ClusterNet: Deep Hierarchical Cluster Network with Rigorously Rotation-Invariant Representation for Point Cloud Analysis

Chao Chen[1] Guanbin Li[1]* Ruijia Xu[1] Tianshui Chen[1,2] Meng Wang[3] Liang Lin[1,2]
[1]Sun Yat-sen University [2]DarkMatter AI Research [3]Hefei University of Technology
chench227@mail2.sysu.edu.cn, liguanbin@mail.sysu.edu.cn, xurj3@mail2.sysu.edu.cn
tianshuichen@gmail.com, wangmeng@hfut.edu.cn, linliang@ieee.org

Abstract

Current neural networks for 3D object recognition are vulnerable to 3D rotation. Existing works mostly rely on massive amounts of rotation-augmented data to alleviate the problem, which lacks solid guarantee of the 3D rotation invariance. In this paper, we address the issue by introducing a novel point cloud representation that can be mathematically proved rigorously rotation-invariant, i.e., identical point clouds in different orientations are unified as a unique and consistent representation. Moreover, the proposed representation is conditional information-lossless, because it retains all necessary information of point cloud except for orientation information. In addition, the proposed representation is complementary with existing network architectures for point cloud and fundamentally improves their robustness against rotation transformation. Finally, we propose a deep hierarchical cluster network called ClusterNet to better adapt to the proposed representation. We employ hierarchical clustering to explore and exploit the geometric structure of point cloud, which is embedded in a hierarchical structure tree. Extensive experimental results have shown that our proposed method greatly outperforms the state-of-the-arts in rotation robustness on rotation-augmented 3D object classification benchmarks.

1. Introduction

Rotation transformation is a natural and common in 3D world, however, it gives rise to an intractable challenge for 3D recognization. Theoretically, since SO(3)[1] is an infinite group, a 3D object possesses rotated clones in infinite attitudes, thus a machine learning model is obliged to extract features from an extremely huge input space. For example, in 3D object classification task, the category label of an object is invariant against arbitrary rotation transformation in majority situations. However, from the perspective of a classification model, an object and its rotated clone are distinct in input metric space, hence the model, such as neural network based methods, should have enough capacity to learn rotation invariance from data and then approximate a complex function that maps identical objects in infinite attitudes to similar features in feature metric space.

To alleviate the curse of rotation, a straightforward method is to design a model with high capacity, such as a deep neural network with considerable layers, and feed the model with great amounts of rotation-augmented data [1] based on a well-designed augmentation pipeline. Although data augmentation is effective to some extent, it is computationally expensive in training phase and lacks solid guarantee of rotation robustness. [12, 19] apply spatial transformer network [6] to canonicalize the input data before feature extraction, which improves the rotation-robustness of model but still inherits all the defects of the data augmentation. [17] proposes a rotation-equivariant network for 3D point clouds using a special convolutional operation with local rotation invariance as a basic block. The method attempts to equip the neural network with rotation-symmetry. However, it is hard to guarantee the capacity of such network to satisfy all rotation-equivariant constraints in each layer.

We address the issue by introducing a novel **R**igorous **R**otation-**I**nvariant (RRI) representation of point cloud. Identical objects in different orientations are unified as a consistent representation, which implies that the input space is heavily reduced and the 3D recognization tasks become much easier. It can be mathematically proved that the proposed representation is rigorously rotation-invariant, and information-lossless under a mild condition. Given any data point in point cloud and a non-collinear neighbor ar-

*Corresponding author is Guanbin Li. This work was supported in part by the State Key Development Program under Grant 2016YFB1001004, in part by the National Natural Science Foundation of China under Grant No.61602533 and No.61702565, in part by the Fundamental Research Funds for the Central Universities under Grant No.18lgpy63. This work was also sponsored by SenseTime Research Fund.

[1] 3D rotation group, denoted as SO(3), contains all rotation transformations in $\mathbb{R}^3$ under the operation of composition.

bitrarily, the whole point cloud can be restored intactly with the RRI representation, even if the point cloud is under an unknown orientation. In other words, the RRI representation maintains all necessary information of point cloud except for the volatile orientation information which is associated with specific rotation transformation. Furthermore, the RRI representation is flexible to be plugged into the current neural architectures and endows them with rigorous rotation invariance. The major difference between rotation-equivariant network and our proposed method is that the former embeds the invariance property as a *priori* into neural network, but the latter separates the rotation invariance from neural network and directly cut down the orientation-redundancy of input space.

Moreover, we propose a deep hierarchical network called ClusterNet to better adapt to our new representation. Specifically, we employ unsupervised hierarchical clustering to learn the underlying geometric structure of point cloud. As a result, we can obtain a hierarchical structure tree and then employ it to guide hierarchical features learning. Similar to CNNs, ClusterNet extracts features corresponding with small clusters, which learns fine-grained patterns of point cloud; the smaller cluster features are then aggregated as larger cluster features capturing higher-level information. The process of embedding is repeated along the hierarchical structure tree from bottom to top until we achieve the global features of the whole point cloud.

We summarize our major contributions as follows:

1. We propose a new point cloud representation that satisfies, both theoretically and empirically, rotation invariance and information preservation;

2. The proposed representation is complementary with the existing neural network frameworks and fundamentally improves their robustness against rotation transformation;

3. We further introduce a novel deep hierarchical network called ClusterNet to better adapt to our new representation. Combing the novel point cloud representation and the elaborate ClusterNet, our method achieves state-of-the-art robustness in standard 3D classification benchmarks.

2. Related Work

Deep Learning for 3D Objects. In general, the development of deep learning for 3D object is closely related to the progress of representation form of 3D object from geometric regular data to irregular one. For the conventional CNNs, it is intractable to handle the geometric irregular data, such as meshes and point clouds. Thus, previous literatures strive to transform such data into voxel representations [10, 13, 21] or multi-images (views) [16, 21]. However, it is inevitable to suffer from loss of resolution and high computational expense during transformation and subsequent processing. In order to escape from the limit of volumetric grid, some methods partition the $\mathbb{R}^3$ space by the traditional data structures, such as k-d trees [7] and octrees [15, 18], to alleviate the issues. PointNet [12] is the most pioneering work that takes point cloud as input and applies MLPs and max pooling to construct a universal approximator with permutation invariance. Since the lack of sensing capability for local information, a variety of hierarchical neural networks for point cloud, such as PointNet++ [14] and DGCNN [19], are proposed to progressively abstract features along a hierarchical structure designed in a heuristic way. Recently, Chen et al. [2] proposed to leverages nonlinear Radial Basis 6 Function (RBF) convolution as basis feature extractor for robust point cloud representation. As far as we known, the existing methods merely design the hierarchical structure by priori knowledge and none of them have made effort to explore the geometric structure underlying the point cloud, which is prone to cause lower capacity of the hierarchical neural network.

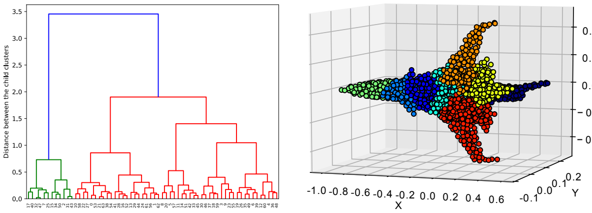

Figure 1: The left figure is a dendrogram of a point cloud learned by hierarchical clustering. The right figure shows partition of the point cloud of plane in a merge-level remaining 8 clusters.

Hierarchical Clustering. In the area of unsupervised learning, hierarchical clustering [11] is a classical method to build a hierarchy (also called dendrogram) of clusters. It generally consists of agglomerative type and divisive type. The first one considers all data points as the smallest cluster and merges the two closest ones with respect to a particular distance metric and a linkage criteria from bottom to top, and the latter performs in an opposite direction. A typical linkage criteria is ward linkage minimizing the total within-cluster variance, which can remedy the degeneration case of uneven cluster sizes. Furthermore, the point cloud in low dimensional space, such as $\mathbb{R}^3$, is quite suitable for hierarchical clustering. A dendrogram and a partition of point cloud is shown in Figure 1.

Rotation-Equivariant Network for 3D Objects. PointNet [12] solves the permutation invariance problem of point cloud by a symmetric pooling operator, which remarkably reduces the $N!$ cases (given a point cloud with N points) of permutation into merely one case. However, rotation invariance is a more challenging problem needed to be solved, since SO(3) is infinite. Previous works have attempted to upgrade the existing neural networks with the property of rotation-equivariance [8, 20]. For example, [17] designs a special convolutional operation with local rotation invariance and applies it as basic block to build a rotation-equivariant network. Besides, [3] proposes a method that transforms the 3D voxel data into spherical representation and then employs a spherical convolution operator to extract rotation-equivariant features. However, it is unavoidable to suffer from loss of information as there is no bijection between $\mathbb{R}^3$ and S^2. [2]

3. Approach

3.1. Rotation-Invariant Representation in $\mathbb{R}^3$

A point cloud with N data points is often expressed as a point set $S = \{(x_i, y_i, z_i) \mid x_i, y_i, z_i \in \mathbb{R}\}_{i=0}^{N-1}$ in Cartesian coordinate system. In another experssion, it can be represented as $S \in \mathbb{R}^{N\times 3}$ in a matrix form. In terms of the point cloud in $\mathbb{R}^{N\times 3}$, rotation transformation is a linear mapping in correspondence with a 3×3 real orthogonal matrix.

In order to precisely describe the rigorous rotation invariance, we conduct a definition as below.

Definition 1 (RRI Mapping)**.** *If $N, D \in \mathbb{N}^+$, a rigorously rotation-invariant* ***(RRI)*** *mapping is a set mapping $\mathcal{F} : \mathbb{R}^{N\times 3} \mapsto \mathbb{R}^{N\times D}$ such that*

$$\mathcal{F}(S) = \mathcal{F}(R(S))$$

holds for all point set $S \in \mathbb{R}^{N\times 3}$ and all rotation mapping $R \in \mathrm{SO}(3)$. Then $\mathcal{F}(S)$ is called as a rigorously rotation-invariant representation of S.

The definition introduces an RRI mapping that not only maintains rotation invariance but also rigorously preserves the cardinality of output point set as same as the cardinality of the input one, i.e., the input set with N points should be mapped to output set with N features.

For example, given a point set $S = \{\boldsymbol{p_i} \mid \boldsymbol{p_i} \in \mathbb{R}^3\}_{i=0}^{N-1}$, it is obvious that $\|\boldsymbol{p_i}\|_2$ is rotation-invariant since the rotation invariance of 2-norm:

$$\|R\boldsymbol{x}\|_2^2 = \|\boldsymbol{x}\|_2^2 \ , \ \forall \boldsymbol{x} \in \mathbb{R}^3. \tag{1}$$

Hence the row-wise vector norm $\|\cdot\|_2$ can be defined as an RRI mapping from $\mathbb{R}^{N\times 3}$ to $\mathbb{R}^{N\times 1}$.

Since rotation transformation has the property of preserving the relative positional relationships among several points, a definition is conducted to describe the rotation-invariance of k-ary operator as follow.

Definition 2 (RRI k-ary Operator)**.** *A rigorously rotation-invariant* ***(RRI)*** *k-ary operator is an operator $\mathcal{G} : \underbrace{\mathbb{R}^3 \times \mathbb{R}^3 \times \cdots \times \mathbb{R}^3}_{k} \mapsto \mathbb{R}^n$ such that*

$$\mathcal{G}(R\boldsymbol{x_1}, R\boldsymbol{x_2}, \ldots, R\boldsymbol{x_k}) = \mathcal{G}(\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_k}) \ ,$$

holds for all $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_k} \in \mathbb{R}^3$ and all rotation mapping $R \in \mathrm{SO}(3)$.

Apparently, the vector norm $\|\cdot\|_2$ is a unary RRI operator from $\mathbb{R}^3$ to $\mathbb{R}$. At the same time, it can be shown that the inner product of two arbitrary points in S is rotation-invariant, because rotation transformation is orthonormal:

$$\langle R\boldsymbol{x}, R\boldsymbol{y}\rangle = (R\boldsymbol{x})^\top (R\boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{y} = \langle \boldsymbol{x}, \boldsymbol{y}\rangle \ , \tag{2}$$

holds for $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$. Thus inner product is an RRI binary operator from $\mathbb{R}^3 \times \mathbb{R}^3$ to $\mathbb{R}$. Note that $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2 \cos\theta_{\boldsymbol{xy}}$ holds when $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$, the formulas (1,2) imply that the relative angle $\theta_{\boldsymbol{xy}}$ between any two points $\boldsymbol{x}, \boldsymbol{y} \in S$ is a rotation-invariant quantity.

Similarly, it can be proved that for any point $\boldsymbol{p} \in S \backslash \{\boldsymbol{0}\}$, if $\mathcal{T}_{\boldsymbol{p}}$ is an orthogonal projection operator of $\mathbb{R}^3$ onto a plane L past the origin and $\boldsymbol{p}$ is orthogonal to L, then the inner product of two arbitrary points in $\mathcal{T}_{\boldsymbol{p}}(S)$ is rotation-invariant. The proof is given as below:

$$\begin{aligned}
&\langle \mathcal{T}_{R\boldsymbol{p}}(R\boldsymbol{x}), \mathcal{T}_{R\boldsymbol{p}}(R\boldsymbol{y})\rangle \\
&= \big(R\boldsymbol{x} - ((R\boldsymbol{x})^\top R\boldsymbol{n}) \cdot R\boldsymbol{n}\big)^\top \big(R\boldsymbol{y} - ((R\boldsymbol{y})^\top R\boldsymbol{n}) \cdot R\boldsymbol{n}\big) \\
&= \big(\boldsymbol{x} - (\boldsymbol{x}^\top \boldsymbol{n}) \cdot \boldsymbol{n}\big)^\top \big(\boldsymbol{y} - (\boldsymbol{y}^\top \boldsymbol{n}) \cdot \boldsymbol{n}\big) \\
&= \langle \mathcal{T}_{\boldsymbol{p}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{p}}(\boldsymbol{y})\rangle \ ,
\end{aligned} \tag{3}$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$, $R \in \mathrm{SO}(3)$ and $\boldsymbol{n} = \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|}$. Hence the composite operator $\mathcal{G}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{p}) = \langle \mathcal{T}_{\boldsymbol{p}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{p}}(\boldsymbol{y})\rangle$ is an RRI ternary operator.

Furthermore, according to the property of cross product, it can be shown that the composite operator $\mathcal{G}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{p}) = \langle \mathcal{T}_{\boldsymbol{p}}(\boldsymbol{x}) \times \mathcal{T}_{\boldsymbol{p}}(\boldsymbol{y}), \boldsymbol{p}\rangle$ is also an RRI ternary operator. The proof is as below.

$$\begin{aligned}
&\langle \mathcal{T}_{R\boldsymbol{p}}(R\boldsymbol{x}) \times \mathcal{T}_{R\boldsymbol{p}}(R\boldsymbol{y}), R\boldsymbol{p}\rangle \\
&= \langle R\mathcal{T}_{\boldsymbol{p}}(\boldsymbol{x}) \times R\mathcal{T}_{\boldsymbol{p}}(\boldsymbol{y}), R\boldsymbol{p}\rangle \\
&= \langle (\det R)(R^{-1})^\top \big(\mathcal{T}_{\boldsymbol{p}}(\boldsymbol{x}) \times \mathcal{T}_{\boldsymbol{p}}(\boldsymbol{y})\big), R\boldsymbol{p}\rangle \\
&= \langle R\big(\mathcal{T}_{\boldsymbol{p}}(\boldsymbol{x}) \times \mathcal{T}_{\boldsymbol{p}}(\boldsymbol{y})\big), R\boldsymbol{p}\rangle \\
&= \langle \mathcal{T}_{\boldsymbol{p}}(\boldsymbol{x}) \times \mathcal{T}_{\boldsymbol{p}}(\boldsymbol{y}), \boldsymbol{p}\rangle.
\end{aligned} \tag{4}$$

Consequently, four rotation-invariant operators have been found in the previous discussion, and we can make use of them to construct a rotation-invariant representation and the construction method is just an RRI mapping.

In order to introduce the proposed representation, we need to construct a K-nearest neighbor (K-NN) graph $\boldsymbol{G} =$

[2] Two-dimensional sphere, denoted as S^2, is the surface of a completely round ball in $\mathbb{R}^3$.

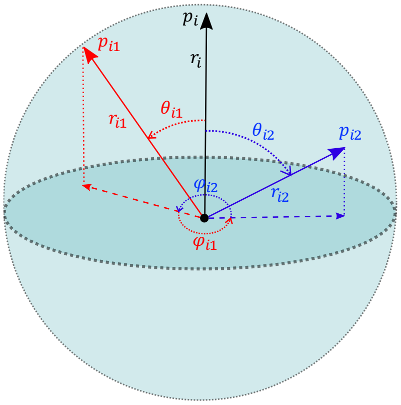

Figure 2: The diagram illustrates each elements in RRI representation (cf. formula (5)) by a trivial case in which we builds 2-NN graph on three points.

$(S,\ \mathcal{E})$ on point set S, where $\mathcal{E} = \{(\boldsymbol{x}, \boldsymbol{y}) \in S \times S \mid \boldsymbol{y}$ is one of the K-NN of $\boldsymbol{x}\}$.

According to the K-NN graph, we can employ the RRI operators to capture the relative positional patterns underlying the K-NN neighborhood of each point in S, and benefit from the property of rotation invariance at the same time.

Specifically, given a K-NN graph $\boldsymbol{G}$ on point set S, the proposed representation of each point $\boldsymbol{p}_i \in S$ is

$$(r_i, (r_{i1}, \theta_{i1}, \phi_{i1}), (r_{i2}, \theta_{i2}, \phi_{i2}), ..., (r_{iK}, \theta_{iK}, \phi_{iK}))\ , \tag{5}$$

where

$$\begin{aligned} r_i &= \|\boldsymbol{p}_i\|_2\ , \\ r_{ik} &= \|\boldsymbol{p}_{ik}\|_2 \quad (\boldsymbol{p}_{ik} \text{ is one of the } K\text{-NN of } \boldsymbol{p}_i \text{ with id } k)\ , \\ \theta_{ik} &= \arccos\left(\langle \frac{\boldsymbol{p}_i}{r_i}, \frac{\boldsymbol{p}_{ik}}{r_{ik}} \rangle\right)\ , \\ \phi_{ik} &= \psi_{j^*} \triangleq \min\{\psi_j \mid 1 \le j \le K,\ j \ne k, \psi_j \ge 0\}\ , \\ \psi_j &= \operatorname{atan2}(\sin\psi_j, \cos\psi_j)\ , \\ \sin\psi_j &= \langle \frac{\mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ik})}{r_{ik}} \times \frac{\mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ij})}{r_{ij}}, \frac{\boldsymbol{p}_i}{r_i} \rangle\ , \\ \cos\psi_j &= \langle \frac{\mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ik})}{r_{ik}}, \frac{\mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ij})}{r_{ij}} \rangle\ . \end{aligned} \tag{6}$$

Note that for a given point $\boldsymbol{p}_i$ and one of its k-nearest neighbor $\boldsymbol{p}_{ik}$, if we apply $\boldsymbol{p}_i$ as normal vector, then ψ_j represents the relative angle between $\mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ik})$ and $\mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ij})$ according to the right-hand rule, thereupon ϕ_{ik} is the relative angle between $\mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ik})$ and its rotation-nearest point $\mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ij^*})$ in anti-clockwise direction. The representation (5) has intuitive geometric meaning which is illustrated in Figure 2. The function $\operatorname{atan2}(\cdot, \cdot)$ in formula (6) is a special $\arctan(\cdot)$ choosing the quadrant correctly.

On the foundation of the four RRI operators, the proposed representation in formula (5) is rigorously rotation-invariant as the claim of the following theorem.

Theorem 1. *The mapping defined by (6) is a rigorously rotation-invariant mapping and the representation (5) is rigorously rotation-invariant.*

Proof. Firstly, the computation method in (6) exactly defines a set of mappings $\mathcal{F} : \mathbb{R}^{N\times 3} \mapsto \mathbb{R}^{N\times(3K+1)}$.

Note that the K-NN neighborhood of arbitrary point $\boldsymbol{x} \in S$ is uniquely determined by $\|\boldsymbol{x} - \boldsymbol{y}\|_2$ with respect to all point $\boldsymbol{y} \in S$, which is proved rotation-invariant by (1), so searching K-NN of point $\boldsymbol{x} \in S$ is a rotation-invariant operation. Besides, since rotation transformation has no influence to the permutation of point cloud, we can obtain consistent order of K-NN by stable sort algorithm that maintains the relative order of points with equal distance.

As the result of (1,2), it is obvious that r_i, r_{ik} and θ_{ik} are rotation-invariant. On the basis of formulas (3,4), $\sin\psi_j$ and $\cos\psi_j$ are both of rotation-invariance, hence ψ_j and ϕ_{ik} are also rotation-invariant.

Therefore formulas (6) define an RRI mapping and the representation defined by (5) is an RRI representation. □

However, RRI mapping probably loses some essential information from the original data because the pursuit of rigorous rotation invariance may result in lower capacity of the RRI representation. For example, the 2-norm $\|\cdot\|_2$ is indeed an RRI mapping as the discussion of formula (1), whereas it only captures the distance information of the points in S and totally discards the relative positional pattern of them.

It is remarkable to point out that the proposed representation not only satisfies the property of rigorous rotation invariance but also preserves necessary information which helps to reconstruct the original point cloud on a weak condition as stated in the following theorem.

Theorem 2. *Given a K-NN graph $\boldsymbol{G} = (S, \mathcal{E})$ on point set S, if $\boldsymbol{G}$ is a strongly connected graph, then for $\forall R \in \mathrm{SO}(3)$, given Cartesian coordinates of a nonzero point and one of its non-collinear K-NN neighbor, the Cartesian coordinates of $R(S)$ can be determined by the RRI representation defined by (5).*

Proof. Given the Cartesian coordinates of arbitrary point $\boldsymbol{p}_i \in R(S)\backslash\{\boldsymbol{0}\}$ and one of its non-collinear K-NN neighbor $\boldsymbol{p}_{ik}$, we can obtain the 2-norm of them and their relative positional information, such as θ_{ik} and ϕ_{ik}. With the representation (5), we will show that the coordinate of another K-NN neighbor $\boldsymbol{p}_{ij^*}$, which is the rotation-nearest point of p_{ik} in anti-clockwise direction after applying orthogonal projection $\mathcal{T}_{\boldsymbol{p}_i}$, can be uniquely determined by the following equation system,

$$\begin{aligned}\langle \boldsymbol{p}_{ij^*},\ \boldsymbol{p}_{ij^*}\rangle &= r_{ij^*}^2\\ \langle \boldsymbol{p}_i,\ \boldsymbol{p}_{ij^*}\rangle &= r_i\, r_{ij^*} \cos\theta_{ij^*}\\ \langle \mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ik}), \mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ij^*})\rangle &= r_{ik}\, r_{ij^*} \cos\psi_{j^*}\\ \langle \mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ik}) \times \mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ij^*}), \boldsymbol{p}_i\rangle &= r_i\, r_{ik}\, r_{ij^*} \sin\psi_{j^*}\ ,\end{aligned} \tag{7}$$

where $\psi_{j^*} = \phi_{ik}$. The quantities in the right hand side of the equation system (7) are all known in the representation (5). In other words, the unique unknown variable is $\boldsymbol{p}_{ij^*}$. On the foundation of the rotation invariance revealed from Theorem 1, it is apparent that there exists at least one solution for the equation system (7) because it has a solution for the original point cloud S.

Suppose that the solution set of the equation system (7) contains at least two different solutions $\boldsymbol{p}_{ij^*}$ and $\widetilde{\boldsymbol{p}}_{ij^*}$, then the equation system (7) would imply that

$$\begin{aligned}\langle \boldsymbol{p}_i\ , \boldsymbol{p}_{ij^*} - \widetilde{\boldsymbol{p}}_{ij^*}\rangle &= 0\\ \langle \mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ik})\ , \boldsymbol{p}_{ij^*} - \widetilde{\boldsymbol{p}}_{ij^*}\rangle &= 0\\ \langle \mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ik}) \times (\boldsymbol{p}_{ij^*} - \widetilde{\boldsymbol{p}}_{ij^*})\ , \boldsymbol{p}_i\rangle &= 0\end{aligned} \tag{8}$$

Since both $\mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ik})$ and $\boldsymbol{p}_{ij^*} - \widetilde{\boldsymbol{p}}_{ij^*}$ are in the plane $L = \{\boldsymbol{x} \in \mathbb{R}^3 \mid \boldsymbol{x} \perp \boldsymbol{p}_i\}$, $\mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ik}) \times (\boldsymbol{p}_{ij^*} - \widetilde{\boldsymbol{p}}_{ij^*}) = \alpha \boldsymbol{p}_i$ holds for some $\alpha \in \mathbb{R}\backslash\{0\}$. However, it would imply that

$$\langle \mathcal{T}_{\boldsymbol{p}_i}(\boldsymbol{p}_{ik}) \times (\boldsymbol{p}_{ij^*} - \widetilde{\boldsymbol{p}}_{ij^*})\ , \boldsymbol{p}_i\rangle = \alpha \langle \boldsymbol{p}_i\ , \boldsymbol{p}_i\rangle = 0\ . \tag{9}$$

Since $\alpha \neq 0$ and $\boldsymbol{p}_i \neq \boldsymbol{0}$, the equation (9) causes a contradiction. Thus, the solution set of (7) contains a unique solution, i.e., given points $\boldsymbol{p}_i$ and $\boldsymbol{p}_{ik}$, the Cartesian coordinate of $\boldsymbol{p}_{ij^*}$ can be uniquely determined by the equation system (7).

Similarly, we can solve the coordinate of the next neighbor which is rotation-nearest from $\boldsymbol{p}_{ij^*}$ in anti-clockwise direction after applying orthogonal projection $\mathcal{T}_{\boldsymbol{p}_i}$. The process is repeated until all the K-NN neighbors of $\boldsymbol{p}_i$ are reconstructed intactly.

Since the graph $\boldsymbol{G} = (S, \mathcal{E})$ is strongly connected, two arbitrary points $\boldsymbol{a}, \boldsymbol{b} \in S$ are connected by at least one path $(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ between them, where n is the path length, and $\boldsymbol{x}_0 = \boldsymbol{a}, \boldsymbol{x}_n = \boldsymbol{b}$ and $(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}) \in \mathcal{E}$ holds for $0 \leq i \leq n-1$. Therefore, starting from the K-NN neighborhood of $\boldsymbol{p_i}$, we can restore the coordinate of arbitrary point in S step-by-step along a path with finite length.

□

3.2. Hierarchical Clustering based ClusterNet

We propose a hierarchical clustering based neural network, called ClusterNet, to learn a hierarchical structure tree for the instruction of hierarchical feature representation of point clouds. With the assistance of unsupervised learning, we can explore and exploit distribution information of point cloud with regard to the hierarchical structure tree.

3.2.1 RRI Representation Processing

We can reformulate the proposed representation (5) of each point $\boldsymbol{p}_i \in S$ as

$$(\underbrace{(r_i, r_{i1}, \theta_{i1}, \phi_{i1})}_{T_{i1}}, \underbrace{(r_i, r_{i2}, \theta_{i2}, \phi_{i2})}_{T_{i2}}, ..., \underbrace{(r_i, r_{iK}, \theta_{iK}, \phi_{iK})}_{T_{iK}}). \tag{10}$$

In other words, we summarize the RRI information between point $\boldsymbol{p}_i$ and its K nearest neighbors as $(T_{i1}, T_{i2}, \ldots, T_{iK})$ to characterize point $\boldsymbol{p}_i$. Hence the new representation of a point cloud $S \in \mathbb{R}^{N\times 3}$ is a tensor $T \in \mathbb{R}^{N\times K\times 4}$. Since the local neighborhood pattern of $\boldsymbol{p}_i$ is probably embedded in its K nearest neighbors, the proposed representation takes advantages of the property and captures the local pattern in the K-NN neighborhood by an RRI and conditional information-lossless mechanism.

Since the RRI representation of point $\boldsymbol{p}_i$ can be regarded as a mini point cloud $(T_{i1}, T_{i2}, \ldots, T_{iK})$, and PointNet is a universal continuous set function approximator, we can apply PointNet as a basic block to learn a representation of the mini point cloud and extract local features to characterize the K-NN neighborhood. In other words, we can transform the RRI representation, an $N \times K \times 4$ tensor, into a $N \times D$ tensor of neighborhood features by means of PointNet as the following formula.

$$\boldsymbol{p}_i' = \max_{1\leq k\leq K} f_\Theta(T_{ik})\ , \tag{11}$$

where $f_\Theta(\cdot)$ is a multi-layer perceptron network with parameters Θ shared with all output features. In other words, we extract a feature $\boldsymbol{p}_i'$ corresponding to the original point $\boldsymbol{p}_i \in S$.

In the view of DGCNN, the formula (11) is a special case of the EdgeConv. However, we utilize an RRI representation to describe the relationship between a point and its K-NN neighbors while DGCNN only uses the difference vector $\boldsymbol{p}_i - \boldsymbol{p}_{ik}$ concatenated with $\boldsymbol{p}_i$, both of them vary with rotation transformation.

3.2.2 Hierarchical Clustering Tree

Since point could embeds in low dimensional space $\mathbb{R}^3$ equipped with Euclidean metric, hierarchical clustering is an appropriate method to analyze the hierarchical structure of point cloud. With the support of hierarchical clustering, we can learn a hierarchical clustering tree which illustrates the arrangement of partition and the relationships between different clusters.

Specifically, we employ the agglomerative hierarchical clustering with ward-linkage criteria to learn the hierarchical structures of point cloud. The ward-linkage criteria minimizes the total within-cluster variance, which tends to partition the point cloud into several clusters with similar

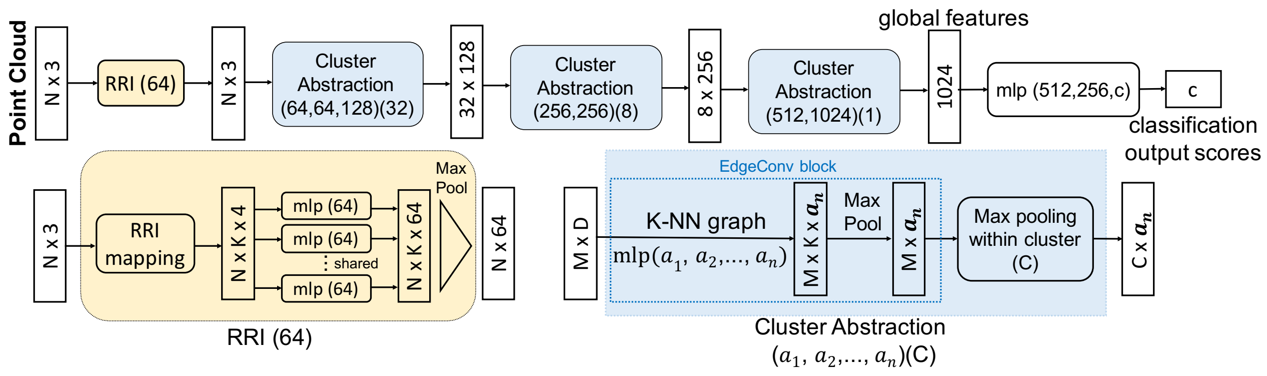

Figure 3: **Model architecture**: it consists of an RRI module, three cluster abstraction modules and the last classifier module. The model takes N points as input, applies the RRI module to extract rigorously rotation-invariant features for each point, extracts hierarchical cluster features using three cluster abstraction modules, and eventually obtains a global feature of the whole point cloud, which is used to generate classification scores for c categories. **RRI module**: the RRI module employs RRI mapping to obtain the RRI representation of point cloud and then aggregate the point features in K-NN neighborhood into local embedding of each point. **Cluster Abstraction module**: the module extracts edge features of each sub-cluster using multi-layer perception (mlp) with the number of layer neurons defined as $\{a_1, a_2, \ldots, a_n\}$ and then applies neighborhood aggregation to obtain super-cluster features. Then it leverages hierarchical structure as a guidance for feature aggregation within each cluster.

sizes. On a particular merge-level in hierarchical clustering tree, hierarchical clustering method will make an optimal partition with respect to the objective function concerning ward's minimum variance.

Similar to CNNs, ClusterNet learns the local features of fine-grained geometric structures from small clusters and then the local features are further aggregated into a higher-level feature of larger cluster according to the cluster relationships revealed in the hierarchical clustering tree. In other words, we can apply the hierarchical clustering tree to instruct the neural network how to extract and aggregate features in a more efficient way.

3.2.3 EdgeConv for Cluster Feature

The EdgeConv layer is first proposed by DGCNN [19], which improves the PointNet++ by dense sampling, i.e., all points are considered as sampled points and the feature of each point is aggregated from its K-nearest neighbors. The K-nearest neighbors are determined by a dynamic K-NN graph since the graph is affected by a similarity matrix of features from previous layer. The dynamic K-NN graph facilitates nonlocal diffusion of similar features in the feature space.

Specifically, given a F-dimensional point set $P = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^F$, we can construct a K-NN graph $\mathcal{E} \subseteq P \times P$ and then the output of EdgeConv can be obtained by

$$\boldsymbol{x}'_i = \max_{j:(i,k)\in\mathcal{E}} f_\Theta(\boldsymbol{x}_i - \boldsymbol{x}_k,\ \boldsymbol{x}_i)\ . \tag{12}$$

Different from DGCNN, the input of EdgeConv is a set of D-dimensional cluster features $\mathcal{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n\} \in \mathbb{R}^D$, where n is the number of clusters in a particular partition. Hence, if we apply EdgeConv to $\mathcal{C}$, the features corresponding to K-nearest clusters of $\boldsymbol{c}_i$ will be aggregated as a higher-level feature to characterize the cluster $\boldsymbol{c}_i$.

3.2.4 Aggregation within Cluster

Since the hierarchical clustering tree contains relationship of clusters, we propose a novel aggregation method for point cloud, which utilizes the relationship to aggregate sub-cluster features into that of a super-cluster. In particular, we can apply max pooling function to the sub-cluster features according to cluster index which records how sub-clusters are merged into a super-cluster in the hierarchical clustering tree. Therefore, it is feasible to learn the hierarchical representation of each cluster passing along the hierarchical clustering tree from bottom to top and finally the global feature of the whole point cloud can be obtained from the root node of the tree.

The proposed aggregation method is similar to the pooling methods in CNNs, since they both downsample the input data and maintain the maximum signal. In terms of the property of downsampling, the proposed aggregation method can improve robustness against mild corruptions of input data. Besides, aggregation reduces the total computational expense and the memory usage of GPU compared with DGCNN which extracts features for all points in the original point cloud.

Method	Input (size)	z/z	z/SO(3)	SO(3)/SO(3)	SO(3)/SO(3)*
PointNet (without STN) [12]	pc (1024 × 3)	88.5	14.4	70.5	72.5
PointNet++ (MSG without STN) [14]	pc+normal (5000 × 6)	91.9	16.0	74.7	78.5
SO-Net (without STN) [9]	pc+normal (5000 × 6)	**93.4**	19.6	78.1	81.4
DGCNN (without STN) [19]	pc (1024 × 3)	91.2	16.2	75.3	76.4
PointNet [12]	pc (1024 × 3)	89.2	16.4	75.5	72.5
PointNet++ (MSG) [14]	pc+normal (5000 × 6)	91.8	18.4	77.4	78.5
SO-Net [9]	pc+normal (5000 × 6)	92.6	21.1	80.2	81.4
DGCNN [19]	pc (1024 × 3)	92.2	20.6	81.1	82.0
Spherical CNN [3]	voxel (2×64^2)	88.9	76.9	86.9	86.9
Ours	pc (1024 × 3)	87.1	**87.1**	**87.1**	**87.1**

Table 1: Comparison of Rotation Robustness on rotation-augmented benchmark.

3.2.5 Permutation Invariance of ClusterNet

Trivial K-nearest neighbor searching is not permutation-invariant, since the K-nearest neighbors will become unstable when there exists some neighbors with exactly the same 2-norm. In such degeneration case, the result of K-NN searching is inevitably affected by the order of input points. However, we can modify the method of K-NN searching to avoid such degeneration. Specifically, if $\boldsymbol{p}_k$ is the k-th nearest neighbor of $\boldsymbol{p}$, then we consider the set of k-nearest neighbors of $\boldsymbol{p}$ as $\{\boldsymbol{q} \in S \mid \|\boldsymbol{q}\|_2 \leq \|\boldsymbol{p}_k\|_2\}$. On the foundation of the modified K-NN searching and the permutation-symmetric aggregation, it is obvious that the proposed ClusterNet is of permutation invariance.

4. Experiments

In this section, we propose a novel benchmark to evaluate the rotation robustness, on which we compare the proposed method with the state-of-the-art methods to empirically validate the effectiveness of the RRI representation and ClusterNet. Furthermore, we conduct an experiment to validate the superiority of ClusterNet over other architectures to learn deep hierarchical embeddings from the RRI representation. Finally, we evaluate the effect of the unique hyperparameter K in the RRI representation if we construct a K-nearest neighbor (K-NN) graph on point cloud.

4.1. Benchmarks

We design a new benchmark to fairly evaluate the rotation robustness of a model. Since the majority of objects in the original dataset are in a fixed postures, we are required to conduct rotation augmentation to enrich the test set. Considering SO(3) is infinite, it is infeasible to cover all the postures thoroughly, so we uniformly sample a reasonable amount of rotation transformations from SO(3).

According to Euler's rotation theorem [4], any rotation can be represented by a Euler axis and a rotation angle. The Euler axis is a three-dimensional unit vector and the rotation angle is a scalar. We can employ the following formulas to solve the rotation matrix R corresponding to the Euler axis $\boldsymbol{e}$ and the rotation angle θ,

$$R = \mathbf{I}_3 \cos\theta + (1-\cos\theta)\boldsymbol{e}\boldsymbol{e}^{\mathsf{T}} + [\boldsymbol{e}]_\times \sin\theta \ ,$$
$$[\boldsymbol{e}]_\times \triangleq \begin{bmatrix} 0 & -\boldsymbol{e}_3 & \boldsymbol{e}_2 \\ \boldsymbol{e}_3 & 0 & -\boldsymbol{e}_1 \\ -\boldsymbol{e}_2 & \boldsymbol{e}_1 & 0 \end{bmatrix} \ . \qquad (13)$$

As [5] stated, Fibonacci lattice is a mathematical idealization of natural patterns with optimal packing, where the area represented by each point is almost identical. Owing to the favorable property, we sample the Fibonacci lattice (points) from unit sphere surface as Euler axes and then uniformly sample the rotation angle in the space $[0, 2\pi)$. We choose such sampling method to generate Euler axes and rotation angles, and then solve the rotation matrix by the formulas (13). Consequently, we obtained a rotaion-sampling method that can sample rotation transformations from SO(3) uniformly.

In terms of dataset, we choose ModelNet40 [21], a widely-used 3D object classification dataset, as our basic dataset. ModelNet40 dataset consists of 12,311 CAD models from 40 manmade object categories, in which 9,843 is used for training and 2,468 is used for testing. Since each CAD model in ModelNet40 is composed of many mesh faces, we sample point cloud from them uniformly with respect to face area and then shift and normalize each point cloud into $[-1, 1]^3$ with centroid on the origin. We employ the sampling method to generate 500 Euler axes and 60 rotation angles for each Euler axes, i.e., 30,000 rotation transformations are sampled uniformly from SO(3) to augment the test set. As a result, we obtain a rotation-augmented test set with 74,040,000 point clouds in total as the benchmark dataset. We employ the augmented test set to evaluate the rotation robustness of each model.

4.2. Comparison of Rotation Robustness

We compare the proposed method with the state-of-the-art approaches on the benchmark for rotation-robustness

Method	Accuracy (%)	Time (h)
RRI+PointNet	85.9	8.5
RRI+DGCNN	86.4	12
RRI+ClusterNet (8, 1)	86.6	9
RRI+ClusterNet (32, 1)	86.8	10.5
RRI+ClusterNet (32, 8, 1)	87.1	9.5

Table 2: Analysis of Architecture Design

evaluation. The results are summarized in Table 1 with four comparison modes: (1) both training set and test set are augmented by azimuthal rotation (z/z); (2) training with azimuthal rotation and testing with arbitrary rotation (z/SO(3)); (3) both training and testing with arbitrary rotation (SO(3)/SO(3)); (4) conditions are almost as same as (3), but test the model with multi-rotation voting strategy (SO(3)/SO(3)*). In order to make the comparison more comprehensive, we make use of the following methods to improve the rotation-robustness of existing methods.

Rotation-augmentation is applied to the training set using two sampling strategies respectively. The first strategy only samples azimuthal rotations for augmentation, i.e., we merely use z-axis as Euler axis. While the second one samples all rotations from SO(3). In a particular epoch, we rotate each object using the sampled rotation transformation so that the model might improve rotation robustness from the objects under different orientations. We can use multi-rotation voting strategy to boost the robustness of model. Specially, we feed the model with test set in several orientation and then sum up the confidence scores as a total one to determine the classification result. Variants of spatial transformer network [6] are used to alleviate the problem caused by rotation transformation. For example, both PointNet and DGCNN employ spatial transformation module to learn a 3×3 rotation matrix which transforms point cloud into the canonical space.

Table 1 consists of four groups of methods. The first group from the top of the table consists of four models without using spatial transformer network (STN), while the methods in the second group are equipped with STN. In the third group, we choose a representative method based on rotation-equivariant network, spherical CNN[3], to compare with our proposed method. As shown in Table 1, the widely used augmentation using azimuthal rotations suffers from a sharp decline on the rotation-augmented test set. Furthermore, it illustrates that rotation-augmentation and STN can improve the rotation robustness of models but still have a large margin with our proposed method without the demand of any data augmentation. Although the spherical CNN is rotation-equivalent, it is also dependent with rotation augmentation and its performance is sensitive to the strategy of augmentation. Besides, our proposed method also outperforms spherical CNN on the rotation-augmented test set.

4.3. Ablation Analysis

4.3.1 Analysis of Architecture Design

Since the proposed RRI representation can be processed to be compatible with many architectures dealing with point cloud data, we enhance PointNet and DGCNN with the RRI representation, and Table 2 shows that ClusterNet outperforms both the enhanced PointNet and the enhanced DGCNN by a large margin on the foundation of the same RRI representation. As is illustrated in Section 3.2, DGCNN is a special case of ClusterNet without cluster aggregation, thus Table 2 shows that the aggregation within cluster along hierarchy indeed facilitates the hierarchical features learning and then extracts more discriminative features for 3D recognization.

4.3.2 Effectiveness of K in RRI Representation

K	40	50	60	70	80	90
Acc. (%)	85.6	86.4	86.8	87.0	87.1	87.1

Table 3: Effectiveness of K in RRI Representation

In terms of the proposed RRI representation, K is the unique hyperparameter, which controls the connectivity of the graph $\boldsymbol{G}$, thus we analyze the effectiveness of different K in RRI representation. As shown in Table 3, the architecture of ClusterNet is robust to diverse values of K even when K is too small to satisfy the connectivity condition in Theorem 2. For example, when $K = 40$, there exists nearly 25% of the point clouds not satisfying strongly connected condition, however, it still achieves comparable classification accuracy. When K is gradually increased to over 70, accuracies of the model remain stable.

5. Conclusion

In this paper, we step forward to enhance the rotation robustness of 3D object recognization model. Specifically, we introduce a novel RRI representation to assign a unique and consistent data form for any identical object in infinite attitude. We theoretically and empirically demonstrate that the representation is rigorously rotation-invariant and conditional information-lossless. Besides, our representation is complementary with prevailing 3D recognition architecture and improves their rotation robustness. Finally, we further design a deep hierarchical network called ClusterNet to better adapt to RRI representation. Extensive experimental evaluation on augmented test split set from widely-used 3D classification benchmark demonstrates the superiority of our novel RRI representation as well as the elaborate ClusterNet.

References

[1] Etienne Barnard and David Casasent. Invariance and neural nets. *IEEE Transactions on Neural Networks*, 2(5):498–508, 1991.

[2] Weikai Chen, Xiaoguang Han, Guanbin Li, Chao Chen, Jun Xing, Yajie Zhao, and Hao Li. Deep rbfnet: Point cloud feature learning using radial basis functions. *arXiv preprint arXiv:1812.04302*, 2018.

[3] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018.

[4] L Euler. General formulas for the translation of arbitrary rigid bodies. *Novi Commentarii academiae scientiarum Petropolitanae*, 20(1776):189–207, 1790.

[5] Álvaro González. Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, 42(1):49, 2010.

[6] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

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

[8] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10117–10126. Curran Associates, Inc., 2018.

[9] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9397–9406, 2018.

[10] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015.

[11] Daniel Müllner. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011.

[12] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

[13] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.

[14] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.

[15] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 3, 2017.

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

[17] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[18] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):72, 2017.

[19] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[20] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In *Advances in Neural Information Processing Systems*, pages 10402–10413, 2018.

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