Spherical Fractal Convolutional Neural Networks for Point Cloud Recognition

Yongming Rao, Jiwen Lu, Jie Zhou
Department of Automation, Tsinghua University, China
State Key Lab of Intelligent Technologies and Systems, China
Beijing National Research Center for Information Science and Technology, China
raoyongming95@gmail.com; {lujiwen, jzhou}@tsinghua.edu.cn

Abstract

We present a generic, flexible and 3D rotation invariant framework based on spherical symmetry for point cloud recognition. By introducing regular icosahedral lattice and its fractals to approximate and discretize sphere, convolution can be easily implemented to process 3D points. Based on the fractal structure, a hierarchical feature learning framework together with an adaptive sphere projection module is proposed to learn deep feature in an end-to-end manner. Our framework not only inherits the strong representation power and generalization capability from convolutional neural networks for image recognition, but also extends CNN to learn robust feature resistant to rotations and perturbations. The proposed model is effective yet robust. Comprehensive experimental study demonstrates that our approach can achieve competitive performance compared to state-of-the-art techniques on both 3D object classification and part segmentation tasks, meanwhile, outperform other rotation invariant models on rotated 3D object classification and retrieval tasks by a large margin.

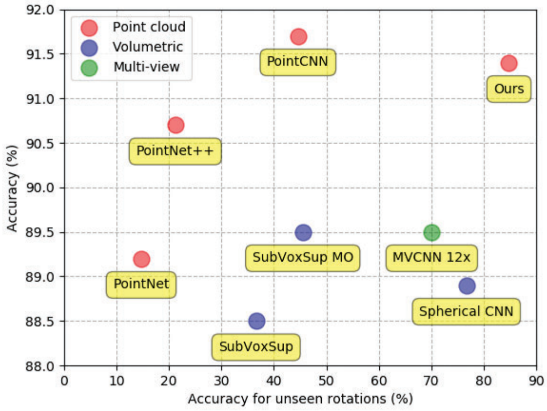

Figure 1. Generalization ability to unseen rotations versus accuracy on ModelNet40. Although previous deep learning algorithms for point cloud show state-of-the-art accuracy, they generalize poorly to unseen orientations. Besides, all other methods suffer a sharp accuracy drop in performance when arbitrary rotations are presented. Our model achieves superior performance on both accuracy and generalization ability.

1. Introduction

Deep learning methods for point cloud processing [16, 18, 22, 6] have attracted great attention recently. Compared to 3D object reasoning techniques based on 3D voxels or collections of images (i.e., views), directly processing 3D points is more challenging. The intrinsic difficulty of point cloud processing comes from its irregular format, which makes capturing local structures of 3D objects costly. To tackle this problem, previous works [18] utilize the set of local points to approximate local structures by dynamically querying the nearest points for each location, which introduces a considerable computation cost during both training and inference, and requires carefully designed module to handle the non-uniform density in different areas.

Point clouds are usually obtained using 3D scanners for real-world applications such as autonomous driving and robotics, where the viewpoints, density and other attributes of points may vary a lot in different scenarios. Therefore, point cloud processing algorithms should be resistant to rotations, perturbations, density variability and other noise coming from sensor and environment. Although several efforts have been devoted to learn robust feature from non-uniform density [18] and 3D rotations [6], the robustness of point cloud processing algorithm is still far from perfect. Existing algorithms usually fail to balance performance and robustness, where models with strong representation capability [16, 18] cannot generalize well to unseen rotations and rotation equivariant algorithms [6, 5] show relatively inferior performance.

Deep convolutional neural networks [12, 20, 9] have led to a series of breakthroughs for image recognition and shown strong representation power and generalization capability in various tasks. One of the reasons for the tremendous success is the hierarchical architecture of CNN, where features from low, middle and high levels are naturally integrated and features can be enriched hierarchically. Benefiting from the regular grid format of image, feature maps can be easily pooled or up-sampled, which allows CNN to learn and enrich features using different receptive fields along a multi-scale hierarchy. Previous success of convolutional neural networks also suggests that it is important to maintain a stable neighboring operation. The stability comes in two ways, a stable selection of neighbors, and the stability of neighbors. For convolutional neural networks, the image grids serve as a good natural regular pattern, which could be easily incorporated with convolutional kernels to guarantee an invariant neighborhood. Such property does not exist in point data, since different point clouds are usually organized in different typologies, where we cannot always maintain a stable selection (e.g., k nearest points) and the stability of neighbors (e.g., points within a radius r) at the same time due to the non-uniform density.

Motivated to address these challenges, we propose an alternative framework for point cloud recognition in this work, named Spherical Fractal Convolutional Neural Networks (SFCNN), to learn deep point cloud features effectively and robustly. Different from existing methods that learning features directly from original set of points or its abstractions, a novel structure that consists of a regular icosahedral lattice and its fractals is introduced to approximate and discretize continuous sphere. More specifically, we design a trainable neural network to project original points onto the fractal structure adaptively, which helps our model resistant to rotations and perturbations while maximally preserve details of the input 3D shapes. Convolution, pooling and upsampling operations can be easily defined and implemented on the lattices. Based on the fractal structure, network structures adopted from CNN based image recognition are proposed to improve the representation power and generalization capability for point cloud recognition. Benefiting from the stability of local operations and spherical symmetry, our model surpasses most previous algorithms on both robustness and effectiveness as presented in Figure 1. Comprehensive experimental study on ModelNet40 classification [27], ShapeNet part segmentation [29] and SHREC'17 perturbed retrieval [19] demonstrates that our approach can achieve competitive performance compared to state-of-the-art techniques on both 3D object classification and part segmentation tasks, meanwhile, outperform other rotation invariant models on rotated 3D object classification and retrieval tasks by a large margin.

2. Related Work

Deep Learning for 3D Object Recognition: Benefiting from deeper and better features, the past few years have witnessed a great development in 3D object recognition. 3D objects can be represented by various formats, which leads to different methods for learning. These methods can be categorized into three categories: view-based methods, volumetric methods and point-based methods. View-based techniques [23] takes a collection of 2D views as input for 3D shape reasoning, where CNNs for image processing can be directly adopted. Typically, a shared CNNs for single view recognition is applied for each view independently and then features from different views are aggregated to a single representation during inference. Volumetric methods [27, 14, 17] apply 3D convolutional neural networks on voxelized shapes, which suffers a lot from the computational bottleneck brought by sparse 3D grids and thus can only built upon relatively shallow networks and low input resolution. Point-based methods is firstly proposed by Qi *et al.* [16], which directly consumes point clouds and thus significantly speed-up 3D shape reasoning. Recent studies on point-based methods [18, 22] show on-par or even better performance on 3D object recognition with much lower computational cost and demonstrate the effectiveness as well as efficiency of this group of methods. However, the robustness of point-based methods has rarely been explored in recent works.

Feature Learning on Irregular Data: Qi *et al.* [16] pioneered a new type of deep learning method on irregular data, which achieves input order invariant feature learning by utilizing symmetry function over 3D coordinates. This work explore feature learning on points via aggregating features individually learned from each point. Local information matters in feature learning, which has been proved by the success of CNN architectures. Follow-up work called PointNet++ [18] improves the original method by exploiting local structures among points, which is achieved by densely querying and fusing neighboring points for each point. Su *et al.* [22] captures local structures in a different way, where original points are mapped into a high-dimensional lattice and thus point clouds can be processed using bilateral convolutional layers. Similar with their method, lattice structure is also introduced in this work to improve the efficiency and stability of point processing, but our method further exploits spherical lattice structure and can generalize to various tasks including classification, part segmentation and retrieval.

Robust Feature Learning: The robustness is essential in real-world applications of point cloud processing systems. There have been some efforts improve the robustness of feature learning algorithm. For example, Qi *et al.* [16] adopted an auxiliary alignment network to predict an affine transformation matrix and applied this transformation on input points and intermediate features to make model resistant to affine transformation. Different from introducing an aux-

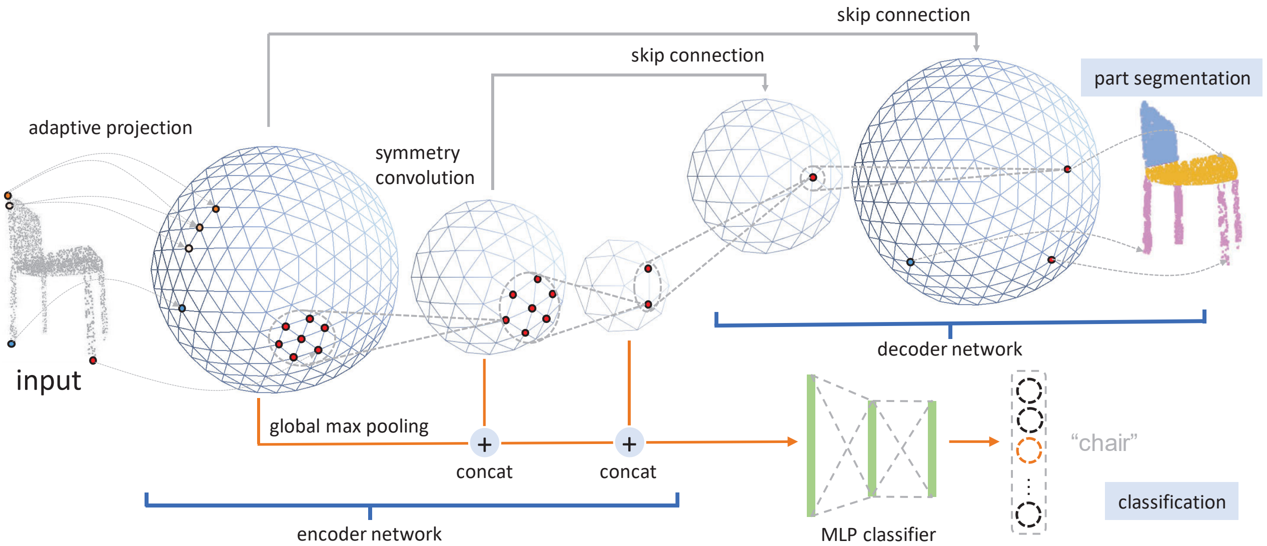

Figure 2. The overall structure of SFCNN. Our proposed feature learning framework can be easily extended to various tasks from point cloud recognition including classification, retrieval and part segmentation. In our framework, input points are adaptively projected onto the discretized sphere. Then, a hierarchical feature learning architecture is designed to capture local and global patterns of point cloud. Features from different hierarchies are summarized to form the representation of input data. Benefiting from the symmetric projection and the hierarchical structure, our framework is effective yet robust.

iliary network, Esteves *et al.* defined several SO(3) equivariant operations on sphere to process 3D data, which can achieve better invariance and generalize well to unseen rotations. However, this model suffers from imperfect projection method and convolution operations defined in spectral domain, which shows poorer capability than spatial convolutions on regular grids. Moreover, spherical CNN is originally designed for voxelized shapes. To the best of our knowledge, this work is the first attempt to study the rotation invariance of point cloud processing algorithm.

Aside from designing robust architecture, data augmentation is also a widely used technique to improve the robustness of neural networks. However, it requires higher model capacity and brings extra computation burdens. Besides, previous study [6] also shows aggressive data augmentation like arbitrary 3D rotations on input data will harm the recognition performance when robust architecture is not used. We show that our model have sufficient capacity to incorporate with different data augmentation methods and it is more robust than others when less augmentations are applied.

3. Approach

We propose an approach inspired by convolutional neural networks for image recognition. Due to the irregular format of point cloud, we firstly map 3D points onto a discretized sphere that is formed by a fractalized regular icosahedral lattice. Convolutional neural networks with multi-scale hierarchy then is defined. Our model can be easily extended to point cloud recognition tasks such as classification and part segmentation. The overall framework of our SFCNN is presented in Figure 2, where a multi-layer perceptron classifier is can be added on features from different hierarchies to perform classification and an encoder-decoder network inspired by similar architecture for image semantic segmentation [1] is designed to conduct part segmentation.

3.1. Preliminaries

The difficulty of point cloud processing mainly comes from the irregular format of points. A natural solution to tackle this challenge is transforming irregular points to a regular format in 2D or 3D, where existing deep learning techniques like 2D and 3D convolutional neural network can be directly used. However, existing volumetric and view-based methods usually suffers from detail losses brought by transformations, where the low resolution of 3D voxelized grids prohibits the usage of local geometric details and the discontinuities across different views leads to poor performance on detail sensitive tasks like shape segmentation. As mentioned above, we project 3D objects onto discretized sphere instead to address these issues. On the one hand, the complexity of conducting neural network algorithms on discretized sphere is $\mathcal{O}(n)$, where n is the number of samples on sphere. Therefore, the complexity of learning on discretized sphere is comparable with point-based method like PointNet and much lower than volumetric and view-based methods. On the other hand, sphere domain is continuous, global and rotation-invariant, allowing

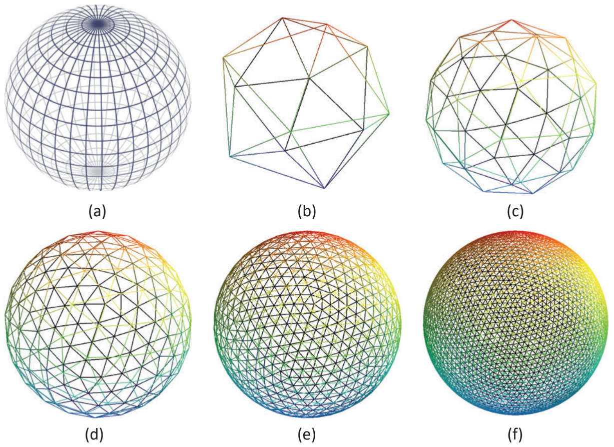

Figure 3. Different spherical discretization methods. (a) is the equiangular sampling. (b)-(f) are discretized spheres produced by the proposed equal-area sampling method with different fractal levels varying from 0 to 4.

our algorithm to capture local structures from complete 3D object while being robust.

Previous works [6, 3] discretize sphere with equiangular sampling, where the cell area varies significantly along latitude. It will lead to significant inconsistency among different rotations and thus requires higher model capacity to learn invariant feature. Instead, we build our model upon spherical lattice with equal area spherical sampling. In practice, we discretize sphere with a regular icosahedron and its fractal to maximally approach sphere, since Platonic solids are the most highly symmetrical among spherical polyhedrons. Note that discretized sphere with perfect symmetry does not exist [6, 25]. Nevertheless, our empirical study shows that is can be overcome by feature learning process with proper data augmentation. The differences between equiangular sampling and ours is shown in Figure 3.

3.2. Detail-preserving Spherical Projection

Consider a point cloud of n points that can be represented as a set of 3D points $X = \{p_1, p_2, ..., p_n\}$, where each point p_i contains 3D coordinates $p_i = (x_i, y_i, z_i)$. In a more generic setting, points can be equipped with additional features representing surface normal, appearance information and so on. Our method projects X to a set of N features $\{F_i | F_i \in \mathbb{R}^n, i = 1, ..., N\}$ on a spherical lattice $L = (V, E)$, where L can be regarded as a undirected graph that comprises N vertices $V = \{v_i | i = 1, ..., N\}$ and a set of corresponding edges E and each feature F_i is associated to an unique vertex v_i.

Different from previous works [6, 25] that project points through a hand-craft rule, a PointNet-like parametric projection module are introduced to maximally preserve the details and structures of the input point clouds. In practice, we learn a shared small PointNet model for all vertices, which takes k nearest points of each vertex as inputs and produces a single feature vector as projected features on vertices. It is worth to notice that different from other methods that requires to search k nearest points dynamically, the spherical lattice structure is shared for different inputs and thus preprocessing algorithms like kd-tree can be applied to significantly accelerate searching. Moreover, since the number of vertices can be pre-defined and is independent with point number, the computational cost of our algorithm will not rapidly increase when more points are sampled.

The rotation-variant point coordinates (x, y, z) make features learned by vanilla PointNet projection module varying with different input rotations. This face motivates us to develop the following *Aligned Spherical Coordinate* representation to improve the robustness of spherical projection modules.

Aligned Spherical Coordinate: Since input points are assigned to vertices on the lattice, we can represent the point coordinates p as the sum of vertex coordinates v and offset vector δ_v:

$$p = v + \delta_v. \tag{1}$$

Consider a rotation R that is applied on the input point cloud. We can donate the rotated point p as $p' = v' + \delta_{v'}$, where v' is a new vertex which p' is assigned to. Since only the nearest k points are assigned to the corresponding vertex, we can assume $||v|| >> ||\delta_v||$. In order to make projection module resistant to rotation, we propose a new coordinate p_v, named aligned spherical coordinate, to replace p as a more robust representation. p_v can be obtained by applying a rotation matrix R_v derived from Rodrigues' rotation formula:

$$p_v = R_v p^T, R_v = 2\frac{(v+u)^T(v+u)}{(v+u)(v+u)^T} - I, \tag{2}$$

where u is a unit vector shared for all vertices and points (we use $u = (0, 0, 1)$ in our implementation), I is the identity matrix and R_v is the rotation matrix that can rotates vector from v to u. This transformation aligns all points that are assigned to v to the local coordinate system of v. Intuitively, because all points are rotated toward u, the difference between p_v and p'_v only depends on the local structure around p and thus p_v is robust when it is assigned to different v due to 3D rotation. Since the degree of freedom is not strictly restricted, the transformed points p_v are not perfectly rotation invariant, but by using the proposed coordinate we can significantly reduce the change of input coordinates when rotation is applied on points. Meanwhile, the local structure of each group of k points can be fully preserved. Actually, the change of offset vector can be viewed as a small random shift on input point cloud, which has been used as a data augmentation method in previous point-based algorithms to avoid overfitting [16, 18]. Therefore, our method

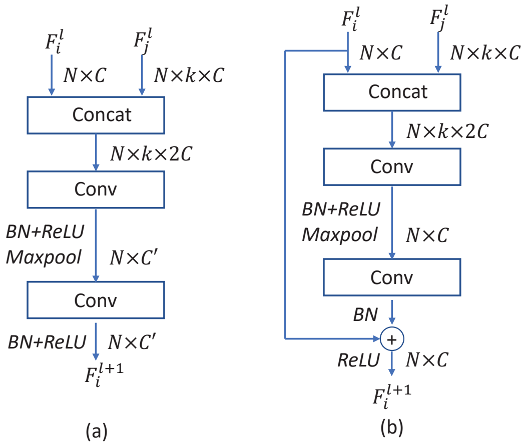

Figure 4. Detailed structure of building blocks. (a) is the basic block for spherical feature learning. The basic block can be used to perform symmetry convolution, feature pooling and up-sampling. (b) is the residual block adopted from [9] to enable deeper feature learning.

can achieve very strong robustness to 3D rotation in applications.

Invertibility Constraint: In our practice, the spherical projection module is jointly trained with the followed CNN model in an end-to-end manner, which greatly increases the difficulty of optimization. We therefore propose a regularization method incorporated with the final objective. Specifically, we constrain the projection to be invertible:

$$L_{inv} = d_{CH}(X, \bigcup_{i}^{N} f(F_i)), \tag{3}$$

where d_{CH} is Chamfer distance, f is a multi-layer perceptron that maps feature on lattice to multiple 3D points. By adding this constraint, the training process can be more stable and models can achieve better generalization capacity and performance.

3.3. Convolutions on Spherical Lattices

Convolution operations can be easily implemented given the regular spherical lattices. Similar with the convolution in 2D CNN, convolution on spherical lattices operates in local regions. For each vertex v_i on spherical lattices, convolution operation takes v_i and its neighboring vertices $\{v_j | d_L(v_i, v_j) = 1\}$ as input, where d_L is the graph distance metric defined on lattice L. Different from convolutions on images, we cannot define a consistent order of neighboring vertices $\{v_j | d_L(v_i, v_j) = 1\}$. Inspired by graph CNN [15] and symmetry function proposed by [16], we achieve symmetry convolution by computing:

$$F_i^{l+1} = \texttt{Conv}(\max_j(\texttt{Conv}(\text{concat}(F_i^l, F_j^l)))), \tag{4}$$

where F_i^l represents feature from the l-th layer at v_i, $\texttt{Conv}$ denotes the convolution with kernel size 1, features from neighboring vertices are concatenated with the feature of v_i along the channel dimension to fuse spatial information while maintain symmetry and channel-wise max-pooling is performed over all neighboring vertices of v_i. The details of our convolutional block is presented in Figure 4, where each block consists of two prevalent Convolution-BatchNorm[10]-ReLU structures and we also adopt the idea of residual learning [9] from image recognition to enable deeper network.

3.4. Spherical Fractal Structure

Given a set of spherical lattices $\{L_i\}, i = 0, 1, .., M$ in different fractal levels, where L_0 represents the regular icosahedral lattice and M is model's the highest fractal level which input points are projected to, we can naturally define a hierarchical feature learning framework based on above-proposed convolution operation. Note that the proposed convolution operation can be directly used for feature learning in the same fractal hierarchy and performing pooling on features from higher fractal level with the number of neighboring vertices as 6. For up-sampling features from lower fractal level, we sample 2 neighboring vertices and use the mean of these two vertices as the new feature if the current vertex does not exist in the last lattice, and just copy the current vertex if it is already in the last level. Because of the imperfect symmetry of spherical lattice, the vertices from the original icosahedral lattice only have 5 neighborhoods satisfying $d_G = 1$. In practice, we do not use the L_0 in the spherical fractal structure to improve the cross-level consistency. Actually, the proposed symmetry convolution is robust to the number of neighboring vertices, and thus defects in lattices will not significantly harm the performance. The network architecture of SFCNN for point cloud classification and retrieval is summarized in Table 1.

For part segmentation task, an encoder-decoder network is used to predict per-point labels. For each points, we concatenate 3D coordinate with features from nearest vertex of different fractal levels to form the final feature of each point.

3.5. Implementation

All of our models can be trained on a single GTX 1080ti GPU. Our models are trained using Adam [11] optimizer with a base learning rate of 0.001, where we decay learning rate by 0.8 every 20 epochs. The models for classification and retrieval tasks are trained for 250 epochs and models for part segmentation are trained for 400 epochs. We fix the mini-batch size to 32 for classification and retrieval tasks and 16 to part segmentation tasks, and set the weight decay as 1e-5 for all tasks. In all of our experiments, we randomly sample points varying from 512 to 1536 to make our models robust to different densities. We randomly dropout [21] the

Table 1. The architecture of SFCNN for classification and retrieval. The number are channels of each block is shown in brackets. Down-sampling is perform at the first block of stage 2, stage 3 and stage 4. N_i represents the number of vertices in the i-th fractal level. We add a `maxpool` layer at the end of MLP projection module to summarize the sampled k neighboring points for each vertex. A `Non-Local` [26] layer is used before the last fully connected layer of projection module to capture the local structures better. C is the number of categories in classification task and K is the channel width.

stage name	output size	architecture
projection	$N_4 \times 16K$	MLP ($8K$, $8K$, $16K$)
stage 1	$N_4 \times 16K$	$\begin{bmatrix} 16K \\ 16K \end{bmatrix} \times B$
stage 2	$N_3 \times 32K$	$\begin{bmatrix} 32k \\ 32k \end{bmatrix} \times B$
stage 3	$N_2 \times 64K$	$\begin{bmatrix} 64K \\ 64K \end{bmatrix} \times B$
stage 4	$N_1 \times 128K$	$\begin{bmatrix} 128K \\ 128K \end{bmatrix} \times 2$
classifier	C	MLP (512, 128, C)

features followed by the classifier with 0.8/0.5 probability for classification/part segmentation task to avoid overfitting. We use 1024 points for all tasks during testing, and voting trick is used to boost performance.

4. Experiments

We conducted experiments on three different benchmark datasets ranging from ModelNet40 classification [27], SHREC'17 perturbed retrieval [19] and ShapeNet part segmentation [29]. The following describes the details of the experiments, results and analysis.

4.1. ModelNet 3D Shape Classification

In this section, we evaluate our model on classification task of ModelNet40 dataset and compare our method with state-of-the-art 3D shape recognition techniques. We also evaluate the robustness of the proposed method through rotated data and perturbations generated by adversarial attack. To better understand the proposed method, we further conducted several ablation experiments.

Main results: ModelNet40 contains 12,311 CAD models of 40 categories. We use the standard split [16, 18], where 9,843 shapes are used for training and 2,468 shapes are selected for testing. Following [6], we evaluated our model using three different settings: 1) training and testing with azimuthal rotations (z/z), 2) training and testing with arbitrary rotations (SO3/SO3), and 3) training with azimuthal rotations while testing with arbitrary rotations (z/SO3).

The results are presented in Table 2. All other models suffer a sharp drop in classification performance in both the z/SO3 and the SO3/SO3 setting, even the SO(3) equivariant method [6] (2% and 12.2% in SO3/SO3 and z/SO3 respectively). It can be observed that our model has a relatively small accuracy drop and consistently outperforms other methods across different settings. Note that some recently proposed point cloud methods like [28] can achieve slightly better performance on the z/z setting than ours. Nevertheless, these algorithms are mainly built upon PointNet and its descendants, which are not robust enough when point cloud is rotated.

We further conducted comprehensive ablation experiments on the proposed framework to examine the effectiveness of our models. Different settings on network architectures and projection modules were tested in our experiments, which is shown in Table 3.

Ablation study on network architecture: We evaluated our model with different numbers of channel and layers. We can see that the performance and generalization ability to unseen rotations consistently increase when deeper and/or wider networks are applied. Our model shows similar property as CNN for image convolutions, which suggests that SFCNN successfully inherits the strong generalization capability of CNN and thus generalize well when the model capacity increases.

Ablation study on projection module: We also conducted experiments on the spherical projection module. Experimental results shows that the number of sampled neighboring points k is crucial and sensitive in our model. When bigger k values are chose, sampling too many points for each vertex harms the locality of vertices and thus this model generalize poorly in both z/z and z/SO3 settings. On the contrary, when much less points are sampled for each vertex, it could be more difficult to capture the local structures of input point cloud but it also improves the locality of vertices. We found models with $k = 16$ achieved superior performance and generalize well to different tasks including retrieval and part segmentation.

Adversarial robustness: The robustness of point cloud algorithm also depends on whether model is resistant to random perturbations. Pervious studies on the robustness of image recognition models show that deep learning algorithm can be easily fooled by adversarial examples, which are some images formed by applying small worst-case perturbations. A natural question is whether 3D recognition algorithm can be fooled by this kind of perturbations. Unsurprisingly, by applying a widely used adversarial attack algorithm, called FGSM [8], we can form adversarial exam-

Table 2. Comparisons of the classification accuracy (%) of our model with state-of-the-art methods on the ModelNet40 dataset. We report the accuracy measured on three benchmarks including z/z, SO3/SO3 and z/SO3. Our model shows superior performance on all three benchmarks. Our model can generalize well even to unseen rotations. † indicates that training data of MVCNN 80x is not restricted to azimuthal.

Method	input	input size	z/z	SO3/SO3	z/SO3
VoxNet [14]	voxel	30^3	83.0	87.3	-
SubVolSup [17]	voxel	30^3	88.5	82.7	36.6
SubVolSup MO [17]	voxel	30^3	89.5	85.0	45.5
Spherical CNN [6]	projected voxel	2×64^2	88.9	86.9	76.7
MVCNN 12x [23]	view	12×224^2	89.5	77.6	70.1
MVCNN 80x [23]	view	80×224^2	90.2	86.0	$81.5^\dagger$
PointNet [16]	xyz	2048×3	89.2	83.6	14.7
PointNet++ [18]	xyz	1024×3	90.7	85.0	21.2
PointNet++ [18]	xyz + normal	5000×6	91.9	85.8	19.7
PointCNN [13]	xyz	1024×3	91.7	84.7	44.5
Ours	xyz	1024×3	91.4	90.1	84.8
Ours	xyz + normal	1024×6	**92.3**	**91.0**	**85.3**

Table 3. Ablation study on ModelNet dataset. All models take 1024 points without surface normal as input. We conducted several ablation experiments to examine the effectiveness of our models. Different settings on channel width K, block number B, sampled neighborhood number k, coordinate alignment and invertibility constraint were tested in our experiments. We show the best results in each group in bold.

Method	z/z	z/SO3
Baseline model (w/ alignment, w/o invertibility)		
Baseline ($K = 4$, $B = 2$, $k = 16$)	90.2	83.2
Architecture		
Wider ×1.5 ($K = 6$, $B = 2$, $k = 16$)	90.5	84.4
Wider ×2 ($K = 8$, $B = 2$, $k = 16$)	90.8	84.7
Deeper ($K = 4$, $B = 3$, $k = 16$)	90.7	83.7
Wider & deeper ($K = 8$, $B = 4$, $k = 16$)	**91.0**	**85.0**
Projection module: k		
Bigger k ($K = 4$, $B = 2$, $k = 64$)	89.5	82.0
Smaller k ($K = 4$, $B = 2$, $k = 4$)	89.7	**83.5**
Projection module: alignment & invertibility		
w/o alignment ($K = 4$, $B = 2$, $k = 16$)	90.3	47.2
w/ invertibility ($K = 4$, $B = 2$, $k = 16$)	**90.8**	**83.7**
Best model		
w/ invertibility ($K = 8$, $B = 3$, $k = 16$)	**91.4**	**84.8**

Table 4. Comparisons of adversarial robustness on ModelNet. Performance of our model, PointNet and PointNet++ against white-box FGSM attacks with different ε is presented. Our model is significantly more robust under adversarial attacks.

	PointNet	PointNet++	Ours
Baseline	89.6	90.7	91.4
FGSM $\varepsilon = 0.002$	44.7	47.5	69.4
FGSM $\varepsilon = 0.01$	32.6	39.2	52.1

ples for point clouds by using the gradient ascent strategy. In Table 4, we show that both PointNet and our model can be fooled by adding small perturbations with $||\delta||_\infty < \varepsilon$, where the maximal absolute value in perturbation δ is restricted to be smaller than ε. Compared to randomly sampled perturbations, adversarial perturbations can be viewed as a more efficient tool to examine the robustness of point cloud algorithms under the worst cases. We can see that although both PointNet and our proposed model suffer from a significant drop in accuracy, our model is more robust.

4.2. SHREC'17 3D Shape Retrieval

We also conducted 3D shape retrieval experiments on ShapeNet Core [4], following the perturbed protocol of the SHREC'17 3D shape retrieval contest [19]. Our model for shape retrieval is trained on training and validation sets provided by the contest. For a fair comparison with previous methods, the model is trained following the practice in [6], where an auxiliary in-batch triplet loss is used together with softmax classification loss. In our implementation, the feature followed by the classifier is L2-normalized and used as invariant descriptor of input point cloud. Cosine similarity is used to compute the distance between samples. Other details are same as [6].

Experimental results are presented in 5. Without tricks, our method can outperform all other algorithms by a large margin, including the winner of this contest. Compared to the most participating methods in SHREC'17, our method and implementation is simple yet efficient, which proves the effectiveness of the proposed method.

Table 5. Comparisons of the 3D retrieval performance of our model with state-of-the-art methods on the *perturbed* dataset of the SHREC'17 contest. We report the performance measured by standard evaluation metrics including precision, recall, f-score, mean average precision (mAP) and normalized discounted cumulative gain (NDCG). The average of the micro macro mAP is used to rank performance following [19]. Without tricks, our method can outperform other methods by a large margin.

	micro					macro					
Method	PN	R@N	F1@N	mAP	NDCG	PN	R@N	F1@N	mAP	NDCG	score
SHREC'17 participating methods											
Furuya [7]	**0.814**	0.683	0.706	0.656	0.754	0.607	0.539	0.503	0.476	0.560	0.566
Tatsuma [24]	0.705	**0.769**	0.719	0.696	0.783	0.424	**0.563**	0.434	0.418	0.479	0.557
Zhou [2]	0.660	0.650	0.643	0.567	0.701	0.443	0.508	0.437	0.406	0.513	0.487
Spherical CNN [6]	0.717	0.737	-	0.685	-	0.450	0.550	-	0.444	-	0.565
Spherical CNN [5]	0.701	0.711	-	0.676	-	0.443	0.508	-	0.406	-	0.541
Ours	0.778	0.751	**0.752**	**0.705**	**0.813**	**0.656**	0.539	**0.536**	**0.483**	**0.580**	**0.594**

Table 6. Part segmentation results on ShapeNet Part Segmentation dataset. We report the mean IoU across all part classes and IoU for each categories are reported, where we use 'EP' and 'SB' to represent earphone and skateboard respectively.

Method	mIoU	aero	bag	cup	car	chair	EP	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	SB	table
PointNet [16]	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++ [18]	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
SyncSpecCNN [30]	84.7	81.6	81.7	81.9	75.2	90.2	74.9	93.0	86.1	**84.7**	95.6	66.7	92.7	81.6	**60.6**	**82.9**	82.1
SPLATNet3D [22]	84.6	81.9	**83.9**	**88.6**	79.5	90.1	73.5	91.3	84.7	84.5	96.3	69.7	**95.0**	81.7	59.2	70.4	81.3
SpiderCNN [28]	85.3	**83.5**	81.0	87.2	77.5	**90.7**	**76.8**	**91.1**	**87.3**	83.3	95.8	**70.2**	93.5	**82.7**	59.7	75.8	82.8
Ours	**85.4**	83.0	83.4	87.0	**80.2**	90.1	75.9	**91.1**	86.2	84.2	**96.7**	69.5	94.8	82.5	59.9	75.1	**82.9**

4.3. ShapeNet Semantic Part Segmentation

As a generic framework, SFCNN can be applied to various tasks for point cloud processing. We can easily extend our framework to 3D shape semantic segmentation by employing the encoder-decoder network architecture.

The ShapeNet Part dataset [29] is a widely used benchmark to evaluate 3D part segmentation, which contains 16,681 objects from 16 categories. Each object have 2-6 part labels. We reported the standard evaluation metrics including mean IoU across all part classes and IoU for each categories following previous works.

Experimental results are shown in Table 6. Our model obtained an mIoU of 85.4, which shows very competitive performance compared to state-of-the-art methods.

Our experiments demonstrate that our framework has strong capacity of capturing and understanding local and global structures in different tasks. Meanwhile, our model is also very efficient. Training PointNet++ and SPLATNet$_{3D}$ for part segmentation tasks on ShapeNet takes 3.5 and 2.5 days [22] respectively on the similar hardware configurations, while our model can converge less than 24 hours on a single 1080ti GPU.

5. Conclusion

In this paper, we present the SFCNN framework, which is a generic, flexible and 3D rotation invariant framework based on spherical symmetry for point cloud recognition. Our framework shows similar properties as CNN for image recognition and extends CNN to learn robust feature resistant to rotations and perturbations. Comprehensive experimental study demonstrates the proposed model is effective yet robust. Our approach can achieve competitive performance compared to state-of-the-art techniques on both ModelNet40 classification and ShapeNet part segmentation tasks. Meanwhile, our model can also show superior performance on rotated ModelNet and SHREC'17 perturbed shape retrieval tasks.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grant 61822603, Grant U1813218, Grant U1713214, Grant 61672306, and Grant 61572271.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 3

[2] Song Bai, Xiang Bai, Zhichao Zhou, Zhaoxiang Zhang, and Longin Jan Latecki. Gift: A real-time and scalable 3d shape search engine. In *CVPR*, pages 5023–5032, 2016. 8

[3] Zhangjie Cao, Qixing Huang, and Ramani Karthik. 3d object classification via spherical projections. In *3DV*, pages 566–574. IEEE, 2017. 4

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 7

[5] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018. 1, 8

[6] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *ECCV*, pages 52–68, 2018. 1, 3, 4, 6, 7, 8

[7] Takahiko Furuya and Ryutarou Ohbuchi. Deep aggregation of local 3d geometric features for 3d model retrieval. In *BMVC*, 2016. 8

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples (2014). *arXiv preprint arXiv:1412.6572*. 6

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 5

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012. 1

[13] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, pages 828–838, 2018. 7

[14] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*, pages 922–928. IEEE, 2015. 2, 7

[15] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, pages 2014–2023, 2016. 5

[16] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*, 1(2):4, 2017. 1, 2, 4, 5, 6, 7, 8

[17] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *CVPR*, pages 5648–5656, 2016. 2, 7

[18] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, pages 5099–5108, 2017. 1, 2, 4, 6, 7, 8

[19] Manolis Savva, Fisher Yu, Hao Su, M Aono, B Chen, D Cohen-Or, W Deng, Hang Su, Song Bai, Xiang Bai, et al. Shrec17 track large-scale 3d shape retrieval from shapenet core55. In *Proceedings of the 10th eurographics workshop on 3D object retrieval*, 2017. 2, 6, 7, 8

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[21] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 5

[22] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *CVPR*, pages 2530–2539, 2018. 1, 2, 8

[23] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, pages 945–953, 2015. 2, 7

[24] Atsushi Tatsuma and Masaki Aono. Multi-fourier spectra descriptor and augmentation with spectral clustering for 3d shape retrieval. *The Visual Computer*, 25(8):785–804, 2009. 8

[25] William P Thurston. *Three-Dimensional Geometry and Topology, Volume 1*, volume 1. Princeton university press, 2014. 4

[26] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *arXiv preprint arXiv:1711.07971*, 10, 2017. 6

[27] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920, 2015. 2, 6

[28] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. *ECCV*, 2018. 6, 8

[29] Li Yi, Vladimir G Kim, Duygu Ceylan, I Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, Leonidas Guibas, et al. A scalable active framework for region annotation in 3d shape collections. *TOG*, 35(6):210, 2016. 2, 6, 8

[30] Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *CVPR*, pages 6584–6592, 2017. 8