Equivariant Analysis of Point Clouds by Message Passing on Simplicial Complexes

Chang Yu
School of Mathematical Sciences, University of Chinese Academy of Sciences
Beijing, China
yuchang22@mails.ucas.ac.cn

Sanguo Zhang
School of Mathematical Sciences, University of Chinese Academy of Sciences
Beijing, China
sgzhang@ucas.ac.cn

Li-Yong Shen*
School of Mathematical Sciences, University of Chinese Academy of Sciences
Beijing, China
lyshen@ucas.ac.cn

ABSTRACT

As a significant characteristic of point clouds, equivariance has attracted increasing attention in various fields such as computer vision and computer graphics. In this paper, we introduce a novel framework for achieving equivariance in point clouds using simplicial complexes and a message passing network. By leveraging the properties of complexes, we obtain equivariance for point clouds represented as graphs with vertices and edges. The message passing network aggregates information from simplices instead of individual points, which is further processed using convolutional layers to produce the output results. Our extensive experiments on point cloud classification and semantic segmentation tasks demonstrate that our method achieves comparable or better results than previous methods, showcasing its robustness.

CCS CONCEPTS

• **Computing methodologies → Computer vision**.

KEYWORDS

Geometric equivariance, point cloud analysis, message passing, simplicial complex, classification, segmentation

ACM Reference Format:

Chang Yu, Sanguo Zhang, and Li-Yong Shen. 2024. Equivariant Analysis of Point Clouds by Message Passing on Simplicial Complexes. In *2024 the 7th International Conference on Image and Graphics Processing (ICIGP 2024), January 19–21, 2024, Beijing, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3647649.3647695

1 INTRODUCTION

The 3D point cloud is a crucial data structure containing geometric information widely utilized in computer vision and computer graphics communities. Analyzing and understanding 3D point cloud data has become increasingly important. In recent years, deep learning methods like PointNet++ [22], PointCNN [11], DGCNN [28], A-CNN [9], and others have become standard tools for modeling point clouds and capturing their geometric relationships.

A characteristic of 3D point cloud is that most scalar features of the point clouds are *invariant* to global translation and rotation, while most vector features are *equivariant* to these transformations. For instance, the geometric structure, graphic categories, and part semantics of point clouds remain unchanged during translation and rotation, while vector features like normal information synchronize with the rotation. However, previous methods for solving point cloud problems, such as point cloud classification and semantic segmentation, have not effectively ensured these properties of point clouds. Therefore, finding a new method to guarantee the invariance and equivariance of point clouds is of utmost importance and significance. Note that invariance is a special case of equivariance.

Currently, several research approaches aim to achieve the invariance and equivariance of point clouds. For instance, learning orientations-based neural networks [15] achieve point cloud equivariance to some extent, but learning orientations can significantly affect the accuracy of the final result. Local-global-representation (LGR)-Net [38] designs a two-branch network to encode local and global features separately, but it requires the normals and higher-order relationships between the local coordinates centered at different points, which is nontrival. Vector-based neural networks [3, 8] use fully connected layers to linearly input point cloud information, achieving point cloud equivariance through linear combinations. However, these methods suffer significant losses in preserving the geometric structure information contained in point clouds. Tensor field-based neural networks [6, 20, 26] employ continuous convolution to process point clouds. Yet, they require calculating a significant amount of spherical harmonics on the fly, making their formulation overly complex, and resulting in high space and time complexity. Another approach defines convolution operators by spherical correlation and SO(3) correlation with circularly symmetric kernels [2, 4]. Some works [29, 30] extend spherical CNNs to 3D voxel grids. Additionally, mesh-based methods for point clouds, such as Geodesic CNN [16], ensure the geometric structure information effectively. However, applying mesh-based methods to large-scale point clouds can be challenging, with high computational costs leading to low algorithm efficiency.

The method we proposed leverages simplicial complexes in geometry to achieve the invariance and equivariance of point clouds through a specialized message passing network and graph neural network. This approach is precise, effective and simple to implement, while preserving the geometric structure information of point clouds. Our method involves three key steps. First, we convert point

*Corresponding author.

ICIGP 2024, January 19–21, 2024, Beijing, China

ACM ISBN 979-8-4007-1672-0/24/01
https://doi.org/10.1145/3647649.3647695

clouds into graphs with point and edge structures, and then further transform them into complexes. Second, we utilize message passing networks to extract and aggregate feature information from various simplicial complexes. Finally, we apply the graph neural network algorithm to obtain the equivariant properties based on the output features from the message passing layers.

In this article, we concentrate on point cloud classification and semantic segmentation tasks, with a specific focus on the impact of rotation and translation on these tasks. In comparison to previous methods, our approach excels at preserving the geometric structure information of point clouds, leading to higher accuracy in the results. Moreover, our method shares similarities with local meshing through simplicial complexes but effectively circumvents the challenges associated with mesh processing.

To summarize, our main contributions are

1. Introducing simplicial complexes from traditional geometry into point clouds, proposing a local mesh-like method that better preserves geometric structure information while avoiding meshing of point clouds.
2. Introducing a novel message passing model on simplicial complexes to capture feature information from point clouds and ensure equivariance.
3. Demonstrating the integration of our message passing model into the traditional graph neural network algorithm for point cloud processing.
4. Conducting extensive experiments on various tasks, including point cloud classification and segmentation, and demonstrating competitive performance of our proposed method.

2 RELATED WORK

Graph neural networks The field of point cloud research has witnessed continuous progress, with methods evolving from voxel-based approaches [17, 27] and multi-view-based approaches [32, 34] to raw point cloud-based approaches [21, 22]. Currently, graph neural network (GNN) methods have emerged as the standard approach for solving machine learning tasks on point clouds. These GNN models, such as DGCNN [28], Point-GNN [25], AdaptConv [39], and others, have demonstrated their effectiveness in various point cloud processing applications. With their ability to capture local and global geometric relationships, GNNs have become increasingly popular and have achieved state-of-the-art performance in point cloud tasks such as classification and segmentation.

GNNs treat point clouds as graphs with vertices and edges. The feature information for each point is transferred and aggregated based on their relative positional relationships within the graph. These aggregated features are then processed through standard neural networks, such as Multi-Layer Perceptrons (MLPs), to solve various machine learning tasks. DGCNN [28] gathers nearest neighboring points in the feature space, followed by the EdgeConv operators for feature extraction, in order to identify semantic cues dynamically. AdaptConv [39] proposes a new graph convolution operator to replace the isotropic kernels, which can adaptively represents the diversity of kernels unique to each pair of points. MoNet [18] defines the convolution as Gaussian mixture models in a local pseudo-coordinate system. However, traditional GNN methods may not ensure the equivariance of point clouds after rotation, and the scale features of point clouds, such as their shape categories, can be affected by their poses in 3D space. Therefore, it is of utmost importance and value to design a new neural network that can achieve equivariance for point clouds, ensuring robustness to transformations and preserving essential geometric information regardless of their orientations.

Rotation-equivariant approaches The sensitivity of CNNs to rotations has sparked interest in exploring rotation-equivariant variants. One approach to achieve equivariance is by learning orientations of the points in point clouds, which effectively decouples global rotation. However, methods like Luo et al. [15] are susceptible to noisy data when learning orientations, and others like GC-Conv [35] rely on handcrafted prior knowledge that may not be available in real-world applications. Another approach proposed by SFCNN [23] involves mapping input point clouds to a sphere and performing operations on the sphere, similar to a multiview representation. This approach aims to achieve equivariance in the presence of rotations.

Vector-based neural networks (VNNs) [3] provide an alternative solution for achieving equivariance by directly mapping rotations applied to input point clouds to intermediate layers. However, some other vector-based approaches [7, 8] may result in the loss of geometric information due to linearly combining their fully connected layers with input vectors. Tensor field-based networks [6, 20, 26] adopt the constraint of convolutional kernels to the spherical harmonics family. Alternatively, some methods are based on the theory of SO(3) representations [4, 24], employing steerable kernel bases for convolution to produce features with equivariant behavior.

Other approaches achieve equivariance through pose estimation [12, 13]. Moreover, in [1], simplicial complexes and color information are used to detect homeomorphism in protein molecular structures. Inspired by this, we introduce simplicial complexes from geometry into point clouds to achieve equivariance. This approach leverages the inherent properties of simplicial complexes to ensure robustness and equivariance in point cloud analysis.

3 PRELIMINARIES

3.1 Simplicial complex

Definition 1. [19] *Let V be a finite non-empty vertex set. A simplicial complex $\mathcal{K}$ on V is a collection of nonempty subsets of V that contains all the singleton subsets of V and is closed under the operation of taking subsets.*

A member $\sigma = \{v_0, \cdots, v_k\} \in \mathcal{K}$ with cardinality $k+1$ is called a k-dimensional simplex or simply a k-simplex. Geometrically, one can see vertices as 0-simplices, edges as 1-simplices, triangles as 2-simplices, and so on. Several simplices can form a complex, see Figure 1. Note that a complex is equivariant in three-dimensional space which can be used in our method.

Definition 2. *We say $\sigma \prec \tau$ iff $\sigma \subset \tau$ and there is no δ such that $\sigma \subset \delta \subset \tau$.*

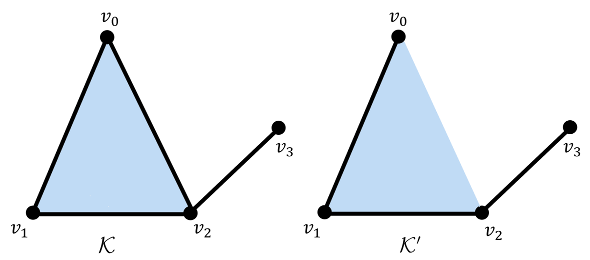

Figure 1: An illustration of the simplicial complexes. The left $\mathcal{K}$ is a simplicial complex and the right $\mathcal{K}'$ is a non-simplicial complex. Because the set $\{v_0, v_1, v_2\}$ is present in $\mathcal{K}'$ but the subset $\{v_0, v_2\}$ is not disqualifies $\mathcal{K}'$ from being a simplicial complex (The blue triangle exists but segment v_0v_2 does not exist).

This relation describes what simplices are on the boundary of another simplex. For instance vertices $\{v_1\}, \{v_2\}$ are on the boundary of edge $\{v_1, v_2\}$ and edge $\{v_3, v_4\}$ is on the boundary of triangle $\{v_3, v_4, v_5\}$. We will make use of this relation to construct our complexes from a graph.

Definition 3. [1] *Consider a simplex $\sigma \in \mathcal{K}$. Four types of adjacent simplices can be defined:*
1. *Boundary adjacencies $\mathcal{B}(\sigma) = \{\tau \mid \tau \prec \sigma\}$*
2. *Co-boundary adjacencies $C(\sigma) = \{\tau \mid \sigma \prec \tau\}$*
3. *Lower-adjacencies $down(\sigma) = \{\tau \mid \exists\delta,\ \delta \prec \tau \wedge \delta \prec \sigma\}$*
4. *Upper-adjacencies $up(\sigma) = \{\tau \mid \exists\delta,\ \tau \prec \delta \wedge \sigma \prec \delta\}$*

Clearly, the boundary simplices of an edge are determined by its vertices. The co-boundary simplices of a vertex are determined by the edges it is a part of. Lower-adjacent edges are identified by common line-graph adjacencies. Lastly, upper adjacencies between vertices correspond to regular graph adjacencies. It is important to acknowledge that a simplex may not necessarily have all four types of adjacent simplices.

3.2 Equivariance

we denote $x \in \mathbb{X}$ as input, $y \in \mathbb{Y}$ as output, and $t \in \mathbb{T}$ as a specific transform in the transformation group $\mathbb{T}$ (*e.g.* permutation group, rotation group, *etc.*), and $\mathcal{T}_t(x) : \mathbb{X} \to \mathbb{X}$, $\mathcal{G}_t(y) : \mathbb{Y} \to \mathbb{Y}$ denote the function that applies transform t to the input and output respectively. Then a function $f : \mathbb{X} \to \mathbb{Y}$ is equivariant if the following equation holds:

$$f\left(\mathcal{T}_t(x)\right) = \mathcal{G}_t\left(f(x)\right).$$

Note that invariance is a case of equivariance when $\mathcal{G}_t(y) = y$. In point cloud analysis, we consider the following types of equivariance:

Permutation equivariance and invariance: These types of equivariance are not the main focus of this article because most research on point clouds has achieved permutation equivariance for global features due to their disorder. Here we will state them formally: let $\mathcal{P}$ be a permutation group, $\theta \in \mathcal{P}$ be a bijective mapping on $\{1, 2, \cdots, N\}$, so $\mathcal{T}_\theta(\boldsymbol{X}) = (\boldsymbol{x}_{\theta(1)}, \cdots, \boldsymbol{x}_{\theta(N)})$, then the outputs are equivariant if $f\left(\mathcal{T}_\theta(\boldsymbol{X})\right) = \mathcal{G}_\theta\left(f(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N)\right)$, and the outputs are invariant if $f\left(\mathcal{T}_\theta(\boldsymbol{X})\right) = f(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N)$.

Rotational and translational equivariance and invariance: The main interest of this work is the equivariance of rotation and translation, the shape category of point clouds and point-wise semantic labels fall into this type equivariance. We also let $\mathbb{T}$ denote the group of rigid transform (including rotation and translational transform), so the transform function on input domain can be defined as $\mathcal{T}_{(\boldsymbol{R},\boldsymbol{t})}(\boldsymbol{X}) = (\boldsymbol{R}\boldsymbol{x}_1 + \boldsymbol{t}, \cdots, \boldsymbol{R}\boldsymbol{x}_N + \boldsymbol{t})$, where $(\boldsymbol{R}, \boldsymbol{t})$ denote the rotation matrix and translation matrix. Then the transform function on the output domain is equivariant if $f\left(\mathcal{T}_{(\boldsymbol{R},\boldsymbol{t})}(\boldsymbol{X})\right) = \mathcal{G}_{(\boldsymbol{R},\boldsymbol{t})}\left(f(\boldsymbol{X})\right)$, namely,

$$\begin{aligned}(f(\boldsymbol{R}\boldsymbol{x}_1 + \boldsymbol{t}), \cdots, f(\boldsymbol{R}\boldsymbol{x}_N + \boldsymbol{t})) = \\ (\boldsymbol{R}f(\boldsymbol{x}_1) + \boldsymbol{t}, \cdots, \boldsymbol{R}f(\boldsymbol{x}_N) + \boldsymbol{t})\end{aligned}$$

and the transform function is invariant if

$$(f(\boldsymbol{R}\boldsymbol{x}_1 + \boldsymbol{t}), \cdots, f(\boldsymbol{R}\boldsymbol{x}_N + \boldsymbol{t})) = \left(f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_N)\right).$$

4 METHOD

The central idea of this work is to obtain a new method for feature extraction of point cloud data through message passing model, and combined with graph neural network to achieve the equivariance of point clouds. Our method follows a three-step process (see Figure 2). First, we take the original or globally rotated point clouds as input and construct the corresponding simplicial complexes. Next, we utilize the message passing model on these simplicial complexes to aggregate information. Finally, by processing the aggregated information through graph convolutional layers, we obtain the output results. Due to the geometric properties of simplical complexes, this approach ensures the equivariance of point clouds and preserves essential geometric information during the analysis process. In this section, we will explain the methods we have adopted separately, organized as follows:

- In section 4.1, we introduce how to convert point clouds into graphs and further transform them into a complex, and explain the meshing like methods used in constructing complex shapes;
- In section 4.2, we detail the principle of message passing model and propose a message passing model based on simplicial complexes to achieve the equivariance of point clouds;
- In section 4.3, we show how to combine our message passing model with DGCNN to get a novel graph neural network with equivariance.

4.1 Construction of complex

Let $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)$ denotes the input point clouds, with each point containing at least its three-dimensional coordinates $\boldsymbol{x}_i = (x_i, y_i, z_i)$, and it is possible includes other features such as color, normal vector and so on. We use the k-nearest neighbor algorithm (K-NN) to construct a graph model $G = (V, E)$ for point clouds $\boldsymbol{X}$, where V and E denote the *vertices* and the *edges* of the graph respectively. In Section 5.3, we discuss the selection of the parameter k.

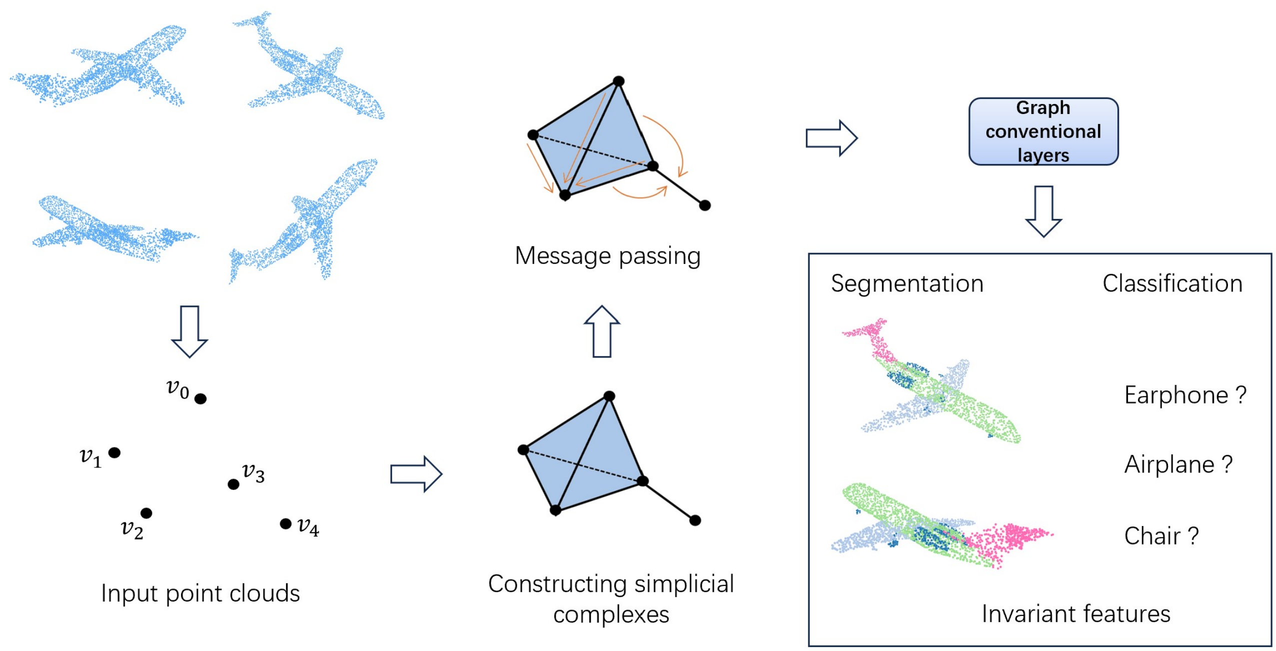

Figure 2: The pipe line of the proposed method.

In geometry, we can consider such a graph G as a simplicial complex. Then we can transform the obtained graph G to a simplicial complex $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{A}, \mathcal{W})$:

$$\mathcal{G} \longleftarrow \mathrm{COM}(G) \tag{1}$$

where COM means the process of transforming the graph into complexes, $\mathcal{G}$ is a set of complexes (such as the quantity is n) with *vertices* $\mathcal{V} = \{v_i^{(n)}\}_{i=1}^{i=k+1}$. *edges* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ in every complexes are part of E in graph G (The collect of $\mathcal{E}$ are same as E). $\mathcal{T}$ is the feature of 2-simplices in complexes, here we make use of the centroid to represent the features. $\mathcal{A}$ is adjacency matrix that specifies the path of information transmission, and $\mathcal{W}$ is edge weight matrix, where the (i, j)-*th* entry $e_{i,j}^{(n)} = H_\Theta(v_i^{(n)}, v_j^{(n)})$ denotes the weight of the edge feature that from node i to node j, and $H_\Theta : \mathcal{E} \to \mathbb{R}$ is a non-linear function with certain parameters Θ. Here we consider the difference between different points as the initial feature of the edges, and utilize *Softmax* function as the non-liner function to process edge features:

$$\mathcal{W} = \mathrm{Softmax}(\beta),\ \beta_{i,j} = ||v_i^{(n)} - v_j^{(n)}||_2 \tag{2}$$

The determination of adjacency matrix $\mathcal{A}$ is the key process to the construction of complex. The *edges* $\mathcal{E}$ obtained directly from K-NN algorithm are fully connected to all nodes in local complexes, and the adjacency matrix $\mathcal{A}$ is determined through $\mathcal{E}$. We can select different adjacency matrix according to the needs, so as to control the generated simplicial complex. For example, the simplest method is to let $\mathcal{A}$ also represent fully connected, while the efficient method is an adaptive one. The process of determining the adjacency matrix $\mathcal{A}$ is somewhat similar to the process of meshing locally, but it is simpler and more efficient, while also preserving the geometric structure information of the point clouds.

4.2 Message passing network

In PyG [5](Pytorch Geometric, a library for deep learning on irregularly structured input data), there is a message passing method that updates the central node information by aggregating adjacent node information, thereby generalizing the convolution operator to irregular domains and connecting the graph to the neural network. The algorithm obtained using this method is called a message passing graph neural network, and its main process can be described as

$$\boldsymbol{X}_i^{(t)} = \gamma_i^{(t)} \left(\boldsymbol{X}_i^{(t-1)}, \bigoplus_{j \in \mathcal{N}(i)} \phi^{(t)} \left(\boldsymbol{X}_i^{(t-1)}, \boldsymbol{X}_j^{(t-1)}, \boldsymbol{e}_{j,i} \right) \right)$$

where $\boldsymbol{X}_i^{(t-1)} \in \mathbb{R}^F$ denotes the feature of node i in the $(t-1)$-*th* layer, $\boldsymbol{e}_{j,i} \in \mathbb{R}^D$ denotes the feature of edge that from node j to node i, $\bigoplus$ denotes a differentiable, permutation invariant function, e.g., sum, mean or max, γ and ϕ denote differentiable functions such as MLPs (Multi Layer Perceptrons).

Here we propose a novel message passing graph neural network for point clouds according to the concept of simplicial complexes in section 3.1. The message passing operations we used are based on the four types of adjacent simplices in Definition 3 (see Figure 3), that is for a simplex σ in complex $\mathcal{K}$, we have:

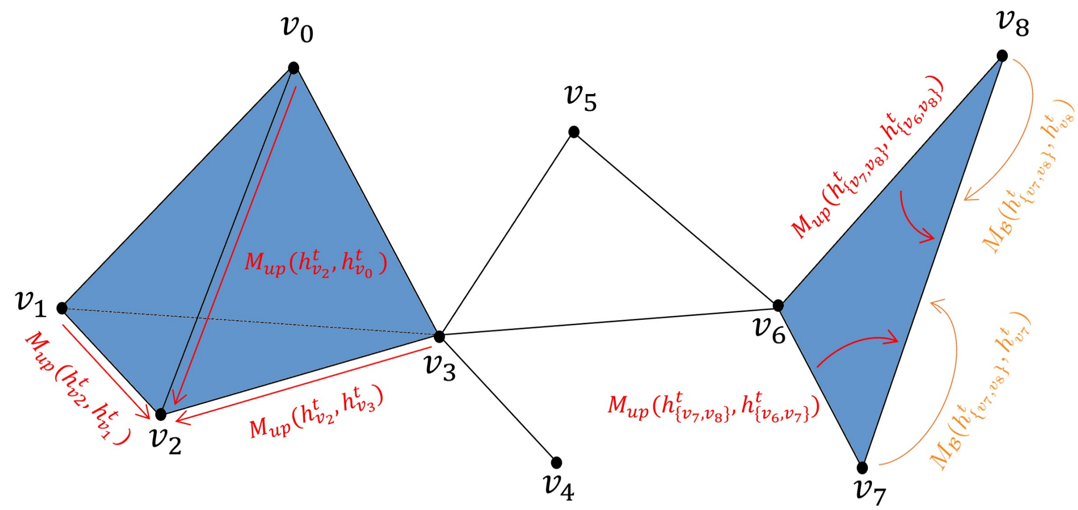

Figure 3: An illustration of the message passing model on simplicial complexes. The model contains four types to aggregate information, and we just introduce two of them as example here. In this figure, message passing with boundary and upper adjacencies to vertex v_2 and edge $\{v_7, v_8\}$. For a certain simplex like vertex v_2, the information conveyed by its non-existent adjacent simplex(e.g. boundary adjacency) is denoted as 0.

$$
\begin{aligned}
m^t_{\mathcal{B}}(\sigma) &= \mathrm{AGG}_{\tau\in\mathcal{B}(\sigma)}\left(M_{\mathcal{B}}(h^{t-1}_\sigma, h^{t-1}_\tau)\right)\\
m^t_{C}(\sigma) &= \mathrm{AGG}_{\tau\in C(\sigma)}\left(M_{C}(h^{t-1}_\sigma, h^{t-1}_\tau)\right)\\
m^t_{down}(\sigma) &= \mathrm{AGG}_{\tau\in down(\sigma)}\left(M_{down}(h^{t-1}_\sigma, h^{t-1}_\tau)\right)\\
m^t_{up}(\sigma) &= \mathrm{AGG}_{\tau\in up(\sigma)}\left(M_{up}(h^{t-1}_\sigma, h^{t-1}_\tau)\right)
\end{aligned} \tag{3}
$$

where σ and τ are the two simplexes mentioned in Definition 1. $M_{\mathcal{B}}(h^{t-1}_\sigma, h^{t-1}_\tau)$ represents feature transferring from τ to σ at iteration $t-1$, which is based on boundary adjacencies $\mathcal{B}$, and the other three functions are similar. $m^t_{\mathcal{B}}(\sigma)$ denotes the feature of simplex σ after $(t-1)$-th iteration based on boundary adjacencies $\mathcal{B}$, the others are similar.

Then the update operation takes into account these four types of incoming messages of simplex and we have:

$$h^t_\sigma = U\left(m^t_{\mathcal{B}}(\sigma), m^t_C(\sigma), m^t_{down}(\sigma), m^t_{up}(\sigma)\right) \tag{4}$$

where h^t_σ denotes the feature of simplx σ at iteration *t*.

For the main process of feature passing function $M(\cdot)$, we take into account it with linear message functions, sum aggregation for all messages and an update function taking the sum of the messages followed by a ReLU activation. Specifically, we consider a *p-dimensional* complex with a set of simplex $\mathcal{S}_n$, then we have the output feature matrix:

$$\boldsymbol{H}^{out}_n = \psi\left(\mathcal{H}(\mathcal{A}_n, \boldsymbol{H}^{in}_n), \mathcal{W}_n\right) \tag{5}$$

where $\mathcal{A}_n$ is the adjacency matrix described above that specifies the path of feature transmission, $\mathcal{H}(\mathcal{A}_n, \boldsymbol{H}^{in}_n)$ is an aggregation mapping, $\mathcal{W}_n$ is the trainable weights, ψ is an entry-wise activation function (*e.g.* ReLU function).

Then we can apply this model to the complexes obtained from the previous section. Thus, we construct our message passing model based on simplicial complexes. Different from the previous message passing model, our model focus on large-scale point cloud data, and uses simplex instead of points as information transfer units, thus making use of the equivariance of simplicial complexes to achieve the equivariance of our model.

4.3 Graph neutral network

In this section, we will illustrate how to achieve equivariance in traditional graph neural networks by integrating our message passing model. We will use DGCNN [28] as an example to demonstrate this process.

DGCNN is a classical graph neural network algorithm, which is related to two classes of approaches, PointNet and graph CNNs. DGCNN builds the model with edge convolution and dynamic graph update as the main body, and its core formulations of the first and subsequent layers are:

$$\boldsymbol{h}^{(1)}_i \longleftarrow \max_{j\in\mathcal{N}(i)} \psi\left(\phi_0(\boldsymbol{x}_j - \boldsymbol{x}_i, \boldsymbol{x}_i)\right) \tag{6}$$

$$\boldsymbol{h}^{(\ell+1)}_i \longleftarrow \max_{j\in\mathcal{N}(\boldsymbol{h}^{(\ell)}_i)} \psi\left(\phi_{(\ell)}(\boldsymbol{h}^{(\ell)}_j - \boldsymbol{h}^{(\ell)}_i, \boldsymbol{h}^{(\ell)}_i)\right) \tag{7}$$

where ψ is an activation function such as ReLU function, ϕ is MLPs function.

DGCNN has shown good performance in solving the classification and semantic segmentation problems of point clouds, but it is unable to achieve equivariance of point clouds. We can combine DGCNN with our message passing model to avoid this deficiency.

Now we consider a certain *p-dimension* complex, each node has d-dimensional feature. In our model, we first need to calculate the initial weight matrix $\mathcal{W}^{(0)}$ of the edges and the adjacency matrix $\mathcal{A}$. We can obtain the result of $\mathcal{W}^{(0)}$ like formula (2) and the result of $\mathcal{A}$ from formula (1). Then we can through the aggregation model *SCAGG* of the above message passing network and graph convolution layers denoted as *Gcov* to update the features of nodes and edges:

$$\left(\boldsymbol{H}^{(1)}, \mathcal{A}^{(1)}, \mathcal{W}^{(1)}\right) \longleftarrow \text{Gcov}\left(\mathit{SCAGG}(\mathcal{V}^{(0)}, \mathcal{A}^{(0)}, \mathcal{W}^{(0)})\right) \tag{8}$$

$$\left(\boldsymbol{H}^{(\ell)}, \mathcal{A}^{(\ell)}, \mathcal{W}^{(\ell)}\right) \longleftarrow \text{Gcov}\left(\mathit{SCAGG}(\boldsymbol{H}^{(\ell-1)}, \mathcal{A}^{(\ell-1)}, \mathcal{W}^{(\ell-1)})\right) \tag{9}$$

When we integrate DGCNN into our model, we just need to replace the formula (7) with

$$\boldsymbol{h}_i^{(\ell)} \longleftarrow \max_{j \in \mathcal{N}(i)} \psi\left(\phi_{(\ell)}(\boldsymbol{H}_j^{(\ell)} - \boldsymbol{H}_i^{(\ell)}, \boldsymbol{H}_i^{(\ell)})\right) \tag{10}$$

and at every iteration, we let

$$\left(\boldsymbol{H}^{(\ell+1)}, \mathcal{A}^{(\ell+1)}, \mathcal{W}^{(\ell+1)}\right) \longleftarrow \text{Gcov}\left(\mathit{SCAGG}(\boldsymbol{h}^{(\ell)}, \mathcal{A}^{(\ell)}, \mathcal{W}^{(\ell)})\right) \tag{11}$$

Thus, we can obtain the model that combines DGCNN and our message passing model. It is worth mentioning that we can also combine other traditional graph neural networks with our method similarly to get a new model.

5 EXPERIMENTS

In the previous sections, we explained how our method obtains equivariant features for point clouds. In this section, we will evaluate the model constructed using simplicial complexes and DGCNN on different tasks, including classification and part segmentation. These tasks will assess the model's ability to model equivariant detail properties effectively.

5.1 Classification

We use the ModelNet40 [31] to estimate our model's performance for point cloud classification task. The dataset contains 12311 shapes from 40 different categories. We followed the previous work [15, 28, 35] to use 9843 shapes for training and other 2468 shapes for testing, and each point cloud contains 1024 points. In addition, to verify the robustness of our model, we also added noise to the data.

In order to evaluate the equivariant properties of our model, we followed the setup in the previous work [3, 10, 15] and adopted three different train-test rotation settings:

(1) Z/Z: both training and test point clouds are rotated around the gravitational axis;
(2) Z/SO(3): training point clouds are rotated around the gravitational axis and test point clouds are rotated arbitrarily. This setting examines the model's quality of equivariance-by-construction.
(3) SO(3)/SO(3): both training and test point clouds are rotated arbitrarily.

Table 1: Classification accuracy on ModelNet40. The upper rows are non-equivariant models, and the lower rows are equivariant models. Our model outperforms all the baselines.

Methods	Z/Z	Z/SO(3)	SO(3)/SO(3)
PointNet[21]	85.9	19.6	74.7
PointNet++[22]	91.8	28.4	85.0
RS-CNN[14]	90.3	48.7	82.6
DGCNN[28]	90.3	33.8	88.6
RI-Conv[36]	86.5	86.4	86.4
GC-Conv[35]	89.0	89.1	89.2
Point-LO[15]	88.4	88.4	88.9
Ours	89.5	89.4	89.4

We compare our model with both equivariant and non-equivariant methods baselines. Table 1 shows the results of mean classification accuracy of different methods. We can find that our method achieves competitive performance compared to other baseline models, and our model exhibits robustness to noise.

5.2 Part segmentation

Point cloud segmentation is a classic point cloud problem. Here, we use the ShapeNet [33] dataset to verify the performance of our method in semantic segmentation problems. For this task, each point from a point cloud set is classified into one of a few predefined part category labels. The dataset contains 16881 shapes, from 16 object categories, annotated with 50 parts in total. Following the convention [11, 15], we sample 2048 points for each shape, and we also add noise to the data to check the robustness of our model. We use two train-set rotation settings for this task: Z/SO(3) and SO(3)/SO(3). We calculate the IoU (Inter-over-Union) metric to measure the segmentation quality for each category.

Table 2 and Table 3 provide a summary of the quantitative comparisons with baseline methods in the Z/SO(3) and SO(3)/SO(3) settings, respectively. Our model outperforms all the baseline models in 11 out of 16 categories in the Z/SO(3) setting and 10 out of 16 categories in the SO(3)/SO(3) setting. This improvement demonstrates the effectiveness of our equivariant model in learning fine-grained details. Additionally, our model exhibits a certain degree of robustness.

5.3 Selection of k in our message passing network

The value of k plays a crucial role in our message passing network, as it controls the generated simplicial complex. Therefore, selecting a suitable value for k is essential. To address this, we conducted a series of experiments on point cloud classification and part segmentation problems. These experiments helped us determine the optimal value of k for our model.

For the classification task, we utilized the ModelNet40 dataset, with 9843 shapes used for training and the remaining 2468 shapes for testing. Each point cloud contains 1024 points. We selected k values of 5, 10, 20, and 40. We followed the Z/SO(3) train-test rotation setting and obtained results as shown in Table 4.

Table 2: Point cloud segmentation results in the IoU (Inter-over-Union) under Z/SO(3) setting.

Z/SO(3)	plane	bag	cap	car	chair	earph.	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate	table
# shapes	2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
PointNet[21]	40.4	48.1	46.3	24.5	45.1	39.4	29.2	42.6	52.7	36.7	21.2	55.0	29.7	26.6	32.1	35.8
PointNet++[22]	51.3	66.0	50.8	25.2	66.7	27.7	29.7	65.5	59.7	70.1	17.2	67.3	49.9	23.4	43.8	57.6
PointCNN[11]	21.8	52.0	52.1	23.6	29.4	18.2	40.7	36.9	51.1	33.1	18.9	48.0	23.0	27.7	38.6	39.9
DGCNN[28]	37.0	50.2	38.5	24.1	43.9	32.3	23.7	48.6	54.8	28.7	17.8	74.4	25.2	24.1	43.1	32.3
ShellNet[37]	55.8	59.4	49.6	26.5	40.3	51.2	53.8	52.8	59.2	41.8	28.9	71.4	37.9	49.1	40.9	37.3
RI-Conv[36]	80.6	80.0	70.8	68.8	86.8	70.3	87.3	84.7	77.8	80.6	57.4	91.2	71.5	52.3	66.5	78.4
GC-Conv[35]	80.9	82.6	81.0	70.2	88.4	70.6	87.1	87.2	81.8	78.9	58.7	91.0	77.9	52.3	66.8	80.3
RI-Fwk[10]	81.4	82.3	86.3	75.3	88.5	72.8	90.3	82.1	81.3	81.9	67.5	92.6	75.5	54.8	75.1	78.9
LGR-Net[38]	81.5	80.5	81.4	75.5	87.4	72.6	88.7	83.4	83.1	86.8	66.2	92.9	76.8	62.9	**80.0**	80.0
TFN[20]	81.1	77.8	79.8	74.5	89.1	77.2	90.8	82.8	77.7	78.6	60.3	93.4	77.0	54.7	74.4	79.5
Point-LO[15]	81.7	79.0	85.0	78.1	89.7	76.5	**91.6**	85.9	81.6	82.1	**67.6**	**95.0**	79.6	**64.4**	76.9	80.7
Ours	**82.4**	**84.0**	**87.8**	**78.5**	**90.2**	**89.6**	91.1	**87.9**	**83.6**	**96.0**	61.3	94.4	**80.9**	52.2	72.3	**82.4**

Table 3: Point cloud segmentation results in the IoU (Inter-over-Union) under SO(3)/S0(3) setting.

SO(3)/SO(3)	plane	bag	cap	car	chair	earph.	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate	table
# shapes	2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
PointNet[21]	81.6	68.7	74.0	70.3	87.6	68.5	88.9	80.0	74.9	83.6	56.5	77.6	75.2	53.9	69.4	79.9
PointNet++[22]	79.5	71.6	**87.7**	70.7	88.8	64.9	88.8	78.1	79.2	94.9	54.3	92.0	76.4	50.3	68.4	81.0
PointCNN[11]	78.0	80.1	78.2	68.2	81.2	70.2	82.0	70.6	68.9	80.8	48.6	77.3	63.2	50.6	63.2	82.0
DGCNN[28]	77.7	71.8	77.7	55.2	87.3	68.7	88.7	85.5	81.8	81.3	36.2	86.0	77.3	51.6	65.3	80.2
ShellNet[37]	79.0	79.6	80.2	64.1	87.4	71.3	88.8	81.9	79.1	95.1	57.2	91.2	69.8	55.8	73.0	79.3
RI-Conv[36]	80.6	80.2	70.7	68.8	86.8	70.4	87.2	84.3	78.0	80.1	57.3	91.2	71.3	52.1	66.6	78.5
GC-Conv[35]	81.2	82.6	81.6	70.2	88.6	70.6	86.2	86.6	81.6	79.6	58.9	90.8	76.8	53.2	67.2	81.6
RI-Fwk[10]	81.4	**84.5**	85.1	75.0	88.2	72.4	90.7	84.4	80.3	84.0	**68.8**	92.6	76.1	52.1	74.1	80.0
LGR-Net[38]	81.7	78.1	82.5	75.1	87.6	74.5	89.4	86.1	83.0	86.4	65.3	92.6	75.2	**64.1**	**79.8**	80.5
TFN[20]	80.8	74.5	82.8	74.4	89.4	75.7	90.6	81.0	77.8	80.5	62.4	93.3	78.5	55.8	74.7	79.5
Point-LO[15]	81.8	78.8	85.4	78.0	89.6	76.7	**91.6**	85.7	81.7	82.1	67.6	**95.0**	79.1	63.5	76.5	81.0
Ours	**82.7**	82.3	**87.7**	**78.3**	**90.0**	**89.2**	91.0	**88.0**	**83.4**	**96.3**	60.9	94.1	**80.5**	51.7	71.9	**82.2**

Regarding the part segmentation task, we used the ShapeNet [33] dataset and focused on the *earphone* category. We sampled 2048 points for each shape in this category and used k values of 5, 10, 20, and 40. We also adopted the Z/SO(3) train-test rotation setting and obtained results as presented in Table 5.

Table 4: The performance of different k in classification task under Z/SO(3) setting.

k	5	10	20	40
accuracy	88.5	89.3	89.4	89.2

Table 5: The performance of different k in part segmentation task of earphone under Z/SO(3) setting.

k	5	10	20	40
accuracy	79.4	82.6	86.4	89.6

As the value of k increases, the accuracy may continue to improve, but the computational workload will increase significantly, leading to a decrease in efficiency. From Table 4, we choose $k = 20$ in our classification task and from Table 5, we choose $k = 40$ in our part segmentation task.

6 CONCLUSION

In this paper, we present a scheme for equivariant point cloud analysis. The key element is a feature extraction model for point clouds using message passing based simplicial complexes, integrated with traditional graph neural networks. Through extensive experiments, we demonstrate the effectiveness and generality of our model.

However, the proposed method currently requires local semi-gridding in the feature extraction model, which demands a considerable amount of computational effort. It is crucial to explore ways to reduce the computational cost of this process. Additionally, while we demonstrate the method's efficacy in a few applications, it would be valuable to investigate its performance in other more challenging tasks as well.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China No.1217154, U19B2940, Beijing Natural Science Foundation under Grant Z190004, National Key Research and Development Program of China under Grant 2020YFA0713703 and Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. 2021. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*. PMLR, 1026–1037.

[2] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. 2018. Spherical cnns. *arXiv preprint arXiv:1801.10130* (2018).

[3] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. 2021. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12200–12209.

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

[5] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

[6] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. 2020. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems* 33 (2020), 1970–1981.

[7] Zan Gojcic, Lorenz Schmid, and Andreas Wieser. 2021. Dense 3D displacement vector fields for point cloud-based landslide monitoring. *Landslides* 18 (2021), 3821–3832.

[8] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. 2020. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411* (2020).

[9] Artem Komarichev, Zichun Zhong, and Jing Hua. 2019. A-cnn: Annularly convolutional neural networks on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7421–7430.

[10] Xianzhi Li, Ruihui Li, Guangyong Chen, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. 2021. A rotation-invariant framework for deep point cloud analysis. *IEEE transactions on visualization and computer graphics* 28, 12 (2021), 4503–4514.

[11] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. 2018. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems* 31 (2018).

[12] Chien-Ming Lin, Chi-Yi Tsai, Yu-Cheng Lai, Shin-An Li, and Ching-Chang Wong. 2018. Visual object recognition and pose estimation based on a deep semantic segmentation network. *IEEE sensors journal* 18, 22 (2018), 9370–9381.

[13] Xiang Liu, Hongyuan Wang, Xinlong Chen, Weichun Chen, and Zhengyou Xie. 2022. Position Awareness Network for Noncooperative Spacecraft Pose Estimation Based on Point Cloud. *IEEE Trans. Aerospace Electron. Systems* 59, 1 (2022), 507–518.

[14] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. 2019. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8895–8904.

[15] Shitong Luo, Jiahan Li, Jiaqi Guan, Yufeng Su, Chaoran Cheng, Jian Peng, and Jianzhu Ma. 2022. Equivariant point cloud analysis via learning orientations for message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18932–18941.

[16] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. 2015. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*. 37–45.

[17] Daniel Maturana and Sebastian Scherer. 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 922–928.

[18] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5115–5124.

[19] Vidit Nanda. March 2021. Computational algebraic topology lecture notes. (March 2021).

[20] Adrien Poulenard and Leonidas J Guibas. 2021. A functional approach to rotation equivariant non-linearities for Tensor Field Networks.. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13174–13183.

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

[22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* 30 (2017).

[23] Yongming Rao, Jiwen Lu, and Jie Zhou. 2019. Spherical fractal convolutional neural networks for point cloud recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 452–460.

[24] Wen Shen, Binbin Zhang, Shikun Huang, Zhihua Wei, and Quanshi Zhang. 2020. 3d-rotation-equivariant quaternion neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer, 531–547.

[25] Weijing Shi and Raj Rajkumar. 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1711–1719.

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

[27] Cheng Wang, Ming Cheng, Ferdous Sohel, Mohammed Bennamoun, and Jonathan Li. 2019. NormalNet: A voxel-based CNN for 3D object classification and retrieval. *Neurocomputing* 323 (2019), 139–147.

[28] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)* 38, 5 (2019), 1–12.

[29] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 2018. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems* 31 (2018).

[30] Daniel Worrall and Gabriel Brostow. 2018. Cubenet: Equivariance to 3d rotation and translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 567–584.

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

[32] Ze Yang and Liwei Wang. 2019. Learning relationships for multi-view 3D object recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*. 7505–7514.

[33] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. 2016. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)* 35, 6 (2016), 1–12.

[34] Tan Yu, Jingjing Meng, and Junsong Yuan. 2018. Multi-view harmonized bilinear network for 3d object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 186–194.

[35] Zhiyuan Zhang, Binh-Son Hua, Wei Chen, Yibin Tian, and Sai-Kit Yeung. 2020. Global context aware convolutions for 3d point cloud understanding. In *2020 International Conference on 3D Vision (3DV)*. IEEE, 210–219.

[36] Zhiyuan Zhang, Binh-Son Hua, David W Rosen, and Sai-Kit Yeung. 2019. Rotation invariant convolutions for 3d point clouds deep learning. In *2019 International conference on 3d vision (3DV)*. IEEE, 204–213.

[37] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. 2019. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1607–1616.

[38] Chen Zhao, Jiaqi Yang, Xin Xiong, Angfan Zhu, Zhiguo Cao, and Xin Li. 2022. Rotation invariant point cloud analysis: Where local geometry meets global topology. *Pattern Recognition* 127 (2022), 108626.

[39] Haoran Zhou, Yidan Feng, Mingsheng Fang, Mingqiang Wei, Jing Qin, and Tong Lu. 2021. Adaptive graph convolution for point cloud analysis. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4965–4974.