, we broadly group existing methods into four categories based on their pretext tasks, including URL methods using data generation, global and local contexts, multimodality data and local descriptors, more details to be discussed in the ensuing subsections.

The major contributions of this work are threefold:

1) It presents a comprehensive review of the recent development in unsupervised point cloud representation learning. To the best of our knowledge, it is the *first* survey that provides an overview and big picture for this exciting research topic.
2) It studies the most recent progress of unsupervised point cloud representation learning, including a comprehensive benchmarking and discussion of existing methods over multiple public datasets.
3) It shares several research challenges and potential research directions that could be pursued in unsupervised point cloud representation learning.

The rest of this survey is organized as follows: In Section II, we introduce background knowledge of unsupervised point cloud learning including term definition, common tasks of point cloud understanding and relevant surveys to this work. Section III introduces widely-used datasets and their characteristics. Section IV introduces commonly used deep point cloud architectures with typical models that are frequently used for point cloud URL. In Section V we systematically review the methods for point cloud URL. Section VI summarizes and compares the performances of existing methods on multiple benchmark datasets. At last, we list several promising future directions for unsupervised point cloud representation learning in Section VII.

II. Background

A. Basic Concepts

We first define all relevant terms and concepts that are to be used in the ensuing sections.

Point cloud data: A point cloud P is a set of vectors $P = \{p_1, \ldots, p_N\}$ where each vector represents one point $p_i = [C_i, A_i]$. Here, $C_i \in \mathbf{R}^{1\times 3}$ refers to 3D coordinate (x_i, y_i, z_i) of the point, and A_i refers to feature attributes of the point such as RGB values, LiDAR intensity, normal values, etc., which are optional and variational depending on 3D sensors as well as applications.

Supervised learning: Under the paradigm of deep learning, supervised learning aims to train deep network models by using labelled training data.

Unsupervised learning: Unsupervised learning aims to train networks by using unlabelled training data.

Unsupervised representation learning: URL is a subset of unsupervised learning. It aims to learn meaningful representations from data without using any data labels/annotations, where the learned representations can be transferred to different downstream tasks. Some literature alternatively uses the term "self-supervised learning".

Semi-supervised learning: In semi-supervised learning, deep networks are trained with a small amount of labelled data and a large amount of unlabelled data. It aims to mitigate data annotation constraints by learning from a small amount of labelled data and a large amount of unlabelled data that have similar distributions.

Pre-training: Network pre-training learns with certain pre-text tasks over other datasets. The learned parameters are often employed for model initialization for further fine-tuning with various task-specific data.

Transfer learning: Transfer learning aims to transfer knowledge across tasks, modalities or datasets. A typical scenario related to this survey is to perform unsupervised learning for

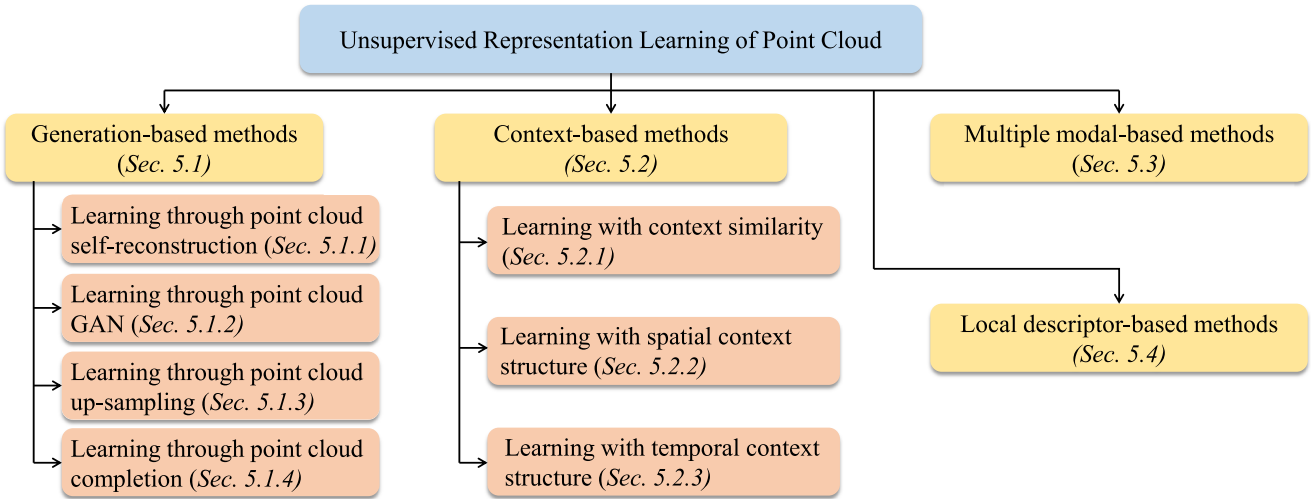


Fig. 2. Taxonomy of existing unsupervised methods in point cloud representation learning.

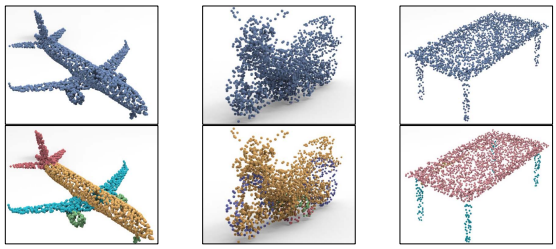
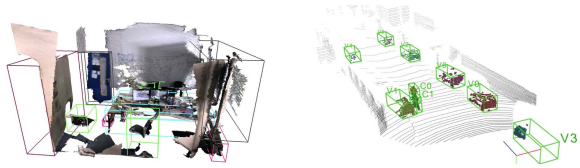


Fig. 3. Illustration of object part segmentation: The first row shows a few object samples including *airplane*, *motorcycle*, and *table* from the ShapeNetPart dataset [14]. The second row shows segmentation ground truth with different parts as highlighted by different colors.



(a) ScanNet-V2 dataset (b) KITTI dataset

Fig. 4. Illustration of 3D bounding boxes in point cloud object detection: The two graphs show 3D bounding boxes in datasets ScanNet-V2 [18] and KITTI [19] which are cropped from [16] and [20], respectively.

pre-training for transferring the learned knowledge from unlabelled data to various downstream networks.

B. Common 3D Understanding Tasks

This subsection introduces common 3D understanding tasks including *object-level tasks* in object classification and object part segmentation and *scene-level tasks* in 3D object detection, semantic segmentation and instance segmentation. These tasks have been widely adopted to evaluate the quality of point cloud representations that are learned via various unsupervised learning methods, which will be discussed in detail in Section VI.

1) Object Classification: Object classification aims to classify point cloud objects into a number of pre-defined categories. Two evaluation metrics are most frequently used: The *overall Accuracy* (OA) represents the averaged accuracy for all instances in the test set; The *mean class accuracy* (mAcc) represents the mean accuracy of all object classes for the test set.

2) Object Part Segmentation: Object part segmentation is an important task for point cloud representation learning. It aims to assign a part category label (e.g., airplane wing, table leg, etc.) to each point as illustrated in Fig. 3. The mean Intersection over Union (mIoU) [15] is the most widely adopted evaluation metric. For each instance, IoU is computed for each part belonging to that object category. The mean of the part IoUs represents the IoU of that object instance. The overall IoU is computed as the average of IoUs over all test instances while category-wise IoU (or class IoU) is calculated as the mean over instances under that category.

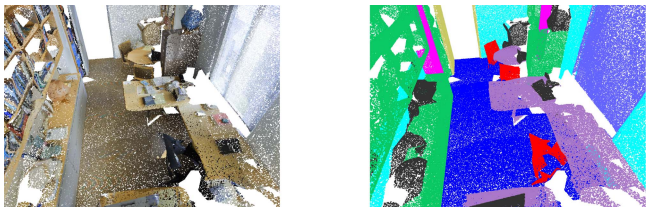
3) 3D Object Detection: 3D object detection on point clouds is a crucial and indispensable task for many real-world applications, such as autonomous driving and domestic robots. The task aims to localize objects in the 3D space, *i.e.* 3D object bounding boxes as illustrated in Fig. 4. The average precision (AP) metric has been widely used for evaluations in 3D object detection [16], [17].

4) 3D Semantic Segmentation: 3D semantic segmentation on point clouds is another critical task for 3D understanding as illustrated in Fig. 5. Different from the object part segmentation that segments point cloud objects, 3D semantic segmentation aims to assign a category label to each point in scene-level point clouds with much higher complexity. The widely adopted evaluation metrics includes OA, mIoU over semantic categories and mAcc.

5) 3D Instance Segmentation: 3D instance segmentation aims to detect and delineate each distinct object of interest in scene-level point clouds as illustrated in Fig. 6. On top of semantic segmentation that considers the semantic category only, instance segmentation assigns each object a unique identity. Mean Average Precision (mAP) has been widely adopted for the quantitative evaluation of this task.

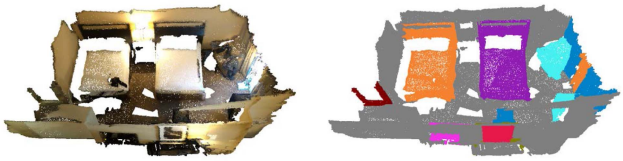
TABLE I
SUMMARY OF COMMONLY USED DATASETS FOR TRAINING AND EVALUATIONS IN PRIOR URL STUDIES WITH POINT CLOUDS

Dataset	Year	#Samples	#Classes	Type	Representation	Label
KITTI [19]	2013	15K frames	8	Outdoor driving	RGB & LiDAR	Bounding box
ModelNet10 [27]	2015	4,899 objects	10	Synthetic object	Mesh	Object category label
ModelNet40 [27]	2015	12,311 objects	40	Synthetic object	Mesh	Object category label
ShapeNet [14]	2015	51,190 objects	55	Synthetic object	Mesh	Object/part category label
SUN RGB-D [28]	2015	5K frames	37	Indoor scene	RGB-D	Bounding box
S3DIS [21]	2016	272 scans	13	Indoor scene	RGB-D	Point category label
ScanNet [18]	2017	1,513 scans	20	Indoor scene	RGB-D & mesh	Point category label & Bounding box
ScanObjectNN [29]	2019	2,902 objects	15	Real-world object	Points	Object category label
ONCE [30]	2021	1M scenes	5	Outdoor driving	RGB & LiDAR	Bounding box



(a) A raw sample (b) Semantic annotations

Fig. 5. Illustration of semantic point cloud segmentation: For the point cloud sample from S3DIS [21] on the left, the graph on the right shows the corresponding ground truth where different categories are highlighted by different colors.



(a) A raw sample (b) Instance annotations

Fig. 6. Illustration of instance segmentation on point clouds: For the point cloud sample from ScanNet-V2 [18] on the left, the graph on the right shows the corresponding ground truth with different instances highlighted by different colors.

C. Relevant Surveys

To the best of our knowledge, this paper is the *first* survey that reviews unsupervised point cloud learning comprehensively. Several relevant but different surveys have been performed. For example, several papers reviewed recent advances for deep supervised learning on point clouds: Ioannidou et al. [22] reviewed deep learning approaches on 3D data; Xie et al. [23] provided a literature review on point cloud segmentation task; Guo et al. [2] provided a comprehensive and detailed survey on deep learning of point cloud for multiple tasks including classification, detection, tracking, and segmentation. In addition, several works reviewed unsupervised representation learning on other data modalities: Jing et al. [24] introduced advances on unsupervised representation learning in 2D computer vision; Liu et al. [25] looked into latest progress about unsupervised representation learning methods in 2D computer vision, NLP, and graph learning; Qi et al. [26] introduced recent progress on small data learning including unsupervised- and semi-supervised methods.

III. POINT CLOUD DATASETS

In this section, we summarize the commonly used datasets for training and evaluating unsupervised point cloud representation learning. As listed in Table I, existing work learns unsupervised point cloud representations mainly from 1) synthetic object datasets including ModelNet [27] and ShapeNet [14], or 2) real scene datasets including ScanNet [18] and KITTI [19]. In addition, various tasks-specific datasets have been collected which can be used for fine-tuning downstream models, such as ScanObjectNN [29], ModelNet40 [27], and ShapeNet [14] for point cloud classification, ShapeNetPart [14] for part segmentation, S3DIS [21], ScanNet [18], or Synthia4D [31] for semantic segmentation, indoor datasets SUNRGB-D [28] and ScanNet [18] as well as outdoor dataset ONCE [30] for object detection.

- *ModelNet10/ModelNet40 [27]:* ModelNet is a synthetic object-level dataset for 3D classification. The original ModelNet provides CAD models represented by vertices and faces. Point clouds are generated by sampling from the models uniformly. ModelNet40 contains 13,834 objects of 40 categories, among which 9,843 objects form the training set and the rest form the test set. ModelNet10 consists of 3,377 samples of 10 categories, which are split into 2,468 training samples and 909 testing samples.
- *ShapeNet [14]:* ShapeNet contains synthetic 3D objects of 55 categories. It was curated by collecting CAD models from online open-sourced 3D repositories. Similar to ModelNet, synthetic objects in ShapeNet are complete, aligned, and with no occlusion or background. Its extension **ShapeNetPart** has 16,881 objects of 16 categories and is represented by point clouds. Each object consists of 2 to 6 parts, and in total there are 50 part categories in the dataset.
- *ScanObjectNN [29]:* ScanObjectNN is a real object-level dataset, where 2,902 3D point cloud objects of 15 categories are constructed from the scans captured in real indoor scenes. Different from synthetic object datasets, point cloud objects in ScanObjectNN are noisy (including background points, occlusions, and holes in objects) and not axis-aligned.
- *S3DIS [21]:* Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset contains over 215 million points scanned from 6 large-scale indoor areas in 3 office buildings, where each area is 6,000 square meters. The scans are represented as point clouds with point-wise semantic labels of 13 object categories.
- *ScanNet-V2 [18]:* ScanNet-V2 is an RGB-D video dataset containing 2.5 million views in more than 1500 scans, which

are captured in indoor scenes such as offices and living rooms and annotated with 3D camera poses, surface reconstructions, as well as semantic and instance labels for segmentation.

• *SUN RGB-D [28]:* SUN RGB-D dataset is a collection of single view RGB-D images collected from indoor environments. There are in total 10,335 RGB-D images annotated with amodal, and 3D oriented object bounding boxes of 37 categories.

• *KITTI [19]:* KITTI is a pioneer outdoor dataset providing dense point clouds from a LiDAR sensor together with other modalities including front-facing stereo images and GPS/IMU data. It provides 200 k 3D boxes over 22 scenes for 3D object detection.

• *ONCE [30]:* ONCE dataset has 1 million LiDAR scenes and 7 million corresponding camera images. There are 581 sequences in total, where 560 sequences are unlabelled and used for unsupervised learning, and 10 sequences are annotated and used for testing. It provides an unsupervised learning benchmark for object detection in outdoor environments.

The publicly available datasets for URL of point clouds are still limited in both data size and scene variety, especially compared with the image and text datasets that have been used for 2D computer vision and NLP research. For example, there are 800 million words in BooksCorpus and 2,500 million words in English Wikipedia that is able to provide comprehensive data sources for unsupervised representation learning in NLP [32]; ImageNet [33] has more than 10 million images for unsupervised visual representation learning. Large-scale and high-quality point cloud data are highly demanded for future research on this topic, and we provide a detailed discussion of this issue in Section VII.

IV. Common Deep Architectures

Over the last decade, deep learning has been playing a more important role in point-cloud processing and understanding. This can be observed by the abundance of deep architectures that have been developed in recent years. Different from traditional 3D vision that transforms point clouds to structures like Octrees [34] or Hashed Voxel Lists [35], deep learning favors more amenable structures for differentiability and/or efficient neural processing which have achieved very impressive performance over various 3D tasks.

At the other end, DNN-based point cloud processing and understanding lags far behind as compared with its counterparts in NLP and 2D computer vision. This is especially true for the task of unsupervised representation learning, largely due to the lack of regular representations in point cloud data. Specifically, word embeddings and 2D images have regular and well-defined structures, but point clouds represented by unordered point sets have no such universal and structural data format.

In this section, we introduce deep architectures that have been explored for the URL of point clouds. Deep learning for point clouds achieved significant progress during the last decade and we see the abundance of 3D deep architectures and 3D models being proposed. However, we do not have universal and ubiquitous "3D backbones" like VGG [36] or ResNet [37] in 2D computer vision. We thus focus on those frequently used architectures in the URL of point clouds in this survey.

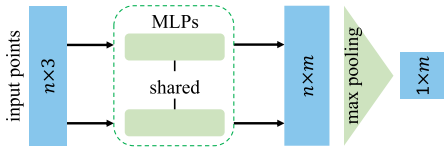


Fig. 7. A simplified architecture of PointNet [15] for point cloud object classification, where parameters n and m denote point number and feature dimension, respectively.

For clarity of description, we group them into five categories broadly, namely, point-based architectures, graph-based architectures, sparse voxel-based architectures, spatial CNN-based architectures, and Transformer-based architectures. Note other deep architectures also exist for various 3D tasks as discussed in [2], such as projection-based networks [38], [39], [40], [41], [42], [43], recurrent neural networks [44], [45], [46], 3D capsule networks [47], etc. However, they were not often employed for the URL task and thus are not detailed in this survey.

A. Point-Based Deep Architectures

Point-based networks were designed to process raw point clouds directly without point data transformations beforehand. Independent point features are usually first extracted by stacking networks with Multi-Layer Perceptrons (MLPs), which are then aggregated into global features with symmetric aggregation functions.

PointNet [15] is a pioneer point-based network as shown in Fig. 7. It stacks several MLP layers to learn point-wise features independently and forwards the learned features to a max-pooling layer to extract global features for permutation invariance. To improve PointNet, Qi et al. proposed PointNet++ [48] to learn local geometry details from the neighborhood of points, where the set abstraction level includes sampling layer, grouping layer, and PointNet layer for learning local and hierarchical features. PointNet++ achieves great success in multiple 3D tasks including object classification and semantic segmentation. By taking PointNet++ as the backbone, Qi et al. designed VoteNet [16], the first point-based 3D object detection network. VoteNet adopts the Hough voting strategy, which generates new points around object centers and groups them with the surrounding points to produce 3D box proposals.

B. Graph-Based Deep Architectures

Graph-based networks treat point clouds as graphs in Euclidean space with vertexes being points and edges capturing neighboring point relations as illustrated in Fig. 8. It works with graph convolution where filter weights are conditioned on edge labels and dynamically generated for individual input samples. This allows to reduce the degrees of freedom in the learned models by enforcing weight sharing and extracting localized features that can capture dependencies among neighboring points.

The Dynamic Graph Convolutional Neural Network (DGCNN) [49] is a typical graph-based network that has been frequently used for URL for point clouds. It is stacked with a graph convolution module named EdgeConv that performs convolution on graph dynamically in the feature space. DGCNN integrates EdgeConv into the basic version of PointNet structures

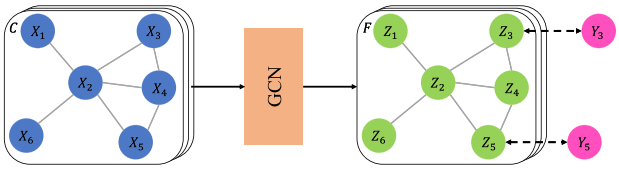


Fig. 8. Schematic depiction of graph convolutional network (GCN): Each graph consists of multiple vertexes representing points X_i or features Z_i (highlighted by circular dots), as well as edges connecting the vertexes representing point relations (shown as black lines). C denotes input channels, F denotes output feature dimensions, and Y_i denotes labels.

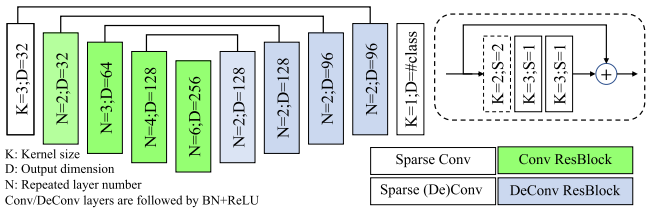


Fig. 9. An illustration of SR-UNet [54] that adopts a unified U-Net [55] architecture for sparse convolution. The graph is reproduced based on [54].

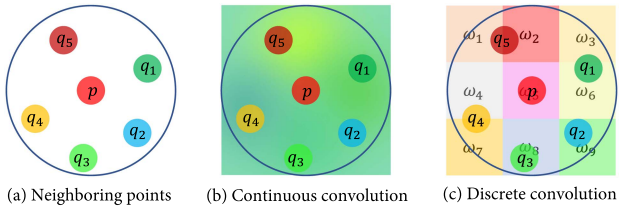


Fig. 10. An illustration of 3D spatial convolution including continuous and discrete convolutions. Parameters p and q_i denote the center point and its neighboring points, respectively. The graph is reproduced based on [2].

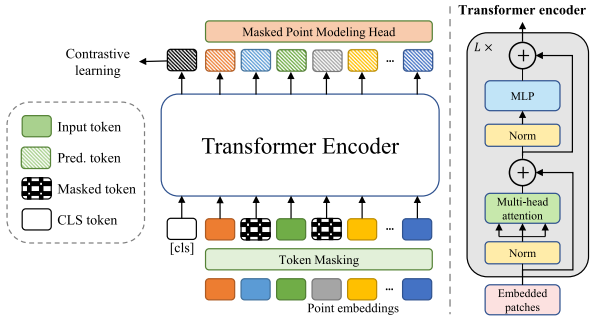


Fig. 11. The architecture of point cloud Transformer that was used for unsupervised pre-training in Point-BERT [57]. More network details can be found in [57]. The figure is reproduced based on [57], [58].

for learning global shape properties and semantic characteristics for point cloud understanding.

C. Sparse Voxel-Based Deep Architectures

The voxel-based architecture v oxelizes point clouds into 3D grids before applying 3D CNN on the volumetric representations. Due to the sparseness of point cloud data, It often involves huge computation redundancy or sacrifices the representation accuracy while processing a large number of points. To overcome this constrain, [50], [51], [52], [53] adopt *sparse tensor* as the basic unit where point clouds are represented with a data list and an index list. Unlike standard convolution operation that employs sliding windows (*im2col* function in PyTorch and TensorFlow) to build the computational pipeline, *sparse convolution* [50] collects all atomic operations including convolution kernel elements and saves them in a *Rulebook* as computation instructions.

Recently, Choy et al. proposed Minkowski Engine [51] that introduces generalized sparse convolution and an auto-differentiation library for sparse tensors. On top of that, Xie et al. [54] adopted a unified U-Net [55] architecture and built a backbone network (SR-UNet as shown in Fig. 9) for unsupervised pre-training. The learned encoder can be transferred to different downstream tasks such as classification, object detection, and semantic segmentation.

D. Spatial CNN-Based Deep Architectures

Spatial CNN-based networks have been developed to extend the capabilities of regular-grid CNNs to analyze irregularly spaced point clouds. They can be divided into continuous and discrete convolutional networks according to the convolutional kernels [2]. As Fig. 10 shows, continuous convolutional networks define the convolutional kernels in a continuous space, where the weights of neighboring points are determined by their spatial distribution relative to the center point. Differently, discrete convolutional networks operate on regular grids and define the convolutional kernels in a discrete space where neighboring points have fixed offsets relative to the center point. One typical example of continuous convolution models is RS-CNN [56] which has been widely adopted for URL of point clouds. Specifically, RS-CNN extracts geometric topology relations among local centers with their surrounding points, and it learns dynamic weights for convolutions.

E. Transformer-Based Deep Architectures

Over the last few years, Transformers have made astounding progress in the research areas of NLP [32], [59] and 2D image processing [58], [60] due to their structural superiority and versatility. They have also been introduced into the area of point cloud processing [57], [61] recently. Fig. 11 shows a standard Transformer architecture for URL of point clouds [57], which contains a stack of Transformer blocks [59] and each block consists of a multi-head self-attention layer and a feed-forward network. The unsupervised pre-trained Transformer encoder can be used for fine-tuning downstream tasks such as object classification and semantic segmentation, etc.

V. Unsupervised Point Cloud Representation Learning

In this section, we review existing URL methods for point clouds. As shown in Fig. 2, we broadly group existing methods into four categories according to their pretext tasks, including generative-based methods, context-based methods, multiple

TABLE II
SUMMARY OF GENERATION-BASED METHODS FOR UNSUPERVISED REPRESENTATION LEARNING OF POINT CLOUDS

Method	Published in	Category	Contribution
VConv-DAE [62]	ECCV 2016	Completion	Learning by predicting missing parts in 3D grids
TL-Net [63]	ECCV 2016	Reconstruction	Learning by 3D generation and 2D prediction
3D-GAN [64]	NeurIPS 2016	GAN	Pioneer GAN for 3D voxels
3D-DescriptorNet [65]	CVPR 2018	Completion	learning with energy-based models for point cloud completion
FoldingNet [66]	CVPR 2018	Reconstruction	learning by folding 3D object surfaces
SO-Net [67]	CVPR 2018	Reconstruction	Performing hierarchical feature extraction on individual points and SOM nodes
Latent-GAN [68]	ICML 2018	GAN	Pioneer GAN for raw point clouds and latent embeddings
MRT [69]	ECCV 2018	Reconstruction	A new point cloud autoencoder with multi-grid architecture
VIP-GAN [70]	AAAI 2019	GAN	Learning by solving multi-views inter-prediction tasks for objects
G-GAN [11]	ICLR 2019	GAN	Pioneer GAN with graph convolution for point clouds
3DCapsuleNet [47]	CVPR 2019	Reconstruction	Learning with 3D point-capsule network
L2G-AE [71]	ACM MM 2019	Reconstruction	Learning by global and local reconstruction of point clouds
MAP-VAE [72]	ICCV 2019	Reconstruction	Learning by 3D reconstruction and half-to-half prediction
PointFlow [73]	ICCV 2019	Reconstruction	Learning by modeling point clouds as a distribution of distributions
PDL [74]	CVPR 2020	reconstruction	A probabilistic framework for point distribution learning
GraphTER [75]	CVPR 2020	Reconstruction	Proposed a graph-based autoencoder for point clouds
SA-Net [76]	CVPR 2020	Completion	Learning by completing point cloud objects with a skip-attention mechanism
PointGrow [77]	WACV 2020	Reconstruction	An autoregressive model that can recurrently generate point cloud samples.
PSG-Net [78]	ICCV 2021	Reconstruction	Learning by reconstruct point cloud objects with seed generation
OcCo [12]	ICCV 2021	Completion	Learning by completing occluded point cloud objects
Point-Bert [57]	CVPR 2022	Reconstruction	Learning for Transformers by recovering masked tokens of 3D objects
Point-MAE [79]	ECCV 2022	Reconstruction	Autoencoder transformer recovers masked parts from input data
Point-M2AE [80]	NeurIPS 2022	Reconstruction	Masked autoencoder with hierarchical point cloud encoding and reconstruction.

modal-based methods, and local descriptor-based methods. With this taxonomy, we sort out existing methods and systematically introduce them in the ensuing subsections of this section.

A. Generation-Based Methods

Generation-based URL methods for point clouds involve the process of generating point cloud objects in training. According to the employed pre-text tasks, they can be further grouped into four subcategories including point cloud self-reconstruction (for generating point cloud objects that are the same as the input), point cloud GAN (for generating fake point cloud objects), point cloud up-sampling (for generating objects with denser point clouds but similar shapes) and point cloud completion (for predicting missing parts from incomplete point cloud objects). The ground truth of these URL methods are point clouds themselves. Hence, these methods require no human annotations and can learn in an unsupervised manner. Table II shows a list of generation-based methods.

1) Learning Through Point Cloud Self-Reconstruction: Networks for self-reconstruction usually encode point cloud samples into representation vectors and decode them back to the original input data, where shape information and semantic structures are extracted during this process. It belongs to one typical URL approach since it does not involve any human annotations. One representative network is *autoencoder* [81] which has an encoder network and a decoder network as illustrated in Fig. 12. The encoder compresses and encodes a point cloud object into a low-dimensional embedding vector (i.e., *codeword*) [66], which is then decoded back to the 3D space by the decoder.

The model is optimized by forcing the final output to be the same as the input. During this process, the encoding is validated and learns by attempting to regenerate the input from the encoding whereas the autoencoder learns low-dimension representations by training the network to ignore insignificant data ("noise") [82]. Permutation invariant losses [83] are widely

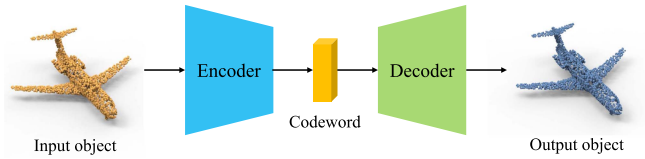


Fig. 12. An illustration of AutoEncoder in unsupervised point cloud representation learning: The *Encoder* learns to represent a point cloud object by a *Codeword* vector while the *Decoder* reconstructs the *Output Object* from the *Codeword*.

adopted as the training objective to describe how the input and output point cloud objects are similar to each other. They can be measured by Chamfer Distance L_{CD} or Earth Mover's Distance L_{EMD} as follows:

$$L_{\text{CD}} = \sum_{p \in P} \min_{p' \in P'} ||p - p'||^2 + \sum_{p' \in P'} \min_{p \in P} ||p' - p||^2 \quad (1)$$

$$L_{\text{EMD}} = \min_{\phi: P \to P'} \sum_{x \in P} ||p - \phi(p)_2||_2 \quad (2)$$

Where P and P' denote input and output point clouds of the same size, $\phi : P \to P'$ is bijection, and p & p' are points.

Self-reconstruction has been one of the most widely adopted pre-text tasks for URL from point clouds over the last decade. By assuming that point cloud representations should be generative in 3D space and predictable from 2D space, Girdhar et al. proposed TL-Net [63] that employs a 3D autoencoder to reconstruct 3D volumetric grids and a 2D convolutional network to learn 2D features from the projected images. Yang et al. designed FoldingNet [66] that introduces a folding-based decoder that deforms a canonical 2D grid onto the underlying 3D object surface of a point cloud object. Li et al. proposed SO-Net [67] that introduces self-organizing map to learn hierarchical features of point clouds via self-reconstruction. Zhao et al. [47] extended the capsule network [84] into 3D point cloud processing and the

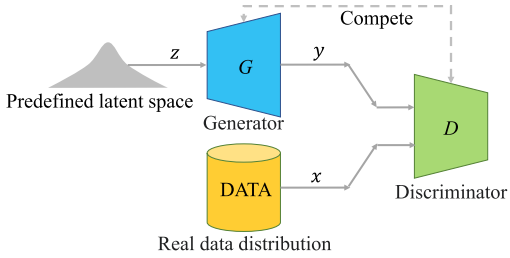


Fig. 13. An illustration of GAN which typically consists of a generator G and a discriminator D that fight with each other during the training process (in the form of a zero-sum game, where one agent's gain is another agent's loss).

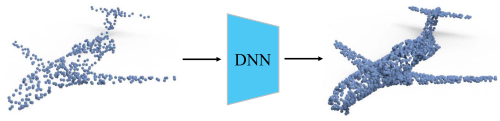


Fig. 14. An illustration of point cloud up-sampling: The network *DNN* learns point cloud representations by solving a pre-text task that reproduces an object with the same geometry but denser point distribution.

designed 3D capsule network can learn generic representations from unstructured 3D data. Gao et al. [75] proposed a graph-based autoencoder that can learn intrinsic patterns of point-cloud structures under both global and local transformations. Chen et al. [85] designed a deep autoencoder that exploits graph topology inference and filtering for extracting compact representations from 3D point clouds.

Several studies explore global and local geometries to learn robust representations from point cloud objects [71], [72]. For example, [71] introduces hierarchical self-attention in the encoder for information aggregation, and a recurrent neural network (RNN) as the decoder for point cloud reconstruction locally and globally. [72] presents MAP-VAE that introduces a half-to-half prediction task that first splits a point cloud object into a front half and a back half and then trains an RNN to predict the back half sequence from the corresponding front half sequence. Several studies instead formulate point cloud reconstruction as a point distribution learning task [73], [74], [77]. For example, [73] presents PointFlow which generates 3D point clouds by modelling the distribution of shapes and that of points given shapes. [74] presents a probabilistic framework that extracts unsupervised shape descriptors via point distribution learning, which associates each point with a Gaussian and models point clouds as the distribution of points. [77] presents an autoregressive model Pointgrow that generates diverse and realistic point cloud samples either from scratch or conditioned on semantic contexts.

Further, several studies learn point cloud representations from different object resolutions [69], [78], [86]. For example, Gadelha et al. [69] designed an autoencoder with a multi-resolution tree structure that learns point cloud representations via coarse-to-fine analysis. Yang et al. [78] proposed an autoencoder with a seed generation module that allows extraction of input-dependent point-wise features in multiple stages with gradually increasing resolution. Chen et al. [86] proposed to learn sampling-invariant features by reconstructing point cloud objects of different resolutions and minimizing Chamfer distances between them.

2) Learning Through Point Cloud GAN: Generative and Adversarial Network (GAN) [87] is a typical deep generative network. As demonstrated in Fig. 13, it consists of a generator and a discriminator. The generator aims to synthesize as realistic data samples as possible while the discriminator tries to differentiate real samples and synthesized samples. GAN thus learns to generate new data with the same statistics as the training set and the modeling can be formulated as a min-max problem:

$$\min_G \max_D L_{GAN} = \log D(x) + \log(1 - D(G(z))), \quad (3)$$

where G is the generator and D represents the discriminator. x and z represent a real sample and a randomly sampled noise vector from a distribution $p(z)$, respectively.

When training GANs for URL of point clouds, the generator learns from either a sampled vector or a latent embedding to generate point cloud instances, while the discriminator tries to distinguish whether input point clouds are from real data distribution or generated data distribution. The two sub-networks fight with each other during the training process and the discriminator learns to extract useful feature representations for point cloud object recognition. The learning process involves no human annotations thus the networks can be trained in an unsupervised learning manner. After that, the learned discriminator is extended into various downstream tasks such as object classification or part segmentation by fine-tuning the model.

Several networks employ GAN for URL for point clouds successfully [11], [64], [68], [88]. For example, Wu et al. [64] proposed the first GAN model applying for 3D voxels. However, the voxelization process either sacrifices the representation accuracy or incurs huge redundancies. Achlioptas et al. proposed Latent-GAN [68] as the first GAN model for raw point clouds. Li et al. [88] further proposed a point cloud GAN model with a hierarchical sampling and inference network that learns a stochastic procedure to generate new point cloud objects. Valsesia et al. [11] designed the first graph-based GAN model to extract localized features from point clouds. These methods evaluated the generalization of the learned representations by fine-tuning them to the high-level downstream 3D tasks.

3) Learning Through Point Cloud Up-Sampling: As shown in Fig. 14, given a set of points, point cloud up-sampling aims to generate a denser set of points with similar geometries. This task requires deep point cloud networks to learn underlying geometries of 3D shapes without any supervision, and the learned representations can be used for fine-tuning in 3D downstream tasks.

Li et al. [89] introduced GAN into the point cloud up-sampling task and presented PU-GAN to learn a variety of point distributions from the latent space by up-sampling points over patches on object surfaces. The generator aims to produce up-sampled point clouds while the discriminator tries to distinguish whether its input point cloud is produced by the generator or the real one. Similar to GANs introduced in Section V-A2, the learned discriminator can be transferred in downstream tasks. Remelli

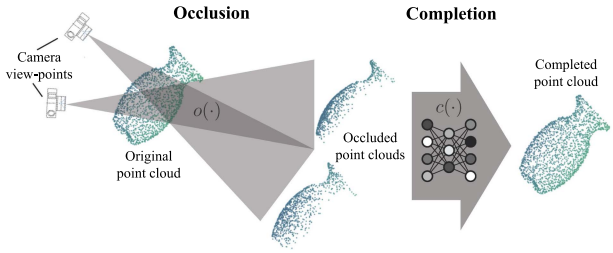


Fig. 15. The pipeline of OcCo [12]. Taking occluded point cloud objects as input, an encoder-decoder model is trained to complete the occluded point clouds, where the encoder learns point cloud representations and the decoder learns to generate complete objects. The learned encoder weights can be used for network initialization for downstream tasks. The figure is from [12] with author's permission.

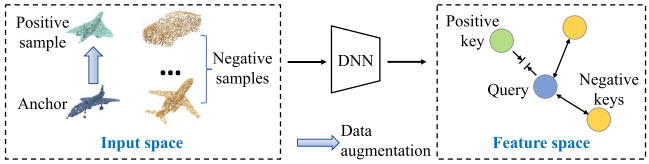


Fig. 16. An illustration of instance contrastive learning that learns locally smooth representations by self-discrimination, which pulls *Query* (from the *Anchor* sample) close to *Positive Key* (from *Positive Samples*) and pushes it away from *Negative Keys* (from *Negative Samples*).

et al. [90] designed an autoencoder that can up-sample sparse point clouds into dense representations. The learned weight of the encoder can also be used as initialization weights for downstream tasks as described in Section V-A1. Though point cloud up-sampling is attracting increasing attention in recent years [89], [91], [92], [93], [94], [95], it is largely evaluated by the quality of generated point clouds while its performance in transfer learning has not been well studied.

4) Learning Through Point Cloud Completion: Point cloud completion is a task to predict arbitrary missing parts based on the rest of the 3D point clouds. To achieve this target, deep networks need to learn inner geometric structures and semantic knowledge of the 3D objects so as to correctly predict missing parts. On top of that, the learned representations can be transferred to downstream tasks. The whole process involves no human annotations and thus belongs to unsupervised representation learning.

Point cloud completion has been an active research area over the past decade [76], [112], [113], [114], [115], [116], [117] with evaluation in different URL benchmarks [12], [62], [65], [76]. A pioneer work VConv-DAE [62] voxelizes point cloud objects into volumetric grids and learns object shape distributions with an autoencoder by predicting the missing voxels from the rest parts. Xie et al. [65] designed 3D-DescriptorNet for probabilistic modeling of volumetric shape patterns. Achlioptas et al. [68] introduced the first DNN for raw point cloud completion which is a point-based network with an encoder-decoder structure. Yuan et al. [113] proposed a Point Completion Network, an autoencoder structured network for learning useful representations by repairing incomplete point cloud objects. Wen et al. [76] proposed SA-Net, which introduces a skip-attention mechanism in the encoder that selectively transfers geometric information from the local regions to the decoder for generating complete point cloud objects. Wang et al. [12] proposed to learn an encoder-decoder model that recovers the occluded points by different camera views as shown in Fig. 15. The encoder parameters are used as initialization for downstream tasks including classification, part segmentation, and semantic segmentation.

Recently, recovering missing parts from *masked* input as the pre-text task of URL has been proved remarkably successful in NLP [5], [6] and 2D computer vision [10]. Such idea has also been investigated in 3D point cloud learning [57], [79], [110], [118]. For example, Yu et al. [57] proposed a Point-BERT paradigm that pre-trains point cloud Transformers through a masked point modeling task. They use a discrete variational autoencoder to generate tokens for object patches and randomly masked out the tokens to train the Transformer to recover the original complete point tokens. The representations learned by Point-BERT can be well transferred to new tasks and domains such as object classification and object part segmentation.

B. Context-Based Methods

Context-based methods are another important category of URL of point clouds that has attracted increasing attention in recent years. Different from generation-based methods that learn representations in a generative way, these methods employ discriminative pre-text tasks to learn different contexts of point clouds including context similarity, spatial context structures, and temporal context structures. The designed pre-text tasks require no human annotations and Table III lists the recent methods.

1) Learning With Context Similarity: This type of method learns unsupervised representations of point clouds by exploring underlying context similarities between samples. A typical approach is *contrastive learning*, which has demonstrated superior performances in both 2D vision [7], [8], [119] and 3D vision [3], [54], [104] in recent years. Fig. 16 provides an illustration of instance-wise contrastive learning. Given one input point cloud object instance as the anchor, its augmented views are defined as the positive samples while other different instances are negative samples. The network learns representations of point clouds by optimizing a self-discrimination task, *i.e.* query (feature of the anchor) should be close to the positive keys (features of positive samples) and faraway from its negative keys (features of negative samples). This learning strategy groups representations of similar samples together in an unsupervised manner and helps networks to learn semantic structures from unlabelled data distribution. The InfoNCE loss [120] defined below and its variants are often employed as the objective function in training:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp\left(q \cdot k_+/\tau\right)}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)}, \tag{4}$$

where q is encoded query, $\{k_0, k_1, k_2, \ldots\}$ are keys with k_+ being the positive key, τ is a temperature hyper-parameter that controls how the distribution concentrates.

Similar to generation-based methods, different contrastive learning methods [99], [100], [102], [121], [122] have been proposed to learn representations on *synthetic single objects*. For

TABLE III
SUMMARY OF CONTEXT-BASED METHODS FOR UNSUPERVISED REPRESENTATION LEARNING OF POINT CLOUDS

Method	Published in	Category	Contribution
MultiTask [96]	ICCV 2019	Hybrid	Learning by clustering, reconstruction, and self-supervised classification
Jigsaw3D [13]	NeurIPS 2019	Spatial-context	Learning by solving 3D jigsaws
Constrast&Cluster [97]	3DV 2019	Hybrid	Learning by contrasting and clustering with GNN
GLR [98]	CVPR 2020	Hybrid	Learning by global-local reasoning for 3D objects
Info3D [99]	ECCV 2020	Context-similarity	Learning by contrasting global and local parts of objects
PointContrast [54]	ECCV 2020	Context-similarity	Learning by contrasting different views of scene point clouds
ACD [100]	ECCV 2020	Context-similarity	Learning by contrasting convex components decomposed from 3D objects
Rotation3D [101]	3DV 2020	Spatial-context	Learning by predicting rotation angles
HNS [102]	ACM MM 2021	Context-similarity	Learning by contrasting local patches of 3D objects with hard negative sampling
CSC [3]	CVPR 2021	Context-similarity	Techniques to improve contrasting scene point cloud views
STRL [1]	ICCV 2021	Temporal-context	Learning spatio-temporal data invariance from point cloud sequences
RandomRooms [103]	ICCV 2021	Context-similarity	Constructing pseudo scenes with synthetic objects for contrastive learning
DepthContrast [104]	ICCV 2021	Context-similarity	Joint contrastive learning with points and voxels
SelfCorrection [105]	ICCV 2021	Hybrid	Learning by distinguishing and restoring destroyed objects
PC-FractalDB [106]	CVPR 2022	Context-similarity	Leveraging fractal geometry to generate high-quality pre-training data
4dcontrast [107]	ECCV 2022	Temporal-context	Learning by contrasting dynamic correspondences from 3D scene sequences
DPCo [108]	ECCV 2022	Context-similarity	A unified contrastive-learning framework for point cloud pre-training
ProposalContrast [109]	ECCV 2022	Context-similarity	Pre-training 3D detectors by contrasting region proposals
MaskPoint [110]	ECCV 2022	Context-similarity	Learning by discriminating masked object points and sampled noise points
FAC [111]	CVPR 2023	Context-similarity	Learning by contrasting between grouped foreground and background

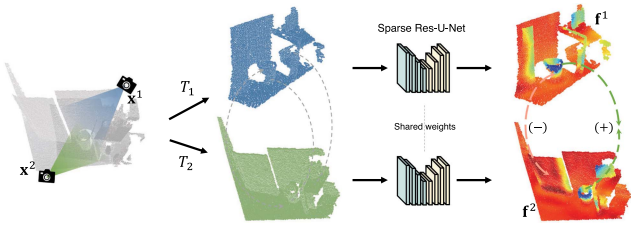


Fig. 17. The pipeline of PointContrast [54]: Two scans x^1 and x^2 of the same scene captured from two different viewpoints are transformed by T_1 and T_2 for data augmentation. The correspondence mapping between the two views is computed to minimize the distance for matched point features and maximize the distance for unmatched point features for contrastive learning. The graph is extracted from [54] with authors' permission.

example, Sanghi et al. [99] proposed to learn useful feature representations by maximizing mutual information between synthetic objects and their local parts. Wang et al. [121] proposed a hybrid contrastive learning strategy that uses objects of different resolutions for instance-level contrast for capturing hierarchical global representations and simultaneously contrasted points and instances for learning local features. Gadelha et al. [100] decompose 3D objects into convex components and construct positive pairs among the same components and negative pairs among different components for contrastive learning. Du et al. [102] introduced a hard negative sampling strategy into the contrastive learning between instances and local parts. Besides, Rao et al. [98] unified contrastive learning, normal estimation, and self-reconstruction into the same framework and formulated a multi-task learning method.

Recently, Xie et al. proposed PointContrast [54], a contrastive learning framework that learns representations of *scene* point clouds as illustrated in Fig. 17. The work shows, for the first time, that network weights pre-trained on 3D scene partial frames can lead to performance boosts when fine-tuned on multiple 3D high-level tasks including object classification, semantic segmentation, and object detection. Firstly, dense correspondences are extracted between two aligned views of ScanNet [18] to build point pairs and point-level contrastive learning is then conducted with a unified backbone (SR-UNet). Finally, the learned model was transferred to multiple downstream 3D tasks including classification, semantic segmentation, and object detection with consistent performance gains.

Since PointContrast brought new insights that the unsupervised representation learned from scene-level point clouds can generalize across domains and boost high-level scene understanding tasks, several unsupervised pre-training works are proposed for scene-level 3D tasks. Considering that PointContrast focuses on point-level alignment without capturing spatial configurations and contexts in scenes, Hou et al. [3] integrated spatial contexts into the pre-training objective by partitioning the space into spatially inhomogeneous cells for correspondence matching. Hou et al. [123] built a multi-modal contrastive learning framework that models 2D multi-view correspondences as well as 2D-3D correspondences with geometry-to-image alignment. While the aforementioned works [3], [54], [123] require 3D data captured from multiple camera views, Zhang et al. [104] proposed DepthContrast that can work with single-view data. Instead of using real point clouds as previous methods, Rao et al. [103] generated synthetic scenes and objects from ShapeNet [14] for network pre-training.

Another unsupervised approach to learn context similarity is *clustering*. In this approach, samples are first grouped into clusters by clustering algorithms such as K-Means [124] and each sample is assigned a cluster ID as pseudo-label. Then networks are trained in a supervised manner to learn semantic structures of data distribution. The learned parameters are used for model initialization for fine-tuning various downstream tasks. A typical example is DeepClustering [125] which is the first unsupervised clustering method for 2D visual representation learning. However, no prior studies adopted a purely clustering strategy for URL of point clouds. Instead, hybrid approaches are proposed by integrating clustering with other unsupervised learning approaches (e.g., self-reconstruction [96] or contrastive learning [97]) for learning more robust representations.

2) Learning With Spatial Context Structure: Point clouds with spatial coordinates provides accurate geometric description

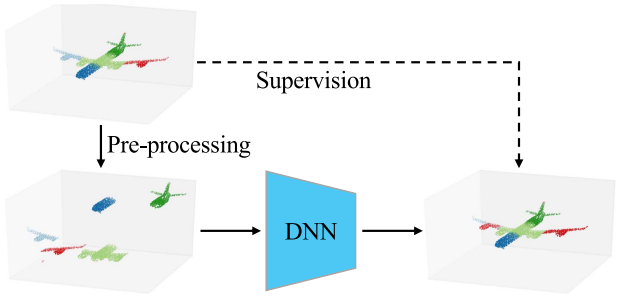


Fig. 18. The pipeline of 3DJigsaw [13]: An object is split into voxels where each point is assigned with a voxel label. The split voxels are randomly rearranged via pre-processing, and a deep neural network is trained to predict the voxel label for each point. The graph is reproduced based on [13].

of 3D shapes of objects and scenes. The rich spatial contexts in point clouds can be exploited in pre-text tasks for URL. For example, networks can be trained to sort out the relation of different object parts. Likewise, the learned parameters can be used for model initialization for downstream tasks. Since no human annotations are required in training, the key is to design effective pre-text tasks to exploit spatial contexts as URL objectives.

The method Jigsaw3D [13] proposed by Sauder et al. is one of the pioneer works that use spatial context for URL of point clouds. As illustrated in Fig. 18, objects are first split into voxels where each point is assigned a voxel label. The network is then fed with randomly rearranged point clouds and optimized by predicting correct voxel label for each point. During the training, the network aims to extract spatial relations and geometric information from point clouds. In their following work [126], another pre-text task was designed to predict one of ten spatial relationships of two local parts from the same object. Inspired by the 2D method that predicts image rotations [127], Poursaeed et al. [101] proposed to learn representations by predicting rotation angles of 3D objects. Thabet et al. [128] designed a pre-text task that predicts the next point in a point sequence defined by Morton-order Space Filling Curve. Chen et al. [105] proposed to learn the spatial context of objects by distinguishing the distorted parts of a shape from the correct ones. Sun et al. [129] introduced a mix-and-disentangle task to exploit spatial context cues.

3) Learning With Temporal Context Structure: Point cloud sequence is a common type of point cloud data that consists of consecutive point cloud frames. For example, there are indoor point cloud sequences transformed from RGB-D video frames [18] and LiDAR sequential data [19], [130], [131] with continuous point cloud scans with each scan collected by one sweep of LiDAR sensors. Point cloud sequences contain rich temporal information that can be extracted by designing pre-text tasks and used as supervision signals to train DNNs. The learned representations can be transferred to downstream tasks.

Recently, Huang et al. [1] proposed a Spatio-Temporal Representation Learning (STRL) framework as illustrated in Fig. 19. STRL extends BYOL [8] from 2D vision to 3D vision and extracts spatial and temporal representation from point clouds. It treats two neighboring point cloud frames as positive pairs and minimizes the mean squared error between the learned feature representations of sample pairs. Chen et al. [107] exploit

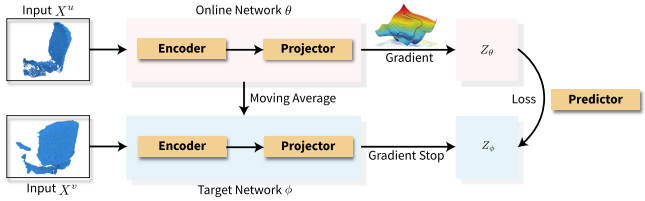


Fig. 19. The pipeline of STRL [1]: An *Online Network* learns spatial and temporal structures from two neighbouring point cloud frames X^u and X^v. The figure is adopted from [1] with authors' permission.

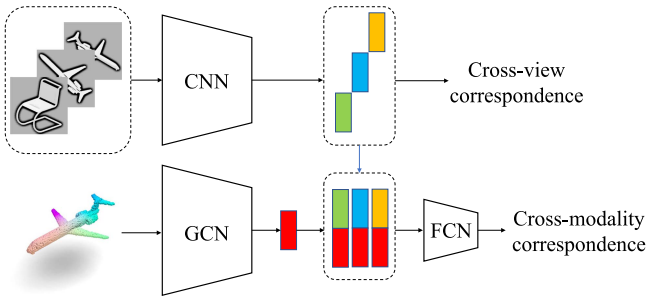


Fig. 20. The pipeline CMCV [4]: CMCV employs a 2D CNN to extract 2D features from rendered views of 3D objects and a 3D GCN to extract 3D features from point clouds. The two types of features are concatenated by a two-layer fully connected network (FCN) to predict cross-modality correspondences. The graph is reproduced based on [4].

synthetic 3D shapes moving in static 3D environments to create dynamic scenarios and sample pairs in the temporal order. They conduct contrastive learning to learn 3D representations with dynamic understanding.

Unsupervised learning with temporal context structures has proved its effectiveness in both 2D computer vision tasks [132], [133], [134], [135] and 3D computer vision tasks [1], [107]. As discussed in Section VII, this direction is very promising but more research is needed for better harvesting the temporal contextual information.

C. Multiple Modal-Based Methods

Different modalities such as images [19] and natural language descriptions [136] can provide additional information for point-cloud data. Modeling relationships across modalities can be designed as pre-text tasks for URL which helps networks to learn more robust and comprehensive representations. Likewise, the learned parameters can be used as initialization weights for various downstream tasks.

Several recent work [4], [137] exploits the correspondences across 3D point cloud objects and 2D images for URL. For example, Jing et al. [4] render 3D objects with different camera views into 2D images for learning from multi-modality data. As Fig. 20 shows, they employ a 2D CNN and a 3D GCN to extract image features and point cloud features, respectively, and then conduct contrastive learning on intra-modal correspondences and cross-modal correspondences. Their study shows that both pre-trained 2D CNN and 3D GCN achieved better classification as compared with random initialization. Differently, Wang et al. [138] project point clouds into colored images and then feed them into an *image pre-trained* model with frozen weights to

TABLE IV
COMPARING LINEAR SHAPE CLASSIFICATION ON MODELNET10 AND MODELNET40 [27]: LINEAR SVM CLASSIFIERS ARE TRAINED WITH REPRESENTATIONS LEARNED BY DIFFERENT UNSUPERVISED METHODS. ACCURACY HIGHLIGHTED BY * WAS OBTAINED BY PRE-TRAINING WITH MULTI-MODAL DATA. [T] DENOTES MODELS WITH MODIFIED TRANSFORMERS. [ST] DENOTES MODELS WITH STANDARD TRANSFORMERS

Method	Year	Pre-text task	Backbone	Pre-train dataset	ModelNet10	ModelNet40
Supervised learning	2017	N.A.	PointNet [15]	N.A.	-	89.2
	2017		PointNet++ [48]		-	90.7
	2019		DGCNN [49]		-	93.5
	2019		RSCNN [56]		-	93.6
	2021		[T]PointTransformer [144]		-	93.7
	2022		[ST]Transformer [57]		-	91.4
SPH [145]	2003	Generation	-	ShapeNet	79.8	68.2
LFD [146]	2003	Generation	-	ShapeNet	79.9	75.5
TL-Net [63]	2016	Generation	-	ShapeNet	-	74.4
VConv-DAE [62]	2016	Generation	-	ShapeNet	80.5	75.5
3D-GAN [64]	2016	Generation	-	ShapeNet	91.0	83.3
3D DescriptorNet [65]	2018	Generation	-	ShapeNet	-	92.4
FoldingNet [66]	2018	Generation	-	ModelNet40	91.9	84.4
FoldingNet [66]	2018	Generation	-	ShapeNet	94.4	88.4
Latent-GAN [68]	2018	Generation	-	ModelNet40	92.2	87.3
Latent-GAN [68]	2018	Generation	-	ShapeNet	95.3	85.7
MRTNet [69]	2018	Generation	-	ShapeNet	86.4	-
VIP-GAN [70]	2019	Generation	-	ShapeNet	94.1	92.0
3DCapsuleNet [47]	2019	Generation	-	ShapeNet	-	88.9
PC-GAN [88]	2019	Generation	-	ModelNet40	-	87.8
L2G-AE [71]	2019	Generation	-	ShapeNet	95.4	90.6
MAP-VAE [72]	2019	Generation	-	ShapeNet	94.8	90.2
PointFlow [73]	2019	Generation	-	ShapeNet	93.7	86.8
MultiTask [96]	2019	Hybrid	-	ShapeNet	-	89.1
Jigsaw3D [13]	2019	Context	PointNet	ShapeNet	91.6	87.3
Jigsaw3D [13]	2019	Context	DGCNN	ShapeNet	94.5	90.6
ClusterNet [97]	2019	Context	DGCNN	ShapeNet	93.8	86.8
CloudContext [126]	2019	Context	DGCNN	ShapeNet	94.5	89.3
NeuralSampler [90]	2019	Generation	-	ShapeNet	95.3	88.7
PointGrow [77]	2020	Generation	-	ShapeNet	85.8	-
Info3D [99]	2020	Context	PointNet	ShapeNet	-	89.8
Info3D [99]	2020	Context	DGCNN	ShapeNet	-	91.6
ACD [100]	2020	Context	PointNet++	ShapeNet	-	89.8
PDL [74]	2020	Generation	-	ShapeNet	-	84.7
GLR [98]	2020	Hybrid	PointNet++	ShapeNet	94.8	92.2
GLR [98]	2020	Hybrid	RSCNN	ShapeNet	94.6	92.2
SA-Net-cls [76]	2020	Generation	-	ShapeNet	-	90.6
GraphTER [75]	2020	Generation	-	ModelNet40	-	89.1
Rotation3D [101]	2020	Context	PointNet	ShapeNet	-	88.6
Rotation3D [101]	2020	Context	DGCNN	ShapeNet	-	90.8
MID [121]	2020	Context	HRNet	ShapeNet	-	90.3
GTIF [85]	2020	Generation	HRNet	ShapeNet	95.9	89.6
HNS [102]	2021	Context	DGCNN	ShapeNet	-	89.6
ParAE [147]	2021	Generation	PointNet	ShapeNet	-	90.3
ParAE [147]	2021	Generation	DGCNN	ShapeNet	-	91.6
CMCV [4]	2021	Multi-modal	DGCNN	ShapeNet	-	89.8*
GSIR [86]	2021	Context	DGCNN	ModelNet40	-	90.4
STRL [1]	2021	Context	PointNet	ShapeNet	-	88.3
STRL [1]	2021	Context	DGCNN	ShapeNet	-	90.9
PSG-Net [78]	2021	Generation	PointNet++	ShapeNet	-	90.9
SelfCorrection [105]	2021	Hybrid	PointNet	ShapeNet	93.3	89.9
SelfCorrection [105]	2021	Hybrid	RSCNN	ShapeNet	95.0	92.4
OcCo [12]	2021	Generation	[ST]Transformer	ShapeNet	-	92.1
CrossPoint [137]	2022	Multi-modal	PointNet	ShapeNet	-	89.1*
CrossPoint [137]	2022	Multi-modal	DGCNN	ShapeNet	-	91.2*
Point-BERT [57]	2022	Generation	[ST]Transformer	ShapeNet	-	93.2
Point-MAE [79]	2022	Generation	[ST]Transformer	ShapeNet	-	93.8

extract representative features for downstream tasks. However, how to learn unsupervised point cloud representations with other modalities such as text descriptions and audio data remains an under-explored field. We expect more studies in this promising research direction.

D. Local Descriptor-Based Methods

The aforementioned methods aim to learn semantic structures of point clouds for high-level understanding, while the local descriptor-based methods focus on learning representations for low-level tasks. For example, Deng et al. [139] introduced PPF-FoldNet that extracts rotation-invariant 3D local descriptors for 3D matching [140]. Several works [141], [142] exploit non-rigid shape correspondence extraction as pre-text tasks for URL of point clouds, aiming to find the point-to-point correspondence of two deformable 3D shapes. Jiang et al. [143] explore unsupervised 3D registration for finding the optimal rigid transformation that can align the source point cloud to the target precisely.

The performances of existing local descriptor-based methods are mainly evaluated on low-level tasks. However, how to adapt the learned feature representations toward other high-level tasks is rarely discussed. We expect more related research in the future.

E. Pros and Cons

Generation-Based Methods. have been extensively studied in 3D URL, thanks to their ability to recover the original data distribution without assuming any downstream tasks. However, most existing research focuses on object-level point clouds, characterized by limited point numbers and data variability, restricting their applicability to object classification and part segmentation tasks. Additionally, these methods demonstrate limited effectiveness in scene-level tasks, such as 3D object detection and semantic segmentation, due to the difficulty of generating scene-level point clouds with complex distribution, rich noises and sparsity variation, and various occlusions. Nonetheless, generation-based methods achieve very impressive progress in 2D images [10] recently, demonstrating their great potential for handling 3D point-cloud data. More efforts are expected in scene-level tasks as well as various downstream applications.

Context-Based Methods. have recently become a prevalent approach in scene-level tasks, such as 3D semantic segmentation, 3D instance segmentation, and 3D object detection, thanks to their ability in addressing complex real-world data. However, they are still facing several challenges. The first is hard-example mining which is crucial to effective contrastive learning. Beyond that, designing effective self-supervision is also challenging for context-based methods, especially while considering generalization across various tasks and applications.

Multiple Modal-Based Methods. allow leveraging additional data modalities for enriching the distribution of point clouds. Pair-wise correspondences between point clouds and other data modalities also offer additional supervision, thereby enhancing the learned unsupervised point cloud representations. However, multi-modality methods are still facing several challenges. For example, acquiring large-scale pair-wise data is often a non-trivial task, and so does the design of effective cross-domain tasks. In addition, how to learn an effective homogeneous representation space across multiple modalities remains a very open research problem.

Local Descriptor-Based Methods. offer distinct advantages in capturing detailed spatial cues and exploiting low-level position information. However, these methods are limited in their ability of transferring learned representations to high-level recognition models, which restricts their application scope in more complex and abstract recognition tasks.

VI. Benchmark Performances

We benchmark representative 3D URL methods with two widely adopted evaluation metrics. The benchmarking is performed over public point-cloud data, where all performances are extracted from the corresponding papers.

A. Evaluation Criteria

There are two metrics that have been widely adopted for evaluating the quality of the learned unsupervised point-cloud representations.

- *Linear classification* first applies a pre-trained unsupervised model to extract features from certain labelled data. It then trains a supervised linear classifier with the extracted features together with the corresponding labels, where the quality of the pre-learned unsupervised representations is evaluated by the performance of the trained linear classifier over test data. Hence, the linear classification can be viewed as a type of representation learning metric which provides *cluster analysis* in an implicit way.
- *Fine-tuning* optimizes a pre-trained unsupervised model using labelled data from downstream tasks. It can assess the quality of the pre-learned unsupervised representations by evaluating the performance of the fine-tuned model over downstream test data, *i.e.* how much performance gains could be obtained by unsupervised pre-training compared to the random initialization.

Note URL can be evaluated with other quantitative metrics. For example, *reconstruction error* [66] can tell how well the learned representations encode the raw point clouds. Different clustering metrics such as *Normalized Mutual Information* [96] could complement the linear-classification metric. However, these metrics are mostly task-specific, e.g., the reconstruction error may not evaluate the representation of scene-level point clouds well due to their inherent noise, occlusion, and sparsity. In fact, few generic metrics can directly and explicitly evaluate the quality of the learned 3D unsupervised representations despite its critical importance to 3D URL studies. More research along this direction is needed to advance this research field further.

Beyond quantitative metrics, unsupervised feature representations can be evaluated in a qualitative manner. For example, t-SNE (t-Distributed Stochastic Neighbor Embedding) [148] has been widely adopted to compress the dimension of the learned feature representations and visualize the compressed feature embeddings.

B. Object-Level Tasks

1) Object Classification: Object classification is the most widely used task in evaluations since the majority of existing works learn point cloud representations on object-level point cloud datasets. As described in Section VI-A, both two types of protocols are widely adopted including the linear classification protocol and the fine-tuning protocol.

Table IV summarizes the performance of the linear classification by existing methods. Specifically, linear classifiers are trained with the representations learned by different unsupervised methods on the ShapeNet or ModelNet40 dataset, and the classification results over the testing set over ModelNet10 and ModelNet40 are reported. For comparison, we also list supervised learning performances of the same backbone models over the same datasets. It can be seen that the performances of unsupervised learning methods keep improving and some methods have even surpassed supervised learning methods, demonstrating the effectiveness and great potential of URL of point clouds.

TABLE V
COMPARISONS OF UNSUPERVISED PRE-TRAINING PERFORMANCE OVER THE OBJECT CLASSIFICATION DATASETS MODELNET40 AND OBJ-BG SPLIT IN SCANOBJECNN. PERFORMANCE NUMBERS ARE PRESENTED IN THE FORMAT OF "A/B," WITH "A" INDICATING TRAINING CLASSIFICATION MODELS FROM SCRATCH WITH RANDOM INITIALIZATION AND "B" INDICATING FINE-TUNING CLASSIFICATION MODELS THAT ARE INITIALIZED WITH UNSUPERVISED PRE-TRAINED MODELS. PERFORMANCE UNDER "A" MAY VARY DUE TO DIFFERENT IMPLEMENTATIONS AS REPORTED IN THE CORRESPONDING PAPERS

Method	Backbone	ModelNet40	ScanObjectNN
Jigsaw3D [13]	PointNet [15]	89.2/89.6(+0.4)	73.5/76.5(+3.0)
Info3D [99]	PointNet [15]	89.2/90.2(+1.0)	-/-
SelfCorrection [105]	PointNet [15]	89.1/90.0(+0.9)	-/-
OcCo [12]	PointNet [15]	89.2/90.1(+0.9)	73.5/80.0(+6.5)
ParAE [147]	PointNet [15]	89.2/90.5(+1.3)	-/-
Jigsaw3D [13]	PCN [113]	89.3/89.6(+0.3)	78.3/78.2(-0.1)
OcCo [12]	PCN [113]	89.3/90.3(+1.0)	78.3/80.4(+2.1)
GLR [98]	RSCNN [56]	91.8/92.2(+0.5)	-/-
SelfCorrection [105]	RSCNN [56]	91.7/93.0(+1.3)	-/-
Jigsaw3D [13]	DGCNN [49]	92.2/92.4(+0.2)	82.4/82.7(+0.3)
Info3D [99]	DGCNN [49]	93.5/93.0(-0.5)	-/-
OcCo [12]	DGCNN [49]	92.5/93.0(+0.5)	82.4/83.9(+1.6)
ParAE [147]	DGCNN [49]	92.2/92.9(+0.7)	-/-
STRL [1]	DGCNN [49]	92.2/93.1(+0.9)	-/-
OcCo [12]	Transformer [57]	91.2/92.2(+1.0)	79.9/84.9(+5.0)
Point-BERT [57]	Transformer [57]	91.2/93.4(+2.2)	79.9/87.4(+7.5)

Table V lists fine-tuning performance on the ModelNet40 and ScanObjectNN datasets. We can see that classification models initialized with unsupervised pre-trained weights always achieve better classification performances as compared with random initialization, regardless of backbone architectures. On the other hand, the performance gaps are still limited, largely due to the limited size and diversity of the pre-training datasets (i.e., ShapeNet and ModelNet40) and the simplicity of existing backbone models. In comparison, thanks to the much larger pre-training datasets ImageNet [33] and the more powerful backbone network ResNet [37], the state-of-the-art methods for unsupervised pre-training of 2D images are able to achieve more significant performance gains in the classification task. As discussed in Section VII, we expect more diverse datasets and more advanced and generous backbone models that can set stronger foundations for this field.

2) Object Part Segmentation: Table VI presents the benchmarking of object part segmentation on the ShapeNetPart dataset [14] using the linear classification protocol (i.e., "Unsup." in Table VI) and the fine-tuning protocol (i.e., "Trans." in Table VI) as described in Section VI-A. As the table shows, the performance gaps between unsupervised and supervised learning (i.e., "Unsup." vs. "Sup.") are decreasing. In addition, unsupervised pre-training achieves better performance in most cases under the fine-tuning protocol (i.e., "Trans." vs. "Sup."), though the improvement is still limited.

C. Scene-Level Tasks

As discussed in Section V-B, unsupervised pre-training in scene-level tasks has recently become prevalent due to its enormous potential in various applications. This comes with a series of 3D URL studies that investigate the effectiveness of pre-training over different scene-level point cloud datasets. We provide a comprehensive benchmarking of these methods with respect to different 3D tasks.

TABLE VI
COMPARISON OF 3D URL METHODS FOR SHAPE PART SEGMENTATION OVER SHAPENETPART [14]. **"UNSUP."** DENOTES LINEAR CLASSIFICATION OF THE LEARNED UNSUPERVISED POINT FEATURES. **"TRANS."** IS PRESENTED IN A FORMAT OF "A/B," WHERE "A" IS OBTAINED WITH SEGMENTATION MODELS TRAINED FROM SCRATCH WITH RANDOM INITIALIZATION, AND "B" IS OBTAINED BY FINE-TUNING SEGMENTATION MODELS THAT ARE INITIALIZED WITH UNSUPERVISED PRE-TRAINED MODELS. WE ALSO PROVIDE SUPERVISED PERFORMANCES (**"SUP."**) OF DIFFERENT BACKBONE MODELS WITH RANDOM INITIALIZATION (EXTRACTED FROM THE ORIGINAL PAPERS)

URL Method	Type	Backbone	class mIoU	instance mIoU
N.A.	Sup.	PointNet	80.4	83.7
	Sup.	PointNet++	81.9	85.1
	Sup.	DGCNN	82.3	85.1
	Sup.	RSCNN	84.0	86.2
	Sup.	Transformer	83.4	85.1
Latent-GAN [68]	Unsup.	-	57.0	-
MAP-VAE [72]	Unsup.	-	68.0	-
CloudContext [126]	Unsup.	DGCNN	-	81.5
GraphTER [75]	Unsup.	-	78.1	81.9
MID [121]	Unsup.	HRNet	83.4	84.6
HNS [102]	Unsup.	DGCNN	79.9	82.3
CMCV [4]	Unsup.	DGCNN	74.7	80.8
SO-Net [67]	Trans.	SO-Net	-/-	84.6/84.9(+0.3)
Jigsaw3D [13]	Trans.	DGCNN	82.3/83.1(+0.8)	85.1/85.3(+0.2)
MID [121]	Trans.	HRNet	84.6/85.2(+0.6)	85.5/85.8(+0.3)
CMCV [4]	Trans.	DGCNN	77.6/79.1(+1.5)	83.0/83.7(+0.7)
OcCo [12]	Trans.	PointNet	82.2/83.4(+1.2)	-/-
OcCo [12]	Trans.	DGCNN	84.4/85.0(+0.6)	-/-
OcCo [12]	Trans.	Transformer	83.4/83.4(+0.0)	85.1/85.1(+0.0)
Point-BERT [57]	Trans.	Transformer	83.4/84.1(+0.7)	85.1/85.6(+0.5)

Tables VII and VIII show the performances of semantic segmentation on the S3DIS [21] dataset. We summarized them separately since different fine-tuning setups have been used in prior works. In Table VII, the unsupervised pre-trained DGCNN is fine-tuned on every *single area* of S3DIS and tested on either Area 5 (the upper part of table) or Area 6 (the lower part of the table). Table VIII instead shows the performance of fine-tuning different segmentation networks with the *whole* dataset by following the one-fold (in the upper part of the table) and six-fold cross-validation setups (in the lower part of the table), respectively.

We also summarize existing works that handle unsupervised pre-training for object detection. Tables IX and X show their performances over indoor datasets including SUN RGB-D [28] and ScanNet-V2 [18] as well as outdoor LiDAR dataset ONCE [30], respectively. In addition, several works investigated unsupervised pre-training for instance segmentation. We summarize their performance over S3DIS [21] and ScanNet-V2 [18] in Table XI.

It is inspiring to see that unsupervised learning representation can generalize across domains and boost performances over multiple high-level 3D tasks as compared with training from scratch. These experiments demonstrate the huge potential of URL of point clouds in saving expensive human annotations. However, the improvements are still limited and we expect more research in this area.

VII. FUTURE DIRECTION

URL of point clouds has achieved significant progress during the last decade. We share several potential future research directions of this research field in this section.

TABLE VII
SEMANTIC SEGMENTATION ON S3DIS [21]: IT COMPARES SUPERVISED TRAINING WITH RANDOM WEIGHT INITIALIZATION AND FINE-TUNING WITH PRE-TRAINED WEIGHTS LEARNED FROM UNSUPERVISED PRE-TRAINING TASKS. IT USES DGCNN AS THE SEGMENTATION MODEL, WHICH IS TRAINED ON DIFFERENT SINGLE AREAS AND TESTED ON AREA 5 (UPPER PART) AND AREA 6 (LOWER PART)

Method	OA on area **5** with different train area					mIoU on area **5** with different train area				
	Area1	Area2	Area3	Area4	Area6	Area1	Area2	Area3	Area4	Area6
from scratch	82.9	81.2	82.8	82.8	83.1	43.6	34.6	39.9	39.4	43.9
Jigsaw3D [13]	83.5(+0.6)	81.2(+0.0)	84.0(+1.2)	82.9(+0.1)	83.3(+0.2)	44.7(+1.1)	34.9(+0.3)	42.4(+2.5)	39.9(+0.5)	43.9(+0.0)
ParAE [147]	91.8(+8.9)	82.3(+1.1)	89.5(+6.7)	88.2(+5.4)	86.4(+3.3)	53.5(+9.9)	38.5(+3.9)	48.4(+8.5)	45.0(+5.6)	49.2(+5.3)
Method	OA on area **6** with different train area					mIoU on area **6** with different train area				
	Area1	Area2	Area3	Area4	Area5	Area1	Area2	Area3	Area4	Area5
from scratch	84.6	70.6	77.7	73.6	76.9	57.9	38.9	49.5	38.5	48.6
STRL [1]	85.3(+0.7)	72.4(+1.8)	79.1(+1.4)	73.8(+0.2)	77.3(+0.4)	59.2(+1.3)	39.2(+0.8)	51.9(+2.4)	39.3(+0.8)	49.5(+0.9)

TABLE VIII
PERFORMANCES FOR SEMANTIC SEGMENTATION ON S3DIS [21]. UPPER PART: MODELS ARE TESTED ON AREA5 (FOLD#1) AND TRAINED ON THE REST OF THE DATA. LOWER PART: SIX-FOLD CROSS-VALIDATION OVER THREE RUNS

Method	Backbone	mACC	mIoU
from scratch	SR-UNet	75.5	68.2
PointConstrast [54]		77.0	70.9
DepthContrast [104]		-	70.6
Method	Backbone	OA	mIoU
from scratch	PointNet	78.2	47.0
Jigsaw3D [13]		80.1	52.6
OcCo [12]		82.0	54.9
from scratch	PCN	82.9	51.1
Jigsaw3D [13]		83.7	52.2
OcCo [12]		85.1	53.4
from scratch	DGCNN	83.7	54.9
Jigsaw3D [13]		84.1	55.6
OcCo [12]		84.6	58.0

TABLE IX
COMPARISON OF PRE-TRAINING EFFECTS BY DIFFERENT UNSUPERVISED LEARNING METHODS. THE BENCHMARKING IS 3D OBJECT DETECTION TASK OVER DATASETS SUN RGB-D [28] AND SCANNET-V2 [18]. "@0.25" AND "@0.5" REPRESENT PER-CATEGORY RESULTS OF AVERAGE PRECISION (AP) WITH IOU THRESHOLD 0.25 (MAP@0.25) AND 0.5 (MAP@0.5), RESPECTIVELY

Method	Backbone	SUN RGB-D		ScanNet-V2	
		@0.5	@0.25	@0.5	@0.25
from scratch	SR-UNet	31.7	55.6	35.4	56.7
PointConstrast [54]		34.8	57.5	38.0	58.5
PC-FractalDB [106]		35.9	57.1	37.0	59.4
from scratch	VoteNet	32.9	57.7	33.5	58.6
STRL [1]		-	58.2	-	-
RandRooms [103]		35.4	59.2	36.2	61.3
DepthContrast [104]		-	-	-	62.2
CSC [3]		33.6	-	-	-
PointContrast [54]		34.0	-	38.0	-
4DContrast [107]		34.4	-	39.3	-
from scratch	PointNet++	-	57.5	-	58.6
PointContrast [54]		-	57.9	-	58.5
RandRooms [103]		-	59.2	-	61.3
DepthContrast [104]		-	60.7	-	-
PC-FractalDB [106]		33.9	59.4	38.3	61.9
DPCo [108]		35.6	59.8	41.5	64.2
from scratch	H3DNet	39.0	60.1	48.1	67.3
RandRooms [103]		43.1	61.6	51.5	68.6

Unified 3D Backbones Are Needed: One major reason of the great success of deep learning in 2D computer vision is the standardization of CNN architectures with VGG [36], ResNet [37], etc. For example, the unified backbone structures greatly facilitate knowledge transfer across different datasets and tasks. For 3D point clouds, similar development is far under-explored, despite a variety of 3D deep architectures that have been recently reported. This can be observed from the URL methods in tables in Section VI most of which adopted very different backbone models. This impedes the development of 3D point cloud networks in scalable design and efficient deployment in various new tasks. Designing certain universal backbones that can be as ubiquitous as ResNet in 2D computer vision is crucial for the advance of 3D point cloud networks including unsupervised point cloud representation learning.

TABLE X
OBJECT DETECTION PERFORMANCE ON DATASET ONCE [30]. THE BASELINE IS TRAINED FROM SCRATCH. UNSUPERVISED LEARNING METHODS ARE USED FOR PRE-TRAINING MODELS. U_{small}, U_{median}, AND U_{large} REPRESENT SMALL, MEDIUM, AND LARGE AMOUNTS OF UNLABELLED DATA THAT ARE USED FOR UNSUPERVISED LEARNING, RESPECTIVELY

Method	Vehicle	Pedestrian	Cyclist	mAP
Baseline [149]	69.7	26.1	59.9	51.9
U_{small}				
BYOL [8]	67.6	17.2	53.4	46.1 (-5.8)
PointContrast [54]	71.5	22.7	58.0	50.8 (-0.1)
SwAV [150]	72.3	25.1	60.7	52.7 (+0.8)
DeepCluster [125]	72.1	27.6	50.3	53.3 (+1.4)
U_{median}				
BYOL [8]	69.7	27.3	57.2	51.4 (-0.5)
PointContrast [54]	70.2	29.2	58.9	52.8 (+0.9)
SwAV [150]	72.1	28.0	60.2	53.4 (+1.5)
DeepCluster [125]	72.1	30.1	60.5	54.2 (+2.3)
U_{large}				
BYOL [8]	72.2	23.6	60.5	52.1 (+0.2)
PointContrast [54]	73.2	27.5	58.3	53.0 (+1.1)
SwAV [150]	72.0	30.6	60.3	54.3 (+2.4)
DeepCluster [125]	71.9	30.5	60.4	54.3 (+2.4)

TABLE XI
PERFORMANCES OF INSTANCE SEGMENTATION ON DATASETS S3DIS [21] AND SCANNET-V2 [18]. IT REPORTS THE MEAN OF AVERAGE PRECISION (MAP) ACROSS ALL SEMANTIC CLASSES WITH A 3D IOU THRESHOLD OF 0.25

Method	Backbone	S3DIS	ScanNet
from scratch	SR-UNet	59.3	53.4
PointContrast [54]		60.5	55.8
CSC [3]		63.4	56.5
4DContrast [107]		-	57.6

Larger datasets are needed: As described in Section III, most existing URL datasets were originally collected for the task of supervised learning. Since point cloud annotation is laborious and time-consuming, these datasets are severely constrained in data size and data diversity and are not suitable for URL with point clouds which usually requires large amounts of point clouds of good size and diversity. This issue well explains the trivial improvements by URL in tables in Section VI. Hence, it is urgent to collect large-scale and high-quality unlabelled point cloud datasets of sufficient diversity in terms of object-level and scene-level point clouds, indoor and outdoor point clouds, etc.

Unsupervised pre-training for scene-level tasks: As described in Section V-B, most earlier research focuses on object-level point cloud processing though several pioneer studies [1], [3], [54], [103], [123] explored how to pre-train DNNs on scene-level point clouds for improving various scene-level downstream tasks such as object detection and instance segmentation. Prior studies show that the learned unsupervised representations can effectively generalize across domains and tasks. Hence, URL of scene-level point clouds deserves more attention as a new direction due to its great potential in a variety of applications. On the other hand, the research along this line remains at a nascent stage, largely due to the constraints in network architectures and datasets. We foresee that more related research will be conducted in the near future.

Learning representations from multi-modal data: 3D sensors are often equipped with other sensors that can capture additional and complementary information to point clouds. For example, depth cameras are often equipped with optical sensors for capturing better appearance information. LiDAR sensors, optical sensors, GPU, and IMU are often installed together as a sensor suite to capture complementary information and provide certain redundancy in autonomous vehicles and mobile robot navigation. Unsupervised learning from such multi-modal data has attracted increasing attention in recent years. For example, learning correspondences among multi-modal data has been explored as pre-text tasks for unsupervised learning as described in Section V-C. However, the study along this line of research remains under-investigated and we expect more related research point clouds, RGB images, depth maps, etc.

Learning Spatio-temporal representations: 3D sensors that support capturing sequential point clouds are becoming increasingly popular nowadays. Rich temporal information from point cloud streams can be extracted as useful supervision signals for unsupervised learning while most of the existing works still focus on static point clouds. We expect that more effective pretext tasks will be designed that can effectively learn spatio-temporal representations from unlabelled sequential point cloud frames.

VIII. Conclusion

Unsupervised representation learning aims to learn effective representations from unannotated data, which has demonstrated impressive progress in the research with point cloud data. This paper presents a contemporary survey of unsupervised representation learning of point clouds. It first introduces the widely adopted datasets and deep network architectures. A comprehensive taxonomy and detailed review of methods are then presented. Following that, representative methods are discussed and benchmarked over multiple 3D point cloud tasks. Finally, we share our humble opinions about several potential future research directions. We hope that this work can lay a strong and sound foundation for future research in unsupervised representation learning from point cloud data.

References

[1] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu, "Spatio-temporal self-supervised representation learning for 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6535–6545.

[2] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 4338–4364, Dec. 2021.

[3] J. Hou, B. Graham, M. Nießner, and S. Xie, "Exploring data-efficient 3D scene understanding with contrastive scene contexts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15 587–15 597.

[4] L. Jing, L. Zhang, and Y. Tian, "Self-supervised feature learning by cross-modality and cross-view correspondences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1581–1591.

[5] A. Radford et al., "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019, Art. no. 9.

[6] J. D.M.-W. C. Kenton and L. K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics - Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[7] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[8] J.-B. Grill et al., "Bootstrap your own latent: A new approach to self-supervised learning," in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 21271–21284.

[9] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[10] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16 000–16 009.

[11] D. Valsesia, G. Fracastoro, and E. Magli, "Learning localized generative models for 3D point clouds via graph convolution," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–11.

[12] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9782–9792.

[13] J. Sauder and B. Sievers, "Self-supervised deep learning on point clouds by reconstructing space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 12 962–12 972.

[14] A. X. Chang et al., "Shapenet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[15] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2017, pp. 652–660.

[16] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3D object detection in point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9277–9286.

[17] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 770–779.

[18] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5828–5839.

[19] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[20] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointNets for 3D object detection from RGB-D data," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2018, pp. 918–927.

[21] I. Armeni et al., "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1534–1543.

[22] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "Deep learning advances in computer vision with 3D data: A survey," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–38, 2017.

[23] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3D: A review of point cloud semantic segmentation," *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 4, pp. 38–59, Dec. 2020.

[24] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 4, pp. 38–59, Dec. 2020.

[25] X. Liu et al., "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, Jan. 2023.

[26] G.-J. Qi and J. Luo, "Small data challenges in Big Data era: A survey of recent progress on unsupervised and semi-supervised methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 2168–2187, Apr. 2022.

[27] Z. Wu et al., "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2015, pp. 1912–1920.

[28] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 567–576.

[29] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1588–1597.

[30] J. Mao et al., "One million scenes for autonomous driving: Once dataset," 2021, *arXiv:2106.11037*.

[31] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3234–3243.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[34] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[35] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D reconstruction at scale using voxel hashing," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–11, 2013.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–10.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[38] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 945–953.

[39] T. Yu, J. Meng, and J. Yuan, "Multi-view harmonized bilinear network for 3D object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 186–194.

[40] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7652–7660.

[41] Z. Yang and L. Wang, "Learning relationships for multi-view 3D object recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7505–7514.

[42] X. Wei, R. Yu, and J. Sun, "View-GCN: View-based graph convolutional network for 3D shape analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1850–1859.

[43] A. Xiao, X. Yang, S. Lu, D. Guan, and J. Huang, "FPS-Net: A convolutional fusion network for large-scale LiDAR point cloud segmentation," *ISPRS J. Photogrammetry Remote Sens.*, vol. 176, pp. 237–249, 2021.

[44] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3D segmentation of point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2626–2635.

[45] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang, "3D recurrent neural networks with context fusion for point cloud semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 403–417.

[46] C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem, "3D-PRNN: Generating shape primitives with recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 900–909.

[47] Y. Zhao, T. Birdal, H. Deng, and F. Tombari, "3D point capsule networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1009–1018.

[48] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–10.

[49] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[50] B. Graham, M. Engelcke, and L. Van Der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9224–9232.

[51] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal convNets: Minkowski convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3075–3084.

[52] H. Tang, Z. Liu, X. Li, Y. Lin, and S. Han, "TorchSparse: Efficient point cloud inference engine," in *Proc. Conf. Mach. Learn. Syst.*, 2022, pp. 302–315.

[53] H. Tang et al., "Searching efficient 3D architectures with sparse point-voxel convolution," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 685–702.

[54] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3D point cloud understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 574–591.

[55] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Intervention*, 2015, pp. 234–241.

[56] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8895–8904.

[57] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3D point cloud transformers with masked point modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 19 313–19 322.

[58] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–12.

[59] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[60] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10 012–10 022.

[61] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16 259–16 268.

[62] A. Sharma, O. Grau, and M. Fritz, "VConv-DAE: Deep volumetric shape learning without object labels," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 236–250.

[63] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 484–499.

[64] 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," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 82–90.

[65] J. Xie, Z. Zheng, R. Gao, W. Wang, S.-C. Zhu, and Y. N. Wu, "Learning descriptor networks for 3D shape synthesis and analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8629–8638.

[66] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 206–215.

[67] J. Li, B. M. Chen, and G. H. Lee, "SO-Net: Self-organizing network for point cloud analysis," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2018, pp. 9397–9406.

[68] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 40–49.

[69] M. Gadelha, R. Wang, and S. Maji, "Multiresolution tree networks for 3D point cloud processing," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 103–118.

[70] 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," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8376–8384.

[71] X. Liu, Z. Han, X. Wen, Y.-S. Liu, and M. Zwicker, "L2G auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 989–997.

[72] 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," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 10 441–10 450.

[73] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, "PointFlow: 3D point cloud generation with continuous normalizing flows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4541–4550.

[74] Y. Shi, M. Xu, S. Yuan, and Y. Fang, "Unsupervised deep shape descriptor with point distribution learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9353–9362.

[75] X. Gao, W. Hu, and G.-J. Qi, "GraphTERr: Unsupervised learning of graph transformation equivariant representations via auto-encoding node-wise transformations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7163–7172.

[76] X. Wen, T. Li, Z. Han, and Y.-S. Liu, "Point cloud completion by skip-attention network with hierarchical folding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1939–1948.

[77] Y. Sun, Y. Wang, Z. Liu, J. Siegel, and S. Sarma, "PointGrow: Autoregressively learned point cloud generation with self-attention," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 61–70.

[78] J. Yang, P. Ahn, D. Kim, H. Lee, and J. Kim, "Progressive seed generation auto-encoder for unsupervised point cloud learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6413–6422.

[79] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Proc. 7th Eur. Conf. Comput. Vis.*, 2022, pp. 604–621.

[80] R. Zhang et al., "Point-M2AE: Multi-scale masked autoencoders for hierarchical point cloud pre-training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1–12.

[81] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, 1991.

[82] Autoencoder, "Autoencoder — Wikipedia, the free encyclopedia," 2022, Accessed: Feb. 16, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Autoencoder

[83] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 605–613.

[84] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3859–3869.

[85] S. Chen, C. Duan, Y. Yang, D. Li, C. Feng, and D. Tian, "Deep unsupervised learning of 3D point clouds via graph topology inference and filtering," *IEEE Trans. Image Process.*, vol. 29, pp. 3183–3198, 2020.

[86] H. Chen, S. Luo, X. Gao, and W. Hu, "Unsupervised learning of geometric sampling invariant representations for 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2021, pp. 893–903.

[87] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014.

[88] C.-L. Li, M. Zaheer, Y. Zhang, B. Poczos, and R. Salakhutdinov, "Point cloud GAN," 2018, *arXiv:1810.05795*.

[89] R. Li, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-GAN: A point cloud upsampling adversarial network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7203–7212.

[90] E. Remelli, P. Baque, and P. Fua, "NeuralSampler: Euclidean point cloud auto-encoder and sampler," 2019, *arXiv:1901.0s9394*.

[91] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-Net: Point cloud upsampling network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2790–2799.

[92] W. Yifan, S. Wu, H. Huang, D. Cohen-Or, and O. Sorkine-Hornung, "Patch-based progressive 3D point set upsampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5958–5967.

[93] Y. Qian, J. Hou, S. Kwong, and Y. He, "PUGeo-Net:: A geometry-centric network for 3D point cloud upsampling," in *Proc. Eur. Conf. Comput. Vis.*. Springer, 2020, pp. 752–769.

[94] G. Qian, A. Abualshour, G. Li, A. Thabet, and B. Ghanem, "PU-GCN: Point cloud upsampling using graph convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11 683–11 692.

[95] R. Li, X. Li, P.-A. Heng, and C.-W. Fu, "Point cloud upsampling via disentangled refinement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 344–353.

[96] K. Hassani and M. Haley, "Unsupervised multi-task feature learning on point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8160–8171.

[97] L. Zhang and Z. Zhu, "Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks," in *Proc. IEEE Int. Conf. 3D Vis.*, 2019, pp. 395–404.

[98] Y. Rao, J. Lu, and J. Zhou, "Global-local bidirectional reasoning for unsupervised representation learning of 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5376–5385.

[99] A. Sanghi, "Info3D: Representation learning on 3D objects using mutual information maximization and contrastive learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 626–642.

[100] M. Gadelha et al., "Label-efficient learning on point clouds using approximate convex decompositions," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 473–491.

[101] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim, "Self-supervised learning of point clouds via orientation estimation," in *Proc. IEEE Int. Conf. 3D Vis.*, 2020, pp. 1018–1028.

[102] B. Du, X. Gao, W. Hu, and X. Li, "Self-contrastive learning with hard negative sampling for self-supervised point cloud learning," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 3133–3142.

[103] Y. Rao, B. Liu, Y. Wei, J. Lu, C.-J. Hsieh, and J. Zhou, "RandomRooms: Unsupervised pre-training from synthetic shapes and randomized layouts for 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3283–3292.

[104] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra, "Self-supervised pretraining of 3D features on any point-cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10 252–10 263.

[105] Y. Chen et al., "Shape self-correction for unsupervised point cloud understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8382–8391.

[106] R. Yamada, H. Kataoka, N. Chiba, Y. Domae, and T. Ogata, "Point cloud pre-training with natural 3D structures," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 21 283–21 293.

[107] Y. Chen, M. Nießner, and A. Dai, "4DContrast: Contrastive learning with dynamic correspondences for 3D scene understanding," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 543–560.

[108] L. Li and M. Heizmann, "A closer look at invariances in self-supervised pre-training for 3D vision," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 656–673.

[109] J. Yin et al., "Proposalcontrast: Unsupervised pre-training for lidar-based 3D object detection," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 17–33.

[110] H. Liu, M. Cai, and Y. J. Lee, "Masked discrimination for self-supervised learning on point clouds," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 657–675.

[111] K. Liu, A. Xiao, X. Zhang, S. Lu, and L. Shao, "FAC: 3D representation learning via foreground aware feature contrast," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 1–11.

[112] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3D surface generation," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2018, pp. 216–224.

[113] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proc. IEEE Int. Conf. 3D Vis.*, 2018, pp. 728–737.

[114] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "PF-Net: Point fractal network for 3D point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7659–7667.

[115] M. Liu, L. Sheng, S. Yang, J. Shao, and S.-M. Hu, "Morphing and sampling network for dense point cloud completion," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11 596–11 603.

[116] W. Zhang, Q. Yan, and C. Xiao, "Detail preserved point cloud completion via separated feature aggregation," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 512–528.

[117] C. Xie, C. Wang, B. Zhang, H. Yang, D. Chen, and F. Wen, "Style-based point generator with adversarial rendering for point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4619–4628.

[118] K. Fu, P. Gao, S. Liu, R. Zhang, Y. Qiao, and M. Wang, "POS-BERT: Point cloud one-stage bert pre-training," 2022, *arXiv:2204.00989*.

[119] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

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

[121] 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," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2773–2781.

[122] J. Jiang, X. Lu, W. Ouyang, and M. Wang, "Unsupervised representation learning for 3D point cloud data," 2021, *arXiv:2110.06632*.

[123] J. Hou, S. Xie, B. Graham, A. Dai, and M. Nießner, "Pri3D: Can 3D priors help 2D representation learning?," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 5693–5702.

[124] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *J. Roy. Stat. Society. Ser. C Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.

[125] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[126] J. Sauder and B. Sievers, "Context prediction for unsupervised deep learning on point clouds," 2019, *arXiv:1901.08396*.

[127] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–14.

[128] A. Thabet, H. Alwassel, and B. Ghanem, "Self-supervised learning of local features in 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 938–939.

[129] C. Sun, Z. Zheng, X. Wang, M. Xu, and Y. Yang, "Point cloud pre-training by mixing and disentangling," 2021, *arXiv:2109.00452*.
[130] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9297–9307.
[131] A. Xiao, J. Huang, D. Guan, F. Zhan, and S. Lu, "Transfer learning from synthetic to real LiDAR point cloud for semantic segmentation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 3, 2022, pp. 2795–2803.
[132] C. Feichtenhofer, H. Fan, B. Xiong, R. Girshick, and K. He, "A large-scale study on unsupervised spatiotemporal representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3299–3309.
[133] X. Song et al., "Spatio-temporal contrastive domain adaptation for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9787–9795.
[134] K. Hu, J. Shao, Y. Liu, B. Raj, M. Savvides, and Z. Shen, "Contrast and order representations for video self-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7939–7949.
[135] H. Kuang et al., "Video contrastive learning with global context," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3195–3204.
[136] D. Z. Chen, A. X. Chang, and M. Nießner, "ScanRefer: 3D object localization in RGB-D scans using natural language," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 202–221.
[137] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9902–9912.
[138] Z. Wang, X. Yu, Y. Rao, J. Zhou, and J. Lu, "P2P: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1–17.
[139] H. Deng, T. Birdal, and S. Ilic, "PPF-foldNet: Unsupervised learning of rotation invariant 3D local descriptors," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 602–618.
[140] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1802–1811.
[141] Y. Zeng, Y. Qian, Z. Zhu, J. Hou, H. Yuan, and Y. He, "CorrNet3D: Unsupervised end-to-end learning of dense correspondence for 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6052–6061.
[142] I. Lang, D. Ginzburg, S. Avidan, and D. Raviv, "DPC: Unsupervised deep point correspondence via cross and self construction," in *Proc. IEEE Int. Conf. 3D Vis.*, 2021, pp. 1442–1451.
[143] H. Jiang, Y. Shen, J. Xie, J. Li, J. Qian, and J. Yang, "Sampling network guided cross-entropy method for unsupervised point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6128–6137.
[144] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16 259–16 268.
[145] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proc. Symp. Geometry Process.*, 2003, pp. 156–164.
[146] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," in *Computer Graphics Forum*. Hoboken, NJ, USA: Wiley, 2003, pp. 223–232.
[147] B. Eckart, W. Yuan, C. Liu, and J. Kautz, "Self-supervised learning on 3D point clouds by learning discrete generative models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8248–8257.
[148] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.
[149] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, 2018, Art. no. s3337.
[150] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 9912–9924, 2020.

Aoran Xiao received the BSc and MSc degrees from Wuhan University, China, in 2016 and 2019, respectively. He is currently working toward the PhD degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include point cloud processing, computer vision, and remote sensing.

Jiaxing Huang received the BEng degree in electronics and electrical engineering from the University of Glasgow, U.K., and the MSc degree in electronics and electrical engineering from the Nanyang Technological University (NTU), Singapore. He is currently a research associate and currently working toward the PhD degree with School of Computer Science and Engineering, NTU, Singapore. His research interests include computer vision and machine learning.

Dayan Guan received the PhD degree from Zhejiang University, China, in Sep. 2019. He is currently a research scientist with the Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates. Before that, he had been a research fellow with Nanyang Technological University from Nov 2019 to Mar 2022. His research interests include computer vision, pattern recognition, and deep learning.

Xiaoqin Zhang (Senior Member, IEEE) received the BSc degree in electronic information science and technology from Central South University, China, in 2005, and the PhD degree in pattern recognition and intelligent system from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China, in 2010. He is currently a professor with Wenzhou University, China. He has published more than 100 papers in international and national journals, and international conferences, including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *International Journal of Computer Vision*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Computers*, ICCV, CVPR, NIPS, IJCAI, AAAI, and among others. His research interests include in pattern recognition, computer vision, and machine learning.

Shijian Lu received the PhD degree in electrical and computer engineering from the National University of Singapore. He is an associate professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His major research interests include image and video analytics, visual intelligence, and machine learning.

Ling Shao (Fellow, IEEE) is a distinguished professor with the UCAS-Terminus AI Lab, University of Chinese Academy of Sciences, Beijing, China. He was the founding CEO and chief scientist of the Inception Institute of Artificial Intelligence, Abu Dhabi, UAE. His research interests include computer vision, deep learning, medical imaging and vision and language. He is a fellow of the IAPR, the BCS and the IET.