Point2SpatialCapsule: Aggregating Features and Spatial Relationships of Local Regions on Point Clouds Using Spatial-Aware Capsules

Xin Wen, *Graduate Student Member, IEEE*, Zhizhong Han, Xinhai Liu, and Yu-Shen Liu, *Member, IEEE*

***Abstract*— Learning discriminative shape representation directly on point clouds is still challenging in 3D shape analysis and understanding. Recent studies usually involve three steps: first splitting a point cloud into some local regions, then extracting the corresponding feature of each local region, and finally aggregating all individual local region features into a global feature as shape representation using simple max-pooling. However, such pooling-based feature aggregation methods do not adequately take the spatial relationships (e.g. the relative locations to other regions) between local regions into account, which greatly limits the ability to learn discriminative shape representation. To address this issue, we propose a novel deep learning network, named Point2SpatialCapsule, for aggregating features and spatial relationships of local regions on point clouds, which aims to learn more discriminative shape representation. Compared with the traditional max-pooling based feature aggregation networks, Point2SpatialCapsule can explicitly learn not only geometric features of local regions but also the spatial relationships among them. Point2SpatialCapsule consists of two main modules. To resolve the disorder problem of local regions, the first module, named *geometric feature aggregation*, is designed to aggregate the local region features into the learnable cluster centers, which explicitly encodes the spatial locations from the original 3D space. The second module, named *spatial relationship aggregation*, is proposed for further aggregating the clustered features and the spatial relationships among them in the feature space using the spatial-aware capsules developed in this article. Compared to the previous capsule network based methods, the feature routing on the spatial-aware capsules can learn more discriminative spatial relationships among local regions for point clouds, which establishes a direct mapping between log priors and the spatial locations through feature clusters. Experimental results demonstrate that Point2SpatialCapsule outperforms the state-of-the-art methods in the 3D shape classification, retrieval and segmentation tasks under the well-known ModelNet and ShapeNet datasets.**

***Index Terms*— Point cloud, shape representation, feature aggregation, spatial relationships, capsule network.**

I. Introduction

3D SHAPE representation learning plays a central role in shape analysis and understanding, which has a wide range of applications such as shape classification [1]–[5], retrieval [6]–[9], semantic/instance segmentation [10]–[14], cross-modal application [15]–[18] and remote sensing [19]–[23]. Among the multiple representation forms of 3D shapes, 3D point clouds, benefited from its easy access, have become one of the most popular 3D shape forms in recent years. Specifically, the point clouds consist of a set of unordered points, each of which is composed of 3D coordinates, possibly with some additional attributes such as normal, color and material.

However, learning discriminative shape representation directly on point clouds is still challenging in 3D shape analysis and understanding. Recent studies for learning point cloud representations usually involve the following three steps. Each input point cloud is first split into some local regions. Then, the corresponding features of local regions are extracted using shared Multi-Layer Perceptron (MLP) [1] or kd-trees [24]. Finally, the extracted local region features are aggregated into a global feature vector as the shape representation [10], [25], [26]. Most of the previous methods mainly focus on how to enhance the process of local region feature extraction, while often employ a simple pooling-based layer [1], [25], [26] to aggregate these extracted features. However, such pooling-based feature aggregation methods do not take adequately the spatial relationships among local regions into account. So far, how to aggregate those learned local region features and their spatial relationships still remain the challenges in existing methods of point cloud representation learning. In this article, we argue the importance of learning spatial relationships for aggregating local region features with respect to the following two reasons. (1) For point clouds with similar local regions, the differences in the spatial arrangements of these local regions are important for learning the discriminative features. (2) Considering the permutation invariant nature of point clouds, it is important to learn the intrinsic spatial relationship between each part and the whole, in order to constitute the permutation invariant knowledge for point cloud recognition.

The common strategy for feature aggregation in previous methods [1], [25], [27], [28] is to extract the most significant

Manuscript received July 27, 2019; revised March 20, 2020 and August 19, 2020; accepted August 19, 2020. Date of publication September 7, 2020; date of current version September 16, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant, 2018YFB0505400, Grant 2019YFB1405703, and Grant TC190A4DA/3; in part by NSFC under Grant 61672307; and in part by the Tsinghua-Kuaishou Institute of Future Media Data. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Riccardo Leonardi. *(Corresponding author: Yu-Shen Liu.)*

Xin Wen and Xinhai Liu are with the School of Software, Tsinghua University, Beijing 100084, China (e-mail: x-wen16@mails.tsinghua.edu.cn; lxh17@mails.tsinghua.edu.cn).

Zhizhong Han is with the Department of Computer Science, University of Maryland at College Park, College Park, MD 20737 USA (e-mail: h312h@umd.edu).

Yu-Shen Liu is with the School of Software, BNRist, Tsinghua University, Beijing 100084, China (e-mail: liuyushen@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TIP.2020.3019925

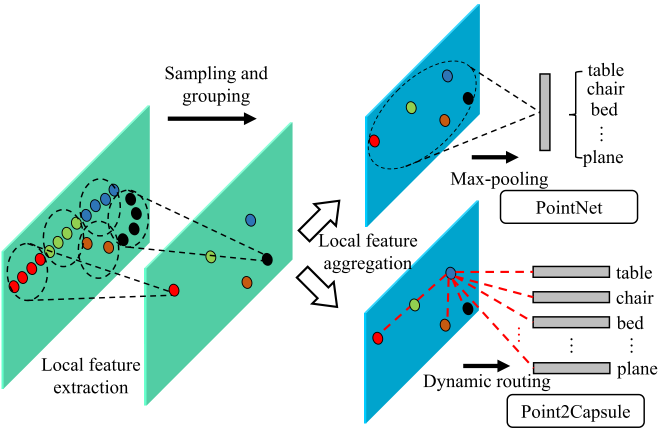


Fig. 1. The illustration of comparison between the max-pooling based PointNet and the dynamic routing based Point2SpatialCapsule in terms of local region feature aggregation. Given an example 2D point cloud like a shape of "P", the point cloud is split into some local regions (left), from which the corresponding local region features are extracted with sampling and grouping (middle). The comparison of two methods is shown at the right side of this figure. Here, denoted by black dotted line, the max-pooling only keeps the most significant geometric characteristics in local regions (top right), while this causes the spatial relationships between local regions are filtered out. In contrast, the dynamic routing based Point2SpatialCapsule can handle both the geometric characteristics and the spatial relationships of local regions (bottom right), denoted by the red dotted line.

characteristics (such as the engine of an airplane) in local regions of the point cloud step by step, through a deep neural network with the pooling structure (such as max-pooling). However, the problem is that the pooling-based methods will filter out the spatial relationships of different areas on the feature map [29]. Thus, it only considers the existences of characteristics in local regions, while the spatial arrangements between these regions will not be preserved. As a result, most of the existing methods usually fail to learn the spatial relationships among the local regions, which further limits the ability of the network for learning discriminative 3D shape representation.

To address the aforementioned problem, we propose a novel deep learning network, named Point2SpatialCapsule, for aggregating geometric features and spatial relationships of local regions on point clouds, which aims to learn more discriminative shape representation. Inspired by the recently developed capsule network [29], Point2SpatialCapsule employs the dynamic routing to aggregate the local region features and their spatial relationships. Fig. 1 illustrates the comparison between our dynamic routing based Point2SpatialCapsule and the previous feature aggregation methods like max-pooling used in PointNet [1]. Note that the max-pooling feature aggregation in PointNet [1] only considers the existences of characteristics in local regions, but in contrast our Point2SpatialCapsule can explicitly handle the spatial relationships between local region features through dynamic routing of capsule network. This advantage encourages us to consider adopting capsule network for 3D point clouds representation learning.

However, the problem is that, the original implementation of capsule network is designed for 2D image recognition, where the log priors in capsule network are bounded to the fixed locations on the 2D feature maps [29]. In contrast, for 3D point clouds, the locations of random sampled input points are disordered and their absolute position coordinates may not always keep consistent. As a result, it is difficult to find a direct mapping that can generate the features encoded with fixed spatial locations. What's worse, the previous capsule based methods failed to address such problem, most of which directly generate the capsules from a single global feature vectors. Such practice leads to the loss of spatial relationships between local regions. As a result, the log priors in routing algorithm between capsules can not learn the spatial relationships of local regions, which greatly limits the representation ability of capsules. In this article, we argue the importance of encoding the fixed spatial locations into capsules, which aims to efficiently utilize the representation ability of log priors for learning the spatial relationships between local regions on point clouds.

In order to solve the above limitations, two novel modules are specially designed in Point2SpatialCapsule to achieve local region feature aggregation as follows. (1) The first module, named *geometric feature aggregation*, aims to aggregate the extracted local region features in the feature space. Here, the term "*geometric*" indicates that this module aggregates the geometric information, like the coordinates of central points and the shapes of local regions represented by the feature vectors, into the centers of local feature clusters, which aims to resolve the disorder problem of local regions. (2) The second module, named *spatial relationship aggregation*, is to apply routing algorithm on the learned feature clusters. The term "*spatial-aware*" indicates that the capsules are encoded with the spatial locations, which is to guarantee the direct mapping between the log priors and the fixed locations in the 3D space. Therefore, we call them the *spatial-aware capsules*, which allows the network to efficiently learn the spatial relationships between local regions. Fig. 2 shows the visualized demonstration of the advantage of spatial relationship aggregation. Because of the shifting and rotation of point clouds, the changing locations of local region features in 3D space also change the log priors. To resolve this issue, the geometric feature aggregation clusters the input local region features into the learnable cluster centers, which are irrelevant to the input points and relatively invariant in the feature space. Therefore, the routing algorithm can efficiently learn the log priors for aggregating the spatial relationships between local regions. Our main contributions are summarized as follows.

- We propose a novel deep network, i.e. Point2Spatial Capsule, for learning more discriminative shape representations of point clouds. Compared with the traditional pooling-based methods, Point2SpatialCapsule can explicitly learn not only geometric features of local regions but also the spatial relationships among them.
- We propose the geometric feature aggregation to resolve the disorder problem of local regions, where the local region features are aggregated into the learnable cluster centers, which are explicitly encoded with the spatial locations from the original 3D space.
- We propose the spatial relationship aggregation to further utilize the spatial locations encoded in the feature clusters. Compared to the previous capsule network based

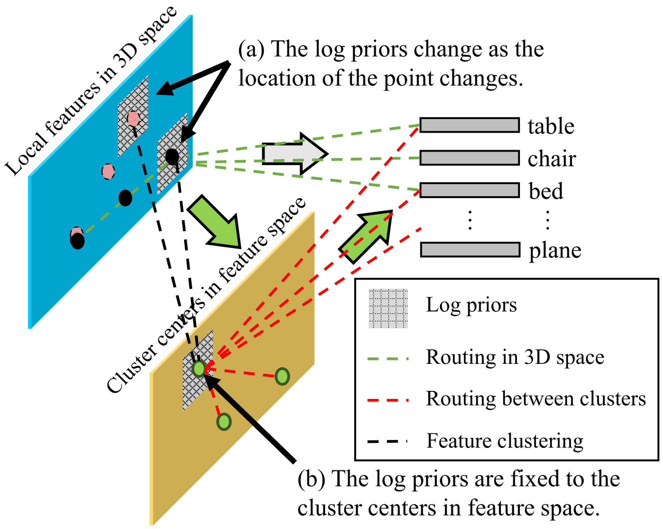

Local features in 3D space
(a) The log priors change as the location of the point changes.
table
chair
bed
⋮
plane
Cluster centers in feature space
▨ Log priors
Routing in 3D space
Routing between clusters
Feature clustering
(b) The log priors are fixed to the cluster centers in feature space.

Fig. 2. The illustration of directly applying capsule network to the point cloud (a) and the clustering based Point2SpatialCapsule (b). The shifting and rotation of point cloud change the locations of local regions in 3D space and also change their corresponding log priors. Therefore, routing in 3D space (green dotted line) will cause the shifting of log priors, making the routing algorithm fail to learn the spatial relationships between local regions. In contrast, the geometric feature aggregation aggregates the input local region features into relatively invariant cluster centers. Therefore, routing between clusters (red dotted line) can efficiently learn the log priors for aggregating spatial relationships between local regions.

methods, the spatial relationship aggregation can learn more discriminative spatial relationships between local regions by establishing a direct mapping between log priors and the spatial locations through feature clusters.

The remainder of this article is organized as follows. First, the related work is introduced in Sec.II. Then we detail the proposed Point2SpatialCapsule in Sec.III. The experiments and the ablation studies are given in Sec.IV. Finally, we conclude this article in Sec.V.

II. Related Work

In this section, we mainly review the methods related to 3D shape representation learning based on deep learning networks. The existing methods can be roughly divided into four categories according to various 3D shape forms that are learned from, including voxels, point clouds, views and meshes.

A. Point Cloud Based Methods

Recent studies of point cloud representation learning mainly focus on the local feature extraction and integration. PointNet [1] is the pioneering work of introducing deep learning into point cloud representation learning, which independently learns the features of each point and aggregates the learned features into a global feature with the max-pooling layer. After that, plenty of the follow-up studies [10], [26], [30], [31] focus on how to better integrate the contextual information of local regions on point clouds. For example, PointNet++ [25] designed the hierarchical feature learning architecture based on PointNet to encode multi-scale local areas. Following the convolutional structure of PointNet++, successors such as PointCNN [26] and SpiderCNN [30] investigated some improved convolution operations which aggregate the neighbors of a given point by edge attributes in the local region graph. Different from the idea of using convolution structure, Point2Sequence [10] introduced the sequential model (i.e. RNN) to capture the fine-grained contextual information of features in local regions. Specifically, Point2Sequence arranges the features into a sequence according to the size of the region scale, and then uses a RNN to capture the contextual information within the local regions. However, the problem is that most of the above methods fail to consider the spatial relationships among different local regions when aggregating the extracted local region features, where the usual practice for these methods is to use the pooling layer to learn the global feature from the local ones.

More recent studies focus on how to improve the local region feature extraction [3], [32], [33]. These methods have shown impressive potentials in the semantic segmentation task on point cloud. For examples, A-CNN [3] was proposed to annularly arrange the neighbor points and apply the convolution network on these arranged points to learn the local region features. RS-CNN [32] designed a shape-aware convolution to learn the local region features from the relation between points. Both DensePoint [34] and ShellNet [35] focus on learning and aggregating the geometric features of local regions under different region scales. SK-Net [36] and KPConv [37] are methods trying to address the local feature extraction by finding the *key points* in the point clouds. As the methods to aggregate different kinds of features, GS-Net [38] learns the geometric relationships between local region features with the Eigen-Graph, and InterpCNN [39] uses a learnable method to integrate explicit geometric information with implicit local region features. However, the max-pooling or concatenation based operation in these methods still limit their ability to learn the spatial relationships between these geometric features of local regions.

The proposed Point2SpatialCapsule mainly focus on how to aggregate the feature and relationships of local regions after extracting local features. The usual practice for previous methods is to apply the strategy of bottom-to-top point cloud feature aggregation [11], [24], [27], [40], [41]. For example, Kd-Net [24] performs multiplicative transformations according to the subdivisions of point clouds based on the kd-trees. SO-Net [27] employs a SOM to build the spatial distribution of the input point cloud, which allows hierarchical feature extraction on both individual points and SOM nodes. However, most of the above methods use max-pooling as a feature aggregation method, which inevitably filters out the spatial relationships among local regions. On the other hand, PVNet [42] is also a notable method that considers the local feature aggregation, which focuses on mining the difference in importance between the local features. It employs high-level global features from the multi-view data of input 3D shapes to mine the relative correlations between different local features from the point cloud data. Same as the above-mentioned methods, PVNet only learns the different contributions among local regions, while the spatial relationships among these regions are not considered.

B. View-Based Methods

The dominant performance of multi-view based methods on the task of 3D shape retrieval comes from the research progress of measuring the similarities between 2D image features [43]–[46]. As one of the pioneering work, GIFT [44] adopted the Hausdorff distance to measure the similarity between the view sets of two 3D shapes. Another notable research direction is to focus on PANORAMA views of 3D shapes, where a PANORAMA view can be regarded as the seamless aggregation of multiple views captured on a circle. For examples, DeepPano [47] introduced a row-wise max-pooling to relief the effect of rotation about the up-oriented direction, and Sfikas *et al.* [48] introduced CNN for learning the global features from the PANORAMA views in a consisitent order. To explore the potential of attention mechanism, the methods like 3DViewGraph [2] have been proposed to integrate the spatial pattern correlations of unordered views with attention weights, and Part4Features [7] developed a novel multi-attention mechanism for aggregating the learned local parts.

More recently, SeqViews2SeqLabels [8] was proposed to learn 3D features via aggregating sequential views by RNN, which aims to eliminate the effect of rotation of 3D shapes. Compared with the previous pooling based methods, the RNN-based SeqViews2SeqLabels suffers less from the content and the spatial location loss. Similarly, as an unsupervised approaches, VIP-GAN [6] trains an RNN-based neural network architecture to solve multiple view inter-prediction tasks for each shape.

C. Voxel-Based Methods

Voxel-based methods often rasterize a 3D shape as a function or distribution sampled on voxels [49]–[51]. For supervised learning the representation of 3D voxels, 3DShapeNets [52] adopted the convolutional restricted Boltzmann machine to learn the representation of 3D voxels. O-CNN [40] learns the representation of 3D voxel based on a novel octree structure. And Han *et al.* [53] proposed a novel permutation voxelization strategy to learn high-level and hierarchical 3D local features from raw 3D voxels. For unsupervised learning, methods like VConv-DAE [54] use the fully convolutional autoencoder for unsupervised learning the voxel representation by reconstruction. However, the problem is, considering the induced complexity and limitations of directly exploiting the sparsity of voxel grids, it is difficult to introduce the large scale or flexible deep networks for representation learning. Therefore, more recent methods such as OctNet [41] and kd-net [24] consider to utilize the scalable indexing structures for solving this problem, where deep neural networks can be further adopted for achieving more impressive results.

D. Mesh-Based Methods

As for mesh-based methods, to explore the effectiveness of the heat diffusion based descriptor, Xie *et al.* [55] proposed a shape feature learning scheme based on auto-encoders, where the model can extract the features that are insensitive to the deformations. By fully utilizing the spectral domain, Xie *et al.* [56] further proposed to learn a novel binary spectral shape descriptor with the deep neural network for 3D shape correspondence. Recently, BoSCC [57] was introduced for a spatially enhanced 3D shape representation based on bag of spatial context correlations. And more recently, Deep Spatiality [58] was also proposed to simultaneously learn 3D global and local features with novel coupled softmax.

E. Capsule Networks

The ability of capsule network [29] for capturing spatial relationships comes from the dynamic routing algorithm and the log priors, which are bound to the absolute location on the input feature maps. Specifically, the capsule network learns the log priors by considering the relationships between the absolute locations on the feature map and the high-level capsules. Then, through the dynamic routing algorithm, which is based on the learned log priors, the high-level capsules can integrate the low-level features and their spatial relationships among different locations on the feature maps. This advantage promotes us to consider applying the capsule network to 3D point cloud representation learning.

So far, the capsule network has shown the great potentials in many research areas, such as image processing [59], [60] and natural language processing (NLP) [61]–[63]. However, as for the application of capsule network in 3D shape representation learning, there are a few methods proposed in recent years. For example, 3D-CapsNet [64] adopts the capsule network for 3D shape classification tasks based on volumetric data, and 3D-Point-Capsule [65] learns the point cloud representation and part segmentations in an unsupervised way. And for supervised learning, 3DCapsule [66] applies the capsule network as an extension of fully-connected layers for point cloud classification.

An important problem of the above methods is that they all build the capsule layers over the global feature (usually produced by the fully-connected layer or max-pooling) of point clouds, where the spatial relationships between local region features have been filtered out by the network. Therefore, the log priors in routing algorithm cannot learn the spatial distribution among the extracted local features, which limits the biggest advantage of capsule network for aggregation spatial relationships of local regions.

Therefore, to address this problem of previous methods, Point2SpatialCapsule aggregates the features into clusters in feature space, and applies the routing algorithm between these aggregated clusters. In the research of point cloud representation learning, methods like PointNetVLAD [67] have adopted the similar clustering strategy, i.e. NetVLAD [68], for feature aggregation. However, different from the previous methods that only cluster features for aggregating regions with similar geometric characteristics (e.g. shapes), our method takes one step further to not only considering geometric characteristics, but also explore the potentials for aggregating spatial relationships between these regions. Specifically, Point2SpatialCapsule produces the clusters for both the features and their coordinates, in order to explicitly preserve the features and their spatial location.

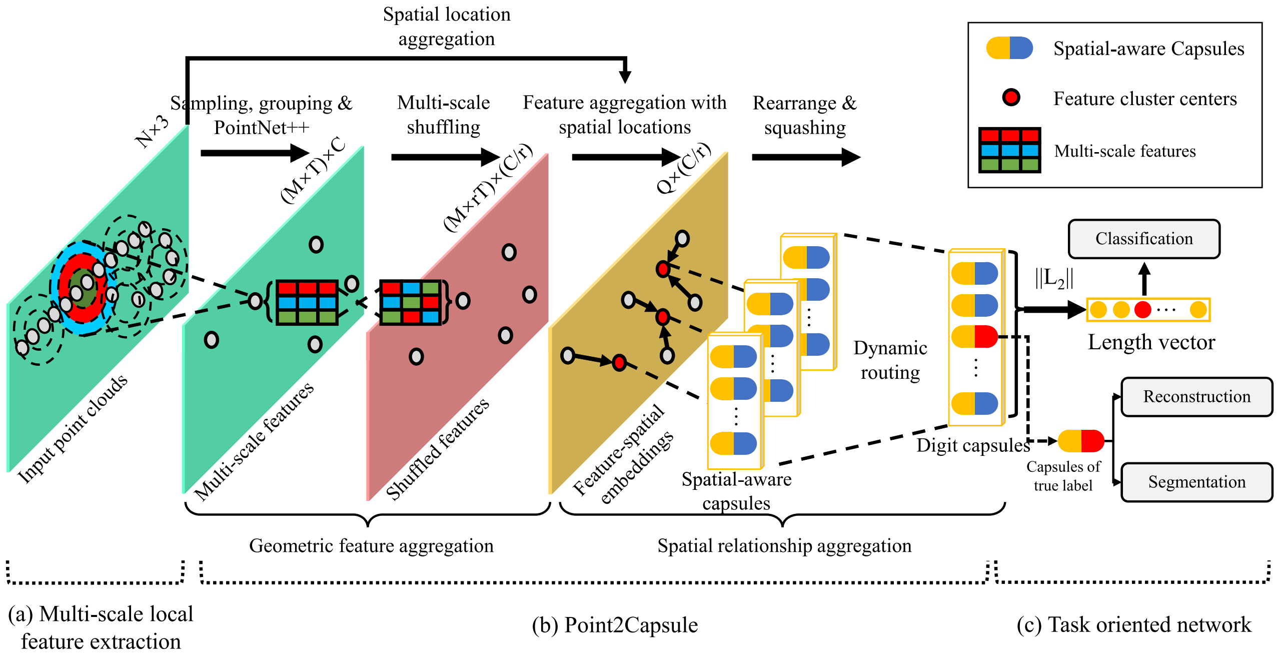


Fig. 3. The architecture of our proposed Point2SpatialCapsule. For input point clouds, (a) the multi-scale local feature extraction first extracts features from multi-scale areas, (b) then the geometric feature aggregation encodes the extracted multi-scale local region features and their locations into the learnable clustering centers to produce the feature-spatial embeddings. The spatial relationship aggregation aggregates the feature-spatial embeddings by considering both the embeddings and their spatial relationships. (c) The task oriented network is adopted for performing on different tasks.

III. Shape Representation Learning With Point2SpatialCapsule

An overview of shape representation learning network with Point2SpatialCapsule is shown in Fig. 3. The whole network consists of three main parts as follows. (1) The first part is the multi-scale local feature extraction, which is a PointNet++ based network for extracting the features from multi-scale local regions on point clouds (see Sec. III-A). (2) The second part is Point2SpatialCapsule, which is composed of two main modules for aggregating the learned features into the global shape representation. Here, the first module, i.e. *geometric feature aggregation*, is to aggregate local region features into clusters (see Sec. III-B). The second module, i.e. *spatial relationship aggregation*, is to aggregate the feature clusters and their spatial relationships into global feature representation (see Sec. III-C). In this section, we will also detail the training procedure of Point2SpatialCapsule (see Sec. III-D). (3) The third part is the task oriented network used for various tasks such as shape segmentation (see Sec. III-E).

A. Multi-Scale Local Feature Extraction

The first part of our network is the multi-scale local feature extraction, as shown in Fig. 3(a). Given a set of input points $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, by following the practice of PointNet++ [25] and ShapeContextNet [69], we iteratively produce a subsampling $\{\boldsymbol{x}_{k_1}, \boldsymbol{x}_{k_2}, \ldots, \boldsymbol{x}_{k_M}\}$ with M points as the centroids of the local regions using farthest point sampling (FPS), such that the newly added point $\boldsymbol{x}_{k_j}$ is the farthest point (in metric distance) from the rest sampled points $\{\boldsymbol{x}_{k_1}, \boldsymbol{x}_{k_2}, \ldots, \boldsymbol{x}_{k_{j-1}}\}$. Then, for each sampled point, the K nearest neighbor (kNN) searching is employed to find $\{K_i|i = 1, \ldots, T\}$ neighbors for this point, under T different scale areas. Followed by a grouping layer, the sampled point and its neighbors are grouped as a $K_i \times 3$ tensor for scale K_i. After that, a simple but effective MLP layer is employed to extract the features of all neighbor points, producing a tensor with shape $K_i \times C$. Finally, a max-pooling layer is applied to integrate the point features in each scale to produce the scale feature of dimension C for scale K_i. For M points in total and T scales for each point, the multi-scale local feature extraction layer produces $M \times T$ multi-scale features, forming a tensor of shape $M \times T \times C$ as its output.

In the implementation, we apply two layers of multi-scale local feature extraction for hierarchically extracting features from point clouds.

B. Point2SpatialCapsule: Geometric Feature Aggregation

In this subsection, we detail the first module of Point2SpatialCapsule, which aims to aggregate the extracted features into clusters and encodes these features with spatial locations.

As shown in Fig. 3(b), before clustering features, the module of geometric feature aggregation first applies the multi-scale shuffling to enhance the diversity of features. Then the features are aggregated into clusters and encoded with the spatial locations (e.g. the absolute locations in the 3D space) from the original 3D space.

1) Multi-Scale Shuffling: Different from the previous methods that apply the pooling-based strategy for integrating the features extracted from multi-scale regions, we propose the multi-scale shuffling layer to build the shuffled features. The reason for adding this layer is demonstrated in Fig. 4, as explained below. When searching the neighbor points in

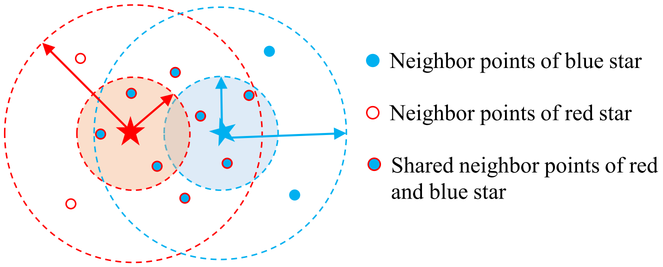

Neighbor points of blue star
Neighbor points of red star
Shared neighbor points of red and blue star

Fig. 4. Illustration of feature similarity between adjacent points. Features of small scales are not very similar because of the small overlap in sampled points. On the contrary, the features of large scales are more similar because of a larger overlap.

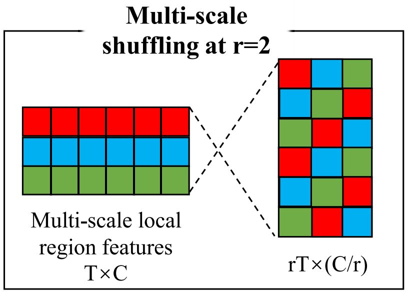

Multi-scale shuffling at r=2
Multi-scale local region features T×C
rT×(C/r)

Fig. 5. Illustration of the multi-scale shuffling, which is the solution to the problem of feature similarity.

a large scale, the searched areas of two adjacent centroids will overlap with each other and output the same neighbor points. As a result, the adjacent points will tend to have the similar features for large scale, which can reduce the diversity of features and introduce an initial clustering center for the subsequent clustering layer. On the other hand, the features of small scales between two centroids are dissimilar because of small overlaps. Therefore, the multi-scale shuffling is introduced to smooth the perceived range of features between different scales and enhance the feature diversity, by mixing the dissimilar features of small scales with the similar features of large scales. As a result, the multi-scale shuffling can promote the network to consider all input features equally and alleviate the problem of similar features.

The effect of multi-scale shuffling is shown in Fig. 5. Specifically, given a point with T scale features of C dimension, which forms a tensor with the shape $T \times C$, the multi-scale shuffling periodically rearranges the elements in the $T \times C$ tensor into a tensor of shape $rT \times (C/r)$, where r is an integer. Thus, for M points in total, the multi-scale shuffling will produce $M \times rT$ shuffled features of dimension C/r, resulting in a tensor of size $(M \times rT) \times (C/r)$.

The multi-scale shuffling is inspired by the subpixel convolution [70] for image upsampling, where the number of area scales T can be considered as the size of image, and the feature dimension C can be regarded as the channels of feature maps. However, different from subpixel convolution which is designed for speeding up the calculations and reducing the amount of parameters in the network, the multi-scale shuffling used in our method aims to enhance the diversity of scale features. We will quantitatively explore the importance of the multi-scale shuffling in ablation studies in Sec. IV-E.

2) Feature Aggregation With Spatial Encodings: The purpose of this layer is to aggregate the shuffled features into the learnable feature cluster centers, which can be regarded as the latent embeddings describing the semantic patterns of the local regions features. To achieve this purpose, we propose to cluster the features in the feature space and their coordinates in the original 3D space. After that, the cluster centers in both the feature space and the 3D space are fused to produce *the feature-spatial embeddings*, as illustrated in Fig. 6(a).

Although the traditional clustering methods like k-means can be adopted to produce the feature cluster centers, their computational cost may be very high because of the huge number of features to be clustered. Therefore, inspired by the recent development of NetVLAD [68], we adopt the soft-assignment for learning the clustering centers for the input shuffled local features. Specifically, the network learns Q cluster centers for input features, denoted as $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_Q | \boldsymbol{q}_k \in \mathbb{R}^{C/r}\}$, as colored by yellow in Fig. 6(a). For each cluster center $\boldsymbol{q}_k$, the layer produces a *feature embedding* $C(\boldsymbol{q}_k) \in \mathbb{R}^{C/r}$, which is an aggregated representation over the whole input shuffled features $\{\hat{\boldsymbol{p}}_i\}$, denoted by

$$C(\boldsymbol{q}_k) = \sum_{i=1}^{n} \frac{e^{\boldsymbol{w}_k^T \hat{\boldsymbol{p}}_i} + b_k}{\sum_{k'} e^{\boldsymbol{w}_{k'}^T \hat{\boldsymbol{p}}_i} + b_{k'}} (\hat{\boldsymbol{p}}_i - \boldsymbol{q}_k), \tag{1}$$

where $\{\boldsymbol{w}_k\}$ and $\{b_k\}$ are the weights and biases, respectively, that determine the contribution of each local feature to the cluster center $\boldsymbol{q}_k$. During training, all the weights, biases and the cluster centers are updated through back-propagation algorithm.

To explicitly encode the spatial locations of local features into their cluster centers, we first cluster the coordinates $\{\boldsymbol{x}_i\}$ of input points into the coordinates cluster centers $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_Q | \boldsymbol{y}_k \in \mathbb{R}^{C/r}\}$, which is the same process as described above and colored by green in Fig. 6(a). The *spatial embeddings* $C(\boldsymbol{y}_k) \in \mathbb{R}^{C/r}$ for coordinates is given as

$$C(\boldsymbol{y}_k) = \sum_{i=1}^{n} \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x}_i} + b'_k}{\sum_{k'} e^{\boldsymbol{w}_{k'}^T \boldsymbol{x}_i} + b'_{k'}} (\boldsymbol{x}_i - \boldsymbol{y}_k). \tag{2}$$

Then, the produced local feature embedding and its corresponding spatial embedding are concatenated to form an explicit *feature-spatial embedding* $C(\boldsymbol{s}_k) = [C(\boldsymbol{y}_k) : C(\boldsymbol{q}_k)]$.

C. Point2SpatialCapsule: Spatial Relationship Aggregation

In Fig. 6(b), we show the overall architecture of previous methods [65], [66] for building the capsules, and compare it with our proposed Point2SpatialCapsule shown in Fig. 6(a). The main difference is that Point2SpatialCapsule builds the spatial-aware capsules based on cluster centers with spatial encodings, while the previous studies simply build the capsules based on the single representation vector generated by fully-connection or pooling based local feature aggregator. As a result, the previous methods fail to preserve the spatial relationships between local regions, which further limits the representation learning ability of dynamic routing.

In this subsection, in order to efficiently learn the prior logs, we first independently generate the spatial-aware capsules from the feature-spatial embeddings using *rearrange*

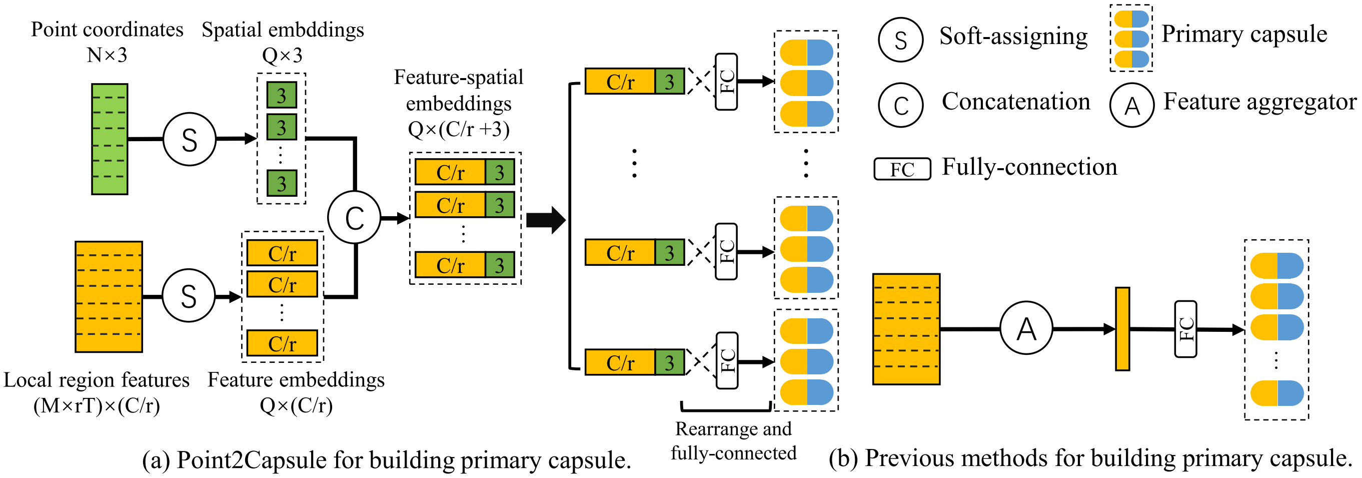


Fig. 6. Comparison of the strategies for applying dynamic routing in local region features between (a) Point2SpatialCapsule and (b) the previous methods [65], [66]. The previous methods directly use a feature aggregator (e.g. max-pooling) to aggregates all local region features, and generate the capsules based on the aggregated global feature. In contrast, Point2SpatialCapsule proposes to cluster the local region features and the point coordinates, and then combines them as spatial-aware cluster centers. The spatial-aware capsules of Point2SpatialCapsule are independently generated according to each cluster centers.

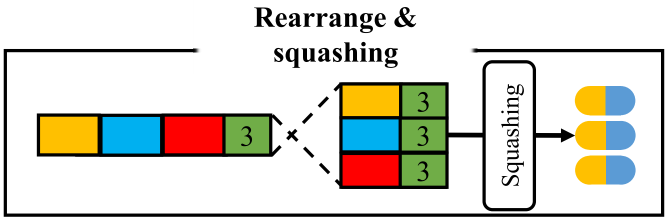


Fig. 7. Illustration of rearrange and squashing layer. The green block is the 3D spatial embedding, while the yellow, blue and red block together constitute a feature embedding.

and *squashing*. Then, we propose to apply routing algorithm between the spatial-aware capsules.

1) Rearrange and Squashing: To build the spatial-aware capsules from the feature-spatial embeddings produced by the geometric feature aggregation module, we deviate from the 2D practice of the original capsule network [29]. In the original capsule network, the spatial-aware capsule aggregates its representation vector by collecting the output logits across different channels at the same location on the feature maps. In Point2SpatialCapsule, since we have built the feature-spatial embeddings encoded with the spatial locations, we can consider that each embedding corresponds to a fixed location, which is the learnable cluster center. Therefore, we can directly rearrange the output and use the fully-connected layer with a squashing activation to produce the spatial-aware capsules. The rearrange layer is to split the feature-spatial embeddings $C(\boldsymbol{s}_k)$ into several short vectors $\{\boldsymbol{u}_i\}$. As shown in Fig. 7, the input feature-spatial embedding is split into $K = 3$ vectors, each of which is combined with the spatial embedding. Then, we follow a squashing layer, as denote by

$$\text{squashing}(\boldsymbol{u}_i) = \frac{\|\boldsymbol{u}_i\|^2}{1 + \|\boldsymbol{u}_i\|^2} \frac{\boldsymbol{u}_i}{\|\boldsymbol{u}_i\|}, \tag{3}$$

where the spatial-aware capsules $\{\boldsymbol{u}_i\}$ are generated as the final output of this layer.

2) Routing Algorithm: Given the input spatial-aware capsules, we follow [29] to apply dynamic routing algorithm to obtain the digit capsule. Specifically, the digit capsule $\boldsymbol{v}_j$ is the output of weighted sum of the prediction vector $\hat{\boldsymbol{u}}_{ij}$ followed by the squashing layer, which can be formulated as

$$\hat{\boldsymbol{u}}_{ij} = W_{ij}\boldsymbol{u}_i, \tag{4}$$

$$\boldsymbol{v}_j = \text{squashing}(\sum_i c_{ij}\hat{\boldsymbol{u}}_{ij}), \tag{5}$$

where $\boldsymbol{u}_i$ is the ith spatial-aware capsule and W_{ij} is a learnable matrix. The *coupling coefficients* [29] c_{ij} is determined by the iterative dynamic routing process, denote by

$$c_{ij} = \frac{e^{b_{ij}}}{\sum_k e^{b_{ik}}}. \tag{6}$$

In 2D capsule network, the $\{b_{ij}\}$ are log priors that only depend on the fixed locations and the type of two capsules. In our network, because the disordered input features are clustered as the feature-spatial embeddings by soft-assignment, these features are bounded to the fixed locations (which are the cluster centers) in the feature space. Therefore, the dynamic routing can learn the log priors between these centers and the digit capsules.

Before training, all of the log priors $\{b_{ij}\}$ are initialized to zero. During training, $\{b_{ij}\}$ are learned discriminatively at the same time with other parameters in the network, by adding the scalar product of $\boldsymbol{v}_{ij}$ and $\hat{\boldsymbol{u}}_{ij}$, i.e.

$$b_{ij} \leftarrow b_{ij} + \boldsymbol{v}_{ij} \cdot \hat{\boldsymbol{u}}_{ij}. \tag{7}$$

D. Point2SpatialCapsule: Training

Following the practice of [29], Point2SpatialCapsule uses the reconstruction loss and the classification loss for supervised point cloud representation learning.

The length of each digit capsule indicates the probability that the characteristic represented by this capsule exists in the input point clouds [29]. During training, the margin loss $\mathcal{L}_{cls}$

is adopted for shape classification defined as

$$\mathcal{L}_{cls} = \sum_j T_j \max(0, m^+ - \|\boldsymbol{v}_j\|)^2 + \sum_j \lambda(1 - T_j)\max(0, \|\boldsymbol{v}_j\| - m^-)^2, \tag{8}$$

where $T_j = 1$ if class j is the true label; otherwise, $T_j = 0$. m^+, m^- and λ are the hyper parameters.

We further reconstruct the input point clouds using four fully-connected layers, with each layer followed by a $relu$ activation and batch normalization except for the last layer. The digit capsule corresponding to the true label is used as the input representation vector to the reconstruction network. The chamfer loss between the original point cloud $\mathbf{X}$ and the reconstructed point cloud $\hat{\mathbf{X}} = \{\hat{\boldsymbol{x}}_i\}$ is adopted as the reconstruction loss $\mathcal{L}_{rec}$, as denoted by

$$\mathcal{L}_{rec} = \frac{1}{|\mathbf{X}|}\sum_{\boldsymbol{x}\in\mathbf{X}} \min_{\hat{\boldsymbol{x}}\in\hat{\mathbf{X}}} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\| + \frac{1}{|\hat{\mathbf{X}}|}\sum_{\hat{\boldsymbol{x}}\in\hat{\mathbf{X}}} \min_{\boldsymbol{x}\in\mathbf{X}} \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|. \tag{9}$$

The total loss for training is the weighted sum of margin loss and the reconstruction loss, as denote by

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{rec}, \tag{10}$$

where $\alpha = 0.0001$ for all the experiments in this article.

E. Model Adjustments for Part Segmentation

The goal of part segmentation is to predict a semantic label for each point in the point cloud. There are two alternative ways for acquiring the per-point feature for each point from the global feature: duplicating the global feature with N times [1], [31], or performing upsampling by interpolation [25], [27]. In this article, we follow the second way to duplicate the vectors in digit capsules belonging to the true label. Then we concatenate the duplicated vectors with the shuffled features. The interpolation layers are used for propagating the features from shape level to point level by upsampling, as shown in Fig. 8.

IV. Experiments

A. Experimental Setup

1) Datasets: The 3D shape classification and retrieval experiments are conducted on two subsets of the Princeton ModelNet dataset [52], i.e. ModelNet40 and ModelNet10. The ModelNet40 dataset contains 12,311 shapes which belong to 40 categories. We follow the same training and split settings as [27], which contains 9,843 shapes for training and 2,468 shapes for testing, respectively. The ModelNet10 dataset is a relatively small dataset which contains the 10 common categories of ModelNet40. Following [27], we split the ModelNet10 into 2,468 training samples and 909 testing samples. Since the original ModelNet provides CAD models represented by vertices and faces, we use the prepared ModelNet10/40 data from [25] for fair comparison. To further evaluate the effectiveness of Point2SpatialCapsule on 3D shapes collected with real scan acquisition, we also conduct experiment on ScanNet dataset [71]. Specifically, we extract 14 out of 20 categories in terms of instance labels from ScanNet dataset, where the labels that we did not use are typically not common classes (e.g. wall/floor). We use the objects in the training split of ScanNet as our training set, and the objects in validation split as our validation set. The total number of objects in training set is 12,412 and in test set is 3,457. The part segmentation task is conducted on the ShapeNet part dataset [72], which contains 16,881 models from 16 categories and is split into training, validation and testing following PointNet++. There are 2048 points sampled for each 3D shape, where each point in a point cloud object belongs to certain one of 50 part classes and each point cloud contains 2 to 5 parts.

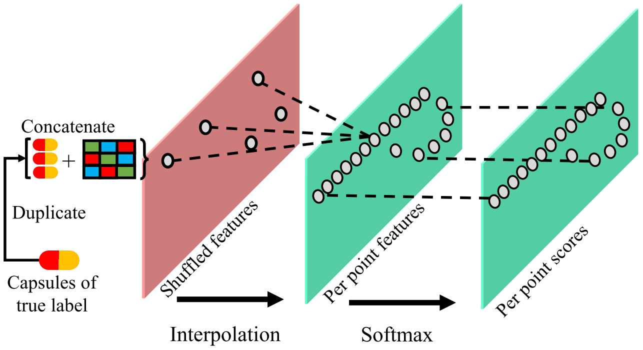


Fig. 8. Illustration of the segmentation network in our Point2SpatialCapsule.

2) Classification and Retrieval Settings: Because the length of representation vector in digit capsule indicates the probability that certain characteristic exists in the input point clouds. In the case of Point2SpatialCapsule, the characteristic of digit capsule is the class label. Thus, we choose the digit capsule $\boldsymbol{v}_j$ with the biggest length $\|\boldsymbol{v}_j\|$ as the predicted label for shape classification. For the shape retrieval task, we use the Euclidean distances between the length vectors $V = [\|\boldsymbol{v}_1\|, \|\boldsymbol{v}_2\|, \ldots, \|\boldsymbol{v}_m\|]$ of point clouds for similarity measurement. Such similarity measurement is in accordance with the way how capsule stores information. What's more, a direct comparison between the length vectors requires less computational cost than comparing representation vectors in capsules.

3) Implementation Details: In this article, we use two multi-scale local feature extraction layer for hierarchically extracting features from point clouds. For the first feature extractor, the input is 1024 points associated with their x, y and z coordinates, from which 512 points is sampled using farthest point sampling. For each sampled point, we select $[8, 16, 32, 64]$ nearest neighbor points of four scales. The MLPs used in the first block have $[32, 32, 64]$ units for each layer. The second feature extractor samples 256 points out of the 512 points. The number of points for kNN search is the same as the first block. The MLPs for the second block have the units of $[64, 64, 128]$ for each layer. The parameter r for multi-scale shuffling is 2. The number of the cluster centers is $Q = 64$ and the dimension is $C = 256$. In the rearrange and squashing layer, we split each embedding into 16 16-dimensional short vectors, which form 1024 16-dimensional spatial-aware capsules in total.

TABLE I

THE SHAPE CLASSIFICATION ACCURACY (%) COMPARISON ON MODELNET10 (MN10) AND MODELNET40 (MN40)

Categories	Methods	Inputs	MN10	MN40
Views	SeqViews2SeqLabels [8]	Multi-views	94.7	93.3
	VIP-GAN [6]	Multi-views	90.8	88.9
	Parts4Feature [7]	Multi-views	96.2	93.4
	3D2SeqViews [43]	Multi-views	94.7	93.4
	3DViewGraph [2]	Multi-views	94.8	93.8
Voxels	VConv-DAE [54]	voxel	-	79.8
	3DShapeNet [52]	voxel	-	83.0
	O-CNN [40]	voxel	-	90.2
Points	PointNet [1]	1024pts	-	89.2
	PointNet++ [25]	1024pts	-	90.7
	SA-Net [73]	1024pts	-	90.6
	SCN [69]	1024pts	-	90.0
	KC-Net [28]	1024pts	94.4	91.0
	PointCNN [26]	1024pts	-	91.7
	DGCNN [31]	1024pts	-	92.9
	PATs [74]	1024pts	-	92.2
	PointWeb [33]	1024pts	-	92.3
	A-CNN [3]	1024pts	95.5	92.6
	RS-CNN [32]	1024pts	-	92.6
	PointConv [75]	1024pts	-	92.5
	3DCapusle [66]	1024pts	-	91.5
	MAP-VAE [45]	1024pts	94.8	90.2
	Point2Sequence [10]	1024pts	95.3	92.6
	Chen et al.[76]	1024pts	95.6	89.7
	Sauder et al.[77]	1024pts	-	92.4
	Point2Node [78]	1024pts	-	93.0
	GS-Net [38]	1024pts	-	92.9
	SK-Net [36]	1024pts	95.8	91.5
	InterpCNN [39]	1024pts	-	93.0
	DensePoint [34]	1024pts	-	92.8
	KPConv [37]	1024pts	-	92.9
	ShellNet [35]	1024pts	-	93.1
	DPAM [79]	1024pts	-	91.9
	3D-GCN [80]	1024pts	-	92.1
	Point2SpatialCapsule(Ours)	1024pts	**95.8**	**93.4**
	Kd-Net [24]	$2^{15} \times 3$	94.0	91.8
	SO-Net [27]	2048pts	94.1	90.9
	Ψ-tree [11]	2048pts	94.6	92.0
	RS-CNN [32]	1024pts + vote	-	93.6
	PointNet++ [25]	1024pts + norm	-	91.9
	SPH3D-GCN [81]	10000pts	-	92.1
	AGCN [82]	2048pts + norm	-	93.2
	SO-Net [27]	5000pts + norm	95.7	93.4
	DGCNN [31]	2048pts	-	93.5
	DensePoint [34]	1024pts + vote	**96.6**	93.2
	Point2SpatialCapsule(Ours)	1024pts + norm	**95.9**	**93.7**

B. 3D Shape Classification

1) ModelNet: Table I compares Point2SpatialCapsule with the existing state-of-the-art methods of point cloud based representation learning in terms of shape classification accuracy under ModelNet10 and ModelNet40, respectively. For convenience, we arrange the results according to the different types of input (e.g. multi-views, 1024 points, 2048 points, and normals). From Table I, we draw two conclusions to demonstrate the superiority of Point2SpatialCapusule over other methods as follows.:

- On the one hand, in the context of point cloud based representation learning, our Point2SpatialCapsule achieves the best result (93.4% on MN40) under the most widely adopted experimental settings (i.e. 1024pts input without normals). Moreover, when comparing Point2SpatialCapsule with other methods across different experimental settings based on point cloud input, it still achieves the best classification performance (93.7% on MN40), which proves the superiority of our method over the current studies. We specially note that Point2SpatialCapusule outperforms the baseline method PointNet++ (i.e. the backbones of multi-scale local feature extraction module in Point2SpatialCapsules) by 2.7%. Specially, Point2Capusule with additional normal vectors achieves the best results (93.7% on MN40), which reach the highest classification accuracy across all experiment settings based on point cloud inputs.
- On the other hand, when comparing Point2Spatial Capusule with some recent multi-view based methods, Point2SpatialCapuslue still has the ability to achieve an equivalent classification performance, where the highest accuracy of multi-view based studies is 93.8% [2] and our Point2SpatialCapsule achieves 93.7%. Note that multi-view based methods can leverage more information (such as texture, illumination and RGB colors) compared with point cloud based methods. As a result, multi-view based methods usually achieve better classification performances than point cloud based methods. Such conclusion can also be drawn from Table I by comparing other point cloud based methods with different multi-view based methods. However, the 2D view inputs significantly limit the potential applications of multi-view based methods in other 3D deep learning tasks (e.g. 3D semantic segmentation, 3D instance segmentation and 3D object detection). In contrast, point cloud based methods can be directly applied to a wider range of 3D applications than multi-view based methods.

In detail, we point out that both PointNet++ and Point2SpatialCapsule use a multi-scale local feature extraction strategy, where the difference lies in the method used for aggregating local features. The PointNet++ applies max-pooling for aggregating the local features, while Point2SpatialCapsule uses the geometric feature aggregation with spatial relationship aggregation for learning the global representation. Therefore, the improvement in classification accuracy of Point2SpatialCapsule proves the effectiveness of the proposed network for local feature aggregations.

3DCapsule [66] is the work most related to our Point2SpatialCapsule in Table I. As already discussed in Sec.II, 3DCapsule simply applies the capsule network on the global features produced by a pooling/full-connected layer, which falls into the scenario of information loss of the spatial locations. In contrast, our Point2SpatialCapsule applies dynamic routing on the feature-spatial embeddings generated by the geometric feature aggregation module, which can aggregate both the features and their spatial location. The experimental results in Table I shows the implementation of capsule network in our Point2SpatialCapsule is more effective than the implementation of 3DCapsule.

Both PointCNN and DGCNN are CNN-based neural networks, which aim to preserve the spatial locations and spatial relationships of local regions. However, both of them use the max-pooling for aggregating the local region features, which filters out the spatial locations and relationships, especially when aggregating the local features into the global features. Therefore, the dynamic routing based Point2SpatialCapsule yields better performance than the PointCNN and DGCNN, which demonstrates the superior advantages of Point2SpatialCapsule for preserving the spatial locations and relationships.

As seen in Table I, our Point2SpatialCapsule outperforms most of the xyz-input methods on point clouds. Specifically, our result is ranked the first place under ModelNet10 (95.8%), and ranked the second place under ModelNet40 (93.4%) which is slightly lower than RS-CNN [32] by 0.2%. As claimed in [32], RS-CNN performed "ten voting tests with random scaling and averages the predictions" during testing. In contrast, we only apply the single model prediction for fair comparison with most of the existing methods [24], [27], [66]. Moreover, when using additional normal vectors as the input, the proposed Point2SpatialCapsule can achieve the best performance among all reported results under ModelNet10 (95.9%) and ModelNet40 (93.7%), respectively. This convincingly verifies the effectiveness of Point2SpatialCapsule.

When compare Point2SpatialCapsule with graph-based neural networks (e.g. AGCN, 3D-GCN and SPH3D-GCN), Point2SpatialCapsule still leads the best performance on ModelNet dataset. Although these graph-based neural network have greatly improved the performance of learning the relationships between discrete points, these networks still fail by using pooling-based feature aggregation, which loses information. And this is exactly addressed by our Point2SpatialCapsule (as we have discussed in the Introduction Section). More specifically, most of these graph-based neural networks still aggregated the learned features/weights between points/regions by pooling-based layer (i.e. both RS-CNN and 3D-GCN used the channel-wise max-pooling to aggregate relationships between center point and its neighbors), which can only preserve the most prominent features. In contrast, the Point2SpatialCapsule uses the dynamic routing technic to allow the network preserve all the features with their spatial relationships. By resolving these issues, Point2SpatialCapsule consistently outperforms these state-of-the-art methods on point cloud representation learning.

2) ScanNet: We typically compare Point2SpatialCapsule with three commonly used baselines (i.e. PointNet, PointNet++ and DGCNN), where PointNet++ is part of the backbone of Point2SpatialCapsule and DGCNN is a graph-based neural network. The results of all compared methods are produced by following their source code released by each paper, as shown in Table II. From the comparison we can find that Point2SpatialCapsule outperforms the other three methods by a non-trivial margin. Since objects in ScanNet are obtained by real scans, they are usually sparse and incomplete (as illustrated in Figure 9). Therefore, in order to correctly predict labels for 3D objects, the network is required to integrate the local region features as a whole (by dynamic routing), instead of only focusing on prominent part (by max-pooling layer). And this is the reason that Point2SpatialCapsule outperforms previous methods on the real scan dataset.

C. 3D Shape Retrieval

In Table III, we compare the proposed Point2Capusules with counterpart methods in 3D shape retrieval task, in terms of *mean average precisions* (mAPs). Since most of the methods focusing on 3D shape retrieval are based on multi-views of 3D models, in this subsection, we also quote the experimental results of the multi-view based methods to verify the effectiveness of Point2SpatialCapsule. Note that, the results of PointNet and PointNet++ are obtained by following the same training procedure as described in their original papers, which are denoted by $*$ in this table.

As shown in Table III, our method has achieved a comparable retrieval accuracy compared with multi-view based methods on both ModelNet10 and ModelNet40. Specifically,

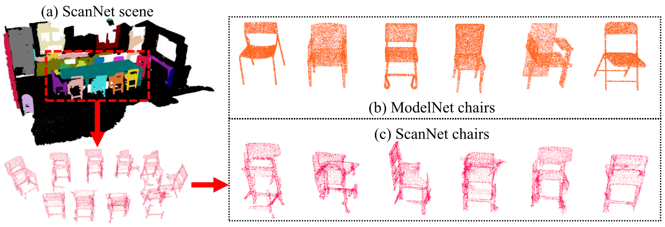


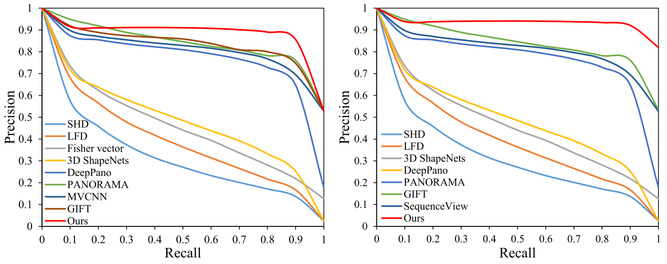
Fig. 9. The visual comparison of shape quality between ScanNet dataset and ModelNet dataset. The classification task on ScanNet objects is more difficult due to shape incompleteness and point cloud sparsity.

TABLE II
THE SHAPE CLASSIFICATION ACCURACY (%) COMPARISON ON SCANNET V2

Method	Ins.
PointNet [1]	85.3
PointNet++ [25]	87.8
DGCNN [31]	88.9
Point2SpatialCapsule(Ours)	**90.8**

TABLE III
THE SHAPE RETRIEVAL ACCURACY IN TERMS OF MAPS ON MODELNET10 AND MODELNET40

Method	Input	ModelNet10	ModelNet40
PointNet* [1]	1024 × 3	67.98	62.41
PointNet++* [25]	1024 × 3	72.52	68.97
3DShapeNets [72]	Multi-View	68.26	49.23
DeepPano [47]	Multi-View	84.18	76.81
MVCNN [83]	Multi-View	-	83.0
PANORAMA [48]	Multi-View	87.39	83.45
GIFT [44]	Multi-View	91.12	81.94
SequenceViews [8]	Multi-View	89.55	89.0
SPNet [84]	Multi-View	**94.20**	85.21
VIPGAN [6]	Multi-View	90.69	89.23
Point2SpatialCapsule(Ours)	1024pts	**93.43**	**89.43**



(a) ModelNet40

(b) ModelNet10

Fig. 10. The comparison of precision and recall curves obtained by different methods under (a) ModelNet40 and (b) ModelNet10.

TABLE IV
THE ACCURACIES (%) OF PART SEGMENTATION ON SHAPENET PART SEGMENTATION DATASET

	mean	Intersection over Union (IoU)															
		air.	bag	cap	car	cha.	ear.	gui.	kni.	lam.	lap.	mot.	mug	pis.	roc.	ska.	tab.
# SHAPES		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
PointNet [1]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++ [25]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
SCN [69]	84.6	83.8	80.8	83.5	79.3	90.5	69.8	91.7	86.5	82.9	96.0	69.2	93.8	82.5	62.9	74.4	80.8
Point2Seq [10]	85.2	82.6	81.8	87.5	77.3	90.8	77.1	91.1	86.9	83.9	95.7	70.8	94.6	79.3	58.1	75.2	82.8
Kd-Net [24]	82.3	80.1	74.6	74.3	70.3	88.6	73.5	90.2	87.2	81.0	94.9	57.4	86.7	78.1	51.8	69.9	80.3
O-CNN [40]	85.2	84.2	86.9	84.6	74.1	90.8	81.4	91.3	87.0	82.5	94.9	59.0	94.9	79.7	55.2	69.4	84.2
KCNet [28]	84.7	82.8	81.5	86.4	77.6	90.3	76.8	91.0	87.2	84.5	95.5	69.2	94.4	81.6	60.1	75.2	81.3
DGCNN [31]	85.1	84.2	83.7	84.4	77.1	90.9	78.5	91.5	87.3	82.9	96.0	67.8	93.3	82.6	59.7	75.5	82.0
SO-Net [27]	84.9	82.8	77.8	88.0	77.3	90.6	73.5	90.7	83.9	82.8	94.8	69.1	94.2	80.9	53.1	72.9	83.0
RS-CNN [32]	86.2	83.5	84.8	88.8	79.6	91.2	81.1	91.6	88.4	86.0	96.0	73.7	94.1	83.4	60.5	77.7	83.6
PointCNN [26]	86.1	84.1	86.5	86.0	80.8	90.6	79.7	92.3	88.4	85.3	96.1	77.2	95.3	84.21	64.23	80.0	83.0
Ours	85.3	83.5	83.4	88.5	77.6	90.8	79.4	90.9	86.9	84.3	95.4	71.7	95.3	82.6	60.6	75.3	82.5

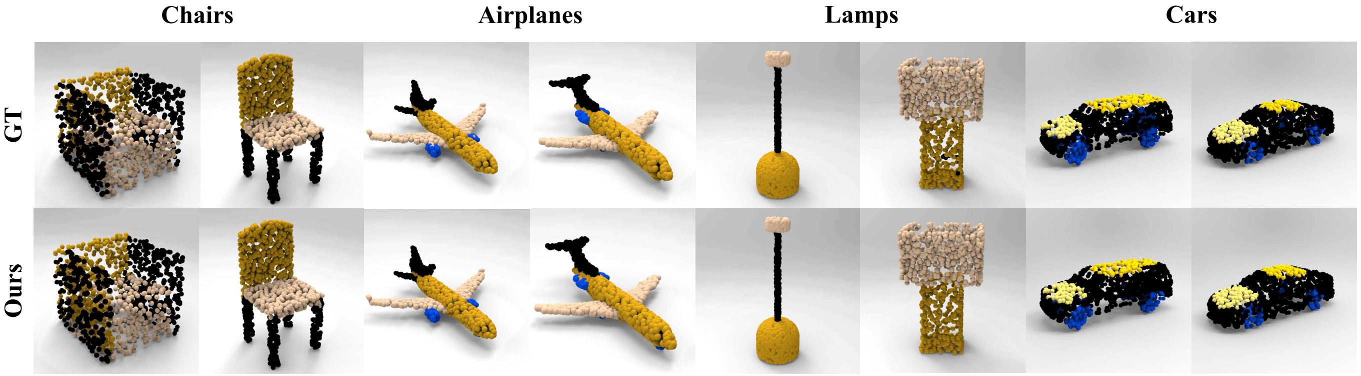


Fig. 11. Visualization of part segmentation results. In each shape pair, the first row is the ground truth (GT), and the second row is our predicted result. Parts with the same color are in the same part class.

Point2SpatialCapsule achieves the best retrieval accuracy 89.43% on ModelNet40, among all reported retrieval results. Point2Capusles achieves the second place result (93.43%) on ModelNet10, which is slight lower than SPNet [84] by 0.77%. However, Point2SpatialCapsule still beats SPNet by 4.22% on ModelNet40 in terms of mAPs, which shows a more balanced performance of Point2SpatialCapsule over different scales of datasets. The comparison of precision-recall (PR) curves under ModelNet40 and ModelNet10 are shown in Fig. 10, where the results of Point2SpatialCapsule show the high performance for the 3D shape retrieval task.

The better performance of Point2SpatialCapsule can be dedicated to the following two reasons. First, the Point2SpatialCapsule is able to learn to encode the spatial locations of local features, which can produce a more discriminative representation for point clouds. Second, the digit capsules provide a more interpretable features for representing the point clouds, which is the length vector. Compared with the traditional single vector representations, in which the high-level characteristics are implicitly encoded in the latent feature space, the length of digit capsule explicitly indicates the probability that the characteristics appear in the point clouds. Therefore, using the distance between length vectors of digit capsules is more effective and interpretable for 3D shape retrieval.

D. 3D Shape Part Segmentation

In Table IV, we also report the performance of Point2SpatialCapsule on the part segmentation task in terms of the Intersection over Union (IoU) [1]. As shown in Table IV, our Point2SpatialCapsule achieves the mean instance IoU of 85.3%, which outperforms the baseline method PointNet++ on 13 categories out of total 16 categories. Note that, same as PointNet++, Point2SpatialCapsule also employs the multi-scale sampling and grouping strategy for local feature extraction. Therefore, the experimental results prove that Point2Sequence improves the quality of local feature extraction, and leads to the better performance on the segmentation task. Fig. 11 visualizes some examples of our segmentation results, where our results are highly consistent with the ground truth.

Note that, segmentation application needs discriminative features of local regions. Although Point2Capsule is proposed for global shape features by encoding the information of spatial locations in local regions, rather than producing more discriminative features of local regions like RS-CNN [32] and PointCNN [26], we still achieve comparable results in segmentation results.

E. Ablation Studies

In this section, we keep the settings of the network the same as described in Sec.III, except for the specified part for ablation study. We first investigate the influence of each part to our model, and then we analyze three important hyper-parameters in terms of classification accuracy on ModelNet40.

1) The Influence of Each Part to Point2SpatialCapsule: In order to investigate the effect of each part in Point2

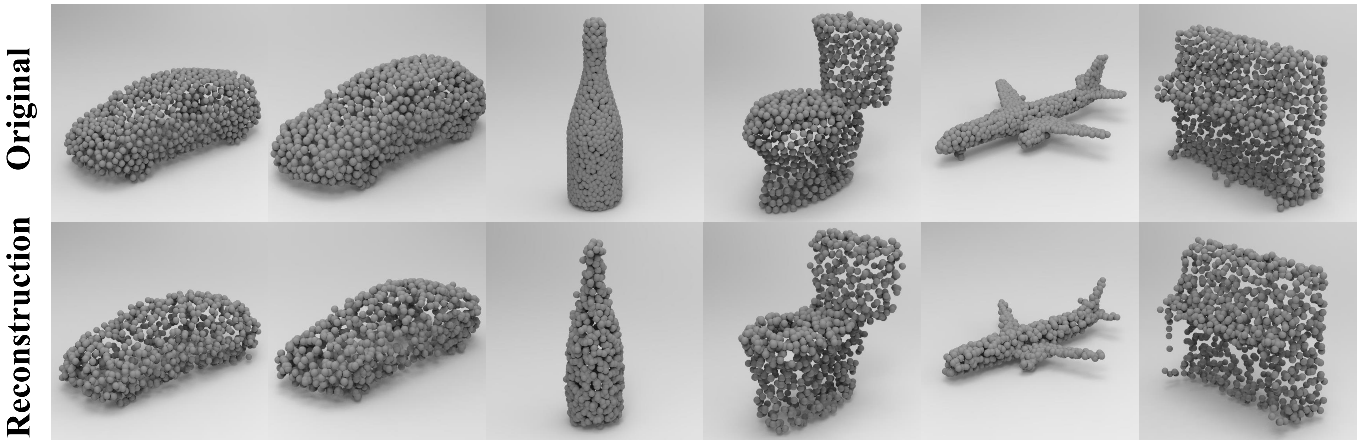


Fig. 12. The visualization of reconstruction results on the test set of ModelNet40. The top roll is the input original point cloud, and the bottom roll is the reconstructed point cloud from the Point2SpatialCapsule's reconstruction network.

TABLE V

THE EFFECT OF EACH PART OF POINT2SPATIALCAPSULE ON MODELNET40

Model	No-Multi	No-VLAD	No-Caps	Full-Model
Acc (%)	92.5	91.4	92.1	**93.44**

TABLE VI

THE EFFECT OF THE ITERATIONS OF DYNAMIC ROUTING ON MODELNET40

Iterations	1	3	5
Acc (%)	**93.44**	92.22	91.98

TABLE VII

THE INFLUENCE OF THE NUMBER OF CLUSTER CENTERS IN GEOMETRIC FEATURE AGGREGATION MODULE ON MODELNET40

Number	16	32	64	128
Acc (%)	92.70	92.93	**93.44**	93.21

TABLE VIII

THE INFLUENCE OF RECONSTRUCTION LOSS ON MODELNET40

α	10^{-3}	10^{-4}	10^{-5}	0
Acc (%)	91.69	**93.44**	92.79	92.46

SpatialCapsule, we develop and evaluate three different variations of our model as follows. (1) 'No-Multi' is the model without multi-scale shuffling, where the output of the multi-scale feature extractor is the direct input to the soft-assignment layer. (2) 'No-VLAD' is the model without the geometric feature aggregation, where the output of multi-scale feature extraction layer is directly reshaped as spatial-aware capsules and input to the dynamic routing layer. (3) 'No-Caps' is the model without the capsule net, where the output of geometric feature aggregation module is concatenated as a single vector and directly fed into the fully-connected layer for shape classification.

The experimental results are shown in Table V. From the results we can find that each part of Point2SpatialCapsule contributes to the model performance. We note that, the No-VLAD model achieves the worst performance among the four models, which means that directly applying the capsule network on the point cloud impairs the model's representational ability. The result of No-VLAD model supports our point of view that dynamic routing cannot learn the log priors directly from the disordered point clouds, and verifies the effectiveness of the proposed geometric feature aggregation. The results of No-Multi proves the importance of applying multi-scale shuffling for smoothing the perceived range between features of different scales. The significant improvement of Full-Model compared to the No-Caps verifies the superior advantage of capsule network for aggregating local features in point cloud recognition.

2) The Analysis of Capsules Net: Following the common practice of [29], we investigate the influence of iterations in dynamic routing. As shown in Table VI, we report the model performance with 1, 3 and 5 iterations of dynamic routing. According to [29], multiple iterations will increase the model's learning ability but may also cause the problem of overfitting. As for Point2SpatialCapsule, we find that dynamic routing with 1 iteration is already enough for learning the point cloud features.

3) The Analysis of Geometric Feature Aggregation: We also analysis the influence of cluster centers in NetVLAD. As shown in Table VII, the model achieves the best result with 64 cluster centers. The explanations are two-fold: (1) the small number of cluster centers could reduce the representational ability of feature embeddings; (2) the slight reduce in performance of the large number of cluster centers is the result of producing similar feature embeddings, which leads to the information redundancy and hinders the model learning more discriminative local features.

4) The Analysis of Reconstruction Loss: In Table VIII, we discuss the influence of reconstruction loss, where α is the weight factor as specified in Eq. (10). From the results, we find that a large α leads to the decreasing of model performance, which in our opinion is the result of a slower learning process cause by the large reconstruction loss weight, especially during the early stage of training. On the other hand, the experimental results also prove the reconstruction loss useful. Compared with a small weight ($\alpha = 10^{-5}$) and the model without reconstruction ($\alpha = 0$), the model with $\alpha = 10^{-4}$ outperforms them by 0.65% and 0.98%, respectively. In Fig. 12, we visualize the reconstruction results on the test set of ModelNet40, from which we can find that Point2SpatialCapsule can learn to produce a relatively satisfactory result, despite of the simple reconstruction network employed in the model.

V. Conclusion

In this article, we propose a spatial-aware network, named Point2SpatialCapsule, to jointly aggregate geometric feature and spatial relationships of local regions on point cloud. The proposed Point2SpatialCapsule has a wide range of potential applications, which can be combined with other local feature extraction methods of multi-scale regions for learning the global shape representation of 3D point clouds. Compared with the previous feature aggregation methods, Point2SpatialCapsule has the ability to integrate both the geometric features of local regions and the spatial relationships among them. The features of local regions are aggregated by spatial-ware capsules with dynamic routing, which can preserve the spatial relationships between the extracted features. Experiments show that our network can achieve superior performance on point cloud classification, retrieval and part segmentation tasks under differen datasets.

References

[1] 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.*, Jul. 2017, pp. 652–660.

[2] Z. Han, X. Wang, C. M. Vong, Y.-S. Liu, M. Zwicker, and C. L. P. Chen, "3DViewGraph: Learning global features for 3D shapes from a graph of unordered views with attention," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp.758–765.

[3] A. Komarichev, Z. Zhong, and J. Hua, "A-CNN: Annularly convolutional neural networks on point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 7421–7430.

[4] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and X. Li, "Unsupervised 3D local feature learning by circle convolutional restricted Boltzmann machine," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5331–5344, Nov. 2016.

[5] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and C. L. P. Chen, "Mesh convolutional restricted Boltzmann machines for unsupervised learning of features with structure preservation on 3-D meshes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2268–2281, Oct. 2017.

[6] 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. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 8376–8384.

[7] Z. Han, X. Liu, Y.-S. Liu, and M. Zwicker, "Parts4Feature: Learning 3D global features from generally semantic parts in multiple views," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 766–773.

[8] Z. Han *et al.*, "SeqViews2SeqLabels: Learning 3D global features via aggregating sequential views by RNN with attention," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 658–672, Feb. 2019.

[9] J. Xie, G. Dai, and Y. Fang, "Deep multimetric learning for shape-based 3D model retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 11, pp. 2463–2474, Nov. 2017.

[10] X. Liu, Z. Han, Y.-S. Liu, and M. Zwicker, "Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network," in *Proc. 33rd AAAI Conf. rtif. Intell.*, 2019, pp. 8778–8785.

[11] H. Lei, N. Akhtar, and A. Mian, "Octree guided CNN with spherical kernels for 3D point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 9631–9640.

[12] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4096–4105.

[13] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4421–4430.

[14] X. Wen, Z. Han, G. Youk, and Y.-S. Liu, "CF-SIS: Semantic-instance segmentation of 3D point clouds by context fusion with self-attention," in *Proc. ACM Int. Conf. Multimedia (ACM MM)*, 2020.

[15] Z. Han, C. Chen, Y.-S. Liu, and M. Zwicker, "ShapeCaptioner: Generative caption network for 3D shapes by learning a mapping from parts detected in multiple views to sentences," in *Proc. ACM Int. Conf. Multimedia (ACM MM)*, 2020.

[16] Z. Han, B. Ma, Y.-S. Liu, and M. Zwicker, "Reconstructing 3D shapes from multiple sketches using direct shape optimization," *IEEE Trans. Image Process.*, vol. 29, pp. 8721–8734, Sep. 2020.

[17] Z. Han, G. Qiao, Y.-S. Liu, and M. Zwicker, "SeqXY2SeqZ: Structure learning for 3D shapes by sequentially predicting 1D occupancy segments from 2D coordinates," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020.

[18] Z. Han, C. Chen, Y.-S. Liu, and M. Zwicker, "DRWR: A differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020.

[19] X. Zhou, Z. Shao, W. Zeng, and J. Liu, "Semantic graph construction for 3D geospatial data of multi-versions," *Optik*, vol. 125, no. 6, pp. 1730–1734, Mar. 2014.

[20] Z. Shao, N. Yang, X. Xiao, L. Zhang, and Z. Peng, "A multi-view dense point cloud generation algorithm based on low-altitude remote sensing images," *Remote Sens.*, vol. 8, no. 5, p. 381, May 2016.

[21] Z. Shao, L. Zhang, and L. Wang, "Stacked sparse autoencoder modeling using the synergy of airborne LiDAR and satellite optical and SAR data to map forest above-ground biomass," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 12, pp. 5569–5582, Dec. 2017.

[22] L. Zhang, Z. Shao, J. Liu, and Q. Cheng, "Deep learning based retrieval of forest aboveground biomass from combined LiDAR and landsat 8 data," *Remote Sens.*, vol. 11, no. 12, p. 1459, Jun. 2019.

[23] X. Li, X. Yao, and Y. Fang, "Building-A-nets: Robust building extraction from high-resolution remote sensing images with adversarial networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 10, pp. 3680–3687, Oct. 2018.

[24] R. Klokov and V. Lempitsky, "Escape from cells: Deep kd-networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 863–872.

[25] 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. 5099–5108.

[26] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 820–830.

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

[28] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4548–4557.

[29] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3856–3866.

[30] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 87–102.

[31] 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, Nov. 2019.

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

[33] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing local neighborhood features for point cloud processing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5565–5573.

[34] Y. Liu, B. Fan, G. Meng, J. Lu, S. Xiang, and C. Pan, "DensePoint: Learning densely contextual representation for efficient point cloud processing," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 5239–5248.

[35] Z. Zhang, B.-S. Hua, and S.-K. Yeung, "ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 1607–1616.

[36] W. Wu, Y. Zhang, D. Wang, and Y. Lei, "SK-Net: Deep learning on point cloud via end-to-end discovery of spatial keypoints," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2020, pp. 6422–6429.

[37] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 6411–6420.

[38] M. Xu, Z. Zhou, and Y. Qiao, "Geometry sharing network for 3D point cloud classification and segmentation," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2020, pp. 12500–s12507.

[39] J. Mao, X. Wang, and H. Li, "Interpolated convolutional networks for 3D point cloud understanding," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 1578–1587.

[40] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, p. 72, 2017.

[41] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3577–3586.

[42] H. You, Y. Feng, R. Ji, and Y. Gao, "PVNet: A joint convolutional network of point cloud and multi-view for 3D shape Recognition," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 1310–1318.

[43] Z. Han *et al.*, "3D2SeqViews: Aggregating sequential views for 3D global feature learning by CNN with hierarchical attention aggregation," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 3986–3999, Aug. 2019.

[44] S. Bai, X. Bai, Z. Zhou, Z. Zhang, Q. Tian, and L. J. Latecki, "GIFT: Towards scalable 3D shape retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1257–1271, Jun. 2017.

[45] 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 IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10441–10450.

[46] Z. Han, M. Shang, X. Wang, Y.-S. Liu, and M. Zwicker, "Y2Seq2Seq: Cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 126–133.

[47] B. Shi, S. Bai, Z. Zhou, and X. Bai, "DeepPano: Deep panoramic representation for 3-D shape recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2339–2343, Dec. 2015.

[48] K. Sfikas, T. Theoharis, and I. Pratikakis, "Exploiting the PANORAMA representation for convolutional neural network classification and retrieval," in *Proc. Eurograph. Workshop 3D Object Retr.*, 2017, pp. 1–7.

[49] Y.-S. Liu, K. Ramani, and M. Liu, "Computing the inner distances of volumetric models for articulated shape description with a visibility graph," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2538–2544, Dec. 2011.

[50] C. Wang, Y.-S. Liu, M. Liu, J.-H. Yong, and J.-C. Paul, "Robust shape normalization of 3D articulated volumetric models," *Comput.-Aided Design*, vol. 44, no. 12, pp. 1253–1268, Dec. 2012.

[51] M. Ren, L. Niu, and Y. Fang, "3D-A-nets: 3D deep dense descriptor for volumetric shapes with adversarial networks," 2017, *arXiv:1711.10108*. [Online]. Available: http://arxiv.org/abs/1711.10108

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

[53] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and C. L. P. Chen, "Unsupervised learning of 3-D local features from raw voxels based on a novel permutation voxelization strategy," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 481–494, Feb. 2019.

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

[55] J. Xie, Y. Fang, F. Zhu, and E. Wong, "DeepShape: Deep learned shape descriptor for 3D shape matching and retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1275–1283.

[56] J. Xie, M. Wang, and Y. Fang, "Learned binary spectral shape descriptor for 3D shape correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3309–3317.

[57] Z. Han *et al.*, "BoSCC: Bag of spatial context correlations for spatially enhanced 3D shape representation," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3707–3720, Aug. 2017.

[58] Z. Han *et al.*, "Deep spatiality: Unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3049–3063, Jun. 2018.

[59] A. Jaiswal, W. Abdalmageed, Y. Wu, and P. Natarajan, "CapsuleGAN: Generative adversarial capsule network," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 526–535.

[60] R. LaLonde and U. Bagci, "Capsules for object segmentation," 2018, *arXiv:1804.04241*. [Online]. Available: http://arxiv.org/abs/1804.04241

[61] M. Yang, W. Zhao, J. Ye, Z. Lei, Z. Zhao, and S. Zhang, "Investigating capsule networks with dynamic routing for text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3110–3119.

[62] L. Xiao, H. Zhang, W. Chen, Y. Wang, and Y. Jin, "MCapsNet: Capsule network for text with multi-task learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4565–4574.

[63] N. Zhang, S. Deng, Z. Sun, X. Chen, W. Zhang, and H. Chen, "Attention-based capsule networks with dynamic routing for relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 986–992.

[64] B. Kakillioglu, A. Ahmad, and S. Velipasalar, "Object classification from 3D volumetric data with 3D capsule networks," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2018, pp. 385–389.

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

[66] A. Cheraghian and L. Petersson, "3DCapsule: Extending the capsule architecture to classify 3D point clouds," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1194–1202.

[67] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4470–4479.

[68] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5297–5307.

[69] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional ShapeContextNet for point cloud recognition," in *Proc. IEEE IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4606–4615.

[70] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1874–1883.

[71] 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. (CVPR)*, Jul. 2017, pp. 5828–5839.

[72] M. Savva *et al.*, "SHREC'16 track large-scale 3D shape retrieval from ShapeNet core55," in *Proc. Eurograph. Workshop 3D Object Retr.*, 2016, pp. 39–50.

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

[74] J. Yang *et al.*, "Modeling point clouds with self-attention and gumbel subset sampling," in *Proc. IEEE IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3323–3332.

[75] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 9621–9630.

[76] 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.

[77] J. Sauder and B. Sievers, "Self-supervised deep learning on point clouds by reconstructing space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 12942–12952.

[78] W. Han, C. Wen, C. Wang, X. Li, and Q. Li, "Point2Node: Correlation learning of dynamic-node for point cloud feature modeling," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2020.

[79] J. Liu, B. Ni, C. Li, J. Yang, and Q. Tian, "Dynamic points agglomeration for hierarchical point sets learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 7546–7555.

[80] Z.-H. Lin, S.-Y. Huang, and Y.-C.-F. Wang, "Convolution in the cloud: Learning deformable kernels in 3D graph convolution networks for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1800–1809.

[81] H. Lei, N. Akhtar, and A. Mian, "Spherical kernel for efficient graph convolution on 3D point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Mar. 31, 2020, doi: 10.1109/TPAMI.2020.2983410.

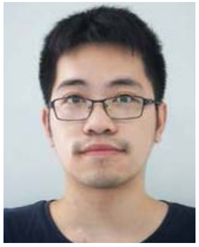
[82] Z. Xie, J. Chen, and B. Peng, "Point clouds learning with attention-based graph convolution networks," *Neurocomputing*, vol. 402, pp. 245–255, Aug. 2020.

[83] C. R. Qi, H. Su, M. NieBner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5648–5656.

[84] M. Yavartanoo, E. Y. Kim, and K. M. Lee, "SPNet: Deep 3D object classification and retrieval using stereographic projection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 691–706.

Xin Wen (Graduate Student Member, IEEE) received the B.S. degree in engineering management from Tsinghua University, China, in 2012, where he is currently pursuing the Ph.D. degree with the School of Software. His research interests include deep learning, shape analysis and pattern recognition, and NLP.

Zhizhong Han received the Ph.D. degree from Northwestern Polytechnical University, China, in 2017. He is currently a Postdoctoral Researcher with the Department of Computer Science, University of Maryland at College Park, College Park, USA. He is also a Research Member of the BIM Group, Tsinghua University, China. His research interests include machine learning, pattern recognition, feature learning, and digital geometry processing.



Xinhai Liu received the B.S. degree in computer science and technology from the Huazhong University of Science and Technology, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Software, Tsinghua University. His research interests include deep learning, 3D shape analysis, and 3D pattern recognition.

Yu-Shen Liu (Member, IEEE) received the B.S. degree in mathematics from Jilin University, China, in 2000, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2006. From 2006 to 2009, he was a Postdoctoral Researcher with Purdue University. He is currently an Associate Professor with the School of Software, Tsinghua University. His research interests include shape analysis, pattern recognition, machine learning, and semantic search.